Co-Designing Heterogeneous Models: A Distributed Systems Perspective

Marius-Constantin Ilau

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of

University College London.

Department of Computer Science

University College London

September 27, 2025

I, Marius-Constantin Ilau, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

The realisation of the absurd cannot be an end, but only a beginning.

Albert Camus

Abstract

Conceptual modelling has existed since the early days of human cognition. However, given the technological and social advancements of today, the object of modelling has increased in complexity. Such objects are no longer singular entities, but heterogeneous socio-technical systems interlinked to form large-scale ecosystems. Furthermore, the underlying components of a system might be based on very different epistemic assumptions and methodologies for construction, interpretation and use. Naturally, consistent, rigorous reasoning about such systems is hard.

This thesis aims at constructing a pragmatic modelling methodology tailored for heterogeneous systems based on four elements: an inferentialist interpretation of what a model is, supported by a model characterization focused on means of construction, a distributed systems metaphor to structure that interpretation, and a co-design cycle to describe the practical design and construction steps of the modelling process.

The underlying idea is that an open world interpretation, supported by a formal, yet generic abstraction facilitating knowledge translation and providing properties for structured reasoning and, used in practice according to the co-design cycle could lead to a better understanding of heterogeneous models, and subsequently, to models that are more likely to achieve their pre-stated goals. Additionally, conceptualizing, interpreting and constructing models using this approach is inherently multidisciplinary, allowing for integration of models from different research traditions.

We explore the suitability of this method in the context of four case studies: a mixed-methods, descriptive case study on the nature of information security models, and three methodological case studies detailing the application of our approach in three different settings: a physical data loss model, an organisational recovery under ransomware model, and an emergency capacity trauma unit model.

Contents

List of Figures

List of Tables

Impact Statement

The nature of the research presented in this thesis can be characterised as conceptual, methodological, and multidisciplinary. Its primary research targets are models in general, or more precisely, the processes of heterogeneous model conceptualisation, design, construction and interpretation. Given this extended, cross-disciplinary scope and general targets, the benefits of the approach can be mapped to a wide range of activities and stakeholders in both academia and industry.

Firstly, this account of modelling impacts areas of philosophy of science, scientific modelling, and representation theory by providing a new conceptualisation of model which addresses the lack of an explicit representation for heterogeneous systems, from a modelling practitioner's perspective. It can be seen as a form of pragmatic inferentialism, placing the utmost importance on the relationship between models, pre-stated goals, and interpretation rather than the more traditional model to empirical world relationship. Technicalities aside, our position aims to impact the above described areas by serving as a framework for multidisciplinary integration, and generation of pragmatic modelling theories.

Secondly, the explicit co-design cycle described in Chapter 5 has an impact on all the participants in the modelling activity, regardless of their field of inquiry. For modellers, it ensures that the resulting model is indeed pluri-perspectivist, and provides explicit ways of conceptualising and integrating the knowledge of stakeholders at the level of the model. For stakeholders and domain experts, it provides a better understanding of the modelling process and a way to ensure that their expertise and domain knowledge are included and accounted for, which in turn can lead to a higher representation quality, increased motivation, and therefore to a higher representation quality and chance of models achieving their goals.

Thirdly, this work is relevant for the field of information security. The case study in Section 3.4 particularly identifies a plethora of security model types, but very little attempts at integration. Interestingly, this seems to contradict the general trend of tool centralisation in the security industry. We posit that this miss-alignment will become an increasingly pressing issue, as more complex security models are developed and deployed without a systematic understanding of what they are, and what they do. Under such assumptions, there should be no surprise when unexpected outcomes are produced. However, if employed during the entire life-cycle of a model, our method ensures that the participants posses an understanding of the model's inner workings, including its possible outcomes.

Fourthly, the application of our method at the level of the models presented in Chapter 6 was seen as valuable by the involved participants and organisations. Particularly in the case of the ransomware recovery model, the modelling process led to the identification of a relevant factor not originally accounted for: the admin deployment policy. This highlights the practical significance and real-world applicability of models constructed using our approach, not only for validating existing assumptions, but also for uncovering critical, yet overlooked factors.

To conclude, the realisation of the above described impacts is highly dependent on further dissemination and increased collaboration between academia, industry and decision-makers. Currently, the work on this thesis has resulted in five academic publications, a modelling workshop, and is actively being disseminated across the UCL Iris Project.

Acknowledgements

At the outset, I would like to express my deepest gratitude to my supervisors, Prof. David Pym and Dr. Tristan Caulfield, whose guidance, encouragement and openness were instrumental in the completion of this thesis. To David, thank you for showing me that a focus on theoretical rigour does not necessarily translate to a positivist interpretation of reality, and yet, no matter how complex an interpretation is, it can still produce some empirical outcomes. To Tristan, thank you for always being there to question my loquacious attempts at conceptual analysis, and to give me a helping hand at times when modern software development manifested itself in-the-world, in rather unexpected ways.

Furthermore, I would like to sincerely thank Dr. Will Venters, and Prof. Susan Scott from LSE for introducing me to the fields of organisational research and information systems. The new perspectives encountered during my LSE visit in 2021 have had a powerful impact in shaping the research trajectory of this work and, more broadly, on my own perception of organisation in general.

Next, I extend my heartfelt gratitude to Dr. Adrian Baldwin from HP Security Labs Bristol for being an extremely active research collaborator, ideal co-design stakeholder and for a short time, direct manager. Your expertise, curiosity, vision of technology, and ability to balance the pursuit of highly complex research targets with organisational needs have been a source of inspiration. I would also like to thank to Dr. Simon Shiu, and Jonathan Griffin for all the interesting and challenging talks we had during my time at the research lab in 2023.

I am also extremely grateful to Prof. Phyllis Illari for helping me with organising the 2022 modelling methodology workshop at UCL and for providing valuable feedback and guidance regarding my incipient epistemic position at the time.

Also, I would like to deeply thank Paul Jennings and Michael Sofowora for providing me with a lot of interesting industry examples of security decision making processes that would benefit from a better integration with modelling approaches.

To my parents, thank you for everything. From the continuous and applied social survival lessons starting in 1996 in post-communist Romania, to the unwavering support that I perhaps too much demanded during all my postgraduate studies in post-brexit Britain. And everything in between. None of it would have literally been possible without you.

Last but absolutely not least, I must thank my partner Simona for shouldering all the years' past efforts alongside me, and for showing me the way towards dissolving the proverbial rock rather than pushing it. For the long philosophical debates that went past midnight. For telling me to take a break, perhaps eat something, and inherently showing me that neither insanity nor starvation should be research outcomes. For new perspectives about-the-world, and for completely different experiences in-the-world. Or, simply for being. For any words I might try to write next could not fully convey my gratitude.

UCL Research Paper Declaration Form

Manuscript 1

(a) What is the title of the manuscript?

Engineering Ecosystem Models: Semantics and Pragmatics

(b) Please include a link to or doi for the work:

`https://doi.org/10.1007/978-3-030-97124-3_21`

(c) Where was the work published?

Simulation Tools and Techniques. SIMUtools 2021. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 424.

(d) Who published the work?

Springer, Cham

(e) When was the work published?

31 March 2022

(f) List the manuscript's authors in the order they appear on the publication:

Tristan Caulfield, Marius-Constantin Ilau, David Pym

(g) Was the work peer reviewed?

Yes

(h) Have you retained the copyright?

Yes. Creative Commons license.

(i) Was an earlier form of the manuscript uploaded to a preprint server (e.g. medRxiv)?

No.

☒ *I acknowledge permission of the publisher named under 1d to include in this thesis portions of the publication named as included in 1c.*

(j) Statement of contribution:

M.I. framed the problem and justified the utility of the proposed solution, conceptually interpreted the properties, introduced the example recovery model. Reviewed and remotely presented the paper at SIMUtools 2021. D.P. designed the formal theory including the small illustrative examples, and formal proofs. T.C. constructed the graphical representations and assisted M.I. when interpreting the properties. All authors contributed to the design, writing, and editing of the final manuscript.

(k) In which chapter(s) of your thesis can this material be found?

Chapters 4 and 6.

Manuscript 2

(a) What is the title of the manuscript?

Modelling Organizational Recovery

(b) Please include a link to or doi for the work:

`https://doi.org/10.1007/978-3-030-97124-3_23`

(c) Where was the work published?

Simulation Tools and Techniques. SIMUtools 2021. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 424.

(d) Who published the work?

Springer, Cham

(e) When was the work published?

31 March 2022

(f) List the manuscript's authors in the order they appear on the publication:

Adrian Baldwin, Tristan Caulfield, Marius-Constantin Ilau, David Pym

(g) Was the work peer reviewed?

Yes

(h) Have you retained the copyright?

Yes. Creative Commons license.

(i) Was an earlier form of the manuscript uploaded to a preprint server (e.g. medRxiv)?

No.

☒ *I acknowledge permission of the publisher named under 1d to include in this thesis portions of the publication named as included in 1c.*

(j) Statement of contribution:

M.I. defined the organisational recovery problem, described the used modelling approach, performed the model conceptual design alongside D.P. and T.C, and the implementation review with the help of A.B. Reviewed and remotely presented the paper at SIMUtools 2021. A.B. described the security problem from an industry perspective, proposed possible modelling directions. D.P. described the underlying theoretical formalism employed and provided managerial supervision. T.C. implemented the model and constructed the graphical representations, with help from M.I. All authors contributed to the results analysis and interpretation, design, writing, and editing of the final manuscript.

(k) In which chapter(s) of your thesis can this material be found?

Chapters 4 and 6.

Manuscript 3

(a) What is the title of the manuscript?

Meta-modelling for Ecosystems Security

(b) Please include a link to or doi for the work:

`https://doi.org/10.1007/978-3-030-97124-3_22`

(c) Where was the work published?

Simulation Tools and Techniques. SIMUtools 2021. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol 424.

(d) Who published the work?

Springer, Cham

(e) When was the work published?

31 March 2022

(f) List the manuscript's authors in the order they appear on the publication:

Tristan Caulfield, Marius-Constantin Ilau, David Pym

(g) Was the work peer reviewed?

Yes

(h) Have you retained the copyright?

Yes. Creative Commons license.

(i) Was an earlier form of the manuscript uploaded to a preprint server (e.g. medRxiv)?

No.

☒ *I acknowledge permission of the publisher named under 1d to include in this thesis portions of the publication named as included in 1c.*

(j) Statement of contribution:

M.I. reviewed relevant philosophical positions, described the 'Triangle Framework', and sketched initial position regarding multi-methodology. Introduced an initial version of co-design cycle — with help from D.P. and T.C. — and reasoned about the possible integration of the 'Triangle Framework' into it. Designed the methodology for the exploratory security models case study. Performed the grounded theory case study, graphically represented the results, and reasoned about future implications for the validation of security models. Reviewed and remotely presented the paper at SIMUtools 2021. D.P. and T.C. provided managerial supervision and reviewed the methodology, results analysis and graphical representations. All authors contributed to the design, writing, and editing of the final manuscript.

(k) In which chapter(s) of your thesis can this material be found?

Chapters 3 and 5.

Manuscript 4

(a) What is the current title of the manuscript?

Modelling and Simulating Organizational Ransomware Recovery: Structure, Methodology, and Decisions

(b) Has the manuscript been uploaded to a preprint server e.g. medRxiv' ? If 'Yes', please please give a link or doi:

Yes. `http://www0.cs.ucl.ac.uk/staff/D.Pym/HP_Security_Recovery_Journal.pdf`

(c) Where is the work intended to be published?

SIMULATION: Transactions of The Society for Modeling and Simulation International

(d) List the manuscript's authors in the intended authorship order:

Marius-Constantin Ilau, Adrian Baldwin, Tristan Caulfield, David Pym

(e) Stage of publication:

In Review.

(j) Statement of contribution:

M.I. defined the organisational recovery problem, reviewed the relevant literature on ransomware behaviour and recovery techniques, described the employed methodology, conceptually designed the model — with help from all the other authors —, implemented the model — with help from T.C and A.B. — ran the model on the UCL cluster — with help from T.C — performed the sensitivity analysis and created all the graphical representations — reviewed by all authors. A.B. described the security problem from an industry perspective, helped M.I. with the literature review, and provided expert knowledge for the model construction. D.P. described the underlying theoretical formalism employed and provided managerial supervision. All authors contributed to the co-design of the conceptual model, results analysis and interpretation, design, writing, and editing of the final manuscript.

(k) In which chapter(s) of your thesis can this material be found?

Chapters 4, 5, and 6.

Manuscript 5

(a) What is the current title of the manuscript?

Co-designing Heterogeneous Models: a Distributed Systems Approach

(b) Has the manuscript been uploaded to a preprint server e.g. medRxiv' ? If 'Yes', please please give a link or doi:

Yes. `https://arxiv.org/abs/2407.07656`

(c) Where is the work intended to be published?

The Journal of Cybersecurity

(d) List the manuscript's authors in the intended authorship order:

Marius-Constantin Ilau, Tristan Caulfield, David Pym

(e) Stage of publication:

Accepted for publication. Currently in revision.

(j) Statement of contribution:

M.I. performed the literature survey, described the methodology, modelling account, comparison between the 2 metaphors, extended the co-design cycle and described, analysed and graphically represented the 3 models. D.P and T.C. designed the trauma unit model, collected the necessary data associated with it, and provided managerial supervision. All authors contributed to the co-design of the conceptual recovery model, analysis, interpretation, design, writing, and editing of the final manuscript.

(k) In which chapter(s) of your thesis can this material be found?

Chapters 2, 3, 4, 5, and 6.

e-Signatures confirming that the information above is accurate:

Candidate: Marius-Constantin Ilau

Date: 25 February 2025

Supervisor: David J Pym

Date: 25 February 2025

Chapter 1

Introduction

All ‘good’ models are alike, yet each ‘bad’ model is ‘bad’ in its own way; a reinterpretation of Tolstoy’s famous introduction to Anna Karenina, serving to remind us that the ability to interpret phenomena around us and construct conceptual models based on such interpretations are subjective and strongly related to human cognition. Indeed, all ‘good’ models are alike in the sense that they achieve a goal, whether that might be understanding, predicting, teaching, serving as a reference or guideline, and so on. In other words, model quality is related to model instrumentality, but perhaps not directly to form. Each ‘bad’ model is ‘bad’ in its own way tells the other part of the story: just as interpretations are subjective, so are models, and although classes of ‘bad’ models can be constructed based on form, this is significantly harder to do so based on interpretation.

These are somewhat synthetic statements about what a ‘good’ model is. We do not imply here that analytic statements exist but that the above statements might simply change over time based on further observation. However, something we can clearly claim without a doubt is that the objects of modelling or the phenomena, entities, and relationships to be modelled have been increasing in complexity. The reasons are multiple: technological advances lead to larger and more interconnected systems on a global scale, social desiderata continuously bring changes to criteria that guide the construction and evaluation of systems and models in general, and new scientific paradigms generate different interpretations of phenomena, which are then translated into new types of models. In a very Kuhnian sense, the paradigmatic shift cycle manifests itself not only at the level of modelling methodology but also at the level of interpretation choice for each underlying component of the model.

Accepting this view brings forward some important implications towards modelling in the future.

First of all, considering this continuous generative process of new theories and interpretations, we can assume that over time, the number and diversity of model epistemological assumptions will increase. Although this is not directly a negative effect, it raises the question of how could an increasing number of epistemically divergent models be integrated in a structured, systematic manner. Approaches such as multimethodology [214, 42] or scenario-based epistemic integration [33] represent attempts at solving this issue, at least at the level of conceptual models.

The situation changes when considering different types of models. For example, in digital economy, a subscription-based service providing customer access to executable models via publicly available network APIs is a very lucrative business model. This allows for faster development of new services by operationally integrating multiple such models in an almost compositional manner. However, such integrations hardly take into account the nature of the resulting model because of factors such as a lack of information regarding the integrated models, a lack of testing the integrated models until a reasonable representation of their behaviour is at least inferred, or a lack of understanding the implications of future updates at the level of the integrated models. It should be needless to say that constructing and deploying models in this way can lead to unexpected outcomes. Although we are aware that the management of such risks is ultimately different depending on the area of activity, the risk appetite of stakeholders, legislation, and so on, we must also be clear in stating that in some areas — such as information security — an in depth understanding of the constructed model is paramount, given the risks involved.

Secondly, considering more procedural implications, the process of uncovering the relevant aspects of each epistemic assumption used in an aggregation of models is a process that requires the participation of multiple parties: modellers, experts, stakeholders, parties affected by it, etc. In this context, a perspective such as knowledge co-creation [254] or participatory design [277] represent good starting points for constructing a process of knowledge translation between the interpretation of empirical realities and model truth. However, the success of any kind of methodological approach to modelling centred on knowledge constructed by multiple participants will be related to the desire of the participants to take part in the modelling activity. Therefore, the modelling process should be conceptualised and designed to ensure the participants' motivation by providing them with an environment in which they obtain something in return.

Furthermore, one must remember that models are not scientific theories, but a possibly instrumental step towards their development. Regardless of type, their construction, and particularly validation can be influenced by time and resource constraints. This can happen particularly when the goal of modelling is to provide a very practical solution to a developing crisis situation, as seen for instance during the Covid pandemic. As crisis situations tend to increase the risk appetite of both stakeholders and people affected — when presented with the possible development of a 'silver bullet' solution — it is especially important for a modelling process to include ways to clearly state and manage assumptions about its target system, and its future behaviour in a timely, structured manner.

The acceptance of these implications leaves the modeller facing the following question: how can models better describe the world and produce better results in the world if the underlying realities and systems become more complex and the number of parties involved in the modelling process and their assumptions about such realities increases and are not always in agreement? In the next subsection, we will detail about how this question can be used to generate additional research question, and on how we attempt to answer such questions using our proposed modelling approach.

In summary, this thesis proposes an integrative modelling methodology in an attempt to answer the above question from a pragmatic perspective. Since the target of this modelling approach are heterogeneous systems, we argue that an inferentialist stance regarding the nature of models supported by a model characterisation focused on means of construction, and a multi-methodological view of model epistemology, ontology, and metaphysics provides an open-enough interpretation for understanding such systems. However, such a stance inherently leads to pluri-perspectivist models based on multiple interpretations, and these interpretations need to be managed somehow to produce practical, understandable, and believable results: we do so by translating the knowledge parts constructed by the participants during the modelling process, at the level of a translation zone, via the distributed systems metaphor and we take into account the different stages a model goes through during design and creation via the extended co-design cycle. Additionally, we ensure that the modelling process is beneficial to the participants by conceptualising the translation zone as a trading zone, then ensuring this trading zone evolves following a specific trajectory. Lastly, we explore the suitability of this method in the context of four case studies: a mixed-methods, descriptive case study on the nature of information security models, and three methodological case studies detailing the application of our approach in three different settings: a physical data loss model, an organisational recovery under ransomware model, and an emergency surge capacity trauma unit model.

1.1 Research questions & thesis structure

This thesis primarily focuses on philosophical and methodological aspects of modelling heterogeneous systems. The motivation behind it can be seen as an attempt to answer the following open-ended research question:

Q: *How can models better describe the world and produce better results in the world if the underlying realities and systems to be modelled become more complex, the number of parties involved in the modelling process increases, and their assumptions about such realities are not always in agreement?*

However, attempting to answer it in that form has only lead us to the realisation that a singular answer might not exist, which in turn led to more questions. We illustrate them below:

Q1: *What philosophical attitude towards modelling would be most suitable to underline a modelling account focused on integration?*

Q2: *How should we conceptualise models of heterogeneous systems?*

Q3: *How can we ensure the consistency of a heterogeneous model?*

Q4: *How can we integrate the knowledge and beliefs of the modelling participants at the level of the model?*

Q5: *How can we increase the probability of the model achieving its goals?*

We attempt to answer the above described research questions, at the level of each individual chapter in the thesis, in the following way:

Chapter 1: Introduction — This chapters describes the implications of the interpretative nature of models in relationship with an increase in the complexity of both model targets and models of heterogeneous systems. Building on this implications, it presents a list of research questions to be addressed by this thesis, and their mapping at the level of thesis structure, a summary of the main arguments of the thesis, the thesis contributions, and the supporting papers published by the author prior, or during the writing of this thesis, which have been partially reused in it.

Chapter 2: Background — This chapter directly addresses the first question from above. By analysing different positions regarding models in the philosophy of science, and representation theory literature, it is able to determine the existence of a significant gap regarding the conceptualisation of heterogeneous systems. Additionally, it showcases that: neither a positivist, nor a relativist perspective on their own are enough for representing heterogeneous systems; some elements of referentialism can lead to useful, generic properties in the act of modelling, but that an inferentialist perspective is more open, and therefore more suitable for heterogeneous systems;

Chapter 3: Modelling Heterogeneous Systems — This chapter focuses on the second research question. Specifically, drawing on the insights gathered during the background literature reviews, it illustrates our own account of modelling. Based on that account, it determines a set of model qualities — namely conceptuality, formality, and executability — which can be used to characterise any kind of model, and it conducts an exploratory case study in the area of information security models, in an attempt to illustrate that different kinds of models are widely employed in that field. Lastly, based on the models encountered in the study and existing directions in the management science, economics and system dynamics literature, we have started the much needed debate on validation methods that the information security field has been avoiding for far too long.

Chapter 4: Metaphors — This chapter provides a part of the answers to the second, third and fifth questions. Firstly, it identifies the trading zone metaphor as a useful approach to achieving knowledge translation. Secondly, it describes the distributed systems metaphor and reasons about its useful properties — namely generality, recognizability, scale-freeness, formal properties, implemented tools, and identity conservation — for a modelling process. Thirdly, it compares the two metaphors and determines that an abstraction of the distributed systems metaphor can be used as an in-between vocabulary tool for achieving knowledge translation in a trading zone fashion, while at the same time conserving the useful properties.

Chapter 5: Methodology — This chapter attempts to provide a complementary answer to the third, fourth and fifth research questions by showcasing our co-design modelling process. Specifically, it describes how the co-design cycle is constructed by extending the classical mathematical modelling cycle with the inclusion of co-design considerations, philosophy implications, our own account of the nature of models, and the distributed systems metaphor as knowledge translation mechanism.

Chapter 6: Model Case Studies — This chapter differs from previous ones in the sense that it does not directly answer any of the above sub-questions. In line with the epistemic position of the thesis, it attempts to justify the utility of the proposed methodology in answering the initial question from the start of the section, from a practical perspective. It does so by illustrating how the methodology was employed in constructing three different security oriented models, then discussing their results, further implications, and benefits.

Chapter 7: General Conclusions — This chapter is used to consolidate the main arguments of the thesis, and reflect on the implications of all the previous chapters, the benefits, limitations, and areas of improvement of the presented methodology, and modelling account.

Appendices A, B, and C — The appendices contain additional information related to the organisational recovery model: the analysis of ransomware strains, the recovery timings used for the network, embedded and usb recovery techniques, the types, values and meanings of both the static and variable parameters, the numerical and graphical representations of the PAWN sensitivity analysis, and the impact of the variable input parameters on the model outputs, respectively.

1.2 Argument summary

The main argument underlying the thesis can be briefly summarised as following: since modelling today is a collective activity involving parties with various beliefs, goals and interpretation criteria attempting to produce a singular, pluri-perspectivist representation of a heterogeneous target, there is no guarantee that such a representation would ever produce unitary interpretations when deployed or interacted with.

Historically, such reductionist attempts at limiting the interpretation potential of a model have been positively perceived as clear, simple design. We do not question the benefits of such a strategy, especially if the system under study is not viewed as heterogeneous. Also, we are not claiming that simple, clear model conceptualisation cannot be achieved in the case of modelling heterogeneous systems. We argue that in the heterogeneous case, a simple and clear model conceptualisation should no longer be attributed to a reductionist take on interpretation, because understanding heterogeneous systems requires precisely multiple interpretations of the components of such a system. This is why an inferentialist, yet sceptical account of modelling is required: to acknowledge the possibility of multiple valid interpretations, but at the same time sceptically consider their implications towards the model outcomes.

Therefore, the question to ask should not be what is the best representation of the target, but how to manage the underlying aspects, the so called unstated assumptions, that so often end up deviating the co-designed model from a more favourable manifested configuration. This bears the implication that the best representation is not a pre-determined, a priori fact, but the result of a decision process that lays with the modelling participants.

Naturally, this management aspect should be included, by design in the modelling process, and we do so when determining the model scope, during the translation zone iterations, during model construction, when deriving model consequences, or during verification and validation. Additionally, we must also take into account the fact that the quality of proposed representations is directly influenced by the participants' motivation to be part of the modelling process, and although that is again subjective, Galison's trading zone approach [125] provides a way of increasing the probability of that happening. The only necessary and sufficient criterion for success requires that the conceptualised model components maintain a common identity across participants, which is achieved via an in-between language.

However, scientific literature does not put additional constraints on the nature of such a language. This allows us to chose or construct a set of concepts, based on a metaphor of distributed systems — which we show to be compatible with the trading zone metaphor —, which provide a set of extremely useful properties, and are used to support structured natural language discourse during the modelling process, without limiting the interpretation potential.

Essentially, we claim that multiple iterations of a modelling process will lead to an improvement in the quality of model results only if they do not end up propagating a misalignment of goals and interpretation introduced by the involved parties.

1.3 Contributions

The research conducted in the context of this thesis makes several research contributions to the areas of scientific modelling, and information security. We note here, however, that scientific modelling is a research area derived from philosophy of science, which does not yet include all the relevant research streams focused on the practice of modelling, or, elements that might be considered as related to modelling as an engineering discipline. In such cases, methodological aspects of modelling are treated as part of the overall discipline encompassing the field of enquiry — for example, as part of economics. Therefore, we could claim multiple contributions towards different disciplines, but we limit ourselves to those with whom we have directly interacted as part of exploratory studies or practical model construction. We categorise the proposed contributions as either empirical or methodological, and illustrate them below:

1.3.1 Empirical contributions

Pragmatic characterisation of models: While conducting exploratory research to better understand the different perspectives regarding the nature of models, we have inevitably been exposed to a wide array of model conceptualisations. The common element of most of those conceptualisations was the fact that models are constructed according to some method or technique. While on its own, this is certainly not a new discovery, most modelling approaches tend to strictly categorise or describe models based on the most used technique employed: in those cases, a model is either formal, physical, executable, conceptual, and so on. However, empirical observation of any heterogeneous system showcases multiple construction techniques being employed: conceptual ones for policy and legal related components, executable ones for coding or physical construction, formal ones for optimisations, or even compositions of multiple of them. In an attempt to bridge the gap between the nature of heterogeneous systems and the models that represent them, we have constructed a way of characterising models, based on the construction technique employed for each component, in which the construction technique is conceptualised as a model quality, and a model may possess multiple such qualities, at the same time.

Study on model usage in information security: The exploratory case study in Chapter 3 provides both security researchers and practitioners with an overall view of the nature of models currently used in the security community in 2020, according to our characterisation. Additionally, it shows that models are an important tool for both the research and practice of security, that simulation based approaches might be a good starting point for constructing models with components of different types, and that some attempts at metamodelling and model interaction exist, but are not yet quantitatively significant.

Compatibility analysis of metaphors: As previously stated, one of the essential elements of our modelling approach is represented by the ability to employ the concepts of the distributed systems metaphor in a trading zone context. In order to do so, we have analysed the two metaphors in terms of entities, the interaction between them, goals, languages, methods, and practices employed, and determined them to be compatible. By doing so, we have shown that at least on a theoretical basis, the two metaphors, and inherently conceptual reasoning frameworks based on them could be used interchangeably.

Methodology testing: As with any scientific hypothesis or methodology claiming to have real-world outcomes, abstract reasoning about its validity should not be the solely reason for justifying correctness, or even utility. Therefore, we showcase the practical applicability and effectiveness of our method by putting it to the test in three different environments that are strongly related to systems' security and resilience. Although additional research studies are required to confirm the validity of the method at a more general level, the analysis of model results and the interaction with stakeholder during the modelling process have shown that the method is suitable for security modelling for a set of reasons: the constructed models helped the participants to better understand the underlying systems, were representationally aligned with the participants' perspectives about the systems, and in the case of the ransomware recovery model, even led to the discovery of a relevant, unaccounted for factor for recovery — the admin deployment policy.

1.3.2 Methodological contributions

New account of modelling for heterogeneous systems: As previously stated, one of the main contributions of this thesis is constituted by the construction of a philosophically justified account of modelling specifically targeting heterogeneous systems. As we show in Chapter 2, modern accounts of modelling in the philosophy of science and representation theory literature tend to focus either towards realist inferentialism, or pragmatic referentialism, and treat heterogeneous systems in a reductionist manner at best. However, our account attempts to integrate aspects of both perspectives by employing a sceptical inferentialist stance on the process of modelling, a pragmatic, multi-methodological stance on the nature of models, and by obtaining a set of useful model properties by using the distributed systems metaphor in a referentialist fashion, at the level of the translation zone.

Co-design cycle updates: Stemming from the above point, our modelling account has further led to a series of alterations of the co-design cycle. These include:

- Ensuring that the nature of model components is being accounted for during all the cycle stages by explicitly including the model characterisation based on means of construction.
- The integration of the distributed systems metaphor at the level of the translation zone.
- The separation of verification and validation processes.
- The explicit inclusion of personal beliefs as element influencing phenomena presentation, and subsequently the model target systems, scope, and validation.

Chapter 2

Background

At the heart of science is an essential balance between two seemingly contradictory attitudes — an openness to new ideas, no matter how bizarre or counter-intuitive they may be, and the most ruthless skeptical scrutiny of all ideas, old and new. This is how deep truths are winnowed from deep nonsense.

Carl Sagan

2.1 Introduction

This chapter attempts to provide a motivational context for the subsequent sections of this thesis and at the same time describe and justify some of the decisions that underpin the construction of our methodological approach to modelling heterogeneous systems. In doing so, it addresses the first research question illustrated in Section 1.1.

Section 2.2 summarises the main aspects of the two most influential philosophy of science traditions of modelling — namely positivism and relativism —, translates them to the level of models underlined by the two traditions, describes both advantages and disadvantages of the two methods, and reflects on the necessity of a methodological framework that incorporates aspects of both of them, in order to construct better representations of heterogeneous systems.

Section 2.3 further analyses relevant positions in the philosophy of modelling regarding the relationship between models and their targets. Specifically, it outlines the two prevalent positions on the matter — referentialism and inferentialism —

illustrates how different authors have constructed modelling accounts based on them in the form of direct referentialism and Kuorikoski inferentialism, and reflects on their suitability for representing heterogeneous systems.

Section 2.4 presents two alternatives to the strictly inferentialist and referentialist positions, in the form of Roman Frigg's DEKI account of modelling, and Bernhard Thalheim's engineering approach to conceptual modelling. It focuses on describing how both inferentialist and referentialist commitments can be found in the two accounts, reflects on why this may be a better approach to modelling heterogeneous systems, and identifies elements in the accounts that suggest the acceptance of a multi-methodological nature of models.

Section 2.5 briefly summarises the issues identified in the previously described approaches to modelling, and then formulates the commitments of the modelling account which will be further developed in the thesis.

As noted in the research paper declaration form, this chapter contains elements of the author's previously published work. More specifically, section 2.2 is an edited version of section 2 from [63]. Similarly, sections 2.3 and 2.4 are extended versions of section 2 from [152].

2.2 Positivism or relativism

There can be no denial that the construction, interpretation, and use of models are processes deeply underpinned by broader philosophical positions about the nature of knowledge and reality. The history of philosophy provides countless possible positions regarding these two notions, yet we are unsure that a complete literature survey of all of them would prove beneficial for this thesis' purpose. However, we can start our inquiry by examining two opposing traditions that have been most influential on modelling, even under current research practices: positivism and relativism. Both of these traditions can be traced back to the rationalist and empiricist schools of thought of the 16th century and have been further developed during the 19th and 20th century philosophical split between continental and analytical philosophy.

We are aware that the terms positivism and relativism are quite general and can be used to describe accounts that might differ on some aspects of epistemology or ontology, yet still follow the overall direction of the tradition in which they have been included. Because of that, we have decided to present these two traditions in a somewhat generic manner, focusing more on the common elements that have been used to define the overall traditions, rather than on the differentiating aspects showcased by positions in the same tradition. We do so based on a belief that the average modeller is not a philosopher of science, but will benefit from an ability to place models, at least on the epistemic level, on a spectrum with positivism on one end and relativism on the other. This spectrum thus operates as a conceptual heuristic that prompts reflection on the epistemic commitments embedded in modelling practices and facilitates more transparent communication of those commitments. At the same time, it functions less as a rigid classificatory scheme than as a comparative tool for articulating the types of knowledge claims that models are taken to support.

2.2.1 Philosophy of science

Positivism, in the broader sense, can be seen as a philosophic tradition characterised by a strong anti-metaphysical commitment and a focus on an external, objective reality whose observability and analysis via the scientific method serves as only foundation for knowledge discovery. Initially proposed in modern form by Auguste Comte in the 19th century and then further developed by the logical positivists of the Vienna Circle in the 20th century, this position has been extremely influential to further developments across sciences — even today, sciences with a strong adherence to the use of quantitative methods tend to be named positivist sciences —, because it explicitly attempted to define a hierarchical relationship between them — culminating with the possible reduction of all science to physics and mathematics in the logical positivist case — and eliminate all practices deemed unscientific. While both Auguste Comte's positivism and the logical positivism of the Vienna Circle share a commitment to empirical science and a rejection of metaphysics, they differ significantly in their methodological approaches. For instance, Comte's positivism is more historical and sociologically oriented, whereas the Vienna Circle's logical positivism is characterised by its rigorous logical analysis and emphasis on the verification principle — which states that a statement is meaningful only if it can be empirically verified or true by definition. These differences reflect the evolution of positivist thought from a broad, observational approach to a more precise, language-focused analysis. Examples of more modern positions that could be categorised as closer to the positivist end of the spectrum include Michael Friedman[118, 119], Thomas Uebel[292, 293], or Richard Creath[81, 82].

Relativism, on the other hand, is a philosophical position that rejects the notion of objective reality. It posits that both reality and knowledge are subjective, constructed — either personally, socially, a priori by an external entity, or in other forms — and based on interpretation. Under this assumption, the goal of scientific inquiry is no longer related to the determination of an objective universal truth but rather to the observation and interpretation of the complexities of human and social experiences in different contexts. More specifically, the general relativist position regarding metaphysics is context-dependent, with the validity and meaning of metaphysical statements being variable across social, linguistic, and cultural contexts and therefore not absolute. Similarly, the concept of verification is seen as a pluralistic and collaborative process, challenging traditional notions of objectivity and universal standards and, moreover, highlighting the importance of reflexivity, ethical responsibility, and respect for diverse ways of determining and validating knowledge. Examples of more modern positions that could be categorised as closer to the relativist end of the spectrum include Ronald Giere[131, 130], Andrew Pickering[243, 242], Martin Kusch[187, 188], Barry Barnes[24, 23], or David Bloor[39, 40].

2.2.2 Modelling

The influence of these philosophical paradigms becomes evident when considering the different approaches to modelling across scientific disciplines. For instance, in natural sciences, the positivist tradition still maintains a dominant position, leading to the development of models that seek to uncover universal laws and principles underlined by an absolute truth. Opposingly, in social sciences and humanities, relativist approaches have fostered the creation of models that explore the intricacies of human behaviour and social interactions from multiple perspectives. However, the models case study and the associated analysis in Section 3.4 illustrate that model diversity is not a property encountered only between separated disciplines.

Similarly to the above, a complete review of possible model types across disciplines is outside the scope of this thesis. Below, we describe some of the main elements of models underlined by either the positivist or relativist traditions:

- *Positivism.* Models are understood as objective and absolute representations of systems. Validation is a process that is formal, algorithmic, and focused on the accuracy of both the structure and outputs of the model. A single structural misrepresentation is enough to invalidate the model, regardless of its outputs. The overall modelling process is believed to reveal the truth if performed adequately.

- *Relativism.* Models are subjective; that is, they are just singular instantiations from a continuum of possible representations of the system. Validation is semi-formal, ‘a gradual process of building confidence in the usefulness of a model’ [21]. Such models do not attempt to reveal absolute truth but rather produce a useful model given the modeller’s goals.

While each tradition offers valuable interpretations to modellers, we posit that neither alone fully addresses the representational needs of heterogeneous systems generally, or heterogeneous systems’ security particularly. Some of the reasons for this derive from some quite basic problems with, or objections to, each of these views.

Problems with positivism: Theoretically, positivism has been historically struggling to overcome the epistemic and methodological implications of Kuhn’s description of the acceptance of scientific theories and Popper’s theory of falsification. Both the acceptance of Kuhn’s thesis — stating that scientific progress is not achieved through the accumulation of knowledge but rather subjective community paradigm shifts — and Popper's view that scientific advancement can only be achieved through falsification rather than proving absolute truths greatly reduce the focus on truth that positivism held of a highest importance. Additionally, some of its practical caveats come from the difficulty of working with knowledge elements that have not been fully proven, completely accepted by the research community, or that are yet unquantifiable because the underpinning theoretical work is not mature enough.

In the specific case of security models, the most common such elements are related to the uncertainty introduced by human actors — either attackers or non-malicious actors — or the discovery of new technical attack vectors.

Also, positivism requires an extremely powerful validation process, which is not always possible in the case of complex cyber-physical systems. Particularly, the structural representation criterion can lead to the invalidation of models that are producing seemingly viable results, which can be considered a quality upper bound, but certain phenomena introduced by humans do not fit this type of approach because they lack the theoretical certainty.

In the best case scenario, a model close to the positivist end of the spectrum can be used in well defined and seemingly stable conditions, for example, when used to determine the trajectory of a rocket given the precise atmospheric conditions, but in today's heterogeneous systems, this is rarely the case.

Problems with relativism: Under a different set of circumstances, relativist perspectives assume a prominent position within contemporary philosophy of science, insofar as notions such as the subjectivity of knowledge and truth, as well as the social construction of reality, are widely acknowledged, though they remain the subject of considerable debate. However, this interpretative way of viewing reality presents practical challenges when applied in isolation to model construction, particularly in contexts requiring rapid decision-making based on diverse sub-models.

First of all a heterogeneous systems model is composed of a high number of sub-models, each with their own primary goal, resources, processes, etc. To be able to obtain the relativistic notion of knowledge about those sub-systems, lengthy processes of data collection — interviews, debates — must be carried out for the better understanding of the reality as seen by all the parts involved in the system under study. Although methodologically this might not be considered an actual problem, we must consider the fact that models are used today for tackling real world issues in reduced time-frames. The early usage of predictive models at the start of the Covid pandemic can be seen as a relevant example.

Secondly, while openness is a key strength of relativist approaches, it may also introduce challenges in contexts requiring unified representations due to operational constraints. If each sub-model is constructed with a different understanding of reality (the ones of the actors involved in it) — this can be seen in studies about the formal and alternative power structure of organisations — their integration becomes a serious issue and reasoning about them may be too complex, or risk oversimplification. Albeit not directly concerned with models constructed based on a relativist philosophy of science, works such as [60], [64] or [59] provide a practical approach of this issue by using interfaces to specify the desired type of output moving between sub-models without trying to alter the underlying notions of knowledge that led to the production of that output.

Thirdly, such models may present stakeholders with difficulties in practical application, particularly when trying to determine why certain decisions have been taken if the assumptions included in the model are not clearly specified. Applying a simple root cause analysis method on a decision taken by such a model might lead either to a return to the real world simulated actors — if they are humans — for further explanations or to uncertainty.

On one hand, although methodologically sound, returning to the simulated actors is a lengthy process that can end up greatly delaying model implementation and should only be used in cases where there is evidence of a lack of understanding.

On the other hand, a model whose decisions cannot be certainly explained will hardly be accepted by decision makers who might prefer to use their own understanding of the system because it manifests a smaller perceived degree of uncertainty. Studying which method would outperform the other is not the goal of this thesis.

2.2.3 Reflections

As seen above, neither positivism nor relativism alone offers a suitable methodology for modelling the dynamic systems of today. In a certain sense, the former approach places models under a set of too-powerful constraints, whereas the latter presents difficulties in choosing a set of constraints or quality measures usable in practice.

The positivist perspective can provide speed and trace-ability by structure and method where the available system knowledge is suitable: the main phenomena to be included in the model have been previously studied by the scientific community and an accepted theory has been formulated, and the phenomena interpretation can be translated to quantifiable data types.

On the other hand, the relativist perspective provides better descriptive power and increases the overall comprehensibility of the model.

Therefore, we believe that the need for a modelling framework that balances both views is justifiable, especially if the overall goal of such a framework is model integration.

2.3 Referentialism or inferentialism

Something currently regarded as common knowledge in the conceptual modelling literature is that any modelling theory must be able to provide an answer to the following two questions: what is the relationship between model and target and, how one can produce accurate inferences about the target using the model. By target, we imply the heterogeneous system or ecosystem that the model is representing. Naturally, a third question can be raised: in which order should the two previous questions be asked and answered? This can be seen as a generative point for different philosophical positions upon which modelling research directions have developed. Based on this interaction between model and object of modelling, we can categorise approaches to modelling into two main accounts: referentialist and inferentialist.

2.3.1 Referentialism

Referentialist views of modelling place utmost importance on defining the relationship between model and target from the very beginning. The relationship is a function mapping the structural elements of the model to the structural components of the target and serves as justifiable explanation for the informational content of the model. Only after this initial step can the relationship be used as means of explaining inferences between model and target and evaluating their accuracy. Nonetheless, the above description can be seen as open to various interpretations, allowing for accounts differing at the metaphysic, epistemic, and ontological levels.

2.3.1.1 Direct referentialism

The direct referentialist positions of Saul Kripke [178, 179, 180], David Kaplan [160, 161, 162], Hilary Putnam [247], and Keith Donnellan [98] are theories of language claiming that the meaning of words and expressions lies in what they point out in the world. Because of that, a relationship between world and language must exist, and the attempts of the authors at defining that relationship can be interpreted as conceptual models. They have been constructed as objective, realist focused reactions against the descriptivism and subjectivism of Frege [115, 116], who was proposing that the meanings of individual words are solely determined by their contribution to the thoughts conveyed within the sentences in which those words are used.

While these philosophers do not explicitly aim to provide a normative framework for modelling, their theories of reference and their interpretations of the meaning of reference can offer modellers a conceptual toolset for understanding the model-target relationship. This does not imply the necessity of considering such a relationship as the most important one — a balanced position, at least from a practical modelling perspective, would accept the existence of the model-target relationship but focus more on the outcomes produced by the interaction of sub-model components and their relationship with model goals. Nonetheless, in order to better understand these positions without delving too deeply into specific aspects of philosophy of language, we have chosen to illustrate their specific commitments or implications towards metaphysics, epistemology, and ontology. We do so below:

Saul Kripke: The understanding of Kripke's stance on metaphysics is dependent on two notions: possible words and rigid designators. In the author's formulation, 'Possible worlds are total ways the world might have been, or states or histories of the entire world.' [180], and 'we think of the actual world as just one of the possible worlds, one among many.' [180]. In this context, rigid designators are simply terms that refer to the same object in all possible worlds where that object exists. Starting from these two notions, we can view Kripke's position as related to a form of essentialism grounded in a realist ontology: objects exists independently of human cognition, and their identity is a set of necessarily true rigid descriptors across all possible worlds, which must be discovered rather than constructed.

Furthermore, the author's concept of a causal theory of reference epistemically implies that references are maintained through social and linguistic practices rather than individual descriptive knowledge, because correct identification of a referent requires knowing the historical chain of communication that led to it.

David Kaplan: Kaplan's metaphysical position is strongly related to the distinction between the character and content of a linguistic expression, in relation to its context. As explained in [161], the character is a rule that determines its referent based on context, while the content is the actual referent in a specific context. In other words, the content refers to the contribution made by an individual term as part of an expression, whereas the character specifies, in a given context, what the content of a particular expression will be. This carries the implication that meaning and reference are context-sensitive and dependent on the linguistic rules governing communication rather than solely on fixed, descriptive properties of objects. Therefore, knowledge of a reference requires an understanding of the situational context in which the term is employed. Yet, since the notion of context is not seen as subjective, we can view Kaplan's stance as a form of contextual realism, supported by an ontology that includes both abstract entities — the rules of the process of reference — and concrete entities refereed to in specific contexts.

Hilary Putnam: Succinctly, Putnam's philosophical stance has been described using two terms: internal realism and semantic externalism. The first one carries the implication that scientific theories are not purely descriptive of reality but play an active part in shaping the understanding of the world — even if we do not have direct access to an objective reality independent of our conceptual frameworks, scientific theories are still objective because they reflect a reality that is mind-independent. The second one indicates that the meanings of terms are not solely determined by internal mental states or definitions but are partly determined by external, causal relations with the world. In this setting, terms can still objectively refer to real entities in the world, and the process of empirical inquiry not a priori reasoning, even if mediated, is used for grounding that reference. The author epistemically implies the importance of communal linguistic practices and scientific expertise in grounding references, but not the necessity of discovering the true nature of an object to perform correct references to it, via his concept of 'semantic division of labor' [247]. Therefore, we can classify Putnam's position as a form of scientific realism.

Keith Donnellan: Donnellan's position can be characterised by the inclusion of personal intentions in relation with the use of linguistic expressions. The author's distinction between referential and attributive uses of descriptions impacts how we understand the metaphysical basis of reference. As explained in [98], 'A speaker who uses a definite description attributively in an assertion states something about whoever or whatever is the so-and-so. A speaker who uses a definite description referentially in an assertion, on the other hand, uses the description to enable his audience to pick out whom or what he is talking about and states something about that person or thing. In the first case the definite description might be said to occur essentially, for the speaker wishes to assert something about whatever or whoever fits that description; but in the referential use the definite description is merely one tool for doing a certain job-calling attention to a person or thing and in general any other device for doing the same job, another description or a name, would do as well. In the attributive use, the attribute of being the so-and-so is all important, while it is not in the referential use.' Therefore, in the referential use, the connection between a term and its referent depends on the speaker's intention and context. Similarly to Putnam, this implies a balance between the influence of conceptual frameworks on the observer and objectivity, while at the same time maintaining the idea of objective reality and direct reference. Yet, both epistemically and ontologically, the position is more complex, because the interpretation of intentions plays a fundamental part in understanding references, which cannot be captured by purely descriptive content alone.

2.3.1.2 Reflections

After briefly describing the above perspectives, we now turn our attention to their relationship with the positivist-relativist debate and the implications for modelling heterogeneous systems.

Firstly, we note that all four accounts seem to acknowledge the necessity of a balanced position between relativism and positivism, but manifest it variably. For example, the positions of Donellan and Kaplan can be viewed as closer to relativism because of their stance on intentions, context, and the importance of linguistic rules governing communication rather than fixed descriptive properties. Opposingly, Kripke can be seen as the most positivist of the four, given his focus on fixed essential properties of objects across possible worlds. The position of Putnam is the most balanced one, because it accepts the influence of conceptual frameworks on understanding and generating knowledge but at the same time posits that scientific theories, even if mediated, still describe an objective reality.

Secondly, given their anti-subjectivist motivations, all of the accounts are committed to a form of realist ontology. However, the authors' attempts at including relativist notions while maintaining an objective positivist baseline still face serious challenges when considered in a modelling heterogeneous systems context.

For example, in the case of Kripke, the idea that objects have inherent properties that are necessary across all possible worlds may require adaptation to suit the complexity of heterogeneous systems. The dynamic nature and emergent properties of such systems can be hard to quantify and reduce to a set of essential characteristics, especially if one attempts to maintain an objective stance regarding them. Furthermore, attempting to trace the historical communication chains in complex systems is almost always impractical due to a lack of data collection in that sense.

When considering Donellan's position, most challenges are related to the notions of context and intentions. Accurate determinations of references require knowledge of both, which is an even harder problem than determining the historical communication chain because it requires knowledge of both subjective and objective aspects involved in the construction of the reference. Additionally, modelling heterogeneous systems under this view requires a conceptual framework supported by implemented tools that can represent both the referential and attributive use of descriptions, possibly resulting in over-complicated models that are harder to understand by the average user. Similar concerns can be raised with respect to Kaplan's position, but this time because of the notion of character: if character can essentially be seen as a rule for accurately determining a reference in a specific context, then heterogeneous systems — which inherently contain multiple contexts — representation require an extensive set of rules for determining all the references in all the contexts. This once again leads to complex models that are more concerned with how to represent the notion of reference, rather than their actual targets.

In the case of Putnam, the main issue is related to consistency. If scientific theories actively shape personal understanding, then there exists the possibility that a heterogeneous model would require integration of multiple opposing such theories for the understanding of different aspects of the same phenomenon. Especially under positivism, this would result in an inconsistent model. However, the idea that terms meaning and inherently reference are partially determined by causal relationships in the world is more practical than simply relating to context or character: although a causal relationship is manifested in a context, the focus is more on the practical outcome of that relationship, which can be determined without a complete understanding of that context.

Although not focused on direct reference, similar theoretical formulations of the relationship between model, or in the previous cases conceptual model of language, and target can be found in structuralist approaches such as [286, 275, 298, 85, 192]. We do not explicitly analyse them in this thesis, but we acknowledge that their conceptualisation of scientific theories as collections of models rather than propositions, and therefore the understanding of reference as a mathematical property such as isomorphism, homomorphism or partial isomorphism represents a better alternative for modelling heterogeneous systems than a purely direct referentialist approach.

2.3.2 Inferentialism

Inferentialist approaches focus more on empirical experimentation with phenomena as basis for model conceptualisation. Through such experimentation, an initial low level of detail model is constructed and further refined along a continuous process of inference. Only once a certain level of inference accuracy has been reached, the relationship between model and target can be determined. Therefore, rather than directly representing real-world entities, models are viewed as tools for generating inferences about such entities, and the quality of inferences is determined by translation to the quality of predictions about the behaviour of such entities. In order to better understand the position, we briefly present the inferentialist account of Jaakko Kuorikoski [186, 185, 184].

2.3.2.1 Kuorikoski inferentialism

We begin our exploration of Kuorikoski's account of modelling with some general considerations regarding his stance on metaphysics, epistemology and ontology.

Firstly, Kuorikoski's perspective is characterised by a mechanistic metaphysics centred on the role of models in uncovering and understanding the causal mechanisms underlying phenomena. This is complemented by a realist perception of such mechanisms: mechanisms are treated not as abstract entities, but rather as real entities that can be discovered, so models representing them are not only descriptive in nature — even if they can include abstract representations of mechanisms —, but provide real-world insights. Interestingly, this does not contradict Kuorikoski's commitment towards a pluralist ontology: the concept of mechanisms is used to structure and ground different perspectives into a realist, outcomes focused baseline.

Secondly, as Kuorikoski puts it, 'We seem to learn something genuinely new about the world by manipulating an artificial surrogate system and then "observing" what the end result is.' [185], but the seeming aspect here is relevant. The model is not viewed as producing a similar type of knowledge as empirical experimentation with phenomena, but rather as a reasoning helping tool whose epistemic value should be analysed in terms of the scope and reliability of inferences. This is achieved by keeping record and managing the doxastic commitments and, inherently, the assumptions and bias introduced by the modellers explicitly, showcasing the author's adherence to the inferentialist stance of Robert Brandom [44] in philosophy of language. Furthermore, the reliability of inferences is not considered in terms of representation alone, but rather as dependent on achieving modelling goals such as understanding, explaining, predicting, or solving practical problems generated by phenomena.

Thirdly, the account is committed to the principle of inference to the best explanation in relation with scientific practice. This essentially means that scientific practice can be viewed as a continuous process of analysing and comparing models in an attempt to obtain more accurate and more plausible predictions about real-world entities.

2.3.2.2 Reflections

The application of this approach at the level of heterogeneous systems still encounters various challenges, some of which have also been presented in previous sections of this chapter.

The first challenge is related to consistency. Even if theoretically, the reduction to the representation of causal mechanisms seems to avoid the overall issue of model consistency, the practical construction of such models remains complicated if the causal mechanisms in case have not been studied before. If that is the case, the decomposition or reduction of a heterogeneous system to a set of causal mechanisms might be inaccurate, because the boundaries of each causal mechanism must also be determined actively during the process of inference.

Secondly, we tend to believe that models constructed under this account might suffer from scalability issues. Although the author seems open to the idea of modular, compositional modelling, he does not explicitly provide a method for ensuring that constructed models can be indeed composed or integrated.

Thirdly, and this is an issue with all realist modelling accounts on an empirical basis, models are extremely dependent or sensitive to available data. We do not claim that, under some other modelling framework, this would not be the case. However, we question whether a strict realist commitment may limit modelling flexibility in cases where empirical data is sparse or inaccessible. In the case of models constructed after an empirically observable phenomenon has occurred, but no data was collected, or the access to data or the possibility to collect data at a later date is denied or impossible, is it even possible to construct models? The overall inferentialist position would suggest that it is possible by inferring from the beliefs of the participants that observed or were directly influenced by the phenomenon, but the realist aspects would contradict this on the grounds that personal belief cannot stand on its own as basis for mechanistic inference. If that is the case, should we infer that model targets must always be empirical phenomena or that a scientific representation of a deeply subjective phenomenon such as the interpretation of the meaning of a sentence for a specific user is impossible?

The consideration of the above described challenges does not determine us to make a claim of unsuitability for representing heterogeneous systems. Our concerns are mostly related to the lack of an explicit inclusion of heterogeneous systems as model targets and to the ability to construct models that can produce accurate inferences about the world in the absence of enough data empirically obtained about the world, from the world. In future chapters of this thesis, we will attempt to overcome these issues at the level of our own modelling account by relaxing the realist commitment and extend the focus on mechanisms to a more encompassing abstraction, namely the distributed systems metaphor.

Lastly, we must note that Kuorikoski's inferentialist position is not singular. Different generally inferential perspectives include [282, 78, 283, 90], with [54, 29, 30] specifically focused on simulation modelling. Interestingly, most of these accounts have pronounced deflationist notes, including Kuorikoski's, in the sense that no deeper or unexplainable meaning is placed on the model-target relationship. For a longer discussion about simulations, although not specifically focused on inferentialism, see [99].

2.4 Between referentialism and inferentialism

Even if referentialism and inferentialism are two important traditions in philosophy of modelling, it is not always possible to determine a complete adherence of modelling accounts to only one of them. Sometimes, modelling accounts simply import elements of both. For instance, Hilary Putnam and Keith Donnellan are defacto classified as referentialists, regardless of their relativists tendencies towards contexts and intentions, but their positions can hardly be considered balanced. In this subsection, we describe two other accounts which include elements of both referentialism and inferentialism: Roman Frigg's DEKI account [122, 224] — which does so explicitly — and Bernhard Thalheim's engineering approach to conceptual modelling [290] — which does so implicitly. Although not explicitly developed with heterogeneous systems in mind — and in the case of Frigg perhaps not even for practical model construction in an organisational context, but rather centred on epistemic clarity — these accounts offer valuable insights and directions to be included in a modelling methodology focused on integration.

2.4.1 Roman Frigg's DEKI account

Roman Frigg's DEKI account of modelling, as described in [122, 224] explicitly integrates aspects of both referentialism and inferentialism, in an attempt to emphasise the complex relationship between models and their target systems. It does so via the four methodological steps that also serve as the account's title:

> **Denotation:** The first step requires the establishment of the link between the model and its target. In other words, the determination of the model's referent.
>
> **Exemplification:** The second step is centred around the selection of relevant features of the target that the model will include. Therefore, the model is never a complete representation of its target.
>
> **Keying-up:** The third step can be described as the translation of model features back to features of the target, via interpretation.
>
> **Imputation:** In the last step, the keyed-up pairs of features are used to attribute insights from the model level to the target level.

The distinction between referentialist and inferentialist commitments at the level of the above steps is quite clear: denotation and imputation align with the referentialist idea that models aim to accurately represent and refer to real-world entities, whereas exemplification and keying-up are inherently inferential processes in opposing directions — from real-world to model and from model to real-world. Additionally, the author's metaphysical stance is pragmatic to some extent, viewing models as tools for inference rather than literal representations of reality.

Another distinction can be observed with regard to the objective and subjective elements of the account. On one hand, all the DEKI steps carry the implication of the existence of an objective reality, supported by a scientific realist ontology that differentiates between models as abstract entities and model targets as real-world entities. On the other hand, the determination of relevant model features and the evaluation of model success are seen as influenced by the subjective epistemic goals and values of the scientific community — here, we diverge from the usual claim of scientific utility as derived from objectivity — and, therefore by the historical, cultural, and social context of scientific practices. This suggests that while models aim to capture objective reality, their development and use are influenced by subjective factors, which is similar to both Putnam and Donnellan's positions on this matter but extended to the level of models, not language constructs.

2.4.2 Thalheim's approach to engineering conceptual models

Bernhard Thalheim's account of modelling, as described in [290], can be seen as an attempt at forming a theoretical basis for conceptual modelling concentrating on engineering and design principles from a practitioner's perspective. Without a clear adherence to either inferentialist or referentialist directions, the author describes the act of modelling as a tripartite activity decomposed into modelling language constructs, application domain gathering, and engineering.

Modelling language constructs: In Thalheim's framework, the modelled language constructs represent the most basic elements. 'Modelling language constructs are applied during conceptual modelling. Their syntactics, semantics and pragmatics must be well understood' [290]. This understanding is achieved via a well defined syntax and semantics, and supported by meta-models and interoperability standards.

Application domain gathering: This involves the collection and analysis of domain related information, including the understanding of requirements, context, and specific needs. In the author's own words, 'Application domain gathering allows to understand the problems to be solved, the opportunities of solutions for a system, and the requirements and architecture that might be prescribed for the solution that has been chosen' [290]. It is explicitly noted that the understanding of the domain requires participation of multiple parties such as stakeholders, domain experts, and users. Because of that, modelling can be seen as an iterative process in which the participants' knowledge must be integrated with the help of co-design principles [268] and meta-models.

Engineering: The last dimension describes the practical application of models to the design, implementation, and maintenance of systems. Explicitly, 'Modelling is also an engineering activity with engineering steps and engineering results. It is thus engineering [...] Engineering is oriented towards encapsulation of experiences with design problems pared down to a manageable scale' [290]. Briefly, the main idea is that generally accepted en-

> gineering principles such as modularity, hierarchical ordering and layering, or more domain specific ones — typically drawn from software engineering, under the assumption that software engineering is a form of conceptual modelling — such as Liskov-substitution-principle, the open-closed-principle, the dependency-inversion-principle, the interface-segregation-principle, and so on, should be extended and generalised to conceptual modelling.

Similarly to Frigg, the integration of inferentialist and referentialist aspects is also present in Thalheim's account. Because of the importance placed on the interaction with the model domain and the definition of model properties such as utility, we tend to believe the author describes an inferentialist practice of model construction. This is also supported by a metaphysical stance committed to the existence of structured, ordered realities that models can help uncover.

Furthermore, the methods for modelling language constructs seem to imply a concept of denotation similar to Frigg's [121], used over both empirical and abstract realities. The difference is at the ontological level: if Frigg's ontological commitments are realist, and the nature of the represented reality is objective, Thalheim's are pragmatic, with models representing a mix of objective structures and subjective contextual influences. Interestingly, Thalheim also hints at the idea of multi-methodology when describing the notion of theory of conceptual modelling, but chooses to maintain the resulting theory and model types separate.

> 'Theories of conceptual modelling must step beyond axiomatic and analytical theories. They must also be operational and 'genetic'. Theories of conceptual modelling can be developed in the frameworks of logical empiricism, of context theories ('context of use', 'language game'), and of constructivism. The first framework supports to define purposes of conceptual modelling, to emphasise threats that should be handled with the help of models, to select appropriate modelling languages and methods, to reason on the quality of the model depending on the purpose of the model, to select measures for the quality of models, and to guide the process of modelling. It may use development experi-

> ments, case studies, evaluation surveys, and implementation studies. The second framework relates models to the application domain, to the stakeholders participating in the development process, to the aspects reflected within a model, and to the resources provided either by the system and by the knowledge from the application domain. It requires to base conceptual modelling on application domain theories. The last framework provides a basis for a general structure by a language of constructs that can be applied for the development of a model, a set of constructors that can be applied to combine models into a new model, and a number of quality properties for characterisation of usage of certain constructs.'[290].

The temporal aspect of models is also considered explicitly. When describing necessary properties for models in the future, [290] includes the ability to perform an adjustable selection of principles depending on modelling goals, model suites with explicit model association, the explicit treatment of model value and adequate representation variants of models.

In the following sections, we shall attempt to answer some of the questions that arise from these requirements. Nonetheless, we acknowledge that the modelling suites and the co-design methodology presented in [268, 267] and exemplified in [266] are attempts worth undertaking in understanding and managing model heterogeneity. A similar approach can be seen in the field of security, in the work of Demjaha [94, 93], with a particular focus on the human oriented factors involved in co-designing security culture.

2.4.3 Reflections

In the context of the above described modelling approaches, it is relatively easy to observe that the acceptance of model instrumentality, explicit inclusion of model goals at the level of model conceptualisation, and desire for future integration lead to the construction of positions that blend both referentialist and inferentialist elements.

Generally, the referentialist aspects are manifested at the level of the definition and validation of the relationship between model and target, whereas the inferentialist ones are employed as means of managing the overall modelling process via iterative calibration. From a modelling perspective, we suggest that balanced positions attempting to include aspects of both paradigms may greatly benefit from explicitly addressing consistency concerns via the processes of the translation zone from Section 5.4. Philosophically, Kuorikoski — following Brandom [44] — describes such a process as a 'doxastic scorekeeping device' [184]. Furthermore, we claim that the acceptance of a multi-methodological nature of models, and the explicit account of the manifestation of that nature in the model during its life-cycle can greatly reduce consistency concerns. Even without a clear adherence from the authors, aspects of both these notions seem to already exist in both Frigg and Thalheim's accounts.

For example, in the case of DEKI, the same abstract type of denotation procedure is used regardless of model target, and imputations from multiple models can be used to draw insights about the same target. Thalheim treats this similarly, at the level of modelling language constructs, and also acknowledges that modelling theories can be constructed over different philosophic positions, and that each of the resulting models are instrumentally valuable. Therefore, both accounts tacitly accept multi-methodology but do not seem to provide ways for its management, except by relating it to objective reality.

With respect to the manifestation of the multi-methodological nature, the situation is more complex. For instance, the DEKI account does not clearly specify how imputation works over multiple sub-models with different underlying assumptions, at the level of the same model. Thalheim, however, chooses to maintain a very strict separation between sub-models, but provides integration via the notion of model suites [289] — which require an explicit collaboration schema between models, controllers to ensure consistency, and an application schema for maintenance and model evolution. The notion of model suites is therefore very powerful, but implies the need to first construct the sub-models in silo, then construct all the necessary elements for the model suite, and only then obtain the final model via a form of composition. Pragmatically, we agree with this, but we ask ourselves if the elements of model suites should be included earlier in the modelling process. In other words, if our primary focus is future integration, shouldn't models be always viewed as model suites?

2.5 Conclusion

This chapter aimed to provide a motivational background for the modelling methodology which will be further developed in this thesis. In order to do so, it presented several modelling traditions and accounts based on them which differed at the level of metaphysics, epistemology, ontology, role of models, relationship with objective or subjective reality, and so on. Then, it reasoned about their suitability for representing heterogeneous systems.

In this context, four main observations can be drawn:

Realism and objectivity: Most of the described modelling accounts manifest commitments to a form of realist ontology and objective reality but balanced with subjective notions such as the influence of conceptual frameworks, context, or intentions.

Multi-methodology: Hints at a multi-methodological practice of model construction exist, but they are constrained by the realist commitment and not explicitly included in the methodologies.

Heterogeneity: There is no explicit focus on heterogeneous systems reflected in the abstractions employed during model construction. Furthermore, the overall notion of model is not viewed as heterogeneous, resulting in a strict separation between model types.

Integration: The desire for future model integration exists, but again does not seem to influence the conceptual frameworks or the abstractions involved.

In the next chapters of this thesis, we will focus on addressing some of the issues related to these observations. In order to prioritise the integration aspect, we argue that the acceptance of model heterogeneity is necessary. Therefore, our metaphysical commitment will not be reduced to causal mechanisms or separate model types, but will rather be centred on the notion of model as collection of sub-models — this conceptualisation is close to Thalheim's notion of model suites, with the main difference being that integration should be included in the modelling process from the very start, not only after the construction of the sub-models. All models can be viewed as composed of sub-models if the level of detail is low enough.

However, accepting model heterogeneity bears the implication of accepting multi-methodology, because the heterogeneous sub-components are constructed using techniques specific to their domain of inquiry. Mingers[214] argues that multi-methodological approaches should be underlined by a philosophical commitment to critical realism, which is a slightly more open position than the different versions of scientific realism we have encountered in the presented accounts. While we agree that a consistent modelling account focused on heterogeneity and integration can be constructed under critical realism, our position is more sceptical: the existence of an objective reality is not a necessary criterion for models to achieve their goals because the process of establishing the link between the observed phenomena in their domain and their targets is a decision making process involving multiple participants, and this process cannot be deemed as objective due to the involvement of personal beliefs. Furthermore, once the conceptual model of the target is established, it is then used for the explicit construction of heterogeneous model components, and the specific configuration or manifestation of the components is, again, directed by a decision making process involving the model goals, deployment environment, and so on.

We must not forget that the realist and objectivity commitments were underlined by a set of abstract structures and procedures that provided useful properties, such as compositionality. These should not be abandoned, but we argue they can be obtained without the commitment towards scientific realism. For example, in the formulation by Partee [236], compositionality requires that 'The meaning of a compound expression is a function of the meanings of its parts and of the way they are syntactically combined.', which does not imply any commitment to realism or objectivity. Therefore, we argue that we can obtain similar properties at the level of models, with the help of a well chosen abstraction — namely the distributed systems metaphor — employed as structuring and guiding element for the decision making processes of the co-design cycle.

Finally, we underline the practical significance of this chapter for modellers, as it establishes the philosophical and methodological foundations that will further guide the design of integrated modelling approaches. By critically examining the strengths and limitations of positivist and relativist traditions, as well as referentialist and inferentialist accounts, this chapter equips modellers with a conceptual toolkit for navigating the epistemic and ontological challenges of representing heterogeneous systems. The discussion highlights why neither extreme positivism nor unrestricted relativism — viewed as the extremes on a modelling perspectives spectrum — adequately support the integration and scalability required in real-world modelling contexts. Instead, it motivates a balanced, multi-methodological stance that preserves the rigour of formal structures while accommodating the flexibility needed for diverse modelling goals. These insights are not merely theoretical: they inform practical decisions about model design, validation strategies, and integration mechanisms, laying the groundwork for the frameworks and tools to be introduced in subsequent chapters.

Chapter 3

Modelling Heterogeneous Systems

There are good reasons to suspect that heterogeneity (i.e. variability within any given set of samples) is an essential characteristic of organic life. This differs widely from the traditional view that heterogeneity is only a nuisance that is to be circumvented or otherwise eliminated. Hence, controversies and mutual misunderstandings of two groups adhering to these diametrically opposed viewpoints are just about inevitable.

W.M. Elsasser [104]

3.1 Introduction

The primary aims of this chapter are to describe the main representation targets of our modelling methodology — namely heterogeneous systems —, introduce the conceptual aspects of our modelling account, and illustrate their utilisation at the level of a security models case study which attempts to provide a general view of the nature of models used in the security research community in 2020. The descriptions, reflections, and analysis provided address the second research question from Section 1.1.

Section 3.2 is centred on the notion of heterogeneous system. In order to construct a general definition of heterogeneous system, it firstly illustrates formulations of the notions of system and heterogeneity across different sciences such as biology, chemistry, physics, computer science, economics, and so on. Then, it draws a series of observations and uses them to extend the notions, and then combines them to produce the definition of heterogeneous system.

Section 3.3 presents the main conceptual aspects of our modelling account. Specifically, it includes our definition of model, reflections on the implications of the definition, and an illustrative, comparative example of two population ecology models with the same target but very different goals and manifestations, allowed by the definition. Furthermore, it reasons about the necessity of conceptualising models as heterogeneous, and describes the multi-methodological implications of this necessity at the level of the modelling account's metaphysical, epistemic, and ontological commitments. Lastly, it introduces our model characterisation based on goals and means of construction under the name 'Triangle Framework', and lists some of it benefits beyond categorisation.

Section 3.4 illustrates the use of our previously introduced model conceptualisation and characterisation at the level of an exploratory security models case study. This includes a description of the employed methodology, presentation and analysis of results, and reflections on the implications towards model validation.

As noted in the research paper declaration form, this chapter contains elements of the author's previously published work. More specifically, Section 3.3 is an edited version of Section 3 from [152]. Similarly, Section 3.4 is an extended version of Sections 5 and 6 from [63].

3.2 Heterogeneous Systems

As stated in previous sections, the primary targets of the modelling methodology and account presented in this work are heterogeneous systems. Therefore, it is only natural that before describing the actual methodology or account in detail, we must first be able to define what a heterogeneous system is. Similarly, to do that, we must first account for the notions of system and heterogeneity.

3.2.1 Definitions

3.2.1.1 Systems

Inspired by [141] and [273], we define the notion of system as follows:

A system is a collection of structured and interrelated entities which achieves a set of goals including, but not limited to maintaining the identity continuity of entities, relationships, and whole, and reducing local entropy.

The definition can be further analysed in terms of what the system is and what it must do.

Firstly, a system is, at the highest level of detail or empiric observational accuracy, a collection of entities. This collection must manifest certain properties at the level of the constituent entities: structure and interrelation. The reasons for this are relatively simple. The lack of structure and inherently organisation would make any random collection of entities a system, which is not exactly a very useful notion. Similarly, if no relationship is manifested between entities, then there is no necessity for a separate notion of system to exist — in such a case, the entities are simply entities.

For example, the notion of structured entities implies the existence of a set of coordinating strategies used for organising the constituent entities and their relationships in ways that improve the ability of the system to maintain identity continuity, achieve its goals, and reduce its local entropy. An example of such a strategy would be functional division of entities, according to which entities with similar functions would be associated to form sub-systems rather than being distributed to different sub-systems.

The interrelation aspect simply states that entities must be able to manifest relationships towards one another. Traditionally, the nature of these relationships has been studied, and different types include hierarchical relationships, causal relationships, functional relationships, feedback relationships, dependency relationships, and many more. However, in order to maintain a degree of generality, we will conceptualise them as influence relationships. This means that a direct or indirect interaction between entities leads to changes in the structure, instantiation — or manifestation of structure —, behaviour, or outputs of either entities involved.

Furthermore, in order for a collection of entities to be considered a system, it must achieve a set of goals — or more abstractly, to justify its own existence as a whole rather than separately. Perhaps the most relevant of them is related to maintaining identity continuity: conceptualising the collection of entities as a system means constructing an identity for that system, and for that to be possible, the entities, relationships, and overall collection must remain recognisable over time. Theoretically, this is achieved by reducing the level of entropy manifested by the entities. Practically, the organisation obtained via the system's structure actively reduces the level of entropy if maintained over time.

As it can be seen, our definition of system is fairly general and does not attempt to separate systems into different types. However, it must be able to account for the different conceptualisation attempts undertaken at the level of multiple scientific disciplines. We illustrate this below by firstly showcasing some of the formulations of the concept of system — which have been inherently associated with different types of systems — and then discussing their relationship with our own proposed definition.

> **Biology:** 'A biological system is a complex network which connects several biologically relevant entities. Biological organization spans several scales and are determined based on different structures, depending on what the system is.' [220]
>
> **Chemistry:** 'Chemical systems are defined as chemical mixtures comprising a network or set of interacting molecules. That is, system-dependent behaviour and the system processes cannot be ascribed to any of the components acting in isolation. For instance, the catalysis by a metal complex in a bulk medium is inherently dependent on the nature of the chemicals (catalyst and substrates). However, if the catalysis is only possible in the presence of a third substance, not per-se involved in the catalytic process, but nevertheless necessary for it because it acts to organize the reactants, then one observes a chemical system.' [165]
>
> **Physics:** 'A system is a portion of the universe that has been chosen for

studying the changes that take place within it in response to varying conditions. A system may be complex, such as a planet, or relatively simple, as the liquid within a glass. Those portions of a system that are physically distinct and mechanically separable from other portions of the system are called phases'. [103]

Computer science: 'A structured set of hardware, software, data, and procedures that interact to perform specific tasks, process information, or provide services in a controlled and coordinated manner.' [246]

Information systems: 'An information system is a formal, socio-technical, organizational system designed to collect, process, store, and distribute information.' [241]

Economics: 'An economic system is a coordinated set of formal and informal institutions. Among the former are such variables as economically important laws and codified rights and duties and their enforcement, the type and features of economic actors that exist, prevail, or are allowed to exist in a given context and their internal structure, normal relations among the suppliers of resources, codified practices and provisions, the nature, role, instruments, and goals of government. Economically important informal institutions are, for example, the role of the family as provider of capital and entrepreneurship, work and payment habits, consumption habits, tax morale and ethical standards, the role of ideology, religion and beliefs — except in dictatorship and confessional societies, in which these latter variables are formal institutions.' [87]

Sociology: 'Reduced to the simplest possible terms, then, a social system consists in a plurality of individual actors interacting with each other in a situation which has at least a physical or environmental aspect, actors who are motivated in terms of a tendency to the "optimization of gratification" and whose relation to their situations, including each other, is defined and mediated in terms of a system of culturally structured and shared symbols.' [235]

> **Generalised systems theory:** 'Because any real "machine" has an infinity of variables, from which different observers (with different aims) may reasonably make an infinity of different selections, there must first be given an observer (or experimenter); a system is then defined as any set of variables that he selects from those available on the real "machine". It is thus a list, nominated by the observer, and is quite different in nature from the real "machine".' [14]

Analysing the above described definitions of systems leads us to four separate observations:

Firstly, all the definitions include the idea of a collection of inter-related entities. The entities are usually explicitly declared — biological entities, mixtures, hardware, software, individual actors, formal institutions, or variables — whereas the relationships are not — they are implied by concepts such as networks, organisation, system-dependent behaviour, or coordination.

Secondly, the nature of both entities and relationships is not unitary across all definitions. For example, the biological system definition clearly specifies that the entities are biological in nature, whereas the information systems definition includes formal, socio-technical, and organisational entities. In the case of relationships, this is more difficult to assess, but the notion of influence relationship seems to be able to encompass all the possibilities from a more general level — it is interesting to note that the chemistry definition clearly states that the relationships can also be indirect. Even in the case of the generalised systems theory definition, the variables chosen for studying must have an effect or influence worth studying, as deemed by the observer — a collection of purely independent variables does not form a system.

Thirdly, the nature of components or relationships does not seem to clearly separate classes of system types based on system goals. This can be observed by comparing the computer science and information systems definitions: even if the nature of the components as described for the information system is more diverse, the aims are very similar. Yet, it is also worth noting that natural systems tend to have mechanisms for the reduction of entropy — in the context of the natural water cycle, the gravitational force can be seen as this sort of mechanism, because it helps collect water in rivers, lakes, and oceans, leading to increased organisation and therefore reducing the local level of entropy. Complex, human constructed systems cannot inherently manifest this property without it being included in their design and construction, which is why we have explicitly included it in our definition.

Lastly, all the definitions seem to implicitly include a notion of level of detail — biological organisation spanning several scales, mixture of molecules, planets, liquid in a glass, observers selecting variables. However, we argue that this is not primarily a property of systems, but rather of the process of conceptualising a system, so therefore of a model of system, which is why it is not explicitly included in the definition.

3.2.1.2 Heterogeneity

Having decided on a definition and notion of system suitable for a modelling account prioritising integration, we can now turn our attention to the notion of heterogeneity.

In the Oxford English Dictionary, the heterogeneity property is defined as following: 'Of a body in respect of its elements: composed of diverse elements or constituents; consisting of parts of different kinds; not homogeneous' [97].

This definition, even if dated considering the formulation focused on the notion of body, still manages to encapsulate the fundamental concept of heterogeneity: the idea that a composite entity can be formed out of a set of sub-entities which manifest differences of various degrees. The scale of such differences is not specified, although it is implied that the differences are related to the nature of the components or constituents.

However, considering the goal of our modelling account, a bit more precision is required. For instance, restricting the notion only to bodies — even if figuratively this could be understood as a collection of entities — would also restrict the scope of our account. Furthermore, the difference aspects require further clarifications in at least three regards:

The first one is related to the nature of system components. Since the definition does not reveal anything specific about the nature of the manifested differences, we have to extend it, but also maintain the generality level. In order to do so, we must consider that differences may be manifested not only at the very explicit level of the components — for example by considering the materials used in the construction of the component —, but also at the level of the knowledge type, epistemic and ontological assumptions, and means of construction involved.

The second one is related to the relationships between components, which unsurprisingly can also differ. As stated previously, studying and conceptualising such relationships as influence relationships seems to be a good initial step because the existence of influence between components is a primary criterion for the existence of a system. Furthermore, this is not necessarily derived only from differences in the nature of the components: even at the level of two hardware components, different degrees of influence can be manifested. Much further work could be carried out in this area, particularly on case studies that could include socio-technical systems for a better understanding of the possible relationships manifested between sub-systems and sub-components.

The third one is concerned with sub-systems goals. Although the definition does not mention anything in this regard, organisational science has shown that organisational systems can be characterised by conflicting personal goals at the level of their sub-systems — for instance organisational departments — even when a mature, standardised organisational posture exists. Since these goals can lead to unexpected outcomes and behaviour in practice, we believe that explicitly acknowledging their existence and heterogeneity is important, particularly since we intend to be able to represent them.

Nevertheless, compared to the systems definitions, isolating specific versions of definitions for heterogeneity across different scientific areas is not that easy. Because of that, we present below some descriptions related to the use of the notion, which are perhaps not precise enough to be considered definitions, but still illustrate some of its characteristics:

Biology: Biological systems are highly heterogeneous. The diversity and variability observed within biological systems at various levels range from molecular and cellular to organismal and ecological scales. This diversity manifests in differences among individual cells, organisms, or populations, contributing to the complexity of biological systems. At the molecular and cellular levels, genetic variations, epigenetic modifications, and stochastic processes lead to diverse phenotypes even among genetically identical cells. This intracellular heterogeneity plays a crucial role in cellular functions, responses to stimuli, and the emergence of distinct cell types within tissues. [189]

Chemistry: Mixtures can be either homogeneous or heterogeneous: a mixture of uniform composition and in which all components are in the same phase, such as salt in water, is called homogeneous, whereas a mixture of non-uniform composition and of which the components can be easily identified, such as sand in water, it is called heterogeneous. [262]

Physics: A material or system that consists of distinct components or phases with different properties, resulting in a non-uniform composition or structure throughout the system. The physical properties (e.g., density, refractive index, conductivity) can vary from point to point within the system. [171]

Computer science: Heterogeneous computing refers to a system that uses multiple types of computing cores, like CPUs, GPUs, ASICs, FPGAs, and NPUs. By assigning different workloads to processors that are designed for specific purposes or specialised processing, performance and energy efficiency are improved. The term “heterogenous computing” may also refer to the use of processors based on different computer architectures, a common approach when a particular architecture is better suited to a specific task due to power efficiency, compatibility, or the number of cores available. An early and still relatively common form of heterogenous computing is the combination of CPU cores and a GPU (Graphics Processing Unit), used for gaming and other graphics-rich applications. [13]

Management: Heterogeneity refers to the diversity or variation in the composition of teams, organizations, or markets. This can pertain to differences in skills, backgrounds, perspectives, or resources among members of a team or within an organization, or to the diversity of products, customer segments, or competition within a market. [77]

Economics: A term used to describe the variation or diversity in characteristics, preferences, or behaviours among economic agents (such as consumers, firms, or workers), goods, or factors of production within an economy. This heterogeneity can lead to different outcomes in economic models and affects market behaviour, distribution of resources, and policy effectiveness. [195]

Sociology: Heterogeneous may refer to a society or group that includes individuals of differing ethnicities, cultural backgrounds, sexes, or ages. [190]

Generalised systems theory: The theoretical interest of systems engineering and operations research is in the fact that entities whose components are most heterogeneous — men, machines, buildings, monetary and other values, inflow of raw material, outflow of products and many other items-can successfully be submitted to systems analysis.[301]

Two important observations can be drawn from the above described paragraphs and their comparison with our extensions to the definition of heterogeneity:

Firstly, concerning the characteristics of the entities or components manifesting heterogeneity, it feels almost unnecessary to say that they are also very diverse. Even the small subset of descriptions included illustrates that differences can exist at the levels of ethnicity, cultural background, sex, age, preferences, perspectives, behaviour, skills, allocated resources, customer segments, rate of competition, power consumption or physical properties such as density, refractive index, conductivity and observability. We do not claim that this list is exhaustive but that it is very likely to increase substantially in the future.

Secondly, enforcing our initial point about the importance of relationships, heterogeneity can be manifested between system components which are similar at the level of underlying entities but varying at the level of manifested relationships. As clearly stated in the biology paragraph, genetically identical cells can manifest different phenotypes — observable characteristics — due to genetic variations, epigenetic modifications, and stochastic processes. Briefly, these can be caused by genetic mutations, environment influence or complex processes which are not yet fully understood at a very high level of detail, which fits out conceptualisation of influence relationship. In Chapters 4 and 5, we will see that our modelling formalism is well suited for explicitly representing such relationships both at conceptual level via the distributed systems metaphor, but also computationally due to the stochastic nature of the resulting executable models.

Lastly, there is hardly any mention of the relationship between heterogeneity and system goals. The only references are in the areas of biology and computer science: the former describes the importance of intercellular heterogeneity in cellular functions, responses to stimuli, and the emergence of distinct cell types, whereas the latter shows that a heterogeneous hardware architecture can achieve better power efficiency and compatibility in graphically intensive applications.

3.2.1.3 Heterogeneous Systems

Having explored some of the existing formulations of the notions of system and heterogeneity and extended them to better suit the modelling formalism that we are attempting to construct in this thesis, we can now illustrate the resulting heterogeneous system definition resulting from their composition:

A heterogeneous system is a collection of structured and interrelated entities which manifests epistemic, ontological, or metaphysical diversity at the level of either entities, relationships, or goals and achieves a set of outcomes including, but not limited to maintaining the identity continuity of entities, relationships, and reducing local entropy.

We note here that this definition of heterogeneous system is also capable of encompassing homogeneous systems if the manifested degree of diversity is close to imperceivable or inexistent. When referring to heterogeneous systems in the remainder of the thesis, it is to this definition that we are referring to.

In the context of cybersecurity systems, the abstract notion of a heterogeneous system can be concretely instantiated by looking at each of its underlying components. This begins with the interplay of diverse entities — such as users, employees, devices, software and hardware components, data assets, organisational policies, and legislation — and the structured relationships that bind them, including access permissions, communication flows, trust hierarchies, and control mechanisms. These entities are not isolated in a real organisation; they are embedded within architectures and operational models that enforce interdependencies, such as identity management systems linking users to credentials or network configurations dictating permissible communication paths. Furthermore, due to the participation of individuals, a secondary layer of less formal relationships exists, including influence, personal preferences, biases, and informal practices. Although it is not our place to advocate for or against such practices, we note that, from both a modelling and security perspective, they must be accounted for, as they can introduce vulnerabilities exploitable via social engineering, side-channel attacks, or insiders.

Epistemic diversity arises from the multiplicity of perspectives and knowledge sources involved in both the system operation and its defence, ranging from human analysts interpreting threat intelligence, to automated tools applying distinct detection heuristics, finance personnel involved in approving security investment budgets, or legislation governing certain high risk industries.

Ontological diversity is evident in the coexistence of fundamentally different types of entities — physical hardware, virtual recovery images, human actors, abstract policies, etc.

Metaphysical diversity, though more subtle, manifests in the varying conceptualisations of core constructs such as identity, trust, threat, or risk, which differ not only across technical subsystems but also between organisational stakeholders and in their legal interpretations.

Together, these layers of diversity and interrelation exemplify the heterogeneous nature of cybersecurity systems and provide a refinement of the general definition toward its practical application. Ultimately, in a practical cybersecurity context, the above described definition of a heterogeneous system must be instantiated in alignment with the specific organisational goals, risk posture, and operational constraints of the entity in which it is employed. The particular configuration of entities, relationships, and governing principles will necessarily reflect the priorities, trade-offs, resources, and threat landscape unique to that organisation.

3.3 Modelling

In Chapter 2, we briefly described the theoretical context into which our research could be placed. In that context, we have identified a series of elements that we wish to further develop in our modelling theory: prioritising model integration, accepting model heterogeneity and multi-methodology, and relaxing the realist commitments that many modelling accounts in the literature have adopted as criteria for accurate representation — perhaps influenced by the simplicity offered by such a paradigm, but omitting the impact this could have on future model integration.

To be able to describe, relate to, and employ such concepts, we first require a basis for what our notion of model means and, more importantly, implies — similarly to our description of heterogeneous systems in the previous section. Therefore, in Subsection 3.3.1, we define what a model is and describe the implications of the definition with regard to model epistemology and ontology with the help of two examples. In Subsection 3.3.2 we reflect on the heterogeneous aspects of models and their relationship with the problem of incommensurable paradigms, and showcase how a multi-methodological conceptualisation of models can be used to increase their descriptive power. Lastly, in Subsection 3.3.3 we describe three relevant model qualities — conceptuality, formality, and executability — that form our triangle framework, together with a list of benefits that can justify its use beyond simple categorisation.

3.3.1 Definition and interpretation

We start with our definition of model. The choice of a simple, yet generic definition here is purposeful, to emphasise the constituent elements.

A model is a representation of a target serving a purpose.

First of all a model is a representation of a target. That is, in the words of [56], a mental object standing for something else. Naturally, various theories of representation have been developed and used under both inferentialist and referentialist approaches, as shown in Chapter 2. To some extent, our position on the matter can be considered as deflationary because of two reasons: we do not impose a series of necessary and sufficient criteria for what representation is and we do believe that representation in a specific domain is deeply connected to the practices of that domain. In the formulation by Suarez, ‘it is impossible, on a deflationary account, for the concept of representation in any area of science to be at variance with the norms that govern representational practice in that area ... representation in that area, if anything at all, is nothing but that practice’ [284]. Although deflationary, we lean towards a cognitivist perspective similar to Oltramari in the sense that we adhere to the separation of presentation and representation — ’the notion of presentation deals with the dynamic forms of cognition while representation is mainly bound to an abstraction level‘[232].

We note here that choosing deflationary pragmatism as underlining tone for our inferentialist methodology is a fairly subjective decision, but motivated by a set of perceived advantages when considering its integrative purpose:

- **Flexibility** — the reduction of overcommitments to pure inferentialism leads to a more balanced position that allows for the acceptance of insights from diverse and possibly opposing approaches.
- **Resistance to absolutism** — if pure inferentialsim risks overemphasising logical relations at the expense of empirical grounding, then pure referentialism can oversimplify the complexity of meaning-making by focussing too

powerfully on external reference.

- **Focus on utility** — prioritising the utility of methodologies rather than abstract commitments ensures that methodological selection is judged by effectiveness in achieving specific goals.

- **Critical dialogue & humility** — recognising the limitations of any single method or perspective in capturing the full complexity of phenomena and avoiding the dogmatic tendencies of classical inferentialism or referentialism.

Furthermore, our interpretation of pragmatism should be seen as closer to Richard Rorty's neopragmatism [260] rather than Charles Peirce [237, 238] or Wilfrid Sellars' [269, 270], given the absence of a focus on objectivity and naturalism and the relationship with normativity. In this context, the nature and complexity of the target are relevant but hard to treat unitarily by a method that avoids multi-methodological selection at a low level of detail. On a theoretical level, that is because the interpretation of the presentation of the target influences the decision of how to represent it. Model targets can be both empirical and non-empirical, vary in complexity from singular phenomena to large scale ecosystems, and accumulative tendencies exist in both directions. Empirically, people construct more complex systems over time to fulfil their needs — this can be seen, for example, in a review of the development of telecommunication systems [150] or other similar historical presentations. Non-empirically, both scientific theories and artistic and literary currents go through accumulative and revolutionary stages in a Kuhnian sense [182], except the criteria for what starts the transition between phases are different. From a modelling perspective, this set of criteria is not relevant, but our definition of heterogeneous system is able to encompass them regardless. The focus should remain on a method of model design and construction which takes into account this increasing variety of model targets. For the specific cases of modelling without a specific target as described by Weisberg [304] — generalized modelling, hypothetical modelling, and targetless modelling — we simply view the target as highly conceptual.

Secondly, a model serves a purpose. While this is generally accepted in the literature, and works such as [113], [96] and [251] explicitly describe the importance of the model goal at the level of validation, they implicitly assume that such a goal is either singular at a given time in the life-cycle of the model or overpowers the other goals. Particularly, Forrester claims that 'the validity of a model should not be separated from the validity and the feasibility of its goal' [113] and Dhrymes concludes that 'validation becomes a problem-dependent or decision-dependent process, differing from case to case as the proposed use of the model under consideration changes' [96]. However, our position on the influence of the modelling goal on the actual model is more extensive and does not include only the validation aspects. Following the presentation and representation distinction, we argue that the modelling goal influences both. At the level of presentation, although the goal cannot affect the empirical manifestation of phenomena, it can surely impact their interpretation. This usually translates into a limitation of decisions with regard to the model representation simply because some aspects are left out.

We briefly illustrate these influences by looking at two models targeting the same phenomenon — namely, the decline of the population of scallops in the French Bay of St Brieuc — but with differently stated goals. The models we look at are [58] and [117]. The first is highly conceptual and attempts to determine possible reasons for the decline in the scallop population, whereas the second is mathematical and simulational and determines generalised equilibrium manifestations for the scallop population under different competition and environmental assumptions.

The first model uses a specific framework, namely the sociology of translation, to represent the actors of this environmental ecosystem — fishermen, scallops, predators, scientists, local communities — as going through a set of phases — problematisation, inter-assessment, enrolment and mobilisation — which lead to the development of social relations between them. We do not question here the possibility for the development of a social relationship between a scallop and a scientist. During this process of representation, the scientists end up as representatives for all the other actors and misrepresent their characteristics, this leading to controversies which in the end destroy the constructed actor-network. Therefore, the model does not produce explicit results, but rather a process to be observed, which could lead to them.

In the case of the second model, the situation is different because the input is mostly numerical data, manipulated by the model through statistical operations, and then interpreted at the end. Briefly, simulations are used to show that controlling invasive species can lead to an increase in the scallop population, and furthermore to quantifiable economic benefits.

In the end, both models managed to achieve their stated goals. The first one inherently shows how a misrepresentation problem can lead to a lack of practical solutions to the population issue and in the end to even more population decline, whereas the second one produces its desired general solution. The chosen interpretation for the phenomenon is clearly different in the two cases and does lead to different model representations and types of solutions, even if considering the same target. Nevertheless, constructing models in an ontological and epistemic silo is not the only possibility.

3.3.2 From heterogeneous systems to heterogeneous models

If heterogeneous systems as we have defined them can be scientifically agreed upon, we believe models are also heterogeneous, in the sense that they should be able to construct inter-operable representations of phenomena with wildly different metaphysical, epistemic and ontological bases.

Even if theoretically, this can be viewed as strongly related to the problem of incommensurability of scientific paradigms as studied by philosophers of science such as Thomas Kuhn [182] or Paul Feyerabend [112], it is important to note that both authors 'repeatedly emphasised that incommensurability does not imply incomparability' [148] — the simplest way of achieving meaningful comparisons seems to be at the level of the predictions constructed using the theories, or in other words, at the level of model outputs. Yet, analysing a model only in terms of outputs can lead to both misrepresentations and a reduced ability to justify or explain why certain outputs have been produced.

Because of that, we argue that a general notion of model should be viewed as 'multimethodological' [214], including the metaphysical, epistemic and ontological aspects. This means that at the level of sub-models, different methodologies, techniques, tools and metaphysical, epistemic, and ontological considerations may be applied. At the general model level, this leads to multi-faceted representations of entities and relationships which offer more descriptive power than a simple reductionist approach. For example, as a result of environmental interaction, a concept might have two different ontological descriptions in two different sub-models — this was hinted at when considering the works of Donellan and Kaplan in Chapter 2. However, we note that this conceptual openness must also be balanced with the sceptical analysis of model results and relationship with model goals, even if considering the additional implications.

To some extent, this is not that different compared to other approaches to modelling, except that in a lot of cases, the underlying complexity is being hidden by altering the level of detail used in conceptualisation. Therefore, one could argue that the heterogeneity of models is simply a consequence of different levels of details co-existing in the sub-models.

Similarly to a distributed system case, the management of these differences allows the model to manifest coherently. Chapters 4 and 5 provide further details regarding this matter.

3.3.3 The triangle framework: a characterisation of heterogeneous models

Having described our notion of model, the modelling goal, the implications of the interpretation of presentation on representation, and inherently on the metaphysical, epistemic, and ontological aspects of a model, we now focus on how to characterise a model. Since our conceptualisation of models is driven by pragmatic and sceptical commitments, we cannot base our characterisation solely on presentation or interpretation. Therefore, we have chosen the manifestation of representation, or the means of construction because of its direct relationship with multi-methodology, language constructs, and their determinable nature at a specific point in time.

Based on its construction and manifestation, a model has 3 main qualities: conceptuality, executability, and formality. In [63], we present them under the name of 'Triangle Framework' and empirically explore their appropriateness for describing the components of models in a cyber-security context.

- **Conceptuality** — The conceptualisation of a model pertains to how its fundamental components and the connections between them are present and explicitly conveyed through precise natural language, pictures or drawings, for instance. To some extent, this can be viewed as the size of the directly expressed ontology.
- **Mathematicality or Formality** — This refers to the degree to which the elements and relationships of a model are expressed using formal constructs. For example, models might be expressed as systems of equations or logical formulae.
- **Executability** — This represents the degree to which the elements and relationships of a model manifest themselves in a physical or computational environment through series of iterative steps.

Various model categorisation attempts have been undertaken in previous years with their focus on the so called most basic constitutive elements. An example can be seen in Weisberg [304], where concrete, mathematical, and computational models are clearly separated. The main difference between such an approach and ours is that in our view a model can manifest degrees of these qualities at the same time. For example, a simulation model may be comprised of two different sets of simulations: one based on a system of equations and another based on a state machine which was constructed ad-hoc from empirical observations. After the simulations are completed, the results might be presented visually through graphs or formulated in natural language. In such a case, we do not believe that labelling the model as computational is enough because each decision to use a certain construction technique is related to the model goal. Furthermore, such an approach would lead to viewing most presentations of a model — interestingly, the constructed representation of a model ends up becoming a surrogate presentation for the model target — as conceptual and therefore omitting its previous configuration and implications from analysis. Not taking into account all these decisions can lead to model misrepresentation and misuse.

As previously said, these qualities coexist and models may have components that exhibit characteristics of all three. Additionally, they trade off against one another; that is, a highly conceptual model will be less formal and executable, or a highly executable model will be less conceptual and formal. This happens because the process of designing and constructing a model has resource constraints, and favouring one of the three qualities based on the context of use can imply a higher chance of achieving the model goal. For example, using a Bayesian model in a court of law will most likely not have a high chance of success — for details see [89] — but a conceptual model based on argumentation with the exact same conclusions might do it. More malicious examples could be constructed here because favouring a specific subset of the elements of the presentation of a phenomenon combined with a construction technique that is considered appealing by a target audience is exactly how manipulation techniques work. We will not further pursue this direction, but we clearly state that the difference between manipulation and focusing on the model goal while considering some elements irrelevant for the context exists. In the case of manipulation, some elements are known to be relevant but specifically omitted so that the final results would be different. In the other case, the elements are omitted precisely because they would not impact the final results.

While we discuss the degree of model qualities, it is important to note that we don't imply a precise measure — there are no units for formality, executability, or conceptuality, and their quantification is subjective. Nevertheless, these qualities, along with the triangular framework in general, provide us with a language to structure and discuss models. Figure 3.1 illustrates how we think of these three types of model and their relationships: in a given model, the relative significance of each of the components determines, by proximity, the position of the model within the triangle; furthermore, the position of the model may change as it evolves during its construction.

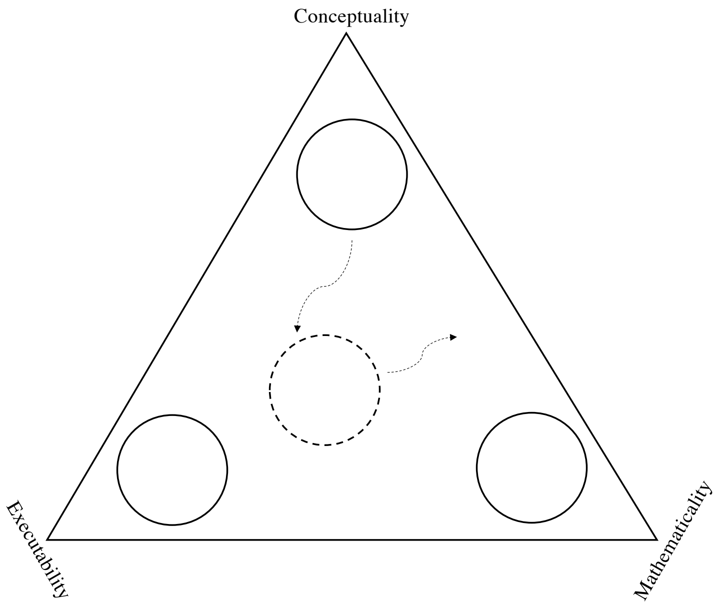


Figure 3.1: The triangle framework [63]

We suggest that the importance of our proposed framework goes beyond its use for categorising, and hence understanding the relationships between, extant models and types of models. Specifically, we suggest that it can:

- Assist the parties involved in modelling in determining which property to emphasise to enhance the likelihood of more accurately representing the target, in alignment with the modelling goals;

- Minimize the likelihood of creating a model that is impractical for achieving the modelling goals;

- Serve as a common base for argumentation, and therefore language for both structuring the process of design and construction and, formulation of requirements;

- Offer a way of analysing a model through all the design and construction stages rather than just at the end;
- Reduce development time because a direction for model evolution can be determined and decided upon, rather than just emerge during construction;
- Lead to a way of comparing models based on the properties of sub-models.

3.4 Triangle case study:

In the previous section, we have described the modelling triangle theoretically and have explained how models move around the triangle during their construction.

In this section, we explore the modelling triangle in the concrete setting of information security, exploring where different types of security models are placed within the triangle. We do this by looking at existing models published in recent security-related conferences and placing them on the triangle. This has several purposes: we want to get a sense of the types of models used in security, we want to understand if there is a relationship between the intended purpose of a model and its location on the triangle, and lastly, we want to test the triangle approach.

3.4.1 Methodology

We have selected papers from five security conferences from 2020: BlackHat USA, NSPW, ACSAC, WEIS, and GameSec. We selected these conferences as they cover a range of topics and security traditions; we looked at all the papers from each conference in 2020. In total, we looked at 212 research papers encompassing a range of security topics: behavioural and security management, security policy, technical exploits, machine learning, economics, and more.

For each paper, we wanted to: (1) determine whether or not it contains a model; (2) understand the purpose and type of model; and (3) determine an appropriate location on the triangle for the model. As a methodological basis, we employed a grounded theory approach, primarily motivated by the relative novelty of the triangular framework and the lack of similar research studies conducted in the area of security models. Grounded theory has two variants: one that focuses on the emergence of properties from the data coding process guided by a theoretical understanding of the domain of study and another that denies the need for any prior domain exploration. Kelle [167] illustrates both approaches. Since we did perform a prior domain exploration by analysing the main philosophical positions regarding models and know the properties we are looking for — conceptuality, executability, and mathematicality — we adopt the first method and perform the selective coding and classification processes with the properties described above in mind.

We have chosen this study methodology for a number of reasons:

1. Grounded theory is integrative as long as the coding process is consistent. This is extremely important, since it allows the analysis of models constructed using various methodologies. 'According to Ralph, Birks, and Chapman [252], grounded theory is methodologically dynamic in the sense that, rather than being a complete methodology, grounded theory provides a means of constructing methods to better understand situations humans find themselves in.'

2. Grounded theory provides ecological validity. This means that the theory produced using this approach is representative of the underlying body of literature surveyed. Although not as powerful as when conducted through interviews with practitioners — since in that case additional questions about the subject of study could have been asked — the novelty of the papers shows the 'state of the art' in the security field at the moment.

3. Grounded theory maintains parsimony. That is, in a situation where multiple hypotheses exist about a certain phenomenon, the one with the smallest number of assumptions is preferred. This allows us to maintain a relatively small number of properties, since we aim to provide practical and simple explanations of complex phenomena by attempting to link those phenomena to abstract constructs and hypothesising relationships among those constructs.

4. Although employing both qualitative and quantitative techniques, the nature

of the analysis remains qualitative and facilitates the interpretation of conceptual aspects of the models under study.

We followed the following process. First, we analysed every paper to decide whether or not it contains a model, according to our definition from Section 3.3. We used a broad understanding of 'model' to ensure we captured conceptual as well as technical and formal models and as such were quite inclusive in the papers we accepted. For example, papers that constructed and reasoned about a structured representation of the phenomenon under study, even if descriptive, were considered models. Papers that did not include such a representation, such as those focused on problem solving or tool building in a very specific and mostly technical focused case were not included. Lastly, some papers included small models used for explanatory purposes, such as showing where their work fits within a system. These are also included in our analysis.

For the papers that contained models, we performed subjective coding focusing on the model description, the techniques employed in the model construction, the model purpose, and the topic. Since we wished to maintain some of the grounded theory ethos, we did not use a pre-established coding scheme in this step, and instead generated the codes as we went through the papers.

After that, we performed selective coding by linking the previously computed codes with our three primary categories — conceptuality, executability, and mathematicality. At this point, we derived the 'triangle configuration' of the model under study, with the important note that we did not simply quantitatively analyse the size of the resulted categories. Deciding the relevance of the underlying codes with respect to the overall modelling goal remains a qualitative process, and therefore the constructed configuration is subjective.

3.4.2 Findings

The intent of this study was to better understand the nature of the models employed in the information security field in 2020. Following the methodology described above, we discovered that 67% (142) of the total 212 surveyed papers did indeed employ models. The initial topic analysis and coding have produced 65 different topics that were further reduced to 35. For example, topic codes such as 'social behaviour', 'social engineering', 'community analysis', 'problem solving' or 'human oriented design' were included in the 'human oriented security' category. Table 3.1 illustrates the most encountered five topics for models in each conference, ranked by their total number of occurrences and having duplicates removed. Subsequently, the most encountered topics were 'attacks/exploits' and 'privacy'.

It is important to note that a model is not limited to a single topic: if one specifically focuses on attacks that affect user privacy, it would be assigned as having both the 'attacks/exploits' and 'privacy' topics. Also, the development of models for the purpose of better understanding machine learning in general can be seen as an interesting attempt at using descriptive models to understand other models.

	Blackhat	Nspw	ACSAC	WEIS	GameSec	Total
Attacks/Exploits	19	1	12	2	3	37
Privacy	5	3	17	1	2	28
IoT	5	0	14	1	3	23
Human oriented	7	5	5	3	2	22
Network Security	7	0	8	1	6	22
Economics	1	0	4	12	3	20
Policy	6	2	7	2	1	18
Hardware Security	10	0	3	0	4	17
Software Security	5	2	8	0	1	16
Machine Learning	6	2	6	0	1	15
Theoretical Security	0	0	2	1	10	13
Systems architecture	6	0	2	0	4	12
Risk	2	0	4	4	1	11
Management	2	0	3	5	0	10
Game Theory	1	0	0	0	8	9

Table 3.1: Top 5 topics per conference [63]

	Blackhat	Nspw	ACSAC	WEIS	GameSec	Total per type
Simulation Models	8	1	25	2	1	37
Descriptive Models	21	6	1	2	0	30
Statistical Models	2	0	15	11	1	29
Deep Learning Models	6	1	15	0	6	28
Game Theory Models	0	0	0	0	18	18
Total per conference	41	8	58	15	26	142

Table 3.2: Model types per conference [63]

By analysing the model's topic, goal, and construction procedure, we have produced triangle configurations for each of the surveyed conferences, which can be observed in Fig 3.2. These configurations largely correspond to the publicly described conference tradition — for example, the models in GameSec had a tendency towards formality — with an interesting aspect identified in Blackhat: even though the models tackled many 'attacks/exploits' and 'hardware security' aspects — as it can be seen in Table 3.1 — they are complemented by models with a tendency towards conceptuality that attempted to explain their functionality, resulting in overall more balanced configurations when considered holistically. Furthermore, we classified the models according to their construction method and modelling goal into five categories.

1. **Descriptive models**: Models in this category are mainly constructed using natural language descriptions, qualitative reasoning and sometimes graphs, charts or other means of visual representation. They construct a subjective representation of reality which can vary in complexity and can include both qualitative and quantitative studies as a starting point. Their primary goal is to simply describe or analyse phenomena that are hard to quantify and therefore focus on topics such as 'human aspects of security', 'security management', 'philosophical aspects of security' or 'security policies'. However, the analysis process has revealed that such models can be used also for describing other models, not necessary phenomena directly. As illustrated in Table 3.2 and Figure 3.3a, models in this category represented roughly 21% of the total models encountered and had a strong tendency to be placed close to the upper corner of the triangle because of their highly descriptive nature.

2. **Simulation models**: This category contains two different types of models that have a common construction method — experimental simulations and practical demonstrations. They are comprised of interpretative executable code, constructed manually by a developer and reflecting a human interpretation of a certain phenomenon. With respect to their goal, they can either be used for experimentation, such as simulation based models coming from

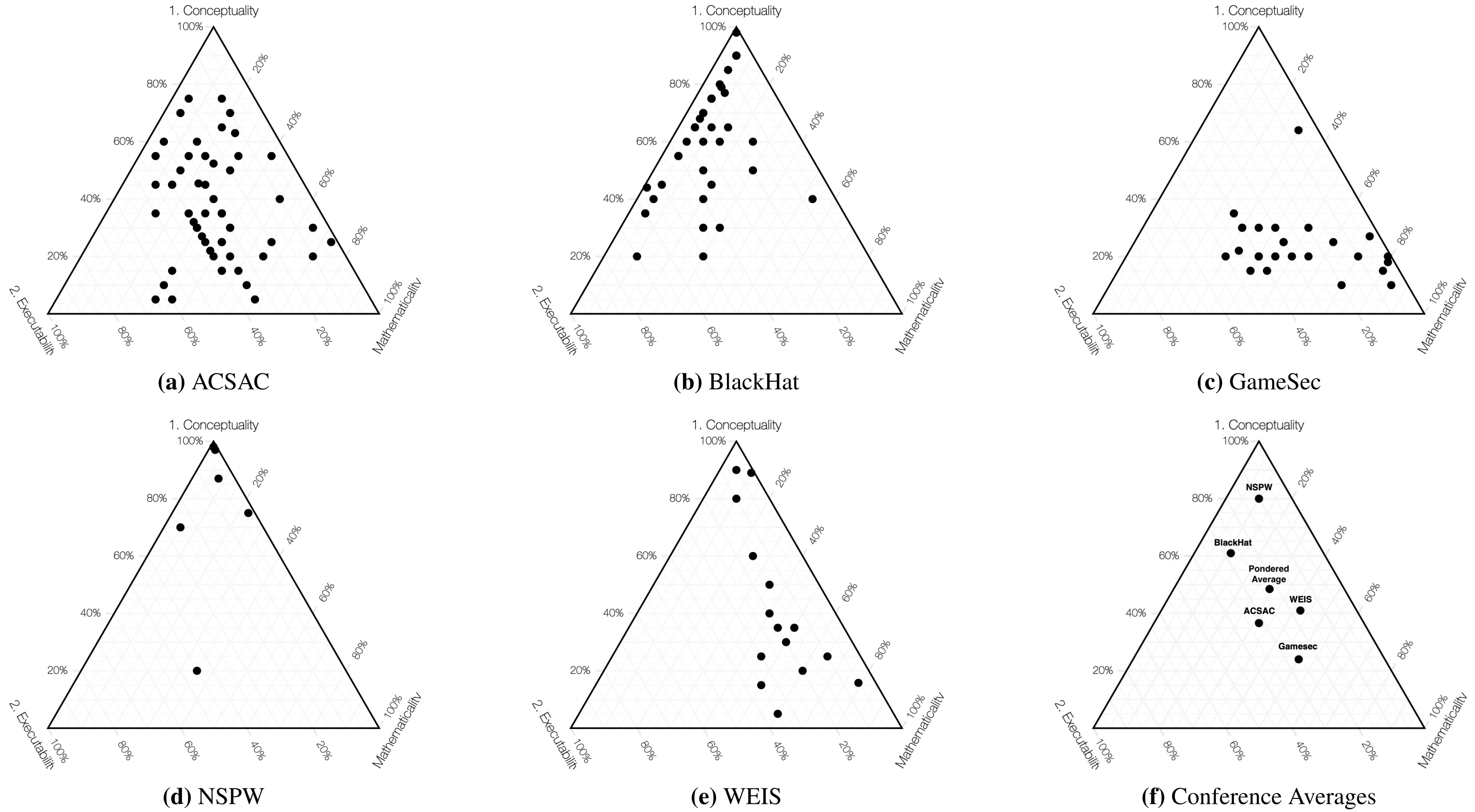
the dynamic systems tradition, attack demonstrations or enforcing qualitative reasoning. They focus on topics such as 'IoT', 'network security', 'software security', 'operation systems emulation', 'malware' or 'attacks/exploits' because the phenomenon under study can be quantified and represented using graph-like structures. As illustrated in Table 3.2 and Figure 3.3b, models in this category represented roughly 26% of the total models encountered — the largest category — and were placed closer to the centre of the triangle. They did not manifest a higher tendency towards the left corner because the model construction was manual and in some cases, the model was run a single time to illustrate that a certain attack was possible.

3. **Statistical models**: Models in this category are constructed using executable code that includes statistical algorithms, data science techniques, natural language processing, and even some traditional machine learning techniques not including deep learning. They construct a complex, stochastic interpretation of reality, usually employed for better understanding or making predictions about a phenomenon that is either extremely complex or would take too much time or analysis power to be understood individually. Furthermore, their most relevant aspect is that their method of producing results can be traced back and understood, with the important note that the results do not directly lead to some automated real world consequence, but require additional interpretation. The most relevant topics to this category were 'privacy', 'security management', 'economics' and 'human oriented security'. As shown in Table 3.2, statistical models represent almost 20% of the total surveyed models and Figure 3.3e depicts them as having balanced triangle configurations, with some slight tendencies towards either conceptuality or mathematicality based on the nature of their input data. For example, a model employing natural language processing and principal component analysis techniques over qualitative data obtained from interviews was placed closer to the conceptuality corner, whereas a Bayesian analysis of security investments was placed closer to the mathematicality corner.

4. **Deep Learning models**: Models in this category are constructed using deep learning and neural networks approaches and they construct a representation of reality that is similarly to statistical models, with the primary difference being the difficulty of interpreting or justifying the produced results. Because of this, they tend to be used for automated problem solving in areas such as offensive security or threat detection. Their triangle placement can be similar to that of simulation models, as Fig 3.3d with the important difference that they do not manifest the slight conceptuality tendency since their phenomena interpretation is particularly hard to understand. However, they do manifest the strongest tendency towards executability, and represent almost 20% of the total models surveyed. Taking into account the current focus on the development of artificial intelligence explainability methods, we could observe a significant amount of models moving from the deep learning category to the statistical models one.

5. **Game-theoretic models**: This category is primarily comprised of mathematical models constructing game-theoretic interpretations of phenomena. Some of the observed models did produce analytical solutions for solving the represented games, whereas others were simply used for problem setting. In the second case, other types of models were used to produce the desired strategies in the game setting. They were usually employed in areas such as ‘network security’, ‘risk’ or ‘attacks/exploits’, but mostly provided the theoretical setting for another type of model to interpret. We observed that the interpretation they produced was mostly used as a setting for either deep learning or simulation models. For example, a game theory model was used to formalise the concept of cyber deception as a multi-party stochastic game and then a simulation model was used to illustrate a successful winning strategy. However, another approach was to construct a deep reinforcement learning model of the involved parties and execute it in multiple epochs such that the actors could develop increasingly better strategies while learning from their own mistakes. As 3.2 and Fig 3.3c illustrate, these models can have both balanced triangle

configurations and heavy tendencies towards mathematicality and represent roughly 13% of the models surveyed.



(a) ACSAC

(b) BlackHat

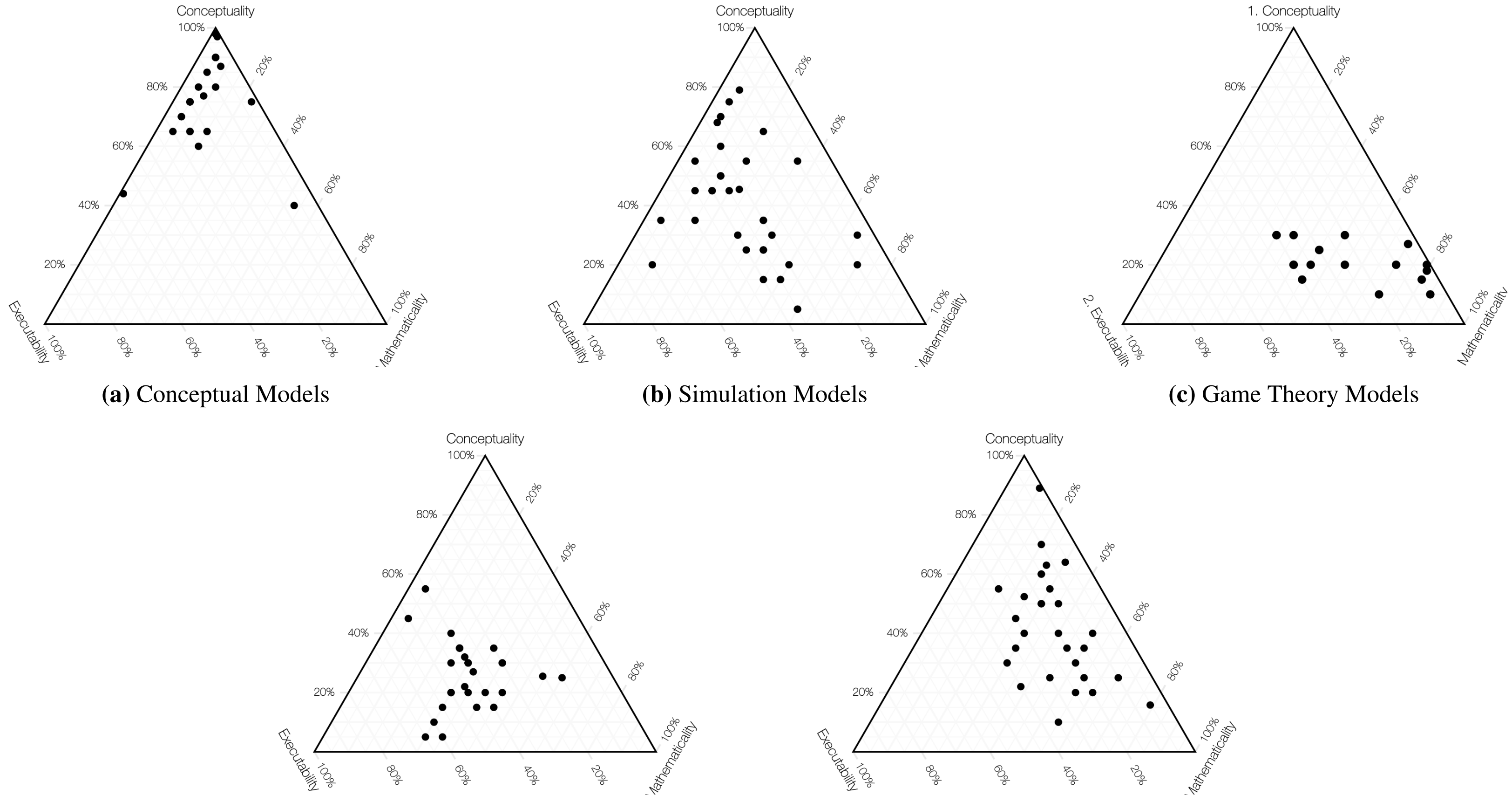
(c) GameSec

(d) NSPW

(e) WEIS

(f) Conference Averages

Figure 3.2: Positions of papers from different conferences on the triangle [63]



(a) Conceptual Models

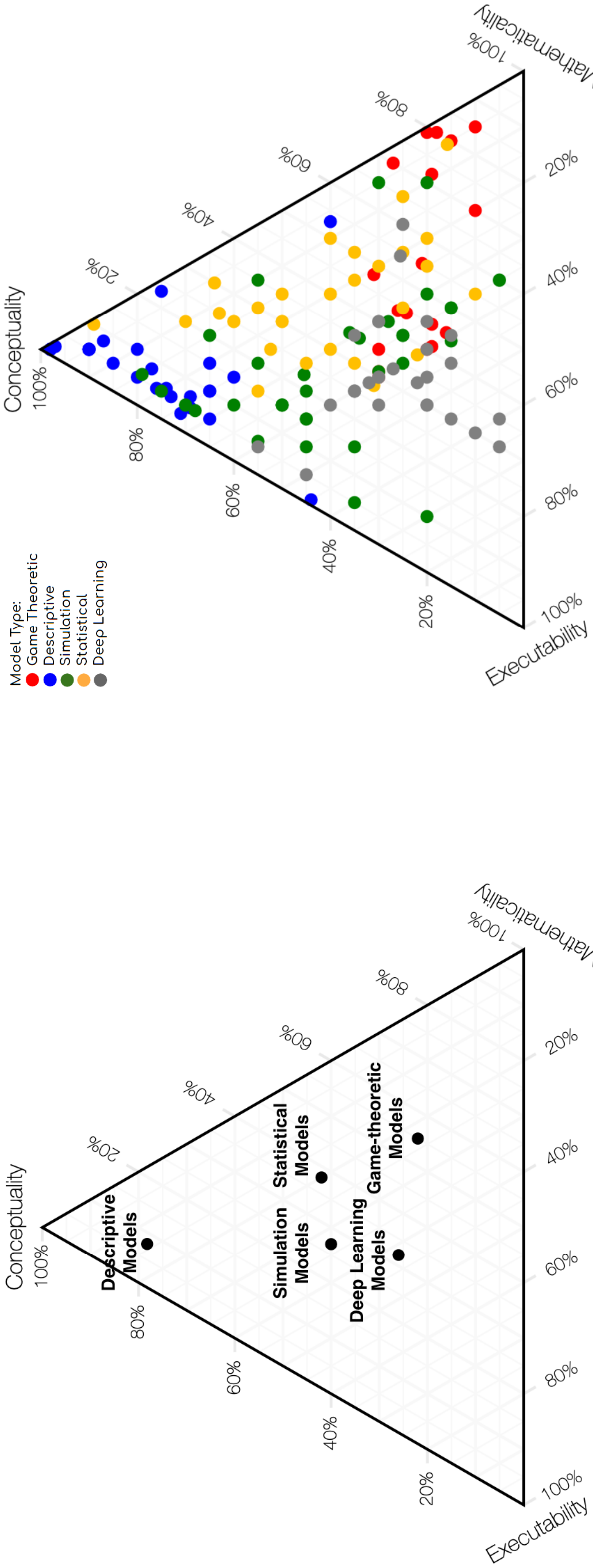
(b) Simulation Models

(c) Game Theory Models

(d) Deep Learning Models

(e) Statistical Models

Figure 3.3: Positions of different model types on the triangle, extended [63]



(a) Average Model Types

(b) Model Types

Figure 3.4: Positions of different model types on the triangle, summarised [63]

Moreover, some additional observations can be drawn when analysing the average and complete model types placement on the triangle in figures 3.4a and 3.4b. For example, the conceptual, deep learning and game-theoretic models can be seen as having the strongest manifested tendencies towards the triangle's corners. Subsequently, the simulation and statistics models manifested the most balanced configurations for two different reasons: simulation models had the most open approach and used internal sub-models that would have been assessed differently in isolation — for example, a stochastic module for agent behaviour or an economics module for determining the risk of an actor's action — but produced balanced overall models, whereas statistical models used extremely varied input data that introduced an additional degree of conceptuality to the mathematical methods used.

However, Fig 3.4b clearly shows that the model categories are not entirely delimited by their triangle configuration: for example, models in Kaczmarczyck et al. [159], Noor et al. [200] and Xiao et al. [307] have similar, central triangle placements even though they are members of the deep learning, statistical, and respectively simulation categories. Furthermore, even though their topics are also different, namely automated malware family identification, illustrating a mechanism for key distribution on automotive networks and analysing the forensic validity of approximated audit logs, they all obtain a more balanced configuration by introducing qualitative reasoning about their inner workings.

We believe that this balancing process leads to models that are easier to understand and therefore more suitable for security decision-making but that require further work to be validated. However, at the end of this exploration of the information security domain, we can draw several conclusions.

1. Models are an important tool for information security today.

2. Usually, models remain focused on very specific problems.

3. Some models directly interact with or complement other models. This raises the question of how models should communicate with one another.

4. Some meta-modelling attempts exist, but only at the theoretical level. For

example, highly mathematical or conceptual models of machine learning algorithms.

5. Simulation models seem to provide a good base for constructing models with components of different types.

3.4.3 Reflections on validation

In the previous sections, we have focussed on constructing an integrated modelling framework for characterising models according to three primary properties — conceptuality, formality, and executability — that are directly linked to the relevant philosophical and modelling literature. We have described how these properties can change during model construction and have attempted to use this approach to explore a part of the present-day space of information security models. Here, we seek to understand some of the implications that our vision might have on the ongoing debate regarding the validation of models, especially in the case of modelling large-scale security ecosystems.

The validation of a model, at a general level, consists in the deployment of a set of processes that is used for testing that the model performs according to its original goal. As expressed in [21], models can be separated into at least three categories according to their primary goal: modelling for the purpose of improving a certain performance indicator of a system, modelling for testing a scientific theory, and modelling for the sake of understanding or learning about a certain system.

However, it is not the goal alone that determines the type of model to be constructed, but also the nature of the system or phenomenon under study and the available input data types and their collection process. These elements — the goal, the available data and the nature of the phenomenon under study — corroborated with the underlying philosophical implications illustrated in chapter 2 and section 3.3 — have led to the development of different and sometimes opposing model validation methodologies in disparate domains of science. To illustrate some of these differences, we present some arguments from the economics, management science, and system dynamics literature that are of great importance for an information security context.

Starting with the management science domain, it can easily be observed that a singular position with respect to validation does not exist. For example, Naylor et. al. [222] argue for a validation process that combines rationalist, empiricist and positive economics [120], but has a primary empiricist assumption: 'A simulation model, the validity of which has not been ascertained by empirical observation, may prove to be of interest for expository or pedagogical purposes, but such a model contributes nothing to the understanding of the system being simulated' [222]. Their position is aligned to the positivist ethos by the utilisation of a truth based criterion for validation. In contrast, work such as Mitroff [217] place validation under the philosophical spectrum of experimentalism and focus on the relativity of the philosophy of science position held by the modeller. As Mitroff [217] states, 'a researcher's philosophy of science is as characteristic of him as it is of the phenomena he typically studies' and the same elements chosen as relevant for the construction of the model should be the same ones used for validation. A more practical position, directly related to the need of validating large scale models in a reasonable amount of time can be seen in House and McLeod [147]. There, the authors follow Friedman's principle [120] that assumptions can serve as scientific hypotheses even if 'unrealistic' as long as they can produce significant predictions and focus on the utility of a model rather than its relationship with truth. In the authors' own words 'A businessman cannot afford to discount a "hoped-for" infinite return as the result of an unknown expenditure for a near perfect model today. Our business world exists in the present, so the businessman will be satisfied to buy a somewhat less than a perfect model for a known cost.'

The same heterogeneity of positions with respect to validation can be seen in the economics literature. As already described, Friedman's position with respect to the need of validation for scientific hypotheses is of great importance. In the author's view, scientific assumptions do not need to be verified, since they can only be validated by their own predictive power. Since models can be considered a preliminary step in the formation of scientific theories — see Grim and Rescher [135] for a more detailed discussion — Friedman's assumption has been translated to models. A subtle distinction from this criterion of predictive power can be seen in the work of Cyert and Grunberg [84]. Following Popper's falsification thesis [245], the authors believe that a model's predictive power does not necessarily validate its position as scientific truth. Therefore, modellers should rather focus on constructing models with high descriptive power. Nevertheless, different variations of these criteria can be found in comprehensive literature overviews such as Dhrymes et al. [96] or Radzicki [251]. Of particular relevance is the conclusion of Dhrymes et al.: 'validation becomes a problem-dependent or decision-dependent process, differing from case to case as the proposed use of the model under consideration changes' [96].

Last but not least, we discuss some of the validation approaches employed in the system dynamics literature. Compared to management science or economics, the system dynamics literature is much more comprehensive in its attempts to tackle philosophical aspects of model validation. An important starting point is the work of Forrester [113], which can be seen as a relativist take on system dynamic validation that was primarily done in a positivist fashion before him. Forrester makes the claim that 'the validity of a model should not be separated from the validity and the feasibility of its goals' [113], and since the feasibility of the goals cannot be determined through a formal process, validation becomes 'a problem of social discussion' [22]. Furthermore, following a thesis similar to Kuhn's [182], he argues that 'Any "objective" model validation procedure rests eventually at some lower level on a judgment or faith that either the procedure or its goals are acceptable without any objective proof.' [113] and that qualitative model validation techniques must be used in practice, given the fact that 'a preponderant amount of human knowledge is in non quantitative form' [113]. However, Forrester's position was contested by works such as Ansoff & Slevin [12] or Nordhaus [225] that deem it unscientific on a positivist basis and ask questions such as 'Does it represent the judgmental approach of a particular scientist?' [12]. Ansoff & Slevin [12] also point out that Forrester does not clearly state a clear criterion of validity or a specific degree of correspondence between the model and the represented system. Additional details about this philosophical debate in the system dynamics field can be found in [9, 31, 32, 271, 258, 280].

As seen above, the fields of economics, management science and system dynamics have been encountering this philosophical debate on model validation for a long period of time, and yet a holistic integrative approach has not been designed. What is even more interesting is that economic, management science, and system dynamics models have been consistently used in information security without the necessary debate about validation.

The field of information security manifests, as shown in section 3.4, the diversity of encompassing all these different types of models, with respect to both goals and method. For example, Yeo et al. [311] construct a system dynamics model for the sake of reducing the negative impact of security policies on the effectiveness of operations in ports, conceptual work such as Inglesant and Sasse [154] complemented by the modelling in Beautement et al. [28] has infirmed the theory that cyclic password ageing techniques lead to increased password security and, the executable, language based model of Jachim et al. [157] can be used for understanding the behaviour changes of Twitter trolls during the Covid-19 pandemic.

Although a handful of examples, these are already enough to manifest different approaches on validation: [311] uses a system dynamics paradigm and both structural and behaviour oriented validation techniques, [28] construct a model that can be placed in between the economics and management science traditions but admits the need for further validation stemming from the subjectivity of the input data obtained from interviews and [157] uses traditional machine learning validation metrics such as k-fold cross-validation.

However, none of the above examples is truly about heterogeneous systems. One can imagine the need of constructing an information security model that combines multiple of the above described directions. Efforts in producing a practical, modular modelling approach that is open to qualitative interpretation and, at the same time, constructed on a rigorous mathematical foundation can be seen in [64] or [60]. The relevant aspect of the method presented is that sub-modules can be constructed using significantly different assumptions as long as the produced output is standardised by the use of interfaces and conceptualised under the distributed systems metaphor.

Nevertheless, validating a model with sub-modules built according to different philosophical assumptions will not be a trivial task. Given the researched streams of literature concerning validation, it is a plausible assumption to make that such a process will not be unitary in nature. Therefore, the model description offered by the framework described in this section, together with the distributed systems conceptualisation and the extended co-design cycle which will be further analysed in Chapters 4 and 5 becomes a road-map for the selection of validation techniques on a case by case basis — [271, 21, 194] offer comprehensive overviews of practical validation tests.

Therefore, our position is close to Dhrymes et al. [96], with the addition that the model description given by our framework can be used to guide the selection of validation tests. In a certain sense, Mingers' 'multimethodology' [214] idea is being translated to the issues of validation. This can facilitate the construction of validation loops from the early model design and implementation phases, thus bringing the advantages of an agile testing methodology. Furthermore, our belief is that the analysis procedure required to construct the model description and the process of choosing validation tests according to it can increase the believability in the usefulness of the constructed model by design.

However, the modularity offered by this method comes with a need of using both sub-module validation and overall model validation. The selection of validation tests for sub-modules is guided by the description offered by the above presented framework. When considering overall validation, more experimentation with the framework is required for an attempt to derive a criterion. Nonetheless, our belief is that such a criterion should take into account both descriptive power and believability and cannot be purely based on positivist premises.

3.5 Conclusion

As the world has evolved to become ever more dependent on complex ecosystems of large interacting systems, it has become ever more important to be able to reason rigorously about the design, construction, and behaviour not only of individual systems — which may include aspects related to all of people, process, and technology — but also of their assembly into ecosystems. In such complex situations, it is inevitable that no one type of model — such as mathematical models of dynamical systems, logical models of languages, or discrete event simulation models — will be sufficient to describe all of the aspects of ecosystems about which rigorous reasoning is required.

Because of that, we have attempted to extend the traditional notions of system and heterogeneity as showcased across different scientific fields and combined them to produce a general notion of heterogeneous system that would support a modelling account focused on model integration. Building on this notions, we have also showcased our concept of model, explained in detail how the nature of heterogeneous systems influences it, and presented multi-methodology as a viable approach for the construction of our methodology.

Furthermore, we have also proposed a meta-theoretical framework, the 'triangle framework', within which different types of models may be categorised and their interactions, especially during the construction of models, can be understood. Specifically, we have identified three qualities of models — conceptuality, mathematicality, and executability — and have explained how in practice models typically have all of these qualities to varying extents. We have conducted an empirical study of the models deployed in a range of security conference papers, have classified these models according to the framework, and lastly, we have started the much needed debate on validation methods that the information security field has been avoiding for far too long.

However, much further work is suggested, including on the following:

- the structure of the triangle and its component models;
- the evolution of models within the triangle as they are developed, especially

in respect of the roles of the stakeholders;

- empirical studies of model design and construction conducted over an extensive period of time, in a deployment environment, in the context of the triangle and the roles of the stakeholders.

Moreover, as we have already discussed, we hope to produce practical results such as correlations between a model's goals, the types of knowledge it employs, the means of design and construction, its triangle configuration, and its success of implementation and deployment in the real world. The results of these studies could be expected to inform a reformulation of the triangle framework and modelling account.

Finally, we emphasise the practical relevance of this chapter for modellers seeking to design effective representations of complex, heterogeneous systems. By introducing a general definition of heterogeneous systems and linking it to a multi-methodological conceptualisation of models, this chapter provides a foundation for constructing models that are both integrative and adaptable. The proposed Triangle Framework offers a practical tool for guiding design decisions, enabling modellers to balance conceptuality, formality, and executability in alignment with modelling goals. This structured approach reduces the risk of misrepresentation, supports modular development, and facilitates informed trade-offs during model evolution. Furthermore, the empirical study of security models demonstrates how the framework can be applied in practice, offering insights into current modelling trends and highlighting strategies for improving model interpretability and validation. Collectively, these contributions make the chapter a valuable resource for practitioners aiming to build models that are rigorous, interoperable, and suited to real-world decision-making contexts.

Chapter 4

Metaphors

We are prisoners of our own metaphors, metaphorically speaking ...

R. Buckminster Fuller

As we have seen in the previous chapter, conceptualising systems and designing and constructing models of systems can become a really complicated endeavour when considering aspects of their heterogeneity. Even if elements such as the means of construction illustrated in the triangle framework, the modelling goals, and the implications derived from the acceptance of a de-facto multi-methodological nature of models are relevant to model understanding, they alone are not enough to ensure that models in the world will perform as expected, or as desired by the people involved in their creation.

However, building on observations from Chapters 2 and 3, we cannot ignore the fact that the first triangle configuration that a model manifests — when considered over time — is highly conceptual. In other words, all models start as conceptual and can then evolve to other configurations, but it feels reasonable to believe that something that cannot be conceptualised, thought about or imagined cannot be built — even in the case of machine learning models constructing other models, the algorithm itself was firstly conceptualised before being formalised and then implemented, the software and hardware architectures were designed, the data types involved were accounted for, etc. In this context, the idea that we cannot separate conceptualisations from the conceptual frameworks used by the people involved in the conceptualisations is even more relevant.

The aims of this chapter are to present the conceptual framework employed by our modelling methodology explicitly — namely the distributed systems metaphor — and to describe a way of utilising it such that the resulting models have a higher probability of achieving their goals and behaving as expected. The chapter addresses the second, third, and fifth research questions from Section 1.1.

Section 4.1 describes the distributed systems metaphor, following the model qualities showcased in the triangle framework, and then showcases a set of useful properties that this conceptualisation offers in a modelling context. Subsection 4.1.1 illustrates the primary elements of the metaphor at a reasonably high level and reasons about the meaning of model construction. Subsection 4.1.2 gives an overview of the formal representation of these concepts, offering a rigorous foundation to the formalism and justifying its correctness. Subsection 4.1.3 demonstrates how executable models can be constructed using a library we have developed for the Julia programming language and then reflects on the stochastic nature of the resulting models as means of dealing with representational uncertainty. Subsection 4.1.4 describes the properties of the DS metaphor.

Section 4.2 introduces the trading zone metaphor and some associated notions such as boundary objects or interactional expertise, places the metaphor in a modelling context, describes types of trading zones and determines a specific trading zone evolution trajectory — from fractionated towards inter-language — as useful for modelling heterogeneous systems.

Section 4.3 does the comparison between the two metaphors at the level of entities, interaction, language, methods, practices, and goals — trading zone and distributed systems metaphor — and determines them as compatible.

As noted in the research paper declaration form, this chapter contains elements of the author's previously published work. More specifically, section 4.1 includes elements of section 5 from [152], sections 2, 3 and 4 from [62], and section 3 from [151]. Similarly, sections 4.2 and 4.3 are derived from section 5 from [152].

4.1 Distributed Systems Metaphor

4.1.1 Conceptual aspects

The expansion of interconnected network systems gave rise to the formulation and advancement of distributed systems theory in the field of computer science. Under this paradigm, a distributed system can be thought of as a collection of physically independent entities that communicate and coordinate with each other to achieve a common goal. The entities have been traditionally related to computing, so technological in nature — in the case of heterogeneous systems, this is no longer the case —, the communication is performed across a network, the coordination is realised via scheduling algorithms, and the common goal is represented by the service the system is supposed to provide to its users.

While this theory has historically been focused on computer systems, its principles and core components can be applied more broadly as a useful metaphor for understanding various types of heterogeneous systems, including ecosystems. In previous work such as [62], we describe the ontology of the distributed systems metaphor as containing the following primary concepts:

> **Location**: Distributed systems inherently involve the idea of multiple locations interlinked with each other. These locations may represent physical entities — such as rooms connected by passageways —, logical entities — like addresses in computer memory connected by memory pointers —, or abstract entities — for instance, the conscious and unconscious areas of cognition interlinked by dreams. Formally, locations are represented by directed graphs.
>
> **Resource**: Resources exist at locations and can be moved between them according to the locations' connections. Generally, they can be used to represent anything that can be manipulated by a process — data, physical and abstract objects, people, etc. For example, computers in rooms, software programs loaded in computer memory, or memories in the unconscious can all be viewed as resources. Formally, resources can be represented as elements

of a pre-ordered, partially commutative monoid, as provided by the semantics of bunched logic [228, 74].

Process: Processes are used to represent collections of actions that manipulate resources in a sequential or parallel manner. Such manipulations can include moving resources from one location to another, generating or removing resources from locations, composing resources, altering the internal structure of a resource via decomposition, and so on. Formally they can be represented — in the spirit of Milner's SCCS [211] — using a monoid of basic actions, a grammar of process terms to describe process interaction and a partial modification function illustrating the co-evolution of processes and resources under actions.

Environment: The systems we model are not isolated entities; instead, they interact with environments that we choose not to represent in detail. Such environments are used to depict the worlds outside the system of interest and the interactions between such worlds and the system. Semi-formally, environments can be represented stochastically by capturing the incidence of events they generate towards the model. For example, given an organisational physical security model, the arrival rate of agents at the entrance that marks the outer boundary of the model may be captured using a negative exponential distribution [64].

Interface: Building on the isolation aspect, models do not interact only with environments, but also with other models — and the same could be said about systems directly. Under our framework, this interaction is enabled by the composition operation — which we will detail in Subsection 4.1.2. To enable composition, models need interfaces that define the locations at which models fit together and which actions, defined at appropriate locations within the interface, are part of the composition. Actions in the interface will nevertheless be able to execute only if the resources they require are available.

This view is obviously very specific to its focus on computer systems, but its concepts can be taken more generally to provide a useful metaphor for understanding all types of systems — and ecosystems. However, we have not actually defined what it means to *build* a model using this distributed systems approach — although the constitutive elements of the triangle framework as described in Chapter 3 already imply it. This, too, is very flexible. Models can be largely conceptual and use the notions described above as a means to help think about the structure and behaviour of a system. Or models under this paradigm can be formal, as we will show in the following sub-section. Finally, this metaphor can be used to build executable models, in the spirit of Birtwistle's Demos [37], where a programmatic description of the system (in terms of locations, resources, processes, interfaces, and environments) is run to simulate the behaviour of the system [61, 64, 59, 74]. In the following subsections, we shall provide more details regarding formal and executable models.

4.1.2 Formal aspects

The formal foundations of this work have previously been developed in [10, 62, 76, 75, 74, 127, 248]. However, to present a complete view of the distributed systems metaphor and to understand its advantages — even if not always in a similar formal setting — we describe them below.

We begin by giving a formal framework for capturing the distributed systems metaphor that we are proposing as a basis for a semantically and logically well founded framework for modelling ecosystems of systems in the absence of locations. The basic theory of processes and their associated logics is technically essentially determined by the interaction between processes and resources, with locations playing a significant conceptual role only when the concepts of interface, substitution, and local reasoning are considered, which we do below. The results presented in this section for states R,E extend to states L,R,E [74].

4.1.2.1 Processes and Resources

The starting points are Milner's synchronous calculus of communicating systems, SCCS [211] — perhaps the most basic of process calculi, the collection of which includes also CCS [208], CSP [142], Meije [91], and their derivatives, as well as the π-calculus [210], bigraphs [212] and their derivatives — and the resource semantics of bunched logic [230, 250, 127, 248]. The key components for our purposes are the following:

- A monoid of actions, Act, with a composition ab of elements a, b and unit 1;
- The following grammar of process terms, E, where $a \in$ Act and X denotes a process variable:

$$E ::= 0 \mid a \mid a : E \mid \sum_{i \in I} E_i \mid E \times E \mid \ldots$$

Most of the cases here, such as 0, action, action prefix, sum, concurrent product, and recursion, will seem quite familiar. Mathematically, this notion of resource — which covers examples such as space, memory, and money — is based on (ordered, partial, commutative) monoids (e.g., the non-negative integers with zero, addition, and less-than-or-equals):

- each type of resource is based on a basic set of resource elements,
- resource elements can be combined, and
- resource elements can be compared.

Formally, we consider pre-ordered, partial commutative monoids of resources, $(\mathbf{R}, \circ, e, \sqsubseteq)$, where $\mathbf{R}$ is the carrier set of resource elements, $\circ$ is a partial monoid composition, with unit e, and $\sqsubseteq$ is a pre-order on $\mathbf{R}$. The basic idea is that resources, R, and processes, E, co-evolve,

$$R, E \xrightarrow{a} R', E',$$

according to the specification of a partial 'modification function', $\mu : (a, R) \mapsto R'$, that determines how an action a evolves E to E' and R to R'.

The base case of the operational semantics, presented in Plotkin's SOS style [244], is given by action prefix and concurrent composition, $\times$, and exploits the monoid composition, $\circ$, on resources:

$$\frac{}{R, a : E \xrightarrow{a} R', E'} \quad \mu(a,R) = R' \qquad \frac{R, E \xrightarrow{a} R', E' \quad S, F \xrightarrow{b} S', F'}{R \circ S, E \times F \xrightarrow{ab} R' \circ S', E' \times F'}.$$

This (rather general [211, 91]) notion of composition at the level of process does not explain the engineering concept of the composition of models, with its requisite notions of interface and substitution, that we discuss further.

Sums, which represent choices, recursion, and other combinators are defined in similar ways.

A modification function is required to satisfy some basic coherence conditions (in certain circumstances, additional structure may be required [10]): for all actions a and b and all resources R and S, and where $\simeq$ is Kleene equality,

- $\mu(1,R) = R$, where 1 is the unit action, and
- if $\mu(a,R)$, $\mu(b,S)$, and $R \circ S$ are defined, then $\mu(ab, R \circ S) \simeq \mu(a,R) \circ \mu(b,S)$.

This function specifies the *signature* of the model.

Sums and recursion are formulated in familiar ways:

$$\frac{R, E_i \xrightarrow{a} R', E'}{R, \sum_{i \in I} E_i \xrightarrow{a} R', E'},$$

where I is an indexing set , and

$$\frac{R, E_i[E/X] \xrightarrow{a} R', E'}{R, fix_i X.E \xrightarrow{a} R', E'},$$

where E_i is the ith component of a tuple of processes.

Of more interest is hiding,

$$\frac{R \circ S, E \xrightarrow{a} R' \circ S', E'}{R, (\nu S)E \xrightarrow{(\nu S)a} R', (\nu S')E'},$$

in which the resource S becomes bound to the process E. This construction replaces and generalises the restriction operation of calculi such as SCCS.

4.1.2.2 Logic

Process calculi such as SCCS, CCS, and others come along with associated modal logics [139, 209, 281, 295]. Similarly, the calculus sketched here has associated modal logic, MBI [75, 74, 10]. The basic logical judgement is of the form

$$R, E \models \phi,$$

read as 'relative to the available resources R, the process E has property ϕ'.

Building on the ideas of the bunched logic BI (e.g., [230, 250, 127, 248]) and its application to Separation Logic [255, 156], MBI has, the usual *additive* connectives, $\top$, $\wedge$, $\rightarrow$, $\bot$, $\vee$.

These are all defined by semantic clauses of a satisfaction relation, where $\mathcal{V}$ is an interpretation of propositional letters in the usual way — see, for example, [297] — beginning as follows:

$$R, E \models \mathrm{p} \quad \text{iff} \quad (R, E) \in \mathcal{V}(\mathrm{p})$$

In addition, MBI also has a *multiplicative* conjunction, $*$,

$$\begin{aligned} R, E \models \phi * \psi \quad \text{iff} \quad & \text{there are } S, T \text{ and } F, G \text{ s.t. } S \circ T \sqsubseteq R, F \times G \sim E, \\ & \text{and } S, F \models \phi \text{ and } T, E \models \psi \end{aligned} \tag{4.1}$$

where $\sim$ is bisimulation (see, e.g., [211, 281, 295]) of processes, together with a multiplicative implication, $-\!\!*$. Note that the truth condition for $*$ — sometimes called a 'separating conjunction', since its conjuncts use separate resources — combines the resources from the truth conditions for its component formulae.

The relationship between truth and action is captured by the clauses of the satisfaction relation for the (additive) modalities, given essentially as follows (recall that $R' = \mu(a,R)$):

$$\begin{aligned} R,E \models \langle a \rangle \phi \quad &\text{iff} \quad \text{there exists } E' \text{ s.t. } R,E \xrightarrow{a} R',E' \text{ and } R',E' \models \phi \\ R,E \models [a]\phi \quad &\text{iff} \quad \text{for all } E' \text{ s.t. } R,E \xrightarrow{a} R',E',\ R',E' \models \phi \end{aligned}$$

A characterisation of hiding is also available [75]. Similarly, in addition to the usual additive quantifiers and modalities, MBI has multiplicative quantifiers and multiplicative modalities [75, 74] (we elide the details of MBI's predication).

The basic connection between the process calculus and the logic is given by a form of van Benthem-Hennessy-Milner theorem that relates process equivalence, as given by bisimulation, and logical equivalence (e.g., [139, 209, 281, 75, 295, 74, 10]), for MBI, defined by

$$R,E \equiv_{\mathrm{MBI}} R,F \quad \text{iff} \quad \text{for all } \phi,\ R,E \models \phi \text{ iff } R,F \models \phi$$

For image-finite processes E and F and any R, [75, 74, 10],

$$R,E \sim R,F \quad \text{iff} \quad R,E \equiv_{\mathrm{MBI}} R,F \tag{4.2}$$

Under stronger assumptions about the nature of resources [10], or with restrictions to the logic [74], this equivalence can be extended to pairs R,E and S,F of states with distinct resources.

Logics based on the language of MBI have proved valuable in program analysis — see the Infer tool [109] — partly by virtue of their deployment of local reasoning, based on the connective $*$.

4.1.2.3 Locations

As we have discussed, a key conceptual component of the distributed systems metaphor that we propose as a basis for a semantically and logically well-founded framework for modelling heterogeneous systems is location, logical or physical.

In general, we can identify a few requirements for a useful notion of location in systems modelling. Specifically,

- a collection of basic locations,

- directed connections between locations,

- a notion of substitution, which respects connections, and

- (optionally) a (monoidal) product of locations (a technical requirement).

In the presence of locations, the judgements for the transition relation for model states and the associated logical truth, respectively, take the forms

$$L,R,E \xrightarrow{a} L',R',E' \quad \text{and} \quad L,R,E \models \phi,$$

where the property ϕ of the process E holds relative to resources R at location L; that is, if a is an action guarding (the rest of) E, then $\mu(a,L,R)$ is defined, but are otherwise defined similarly as above [74, 10].

4.1.2.4 Interfaces, environments and models

Closely following [64, 59], we describe interfaces more formally using well-motivated simplifications (that are, in fact, convenient to implement [64]) of the general semantic set-up [75, 74, 10].

Models in this methodology are designed to be composed with other models or environments. Composition allows two or more models or environments to be combined and the resulting behaviour explored. When models are composed there are interactions at the location, process, and resource levels, and the role of their intended environments is critical. Processes evolve (transition) and resources are moved between models at locations shared between the models. To enable composition, models need interfaces, which define the locations at which models fit together and which actions, defined at appropriate locations within the interface, are party to the composition. Actions in the interface will nevertheless be able to execute only if the resources they require are available.

As an illustrative example, Figure 4.1 depicts three models which compose together. When models with interfaces are not composed, the environment generates the events expected by the interface; when composed, the environment is replaced by a model. Also shown is an example of substitution: `Model C` can be substituted for `Model B` as the interfaces of the two models match; this allows a modeller to refine or increase the level of detail in parts of a larger model.

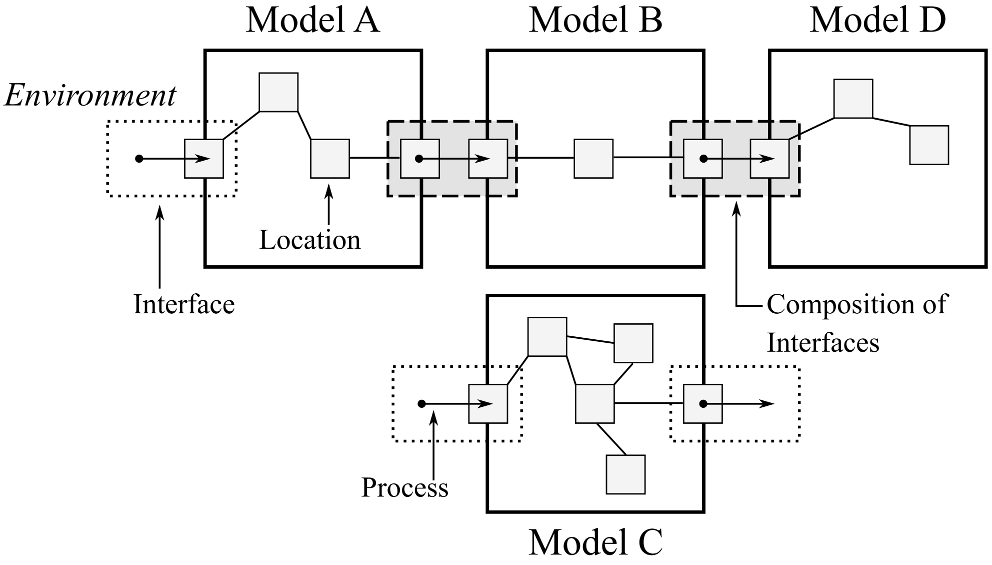


Figure 4.1: Interfaces, Composition, and Substitution [62]

The locations and resources of a model are represented using a location graph, $\mathscr{G}(\mathscr{V}[\mathscr{R}],\mathscr{E})$, with a set of vertices, $\mathscr{V}$, representing the locations of the model, and a set of directed edges, $\mathscr{E}$, giving the connections between the locations. Vertices are labelled with resources $\mathscr{R}$. Rather than thinking of actions evolving processes, it is convenient to think of a process as a trace of actions — the history of actions that have evolved a process during the execution of the model. All of the actions in a model are contained in a set, $\mathscr{A}$, and process traces are comprised of these.

The environment a model sits inside causes actions within the model to be executed, at a particular location. A model contains a set of located actions, $\mathscr{L}$, and a located action, $l \in \mathscr{L}$, is given by an ordered pair $l = (a \in \mathscr{A}, v \in \mathscr{V})$. The environment associates these located actions with probability distributions: $Env : \mathscr{L} \to ProbDist$. During the execution of the model, the located actions are brought into existence by sampling from these distributions.

Writing I for the set of interfaces on a model, then an interface $I \in I$ on a model is a tuple (In, Out, L) of sets of input and output vertices, where $In \subseteq \mathscr{V}$ and

$Out \subseteq \mathscr{V}$, and a set of located actions $L \subseteq \mathscr{L}$. The sets of input vertices and output vertices in interfaces must be disjoint; that is,

$$\bigcap_{i \in \mathscr{I}} In_i \in In = \emptyset \quad \text{and} \quad \bigcap_{i \in \mathscr{I}} Out_i \in Out = \emptyset.$$

Given this set-up, we can define a model as follows:

Definition 1 *A model $M = (\mathscr{G}(\mathscr{V}[\mathscr{R}], \mathscr{E}), \mathscr{A}, \mathscr{P}, \mathscr{L}, \mathscr{I})$ consists of a location graph $\mathscr{G}$, a set of actions $\mathscr{A}$, a set of processes $\mathscr{P}$, a set of located actions $\mathscr{L}$, and a set of interfaces $\mathscr{I}$. (Note that we can still consider the evolution of model states to be described as above.)*

Our notion of interface is related to Lynch and Tuttle's input/output automata [193].

4.1.2.5 Composition

Two models, M_1 and M_2 are composed using specific interfaces$I_{1,1}$, …, $I_{1,j}$, …, $I_{1,n} \in \mathscr{I}_1$ and $I_{2,1}, \ldots, I_{2,k}, \ldots, I_{2,m} \in \mathscr{I}_2$ using the composition operator, to give $M_{1 I_{1,j}}|_{I_{2,k}} M_2$, which is defined using an operation $\oplus$ on each of the elements of a model. First, we define the $\oplus$ operator for vertices and edges, $\mathscr{V}_1 \oplus \mathscr{V}_2 = \mathscr{V}_1 \cup \mathscr{V}_2$ and, for each $v \in \mathscr{V}_1 \oplus \mathscr{V}_2$, and then

$$v[\mathscr{R}_1 \oplus \mathscr{R}_2] = \begin{cases} v[\mathscr{R}_1] & \text{if } v \in \mathscr{V}_1 \wedge v \notin \mathscr{V}_2 \\ v[\mathscr{R}_2] & \text{if } v \in \mathscr{V}_2 \wedge v \notin \mathscr{V}_1 \\ v[\mathscr{R}_1 \cup \mathscr{R}_2] & \text{otherwise.} \end{cases} .$$

Composition of edges, actions, and processes are straightforward: $\mathscr{E}_1 \oplus \mathscr{E}_2 = \mathscr{E}_1 \cup \mathscr{E}_2$, $\mathscr{A}_1 \oplus \mathscr{A}_2 = \mathscr{A}_1 \cup \mathscr{A}_2$, and $\mathscr{P}_1 \oplus \mathscr{P}_2 = \mathscr{P}_1 \cup \mathscr{P}_2$.

To define the $\oplus$ operator for locations and interfaces, we first need to introduce some notation. The interfaces on a model are a set of tuples; for example, the interfaces of M_1: $\mathscr{I}_1 = \{(In_1, Out_1, L_1)_i\}$. A particular interface from $\mathscr{I}_1$ is referred to as $I_{1,i}$, and the input locations from that interface are referred to as $In_{1,i}$, the outputs as $Out_{1,i}$, and the located actions as $L_{1,i}$.

When models are composed, the located actions in the interface that were executed by the environment in the uncomposed model are now executed as a consequence of the other model instead. As such, the composition of located actions is the union of both sets of located actions, minus those that are in interfaces used in the composition: $\mathscr{L}_1 \oplus \mathscr{L}_2 = \mathscr{L}_1 \cup \mathscr{L}_2 \setminus \{L_{1,j}, L_{2,k}\}$.

Interfaces can be used in just one composition, and the input and output locations of the interfaces from the two models must correspond, so their composition is $\mathscr{I}_1 \oplus \mathscr{I}_2 = (\mathscr{I}_1 \cup \mathscr{I}_2) \setminus \{I_{1,j}, I_{2,k}\}$, where we require $\bigcup_{j=1}^{n} In_{I_{1,j}} = \bigcup_{k=1}^{m} Out_{I_{2,k}}$ and $\bigcup_{j=1}^{n} Out_{I_{1,j}} = \bigcup_{k=1}^{m} In_{I_{2,k}}$. Models must be composed completely: any location that is in both of the models must belong to the interfaces used in the composition.

Definition 2 *With the data as established above, the composition of models M_1 and M_2 is given by*

$$M_1{}_{I_{1,j}}|_{I_{2,k}}M_2 = (\mathscr{G}((\mathscr{V}_1 \oplus \mathscr{V}_2)[\mathscr{R}_1 \oplus \mathscr{R}_2], \mathscr{E}_1 \oplus \mathscr{E}_2), \mathscr{A}_1 \oplus \mathscr{A}_2, \mathscr{P}_1 \oplus \mathscr{P}_2, \mathscr{L}_1 \oplus \mathscr{L}_2, (\mathscr{I}_1 \oplus \mathscr{I}_2))$$

with the constraint that $\mathscr{V}_1 \cap \mathscr{V}_2 = In_{1,j} \cup In_{2,k}$. (This constraint above represents a significant design choice in the definition of interfaces.)

Proposition 1 ([64, 62]) *$M_1{}_{I_{1,j}}|_{I_{2,k}}M_2$ is a model.*

Proposition 2 ([64, 62]) *For any models M_1 and M_2, let $I_{1,2}$ be the subset of interfaces in $\mathscr{I}_1$ that compose with M_2. Composition of models is commutative and associative: $M_1{}_{I_{1,j}}|_{I_{2,k}}M_2 = M_2{}_{I_{2,k}}|_{I_{1,j}}M_1 and$ $(M_1{}_{I_{1,2}}|_{I_{2,1}}M_2)_{I_{1,3} \cup I_{2,3}}|_{I_{3,1} \cup I_{3,2}}M_3 = M_1{}_{I_{1,2} \cup I_{1,3}}|_{I_{2,1} \cup I_{3,1}}(M_2{}_{I_{2,3}}|_{I_{3,2}}M_3)$*

So far, this definition of interface says little about how a model becomes animated. How this actually works is that a model is animated when events occur at its boundaries. These ideas can be conveniently illustrated by considering the concept of substitution, together with a simple example. As we have seen,models exist within environments and, as we have remarked, environments are captured within our framework stochastically. In fact, our treatment of environment — that is, that part of a model that is not captured in detail, using the distributed systems structure of locations, resources, and processes — is rather simple.

These issues will be clear by a simple example: the conveyor belt, represented using the language of our distributed systems metaphor, and explain how it can decomposed into two component subsystems using an appropriate choice of interface. Figure 4.2 depicts a conveyor belt in which resources *r* are moved along from right to left, with *in* and *out* locations at either end.

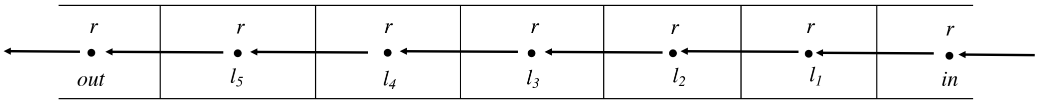


Figure 4.2: A conveyor belt [62]

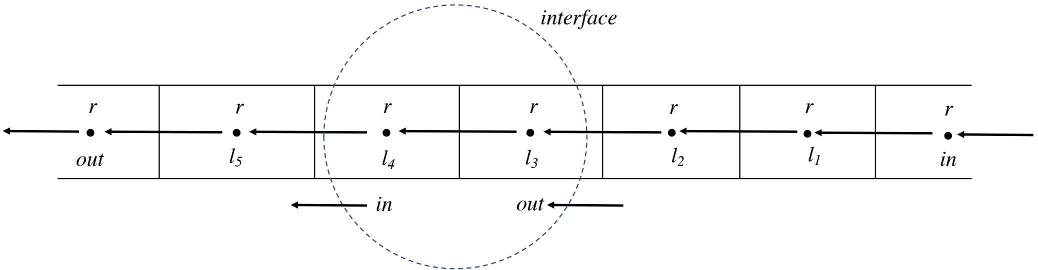


Figure 4.3: A composite conveyor belt [62]

The signature for this model, as described by its modification function, can be specified as follows:

$$\begin{array}{rclrcl} \mu(move(r,in,l_1),in,r) & = & (l_1,r) & \mu(move(r,l_5,out),l_5,r) & = & (out,r) \\ \mu(move(r,l_k,l_{k+1}),l_k,r) & = & (l_{k+1},r) & otherwise & & \uparrow \end{array}$$

where, as usual, $\uparrow$ denotes ‘undefined’.

The process-component of the model is then defined, recursively, as follows:

$$\begin{array}{rcl} ConBelt & ::= & (move(r,in,l_1):ConBelt \times move(r,l_1,l_2):ConBelt \times \\ & & \ldots \times move(r,l_5,out):ConBelt) + 0 \end{array}$$

Then the system *Ls* , *Rs* , *ConBelt*, where we right *L*s and *R*s for the evident lists of locations and resources, describes the basic operation of a conveyor belt, as depicted in Figure 4.2. Either the belt moves with each section in lockstep or it stops (0 denotes termination).

Consider now a conveyor belt that consists of one belt passing on its items to another, perhaps because different machines are used to process the items on different belts. We can use our idea of interfaces to describe how to think of our *ConBelt* as the composition of two, component, ConBelts. The set-up is depicted in Figure 4.3. Here we can see that the conveyor belt can be understood as the composition of two such belts, the right-hand one of which has l_3 as it's *out* location, which then leads to the *in* location, l_4, of the right-hand one. The interface consists of the two locations, l_3 and l_4, together with their associated data.

4.1.2.6 Substitution

As we have briefly discussed, the construction of models of complex systems may require the substitution of one component model for another; for example, perhaps, to either increase or reduce the level of detail; or, perhaps, to explore a quite different design for a part of a model; or, perhaps, to replace part of the environment with a specific model. The typical situation is, more or less, as depicted in Figure 4.4: a model N has components M_1 and M_2 connected by a model Q. We seek to replace Q with the model P.

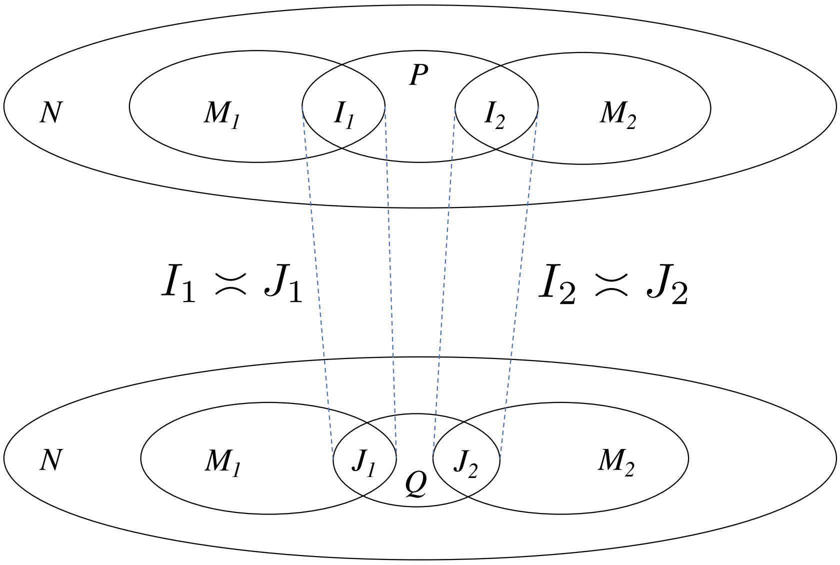


Figure 4.4: Interfaces and Substitution [62]

For simplicity, denote the interfaces between M_1 and Q and Q and M_2 — formally defined as composites, as above — by J_1 and J_2, respectively. Similarly, suppose that P, which replaces Q, has interfaces I_1 and I_2 to M_1 and M_2.

For substitution to behave as required, what must we require of P, I_1, and I_2? We identify the following requirements: (i) the pairs of substituting and substituted interfaces should be able to simulate one another; (ii) the distributions of the events that are incident upon the corresponding boundaries of the interfaces should be the same, up to choices of parameters. These two conditions together give us what we need: let $\mathscr{D}_M(L)$ denote a set of pairs of probability distributions and locations in a model M. We write $d_1 \asymp d_2$ if d_1 and d_2 are distributions that are the same up to choices of parameters and extend this notation to sets $\mathscr{D}_M(L)$. Then, we require:

- $\mathscr{D}_N(In_{I_1}) = \mathscr{D}_N(In_{J_1})$ and $\mathscr{D}_N(Out_{I_1}) = \mathscr{D}_N(Out_{J_1})$
- $\mathscr{D}_N(In_{I_2}) = \mathscr{D}_N(In_{J_2})$ and $\mathscr{D}_N(Out_{I_2}) = \mathscr{D}_N(Out_{J_2})$
- for $i = 1, 2$, abusing notation a little, $I_i \asymp J_i$.

Consider the example of a substitution depicted in Figure 4.5, in which a small-scale road map of the roads in and out of a city is replaced by a larger scale map, which has more detail of the topography of the city. The relevant interfaces here are simply the points of contact between the roads within the city and their connections in the environment, together with their associated probability distributions.

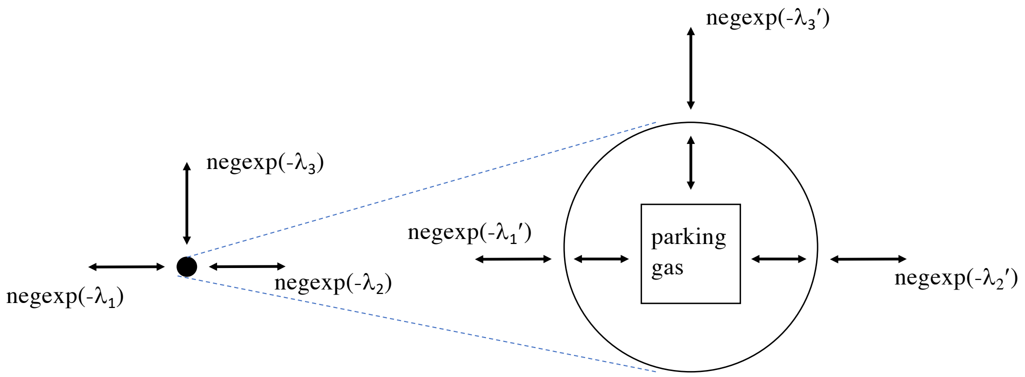


Figure 4.5: Substitution [62]

The logic MBI allows us to assert some useful properties. For example, if $\mathscr{S}$ (i.e., some L, R, E) and $\mathscr{S}'$ (i.e., some L', R', E') denote states (we elide details) of the smaller and larger scale models, then we can write

$$\mathscr{S} \models \phi \quad \text{and} \quad \mathscr{S}' \models \phi'$$

where — writing c for a car, g_1, g_2, g_3 for the three city gates, and t and u for time periods, all as parameters for actions in the evident way — we can assert $\phi = [enter_{c,g_1}]\top \rightarrow (\langle exit_{c,g_2}\rangle\top \vee \langle exit_{c,g_3}\rangle\top)$ and

$$\phi' = [enter_{c,g_1}](\langle park_{c,t}\rangle\top \vee \langle gas_{c,u}\rangle\top) \rightarrow (\langle exit_{c,g_2}\top\rangle \vee \langle exit_{c,g_3}\rangle\top)$$

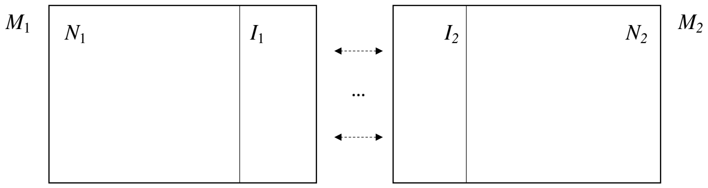
Here, just as in the transition from $\mathscr{S}$ to $\mathscr{S}'$, we give greater detail of the properties that may hold of a city location. Note that the exit possibilities are not the only such possibilities (e.g., a car may remain in the town and never leave).

4.1.2.7 Local Reasoning

In this section, we introduce the concept of local reasoning, first introduced in the context of Separation Logic [156, 256, 309]. This conceptual design facilitates the ability to reason locally about the underlying components of systems or ecosystems.

The primary advantage of this is that the properties of a specific component in a decomposition of a models can be reasoned about without the need to reason about other components other than in respect of the interfaces to the specific component. Consequently, modularity (and substitution) are supported, with the conceptual and computational complexity of reasoning constrained.

With respect to local reasoning, we argue that the combination of the mathematical foundations sketched in this subsection and the conceptual separation of components, as described in subsection 4.1.1, offers the ability to focus analyses on specific components and simply state the relevant aspects for intercommunication at the level of interfaces.



$$M = M_1 \; {}_{I_1}|_{I_2} \; M_2$$

Figure 4.6: Interfaces and Local Reasoning [62]

We can identify here a local reasoning principle, or *frame rule* [156, 229, 310, 227, 248].We begin by setting up some notation for the states of the various component models depicted in Figure 4.6:

- let the model $M = M_{1 I_1}|_{I_2} M_2$ have state $\mathscr{S}$;

- let the component models (of the composition of interest) M_i have states $\mathscr{S}_i$, respectively;

- let the submodels N_i have states $\mathscr{U}_i$, respectively; and

- let the interfaces I_i have states $\mathscr{I}_i$, respectively.

Now, using $\circ$ for composition of states, we assume the following, for $i = 1, 2$:

- $\mathscr{S}_i \sim \mathscr{U}_i \circ \mathscr{I}_i$, $\mathscr{S} \xrightarrow{a} \mathscr{T}$, and $\mathscr{I}_i \xrightarrow{a} \mathscr{J}_i$

- $a\#N_i \backslash I_i$; that is, that the action a is 'separated from' that part of the model N_i that is not coincident with the interface I_i in that the execution of a does not affect N_i.

Now, suppose that $\mathscr{U}_i \models \phi_i$, for $i = 1, 2$. Then we have the following frame rule:

$$\frac{\mathscr{J}_1 \models \psi_1 \quad \mathscr{J}_2 \models \psi_2}{\mathscr{T} \models (\phi_1 * \psi_1) * (\psi_2 * \phi_2)} \quad \begin{array}{l} \mathscr{S} \xrightarrow{a} \mathscr{T} \text{ and } \mathscr{I}_i \xrightarrow{a} \mathscr{J}_i \\ a\#N_i \backslash I_i \end{array}$$

This rule is sound with respect to bisimulation equivalence:

Proposition 3 (Soundness of the frame rule) *Suppose, for* $i = 1,2$, $\mathscr{J}_i \sim \mathscr{J}'_i$, $\mathscr{I}_1 \sim \mathscr{I}'_i$, *and* $\mathscr{S} \sim \mathscr{S}'$ *and* $\mathscr{T} \sim \mathscr{T}'$. *Then* $\mathscr{T}' \models (\phi_1 * \psi_1) * (\psi_2 * \phi_2)$.

Proof sketch. By (4.2), we have that that, for $i = 1,2$, $\mathscr{U}_i \models \phi_i$ and $\mathscr{J}_i \models \psi_i$. Then, note that separation condition, $a\#N_i \backslash I_i$, and the definition (4.1) of satisfaction for $*$ are respected by bisimulation. Finally, further applications of (4.2) then give the required conclusion.

To understand how all this works, consider again Figure 4.5 and suppose we have a model M for the part of the city that includes the parking and the gas station. That model is connected by interfaces — here again they are just point-to-point, respecting stochastic flows — to the rest of the more detailed model of the city. The facilities of the gas station and their operating capacities, which can be expressed logically, are properties of M that are independent of the model of the surrounding city. In this example, these properties correspond to the ϕ_is in the frame rule: separated by the multiplicative conjunction, $*$, they are invariant under changes to the surrounding model and the interfaces to it when the overall model evolves. The primary advantage of such a setting is that the modeller can confidently focus his analysis on a singular model component without the need to reason about its relationships with other components — the relevant aspects of intercommunication remain located at the interface level, acting as contracts that submodels have to fulfill in order for the composition to be possible.

Returning to our example of the conveyor belt, as depicted in Figures 4.2 and 4.3, suppose the two component belts are there to support two different sequences of operations:

- The right-hand belt performs actions op_1 and op_2 on the resources at locations l_1 and l_2, respectively;
- At l_3, in the interface, the correct completion of the operations op_1 and op_2 is verified;
- At l_4, in the interface, the readiness of the resources for the operations of the left-hand belt is verified;

- The left-hand belt performs the operation op_5 on the resource at l_5.

What can a frame rule say about this situation? First, we give the conveyor belt a bit more to do. Suppose that at locations l_1, l_2, and l_5, the operations op_1, op_2, and op_5, respectively, may — provided the machines servicing the belts are functioning correctly — be performed. Then, using MBI's modalities, as defined in Section 4.1.2.2,

$$in\,,\,r\,,\,move(r,in,l_1):ConBelt \quad\models\quad [move(r,in,l_1)]\,\langle op_1\rangle\top$$

since $move(r,in,l_1)$ takes our focus to location l_1, at which point op_1 may be performed, and nothing else happens to the resource r until it moves to l_2. Similarly,

$$l_1\,,\,r\,,\,move(r,l_1,l_2):ConBelt \quad\models\quad [move(r,l_1,l_2)]\,\langle op_2\rangle\top$$

These properties hold of that part of the right-hand belt that lies outside its interface to the left-hand belt. A similar logical judgement holds for the left-hand belt:

$$in\,,\,r\,,\,move(r,l_4,l_5):ConBelt \quad\models\quad [move(r,l_4,l_5)]\,\langle op_5\rangle\top$$

Again, this property holds independently of properties of the right-hand belt. Here we are assuming, for simplicity, that the belt(s) cannot stall or otherwise prevent the passing of resources from one location to the next — such a possibility would break our separation condition. This assumption, however, provides a clue to the use of the frame rule.

So far, our discussion of interfaces has been purely at the operational level: locations, actions, and so on. But the composition of models through interfaces might also be subject to some requirements that certain properties of the component models hold. That is, the composition

$$M_{1\,I_{1,j}}|_{I_{2,k}}M_2$$

might be made subject to conditions, following the notational convention set out above, as follows: $\mathscr{I}_{1,j} \models \phi_{1,j}$, for each j and $\mathscr{I}_{2,k} \models \phi_{2,k}$, for each k, specifying the required properties of the output from one model and input to the other.

Within our conveyor belt(s) example, we can set up an example of such a situation. Let $Op_1(r)$ and $Op_2(r)$ be propositions that denote that the resource has received the operations op_1 and op_2, respectively. Then we may impose the conditions

- On the output of right-hand belt:

$$l_3\,,r\,,move(r,l_3,l_4):ConBelt \models Op_1(r) \wedge Op_2(r)$$

- On the input to the left-hand belt:

$$l_4\,,r\,,move(r,l_4,l_5):ConBelt \models Op_1(r) \wedge Op_2(r)$$

In order to check that the two conveyor belts can be composed, we need only check that the resources arriving at l_3 have received the operations op_1 and op_2. Of course, the left-hand belt may require that the resources it receives also carry a certification that these operations have been performed. Such a certification might be delivered as part of a check at l_3 and a verification at l_4:

- Check: $l_3\,,r\,,move(r,l_3,l_4):ConBelt \models Check(op_1,op_2)$

- Validate: $l_4\,,r\,,move(r,l_4,l_5):ConBelt \models Validate(op_1,op_2)$

Again, checking these properties would be independent of those parts of the belts outwith their interfaces.

4.1.3 Executable aspects

The basic concepts that form the distributed systems metaphor have been used to construct executable frameworks for practical model construction. The earliest implementation attempted was Dahl & Nygaard's Simula [86], which was an Algol simulation framework that mainly focused on the notion of processes. Further implementations such as Birtwistle's Demos [37, 38] or Gnosis [74] focused on extending the conceptual tool-set to notions such as resources or locations. In the context of this thesis, we employ an implementation in the Julia language: the Julia `SysModels` package [61] is an improved, more modern implementation of these ideas that includes new capabilities such as composition of models, while at the same time simplifies the actual process of constructing models, since Julia does not maintain similar syntactical complexities to C or Algol.

4.1.3.1 The Julia implementation

The `SysModels` package provides the constructs needed to build models — locations, resources, processes, environments, and interfaces — as well as the ability to execute them. We will use a simple example to demonstrate how models are written and executed, showing how the basic components are combined to form a representation of a system.

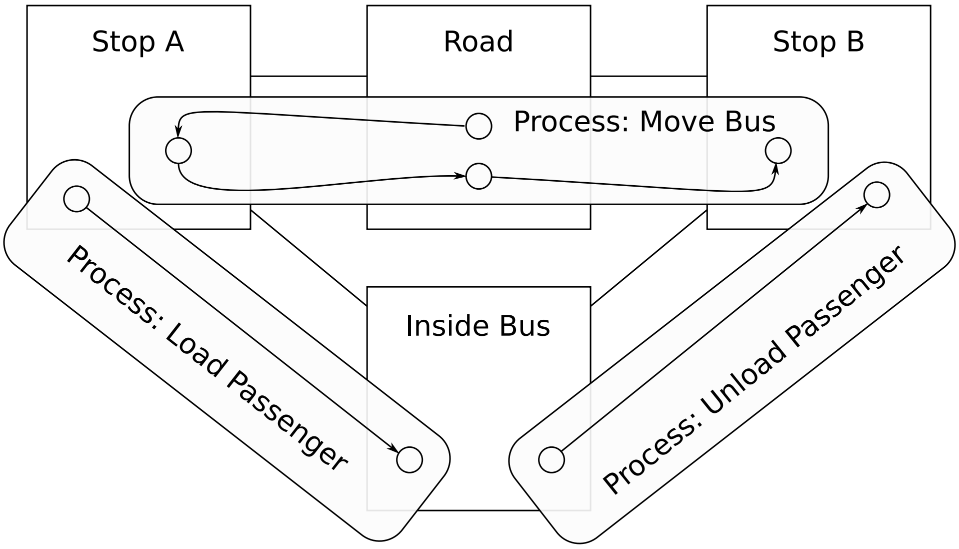


Figure 4.7: Example model: passenger travelling on a bus. Squares represent location and the rectangles depict the processes moving resources between the locations. [151]

In this example, a bus moves between two stops and a passenger waits for the bus to arrive, boards the bus, and gets off the bus when it arrives at the next stop. Figure 4.7 depicts this scenario. The first thing we define in our model is the location structure:

```
using SysModels
loc_stopA  = Location("Stop A")
loc_stopB = Location("Stop B")
loc_road  = Location("Road")
loc_bus  = Location("Inside Bus")
link(loc_stopA, loc_road)
link(loc_stopB, loc_road)
link(loc_stopA, loc_bus)
link(loc_stopB, loc_bus)
```

Here, we have defined the four locations in this model: two bus stops, a road between them, and a location representing the inside of the bus. We then use `link` to define how these locations are connected, which determines how resources may be moved from one location to another.

After this, we need to define the resources that are used in the model. In this case, there are two: a passenger resource and a bus resource, which are defined using Julia `structs`. We also use `distrib` to set the starting locations of each of these resources:

```
struct Passenger <: Resource
end
struct Bus <: Resource
end
pax = Passenger()
bus = Bus()
distrib(pax, loc_stopA)
distrib(bus, loc_road)
```

Next we define the processes in this model. We start with the process responsible for moving the bus:

```
function move_bus(proc :: Process)
    @claim(proc, (loc_road, bus))
    move(proc, bus, loc_road, loc_stopA)
    release(proc, loc_stopA, bus)
    hold(proc, 2minutes)
    @claim(proc, (loc_stopA, bus))
    move(proc, bus, loc_stopA, loc_road)
    hold(proc, 5minutes)
    move(proc, bus, loc_road, loc_stopB)
    release(proc, loc_stopB, bus)
end
```

Processes are written as Julia functions. The `SysModels` package provides functions for controlling processes and manipulating resources:

- `@claim` — before a process can manipulate (move, remove) a resource, it must claim it. If the requested resources are not present, the process waits until they are; if multiple processes are trying to claim the same resource, they implicitly queue for it.

- `move` — after a process has claimed a resource, it can be moved from one location to another.

- `release` — when a process has finished with a resource, it releases it. Other processes may then claim it.

- `hold` — this pauses the process for a specified amount of simulation time.

In the code above, the process first claims the bus resource at the road location, before moving it to stop A and then releasing it. It waits 2 minutes, claims it again, moves the bus to the road, waits 5 minutes, moves the bus to stop B, and finally releases it.

The processes for loading and unloading the passenger are written in the same way:

```
function load_passenger(proc :: Process)
    @claim(proc, (loc_stopA, pax) && (loc_stopA, bus))
    move(proc, pax, loc_stopA, loc_bus)
    release(proc, loc_bus, pax)
    release(proc, loc_stopA, bus)
end

function unload_passenger(proc :: Process)
    @claim(proc, (loc_stopB, bus) && (loc_bus, bus))
    move(proc, pax, loc_bus, loc_stopB)
    release(proc, loc_stopB, pax)
    release(proc, loc_stopB, bus)
end
```

Here, the load process waits until it has claimed *both* the passenger and bus resources. It then moves the passenger to the Inside Bus location, and then releases both resources. Similarly, the unload process waits until it has claimed both resources, and then moves the passenger to Stop B. Claiming the passenger resource means the processes can move that resource into or out of the bus; claiming the bus resource causes the processes to wait until the bus resource—moved by the bus process above—arrives at the correct location.

Finally, we have to set up the model and run it:

```
model = Model()
proc_bus = Process("Move Bus", move_bus)
proc_load = Process("Load Passenger", load_passenger)
```

```
proc_unload = Process("Unload Passenger", unload_passenger)
add_startup_process(model, [proc_bus, proc_load, proc_unload])
sim = Simulation(model)
SysModels.start(sim)
SysModels.run(sim, 2hours)
```

This creates the model object, creates the three processes, and sets them to start when the model begins executing. Then it creates the simulation, which handles the execution of the model, and runs it for 2 hours of simulation time.

Although not used in this simple example, the Julia implementation supports composition of models. Models can define interfaces, which specify the locations and actions involved in composition. Then, two models can be composed:

```
composed_model = compose(model1,interfaceA,model2,interfaceB)
```

The `compose` function here returns a new model from the composition of `model1` (using its `interfaceA`) and `model2` (using its `interfaceB`).

The ransomware recovery model presented later in this paper is naturally much larger and more complex than the example given here. However, it is still constructed in a similar manner, using locations, resources, and processes to represent the different elements of the systems being modelled, and composition to construct the whole model from smaller sub-models.

In the following lines, we present the Julia definitions for the notions of resource, location, process, interface, and models. For additional information, please check the Julia `SysModels` package available at ([61]).

- *Resource* — simply an abstract type used to derive resources with different characteristics.

  ```
  abstract type Resource end
  ```

- *Location* — any location must have a name, a dictionary of stores that contain entry points to the priority queue used by processes to claim resources, a dictionary of links with other locations and a vector of resources that can be found at that location. The js_properties attribute is related to the framework's logging capability and is not relevant from a distributed systems perspective. Furthermore, two additional types of locations, respectively input and output ones exist and will be used to define interfaces. Both of them contain a dictionary of functions that represent the operations available at the interface level, but the input ones also have access to a vector of environment processes that are used to perform actions in case the model is not composed with another.

```
mutable struct Location

name :: String
stores :: Dict{String, Store}
links :: Dict{Location, Bool}
resources :: Vector{Resource}
js_properties :: Dict{Any, Any}

function Location(name :: String)
        loc = new()
        loc.name = name
        loc.stores = Dict{String, Store}()
        s = Store()
        loc.stores["default"] = s
        loc.resources = Resource[]
        loc.links = Dict{Location, Bool}()

        loc.js_properties = Dict{Any, Any}()
        return loc
    end
```

```
end

 mutable struct InputLocation
    functions :: Dict{Type, Function}
    env_processes :: Vector{Process}

    function InputLocation()
        il = new()
        il.functions = Dict{Type, Function}()
        il.env_processes = []
        return il
    end
end

mutable struct OutputLocation
    functions :: Dict{Type, Function}

    function OutputLocation()
        ol = new()
        ol.functions = Dict{Type, Function}()
        return ol
    end
end
```

- *Process* — processes are built upon the already shown definitions. They must have an identifiable name, a flag to show if the process is currently being used by the process scheduler routine, a function to start and a task to complete, which are the Julia required types for handling process scheduling, an array of claimed resources that is being used to avoid race conditions and deadlocks between processes that attempt to claim the same resource and a simulation to be associated to.

```
mutable struct Process


name :: String
scheduled :: Bool
start_func :: Function
task :: Task
claimed_resources :: Vector{Resource}
simulation


function Process(name :: String, start_func :: Function)
        p = new()
        p.name = name
        p.scheduled = false
        p.start_func = start_func
        p.claimed_resources = Resource[]
        return p
    end
end
```

- *Interface* — interfaces are essentially just dictionaries of input and output locations. The operations that can be performed at the borders of the interface are present in the underlying data structures for input and output locations.

```
mutable struct Interface
input_locations :: Dict{String, InputLocation}
output_locations :: Dict{String, OutputLocation}


function Interface()
    i = new()
    i.input_locations = Dict{String, InputLocation}()
    i.output_locations = Dict{String, OutputLocation}()
    return i
    end
end
```

- *Model* — models contain dictionaries of interfaces and functions internal to the interfaces, a setup function to be used for operations required prior to the model construction, a list of environment processes since a model might not be composed with others on all its available interfaces and even then it could still be impacted by environment actions, a dictionary with all the available locations in the model, a dictionary of parameters to be used when executing and a dictionary to hold the relevant generated data.

```
mutable struct Model
interfaces :: Dict{String, Interface}
interface_funcs :: Dict{String, Dict{Type, Function}}

setup:: Function

env_processes :: Vector{Process}

locations :: Dict{String, Location}

params :: Dict{String, Any}
data :: Dict{String, Any}

function Model()
    m = new()
    m.interfaces = Dict{String, Interface}()
```

```
        m.env_processes = []
        m.locations = Dict{String, Location}()
        m.interface_funcs = Dict{String, Dict{Type, Function}}()
        m.params = Dict{String, Any}()
        m.data = Dict{String, Any}()
        m.setup = (mod :: Model) -> begin end
        return m
        end
    end
```

So far, we have not described how the stochastic environment, within which a model is situated, is represented and integrated with the structural definition of models that is described theoretically in Subsection 4.1.2 and implementated as above.

Our modelling set-up makes the choice not to incorporate stochastic definitions into the definition of the process algebra and its logic — as is done, for example, in PEPA [132]. Rather, the stochastic existence of actions is represented at the level of the implementation. That is, suppose an action *a* is defined in a model as in Subsection 4.1.2. In an implementation of that model, *a* may 'fire' when a specified distribution for it is sampled, and the model executes according to the execution of a scheduler.

While this approach simplifies the definition of the semantics, as sketched in Subsection 4.1.2, it necessitates providing an interpretation of the scheduler in the underlying semantics. This is done, for Gnosis, in [74]. The analysis for the Julia packages would be essentially the same.

4.1.3.2 Pragmatics, simulations, distributions

Up to this point, we have only sketched how the stochastic aspects of our models integrate with the approach above at the theoretical level. We now describe how the stochastic aspects of our models are used in practice. This is closely related to the approach of Demos [37, 95].

In elementary terms, executable models constructed using the above described approach behave like a structured Monte Carlo simulation [101, 314]. Stochastic processes are employed as means of dealing with uncertainty, but at the same time maintain the generic aspect of the model. For example, in the ransomware recovery model to be seen in Chapter 6, we are interested in the process of ransomware infecting devices. This might be influenced by the security posture of the organisation at the level of both employee training and awareness, the technical defence mechanisms present, the ransomware strain involved including the entry point of the attack, the industry in which the organisation that owns the devices operates, etc. However, not all these aspects are modelled explicitly: from the perspective of the ransomware and given our model goal, the relevant aspects are the infection probability, the attack duration and the ransomware behaviour — in our case the spreading mechanism. By experimenting with numerous combinations of parameters, we get to unpack the conceptual complexity of the features: a very high infection probability, a small attack duration, and a fast spreading behaviour can be used to represent a worm-like ransomware that is using a zero-day exploit, while a high infection probability longer attack duration and no spreading behaviour could mean a phishing attack on an organisation with poor security posture. This represents an implicit stochastic aspect of simulation models.

Furthermore, stochastic processes are being used explicitly to describe behaviour by means of sampling from probability distributions. This is generally used in the case of complex or uncertain behaviours. For example, ransomware spreading behaviour can be represented using two processes: one that chooses attack targets and another that determines the time of attacking the targets. To describe a slow paced phishing attack that does not employ particular internal knowledge about the organisation, we can use a uniform distribution for choosing targets at a regular, constant incremented time. Combined with the parameter describing infection probability, this already offers us the ability to represent slow-paced phishing attacks, but also fast-paced worm-like spreading ransomware in organisations with varying security postures.

On a final note, this approach is computationally intensive. The experimental space, with a large number of parameter combinations, requires a lot of computational resource—in the case of the ransomware recovery model presented in Chapter 6, a computing cluster—to execute in a reasonable amount of time. This is important, particularly since the number of model iterations and the size of the parameter space greatly influence the believability and representation power of the model.

4.1.4 Properties of DS Metaphor

In the previous subsections, we have described in detail the primary aspects of our distributed systems metaphor, along the conceptual direction of the triangle framework from Section 3.3.3. Yet, the utility of a metaphor lies not solely in its literal accuracy but in the extent to which it enhances understanding, facilitates communication, and elicits deeper insights. While the direct correspondence between the metaphor and its subject may be imperfect, the benefits derived from its usage can outweigh mere literal precision. Below, we outline the benefits of using the distributed systems metaphor approach in the context of modelling heterogeneous systems:

> **Generality**: This conceptualisation offers a set of ideas applicable for modelling virtually any type of system. Historically, notions of systems can be traced back to Plato, Descartes or the development of cybernetics and generalised systems theory — which specifically included various types of systems such as ecological, technological, biological, cognitive and social systems. In Chapter 3, we have illustrated multiple variations of the notion of system across different scientific fields. For a more detailed account, the reader can consult [114].
>
> **Recognizability**: Based on the generality aspect, there is no surprise that the above concepts have been studied and can be mapped across a wide array of scientific areas. For example, definitions of a notion of process — similar to the more general systems case — can be easily found in the fields of business

and management, economics, law, psychology, philosophy, physics, chemistry, computer science, mathematics, logic and others. It is worth noting that although these definitions are specialized for their corresponding areas — and expertise from many such areas might be required to model a heterogeneous system —, they still maintain the core idea of a collection of actions that leads to a form of result.

Scale-freeness: The above described set of concepts can be used for constructing representations at any level of abstraction. For example, a location could be an area in computer memory, a room, an office building, an entire city. Similarly, a resource could be a network data packet, a configuration file, an unit of energy, an amount of money or an entire fleet of IoT devices.

Formal Properties: The formal theory underlying these concepts describes three extremely useful properties for a modelling formalism: *composition*, *substitution* and *local reasoning*. *Composition* describes a structured way of constructing model components by connecting or combining smaller sub-components that inherently helps with managing system complexity — for example, the organisational ransomware recovery model described in Chapter 6 is composed of four underlying sub-models: a storage server model, a network model, a physical organisational model and a ransomware behaviour model. *Substitution* illustrates the necessary and sufficient structural conditions for a model component to be replaced with another. This ensures that a replacement of model sub-components would not bring the overall model into a misconfigured state and can be particularly relevant when adjusting the level of detail, exploring different designs for parts of the model, or replacing a portion of the environment with a specific model. For example, in the case of the emergency trauma unit model in Chapter 6, the environment generating patients based on arrival rates could be substituted with an explicit triage model. Last but not least, *local reasoning* refers to the process of making inferences or drawing conclusions about a specific part of a model without considering the entire model. The main benefit of this lies in the ability to analyse the properties of a particular component in a model's decomposition without the necessity to consider other components, except in relation to their interfaces or connection points with that specific component. A precise, formal description of these properties has been provided in Section 4.1.2.

Implemented Tools: The foundational principles underlying the distributed systems metaphor have been utilized to create practical frameworks for model construction. One of the earliest attempts at implementation was Simula [86], an Algol simulation framework that primarily emphasised processes. Subsequent implementations like Birtwistle's Demos [37, 38] or Gnosis [74] aimed to expand the conceptual toolkit to include elements like resources or locations. Furthermore, the authors propose a newer implementation in the Julia Language — specifically the publicly available SysModels package as described in [61] — incorporating more of the above described properties. Extended explanations about the resulting executable models have been provided in Section 4.1.3.

Identity Conservation: Constructing models using this approach satisfies the main criterion for the functioning of a trading zone, as shown in [278]. As long as the modelling participants have a common understanding of the concepts — process, resource, location, etc. —, a common identity for model components can be maintained in relation with them. We note here that maintaining this common sub-component identity implies first constructing it collaboratively. This facilitates the development of interactional expertise and knowledge sharing about the underlying sub-components: when debating the possible configuration of model sub-components, the participants inherently construct a personal understanding of them which can be directly shared because the set of concepts acts as an in-between language. More details about this in Sections 4.2 and 4.3.

4.1.5 Reflections on formal verification

As seen in the previous subsections, the constitutive elements of the distributed systems metaphor have been carefully selected, defined, and explained in alignment with the Triangle Framework introduced in Section 3.3.3.

In Section 5.3, we further elaborate on the possibility of translating between different model components within our methodology. For example, an executable component can be used to derive results, translate these results into executable consequences, interpret those consequences at the conceptual level, update the conceptual representation, and ultimately transform the executable model component into a formal one. We note here that such translations are possible across all corners of the triangle, meaning that any model subcomponent can be formalised without introducing external ontological commitments. As a result, formal verification — and rigorous conceptual verification of model consequences — is embedded by design. This illustrates how the Triangle Framework supports formal verification: conceptual models define the properties, formal components express them as logical constraints, and executable components enforce them dynamically.

This balance between flexibility and rigour is particularly important in cybersecurity contexts, where systems must provide guarantees around properties such as safety, security, policy compliance, reachability, resilience, and so on. Such guarantees are generally constructed at the level of models, not at the level of real systems, unless the systems themselves are equipped with mechanisms that enforce such guarantees. Even then, we argue that these mechanisms were necessarily preceded by a model — explicit or implicit — of their functioning and integration. For an example of this principle applied in industry, see [79] on how formal verification methods were integrated into the Amazon Web Services (AWS) technology stack. Unlike purely cooperative systems, cybersecurity models must account for adversarial agents, and our approach allows specifying constraints that capture both cooperative and malicious behaviours — for example, by ensuring that recovery processes remain available even under targeted denial-of-service attempts.

Importantly, the internal structure and translatability of our approach make it

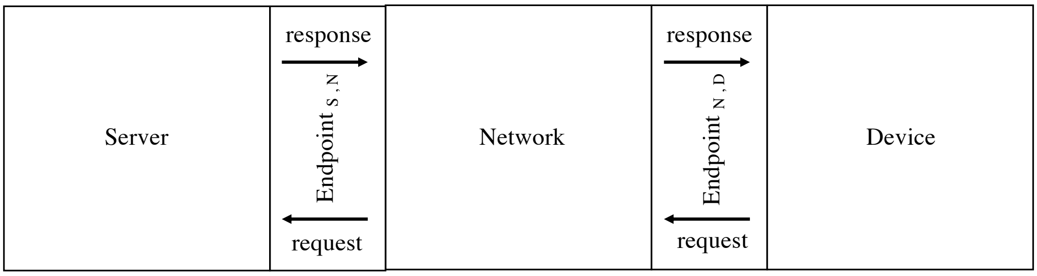


Figure 4.8: The Simplified Organisational Recovery Model [62]

straightforward to incorporate such guarantees directly into a model, without relying on external tools or software packages. We illustrate this point below with an example derived from the organisational ransomware recovery model presented in Section 6.3. The corresponding graphical representation can be seen in Figure 4.8. Because models are constructed compositionally, verification can be applied locally to sub-models — i.e. the network or device models — and then extended to the composed system, reducing complexity and improving scalability.

In this scenario, a number of devices may get infected with ransomware packages over a network, and use the same network to obtain recovery images from a server storing them. For this example, we consider two properties that we want to verify using dynamic formal verification. The first is that the number of infected devices never exceeds a specified threshold κ, which would endanger the network's operational integrity. The second is that the number of recovery requests being actively handled by the server does not exceed its concurrent processing capacity.

For the first property, we define the logical formula:

$$\phi_1 = (\mathrm{State}(\mathrm{monitoring\_proc}, \texttt{"active"}) \wedge \mathrm{Count}(\mathrm{InfectedDevice}, \mathrm{loc\_network\_ep}) \leq \kappa) \tag{4.3}$$

This property checks that the process responsible for monitoring the packets across the network is active and that the number of infected devices connected to the network at the network interface level does not exceed the safe threshold κ. In the Julia executable model, this would be implemented as:

```
state(monitoring_proc, ‘active’) &&
length(at(loc_network_ep, r -> r isa InfectedDevice)) <= k
```

The second property is that the number of active recovery requests being handled by the server must not exceed its maximum concurrent processing capacity. This ensures that devices are performing their recovery processes without overloading the infrastructure at the server endpoint. The requests are counted at the level of the network interface endpoint to which the server is connected. Formally:

$$\begin{aligned}\phi_2 = \big(&\mathrm{State}(\mathrm{recovery\_proc}, \texttt{"serving"}) \wedge \\ &\mathrm{Count}(\mathrm{ActiveRequest}, \mathrm{loc\_server\_ep}) \leq \mathrm{max\_concurrent\_requests}\big)\end{aligned} \tag{4.4}$$

Its implementation in Julia would be:

```
state(recovery_proc, ‘serving’) &&
length(at(loc_server_ep, r -> r isa ActiveRequest))
<= max_concurrent_requests
```

The '`state`‘ function is used to check process states, '`at`‘ filters resources at specific locations, and '`length`‘ returns the number of such resources from the corresponding vector, allowing us to express constraints over the dynamic configuration of the system. Additionally, processes use a pair of functions '`set_state`‘ and '`reset_state`‘ to declare when they enter and leave a particular state, essentially giving a label to an action or sequence of actions for reference by formulae.

Similarly, additional invariants can be specified to capture adversarial scenarios, such as preventing privilege escalation by compromised devices or ensuring that no unauthorised process gains access to critical resources. These constraints can be expressed as state-based invariants and dynamically verified during simulation.

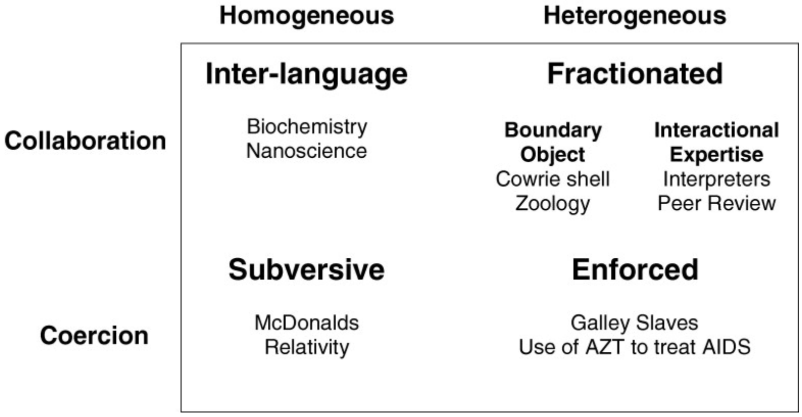
Though minimal, the example above illustrates how executable models built under the distributed systems metaphor enable both the formal verification and simple specification of cybersecurity properties. Integrating logical specification directly into executable model code in this way supports dynamic formal verification and operationally reinforces our methodology. However, we note that, when stochastic elements are included, the verification results may vary between model executions. In such cases, repeated executions are required to gain sufficient confidence that the model satisfies the specified formulae in the same Monte Carlo manner described in 4.1.3.2. For additional examples and an extended discussion on dynamic formal verification, see [55].

4.2 The trading zone metaphor

As coined by Peter Galison in [126, 125, 124], the term 'trading zone' describes situations where people from different disciplines or cultural backgrounds collaborate and communicate, despite having distinct languages, methods, and practices to achieve a high degree of understanding across a multidisciplinary domain of inquiry. The idea has been inspired by anthropological practice and was used by Galison to describe how physicists focused on different paradigms managed to jointly collaborate with engineers to construct the radar and particle detectors. However, its degree of generality is higher than that. In the author's own words:

> 'Two groups can agree on rules of exchange even if they ascribe utterly different significance to the objects being exchanged; they may even disagree on the meaning of the exchange process itself. Nonetheless, the trading partners can hammer out a local coordination, despite vast global differences. In an even more sophisticated way, cultures in interaction frequently establish contact languages, systems of discourse that can vary from the most function-specific jargons, through semi-specific pidgins, to full-fledged creoles rich enough to support activities as complex as poetry and metalinguistic reflection' [124].

Building on Galison's metaphor, [72] further extends the notion and classifies different types of trading zones by analysing the nature of the cooperation between participants and of the resulting culture. This results in the conceptualisation of four types of trading zones: inter-language, subversive, enforced, and fractionated. Figure 4.9 illustrates the four types of trading zones. For the purpose of this thesis, we will focus our attention on the fractionated trading zone. A more in depth analysis of each of the zone types and relationships with constructed and deployed models is deferred to further work.



	Homogeneous	Heterogeneous	
Collaboration	**Inter-language**	**Fractionated**	
	Biochemistry Nanoscience	**Boundary Object** Cowrie shell Zoology	**Interactional Expertise** Interpreters Peer Review
Coercion	**Subversive**	**Enforced**	
	McDonalds Relativity	Galley Slaves Use of AZT to treat AIDS	

Figure 4.9: The 4 types of trading zones [72]

The primary descriptors of a fractionated trading zone are high collaboration, heterogeneity of resulting culture, and fractions of culture as medium of interchange. Given the materiality aspect of the fractions of culture, the resulting trading zone can be based on either boundary objects or interactional expertise.

As described in [278], boundary objects are an ‘analytic concept of those scientific objects which both inhabit several intersecting social worlds and satisfy the informational requirements of each of them. Boundary objects are [...] both plastic enough to adapt to local needs and the constraints of the several parties employing them, yet robust enough to maintain a common identity across sites’. Furthermore, they can be both abstract and concrete as long as they remain recognisable at the level of each intersecting social worlds.

Interactional expertise, on the other hand, represents a form of expertise in understanding and using the language, concepts, and practices of a particular domain or community without possessing practical or hands-on skills in that domain. In [73], this is expressed as ‘enough expertise to interact interestingly with participants and carry out a sociological analysis’, but the sociological analysis element should be interpreted as specific to their topic of inquiry — to maintain generality, we interpret it as an analysis of the topic. Interestingly [72] also describes interactional expertise as a linguistic complement of boundary objects developing through linguistic socialisation.

The above described concepts can be easily observed at the level of modelling. In this setting, perhaps the most relevant aspect to notice is that the model to be constructed acts as a boundary object — the nature of heterogeneous models does not contradict [278], as long as a common identity for the model is maintained by the participants. Since the notion of heterogeneous model inherently implies a multidisciplinary domain, we can argue that the process of designing and constructing such models can be viewed as a fractionated boundary object trading zone.

However, as [72] notes, trading zones do not remain stable but tend to evolve and transition to different types over time. Figure 4.10 illustrates a possible trajectory for the evolution of a trading zone. In our case, a transition towards a subversive or enforced trading zone would not be desirable: the former implies one culture overwhelming the other participants', which brings the risk of biases being more easily introduced in the model, whereas the latter usually implies a lack of cultural interchange and therefore a reduction in multidisciplinary understanding of the phenomena under study. Furthermore, a transition towards an inter-language trading zone, although desirable in later stages of the modelling process due to the increased collaboration, is most of the times impossible due to time constraints and differences in goals of the participants. Stakeholders, users, and domain experts usually partake in modelling activities alongside other projects and do not inherently aspire to become modellers. Last but not least, attempting to maintain a classic boundary object trading zone also has its disadvantages, since the participants would essentially work 'in silo' and not share their knowledge and expertise — this can be very problematic when attempting translation and implementation of model components and lead to biases remaining hidden and conserved.

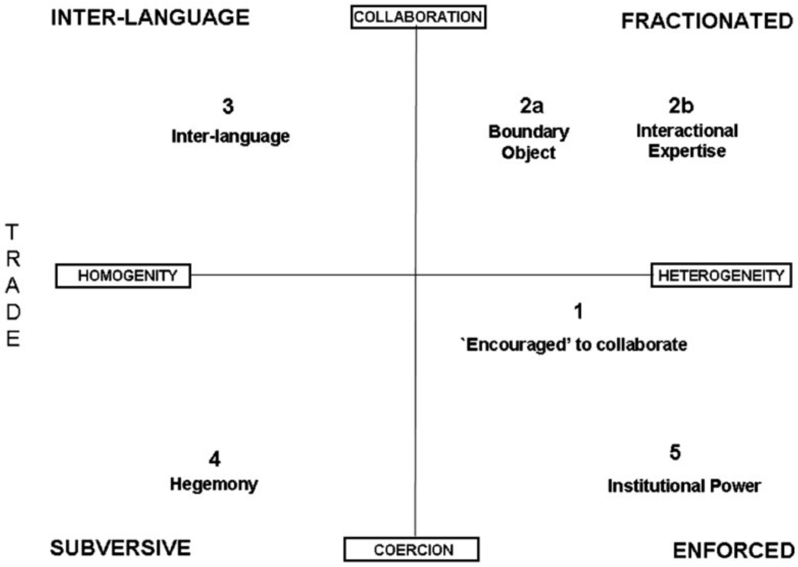


Figure 4.10: Evolution of a trading zone [72]

Therefore, the following trading zone evolution trajectory could be considered desirable: starting as a fractionated, boundary object trading zone — with or without initial coercion in the form of ‘encouragement’ during the start of the project — increasing the degree of collaboration and interest of the participants regarding the topic up to the point of development of interactional expertise to also facilitate knowledge sharing and then striving towards an inter-language trading zone by attempting to increase the homogeneity of the resulting culture via new cultural tools as described by [72]. One such conceptual tool is represented by the distributed systems metaphor that we have presented in detail in Section 4.1.

4.3 Comparing the 2 metaphors

In previous subsections, we have described the distributed systems metaphor and made arguments for its usefulness in a modelling context. We have also determined that it would be desirable for the modelling processes — described in more detail by the co-design cycle in Chapter 5 — to follow a specific trading zone trajectory: from fractionated towards inter-language. This implies homogenising the resulting culture via new cultural tools [72] and is associated with the development of in-between vocabularies along a relatively simple heuristic: the higher degree of homogeneity of the resulting culture, the higher the complexity of the resulting language — for example, a fractionated boundary object trading zone could be supported by a jargon, whereas an established inter-language trading zone is usually associated with a creole. We argue that the distributed systems conceptualisation can be considered as such a cultural tool and could act as an in-between vocabulary, but employing it in a trading zone context should only be done if the two metaphors are essentially describing similar situations.

The situation is slightly less complicated when considered only in a modelling context. Although, in most cases, the participants do not wish to become modellers and time and personal constraints hinder the establishment of a creole language — so achieving a complete inter-language trading zone is not usually possible — they can still achieve a basic level of understanding of each other via natural language. Because of that, one could argue that a common identity for model components could be maintained by a simple naming convention. However, that would not necessarily help with increasing the interactional expertise of the participants or the homogenisation of the resulting culture.

Our belief is that structuring the use of natural language with a set of concepts inspired by the theory of distributed systems represents a valid attempt at maintaining a common identity at the level of model sub-components while at the same time increasing the development of interactional expertise and knowledge sharing by design. We note that maintaining this common sub-component identity implies first constructing it. Involving all the modelling participants in this process not only facilitates direct knowledge sharing about the structure and behaviour of the components but also the development of interactional expertise, thus improving the homogeneity of the resulting culture.

The specific choice of a distributed systems conceptualisation is not arbitrary: a certain similarity exists between trading zones and distributed systems. Based on the definition of trading zone from Section 4.2, we illustrate the similarities with distributed systems metaphor below:

Entities: The primary constitutive elements of a trading zone are the people partaking in the ‘trading’. For distributed systems, the subsystems are composed of a mixture of people, technology related components, and policies. In both cases, the people can have different cultural backgrounds and areas of expertise; the technology components can be developed based on different scientific traditions; the policies might affect any of the components.

Interaction: The trading zone metaphor, being specifically focused on people, describes the main attributes of the interaction between parties as communication and collaboration. Similarly, the sub-components of a distributed system must communicate and coordinate their operational activity. The situation is more complex in the case of distributed systems, because the underlying entities can be different in their nature: people might have to interact not only with other people, but also with policies and technology based components.

Language, methods & practices: Based on the cultural and expertise differences, the participants in a trading zone might speak different languages and adhere to different sets of methods and practices. The same can be said for the distributed systems case, with the note that this is manifested also for technological components and policies. For example, computers might run different operating systems, execute code in various programming languages, vary in development and testing methodologies and have very specific sets of policies.

Goals: In both cases, multiple sets of goals exist. For example, trading zones exist because the participants obtain something by partaking in them — knowledge, economic benefits, reputation, etc. — but at the same time have an overall goal — for example to produce a boundary object — which would be hard to achieve in a different setting. At the level of a distributed system, the same can be observed: people have different incentives for being part of the system — a lot of them related to being part of the organisation that owns the system — technology components are usually designed for specific

> purposes, but their practical use can differ from that purpose and policies can sometimes even serve purposes that are not directly related to the system, but to some higher level goal — for example, a global organisational spending policy might drastically reduce design options for a system, without considering the implications for its construction. Yet, all the different components serve a common goal, to produce a service or product that is significantly harder or even impossible to produce in isolation.

Based on the above descriptions, we can argue that trading zones and distributed systems manifest a high degree of similarity at a conceptual level. Therefore, using a distributed systems conceptualisation in a trading zone context is not far fetched.

4.4 Conclusion

In this chapter, we have extensively and explicitly described the distributed systems metaphor, including its properties, along the directions of the triangle framework from Chapter 3. The reason for that was simple: if models can manifest the constitutive qualities of the triangle framework — conceptuality, formality, and executability — in various configurations, then the conceptual framework underlying the modelling process — namely the DS metaphor — must be usable in situations involving construction techniques related to all those qualities. In Section 4.1 we have indeed shown that the DS metaphor can be used to construct and reason about conceptual, formal, or executable models. Given the compositional nature of models conceptualised under this paradigm, this also shows that the distributed systems metaphor is suitable to handle cases of more complex, highly heterogeneous models.

Furthermore, we have described a second metaphor — the trading zone — and determined that it is compatible with the distributed systems metaphor, and that one can be employed in situations that have been described in the literature as representative for the other. Since the trading zone metaphor has not been, to the best of our knowledge, extended to formal or executable cases — in the sense that formal syntax, semantics, or programming languages or compilers explicitly developed from the trading zone metaphor have not been constructed —, the comparison was carried out at the conceptual level. This was motivated by the fact that a theory — or at least a set of justifiable empirical observations — describing the evolution of trading zones exists [72], and includes a criterion for the functioning of the trading zones — the identity conservation criterion [278]. Models underlined by the distributed systems metaphor satisfy this criterion because conceptually, the ontology of the metaphor acts as an in-between language which conserves sub-models identity. Following this argument, we have used examples about the evolution of trading zones to derive a possible trajectory of the evolution of a model during its design and construction phases that would have an increased chance of achieving its goals — because the modelling participants would better understand the pluri-perspectivist aspects of the phenomena under study. We note that this process is still somewhat limited by the knowledge and abilities of the people involved in it, although one could imagine a future in which such limitations could be lifted, or at least reduced via intelligent, but artificial agents also involved in the modelling process. For the time being, however, we posit that the inclusion of such agents would increase the degree of uncertainty involved in the modelling process and therefore reduce its believability.

Finally, we underscore the chapter's practical value for real-world modelling projects, where its methods and tools directly support the construction of models that are both rigorous and operationally effective. This chapter provides modellers with a practical toolkit for addressing the complexity of heterogeneous systems. By operationalising the distributed systems metaphor across conceptual, formal, and executable layers — by following the Triangle Framework —, it offers a coherent method of conceptualising heterogeneous systems that supports modular design, controlled refinement, and scalable integration during modelling. The inclusion of executable tools, such as the Julia-based SysModels package, bridges theory and practice, enabling simulation, stochastic analysis, and the ability to perform dynamic verification of critical properties. Combined with strategies for managing multidisciplinary collaboration following the trading zone evolution trajectory, these contributions make the chapter a valuable resource for building models that are both analytically rigorous and adaptable to real-world constraints.

Chapter 5

Methodology

My method leads me to posit the idea that existence precedes essence, which means that we must start from the subjective experience of the individual.

Jean-Paul Sartre

Up to this point, we have explored the nature of heterogeneous systems and models — in Sections 3.2, 3.3.1 and 3.3.2 — constructed a conceptual, descriptive framework for better understanding models based on three primary qualities related to their means of construction — in Section 3.3.3 —, and augmented it with the distributed systems metaphor view, in order to ensure a model evolution trajectory that would increase the modelling participants' interactional expertise, knowledge sharing, and produce a very useful set of properties — in Chapter 4 — resulting in models with higher chances of achieving their goals. However, to complete our methodology, we need to explicitly describe the modelling process that was extensively alluded to in previous chapters.

Therefore, in this chapter, we focus on describing our methodological approach to modelling based on updating the classical, mathematical modelling cycle by including ideas derived from the principles of co-design, the distributed systems metaphor, and the triangle framework. In doing so, we address the third, fourth, and fifth research question illustrated in Section 1.1.

In Section 5.1 we explicitly describe what we mean by co-design, make the distinction between co-design and co-creation, and place our research in the context of user centred design, participatory design, and participatory modelling by analysing some of the similarities and differences.

In Section 5.2, we briefly analyse the classical mathematical modelling cycle and showcase some of its caveats that need addressing, particularly regarding a set of key unstated assumptions about the domain of inference.

In Section 5.3, we present our version of co-design cycle which incorporates the definitions in Sections 3.3.1, 3.3.2 and the model qualities described in Section 3.3.3.

Lastly, in Section 5.4, we illustrate our notion of translation zone that mediates the dual evolution of domain knowledge and model focus, and explain its integration with the co-design cycle from the previous section.

As noted in the research paper declaration form, this chapter contains elements of the author's previously published work. More specifically, all sections include elements of Section 4 from [152], and Section 4 from [151].

5.1 Co-design

As defined in [172], 'co-design is the process in which actors from different disciplines share their knowledge about both the design process and the design content. They do that in order to create a shared understanding on both aspects, to be able to integrate and explore their knowledge and to achieve the larger common objective: the new product to be designed'. This formulation focused on the notion of process was developed as a response to the need of structuring and addressing collaboration and communication problems — in areas such as information gathering and sharing, problem analysing and understanding, concept generation and adoption, conflict resolution and others — observed in the design literature at the level of multidisciplinary teams. For a thorough analysis of the topic, see [51, 52, 83, 16, 53, 17].

For the purpose of clarity, we make the distinction between co-creation and co-design. As defined in [264], co-creation represents 'any act of collective creativity, i.e. creativity that is shared by two or more people' and is a more general concept than co-design because it refers to creative collaboration throughout more than the design oriented stages in the life-cycle of a product. From a modelling perspective, although co-creation could be viewed as a desiderata [300], usually it might be simply impractical to involve stakeholders, domain experts and users in all the model construction phases. Typical reasons for this impracticality are related to either resource constraints — for example, personal time constraints or remuneration issues because modelling is not always part of stakeholders' or domain experts' workload — or a lack of understanding of the practical construction methods, especially in the case of formal or executable models. Nevertheless, since model design stages heavily lean towards conceptuality as explained in Section 3.3.3, the above issue of lack of understanding becomes more addressable. Because of this pragmatic reason, we choose to focus on co-design rather than co-creation.

However, the idea of a co-design focused modelling process is not entirely new. Approaches such as user-centered design [4] or participatory design [277] have led to the appearance of participatory modelling [26], which represents an attempt at applying participatory design principles at the level of modelling. As defined in [299] participatory modelling is a 'purposeful learning process for action that engages the implicit and explicit knowledge of stakeholders to create formalized and shared representations of reality". Authors such as [6, 183, 221] associate a plethora of advantages such as a higher quality of system requirements, higher system quality, a better fit between the system and users' needs, improved satisfaction and mutual understanding of users or customers, development of differentiated new services with unique benefits, reduced development time, education of users, enhancing communication and cooperation between different people, and joint creation of new ideas, to the use of such collaborative design methodologies, particularly when related to service design and development.

In [94] model co-design is defined as follows: 'Model co-design is a process that engages modellers and system stakeholders cooperatively in the acts of objective identification and model specification, design and construction with the aims of aligning model objectives with the needs of the stakeholders, and designing a model that is feasible given the limits of data availability, which are discovered as parts of the process.'. While in principle we agree with this definition, we would like to make some additions to it. Firstly, it is not only modellers that should take part in the process, but also domain experts, users, and possibly people influenced by the future model implementation, deployment, and so on. Secondly, we do not conceptualise the acts of identification, specification, design and construction as objective, but rather as subjective, but justifiable, debated, and agreed upon by the participants in the process. Thirdly, while data availability is an important constraint, it is not the only one. For instance, three other constraints that easily come to mind are related to the degree of knowledge that the modelling participants have access to in regard to the object of modelling, and financial and time limitations — surely, in an ideal scenario the last two will not be an issue, but our focus is not on ideal scenarios.

The essential difference between our approach to co-design and participatory design [277] or participatory modelling [299] is that the participants do not provide expertise only at certain stages, but rather contribute to the whole process, or at least are aware of the whole process and participate in multiple stages that are not only design oriented — in our case, as we will see in Chapter 6 the stakeholders were involved in all the modelling stages, from providing relevant information about the domain, to contributing to choosing the model structure, interpreting model results and even participating in code debugging sessions. The insights provided by stakeholders and other participants directly contributed to the models, making this method, in a sense, closer to agile software development [1] than to participatory modelling.

In the context of this thesis, given our deflationist views, we argue that the representations constructed through our modelling process need not necessarily be fully formalised, but rather discussed, justified, and clearly agreed upon during the modelling process. Still, in Section 4.1, we have described some of the requirements and benefits of employing a formalised distributed systems metaphor for structuring the co-design aspects of model design. We consider this step as process optimisation rather than baseline necessity, particularly because methods such as SSM(Soft Systems Methodology) [68] have been successfully used in the collaborative creation of conceptual models without formalisation. Even though a formal description of SSM has been achieved in [261] via Petri Nets, this has not led to a significant transformation of the process of constructing conceptual models of systems using SSM.

5.2 Classical cycle

Having previously focused on co-design and its advantages, we now turn our attention to the actual process of designing and constructing models in the context of heterogeneous systems.

Although from a historical perspective, conceptual models have existed since at least the early days of humanity, a generally agreed upon process for their design and construction still remains an active topic of research today. However, this is not the case for formal or mathematical models, due to their usually positivist epistemic tendencies. Thus, the process of model design and construction in the case of formal models can be considered in a more mature state than its conceptual counterpart, and therefore, more properties of models constructed in such a way are known. Because of this, we have chosen to construct our integrative methodology starting from this classical, mathematical modelling cycle and we are attempting to expand it for the more complex case of heterogeneous systems. We acknowledge the fact that the process could be reversed, perhaps by starting from a methodology less focused on formality, but we argue that the set of properties provided by the classical mathematical modelling cycle, if underpinned by a ‘good’ metaphor — in our case, the distributed systems metaphor —, reasoning framework and tools for practical implementation are significantly harder to produce if starting for example from SSM.

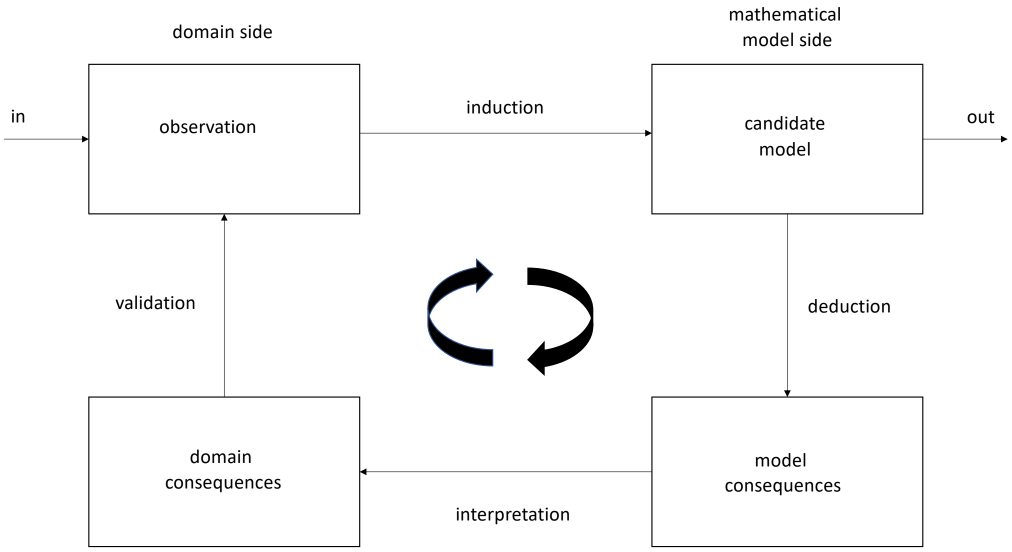


Figure 5.1: The classical mathematical modelling cycle [63]

Traditionally, mathematical models have been constructed using a multi-stage iterative process named ‘the classical mathematical modelling cycle’ [196, 63, 94]. As it can be observed in Figure 5.1, this is comprised of six different stages:

- Constructing a conceptual representation of the phenomenon under study based on observing its domain. In the case of the organisational ransomware recovery model from Chapter 6, this would mean determining the relevant aspects of ransomware, organisations and recovery mechanisms that provide us with enough information about the recovery problem in this setting as to construct our model.

- Abstracting a candidate model based on the observations, using induction. The candidate model is based on the distributed systems metaphor to provide modularity, customizability, the ability to reason locally about its components, and all the other properties described in Section 4.1.

- Deducing the mathematical consequences of the model. Succinctly, this means interacting with the model to produce a list of abstract properties or consequences relevant to understanding the system.

- Translating the mathematical consequences of the model by interpretation in the domain side. In our case, it follows that abstract consequences of the candidate model should be reflected back to the domain side in an explicit form. For example, a model consequence could describe the input parameter admin_nr as being an influential component of the output variable recovery_time. By interpretation, this would be translated in the domain as the number of admins corresponding to a help-desk in an office being an important factor in determining the recovery time of devices in that office.

- Validate the correspondence between the interpreted consequences in the domain side and the actual observed reality of the domain. Continuing with our example, this would mean confronting the relationship between the number of admins and the recovery time of devices with the observations about the domain in an attempt to either confirm or deny it.

- Update the conceptual representation of the phenomenon, based on the previous step.

These stages are repeated until a criterion of adequacy for the intended purpose of the model, often determined by the judgement of the modelling participants, is passed and the model is considered to be a good enough representation of the system under study. For additional information about this criterion in the system dynamics simulation literature, see [113]. Nonetheless, the efficacy of this modelling process depends on certain key, usually unstated, assumptions about the modelling task:

- The structure and behaviour of the domain is clearly understood in conceptual or engineering terms. For example, the increase of device recovery time given an increase in device recovery requests is a well known phenomenon attributed to the inability of a network to transmit requests with the same speed after a certain congestion level has been reached. Such a phenomenon is a good candidate for mathematical modelling, perhaps in the context of testing network upgrades or extensions.

- The data that can be collected about the domain is essentially unambiguously identified. For example, modelling an organisation's employees productivity levels for the sake of improving them might prove an extremely complicated task for mathematical modelling since, essentially, employees might be motivated by extremely subjective concerns and the interpretation of those concerns might differ from person to person.

- The questions that the model is intended to address are identified independently of the detailed design choices required for the construction of a model. For example, building a model for the purpose of optimizing the production time of hardware components in a fully automated manufacturing environment is well suited for the traditional modelling methodology described above. Contrarily, simulating the same system for the purpose of understanding its behaviour and only then deciding what can be optimized would be more suited to a different approach.

Such assumptions are usually not compatible with heterogeneous systems simply because the underlying components of those systems differ in their nature, providing the modeller with much richer challenges in terms of the design, construction and interpretation. For example, the structure and behaviour of the domain are harder to grasp because the domain is composed of multiple sub-domains with different structures, behaviours, incentives and so on. Naturally, a clear understanding of such domains requires access to knowledge from different scientific disciplines, if such knowledge already exists. Furthermore, this issue extends to the domain data collection: multiple data types might be available, and the process of unambiguously identifying the relevant aspects from it can be more complex, especially when cross-checking between different types. This can be particularly difficult if the data has already been recorded before that start of the modelling process, since relevant features might be missing from quantitative data and additional clarifications might be required for qualitative data. Last but not least, independently identifying and fixating the questions the model should address before knowing the available resources — data, expertise, even deployment environment — might not always be the best idea. We do not attempt to diminish the importance of setting a clearly defined goal from the very beginning here.

However, goals can be attained in different ways: for example, if the goal of a model is to automatically stop ransomware from infecting a computer in a specific organisation, one could ask how does ransomware behave in general, or what are the top three indicators that an email attachment contains ransomware. The distinction here is clear: although the first question is not incorrect, it shows a lack of information regarding the environment in which the model will be deployed. If the only available entry point is via email, there is no need to represent possible infection via USB ports. In other words, even if the overall model goal is pre-stated, the model scope and, inevitably the exact questions to be answered should co-evolve with the model. As described by [279], 'in design thinking, problems and possible solutions are explored and developed and evaluated simultaneously in an iterative process: A "design process involves finding as well as solving problems" so that "problem and solution co-evolve." Design thinking is needed to cope with "wicked problems" — problems that cannot be clearly defined using "facts" at the start of a project and that cannot be solved by selecting a "best" solution'.

For example, given a generic organisational phenomenon, we might have available the following data assets: a scientifically agreed upon organisational theory about it, expert knowledge from practitioners in plain text, a set of CCTV recordings where the phenomenon is empirically observable, a numerical data set from previous attempts at quantifying such a phenomenon and user feedback which was collected via a survey with both qualitative and quantitative questions. Without an in depth analysis, two questions could be asked: what happens if the quantitative data does not include all the relevant variables for the analysis and, in the case of qualitative data, is it even possible to fully and unambiguously identify the entities and relationships relevant to the modelling task?

As we can easily observe with only the above example, the traditional mathematical modelling approach is suited for phenomena that already have a significant theoretical understanding behind them, and that is particularly why the third assumption can be considered as legitimate in such cases. However, if the domain is still an active area of research, we argue that the knowledge about the domain and the questions the model can provide answers for are strongly related, and should be updated while iterating through the modelling cycle. In a sense, the knowledge about the domain provides an array of possible questions that the model could answer, but choosing a specific set of questions limits then the necessary information from the domain required for answering. This observation has led us to a methodological development called the 'Co-design modelling cycle' [63, 94], which will be explained in detail in the next section, and is based on the explicit concept of a 'translation zone' — explicitly described in Section 5.4 — that mediates the dual evolution of domain knowledge and model focus.

5.3 Co-design cycle

Building on the previous subsection, we can clearly state our starting point — the classical mathematical modelling cycle — and our desired outcome: a modelling cycle that would explicitly acknowledge and facilitate the understanding of both the structure and behaviour of the domain and phenomena under study, the unambiguous identification of relevant entities and relationships from the available data and the co-evolution of model and scope while conserving the pre-stated goals.

The author's previous work in [63, 152] represents preliminary attempts at constructing a modelling cycle based on the principles of co-design, which can be observed in Figure 5.2. In the following paragraphs, we further develop this cycle by explicitly accounting for the definitions in Sections 3.3.1 and 3.3.2 and integrating it with the qualities described in Section 3.3.3. The resulting co-design cycle can be observed in Figure 5.3. We further describe the components:

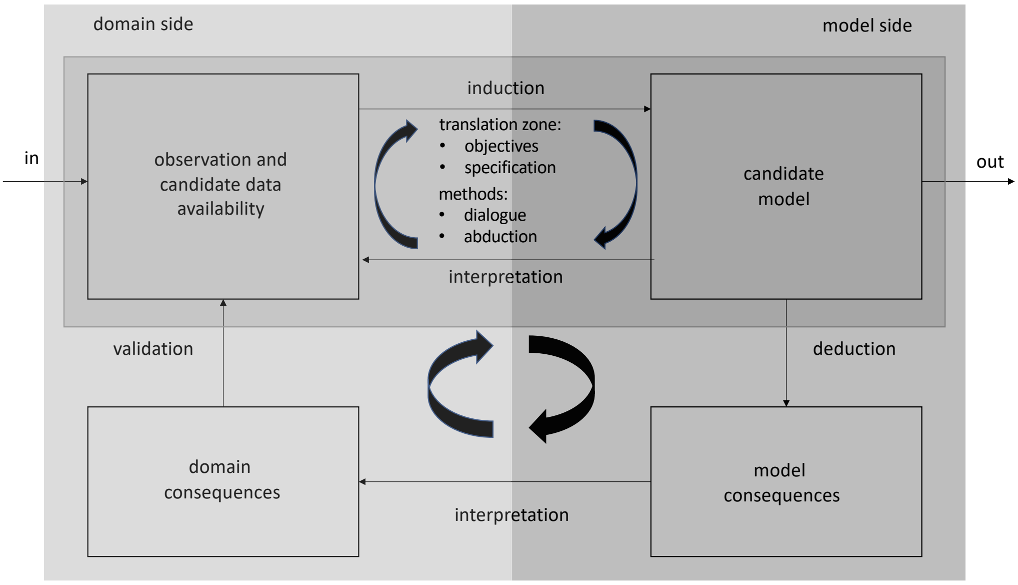


Figure 5.2: The simplified co-design cycle [63, 152]

5.3.1 Domain exploration

The primary objective of the domain exploration phase is represented by the construction of an initial model scope. Since the model scope is constituted by the list of target systems, their relevance and representation quality criteria, this implies the participants posses or have acquired a degree of understanding of the phenomena under study. Each participant constructs an initial conceptualisation of the phenomena — the phenomena presentation — based on direct observation and inference, prior knowledge, beliefs & interpretation, and analysis and interpretation of available data. By interpreting each presentation, the participants propose different target systems, quality criteria and justifications for relevance. Following a joint debate, an initial model scope is agreed upon. It is important to note here that during multiple cycle iterations, alterations of the model scope can lead to modifications in the presentation for each participant. This can further propagate into changes at the level of personal knowledge and belief, changes in how direct observation is interpreted or changes in data requirements and analysis — which can require a new data collection process.

Once the model scope has been constructed, the model will transition to the next phase of the cycle, namely the candidate model construction phase. This transition happens via the processes of the translation zone, which are explained in greater detail in Section 5.4. Briefly, this ensures that a consistent conceptual representation of the heterogeneous system under study is being constructed, based on the distributed systems metaphor and taking into account all the elements of the model scope.

> **Phenomena**: the phenomena under study. For example, in the case of the ransomware recovery model, the infection, movement, and recovery of devices in different work environments.
>
> **External knowledge & beliefs**: the knowledge and beliefs of the participants in the modelling task about or related to the phenomena under study. These can be based on personal experience, cultural and societal norms or different forms of education and are different from one person to another. For example, the security research manager identified network congestion as a relevant factor for the decrease in recovery time early in the recovery model conceptualisation.
>
> **Phenomena presentation**: initial conceptualisation of the phenomena constructed after the start of the modelling task and based on direct observation and inference, prior knowledge, beliefs & available data. Given the direct influence of the beliefs, this is again different for each participant, and only available to them.
>
> **Data**: the available data assets regarding the phenomena. Can be both qualitative and quantitative and recorded before or during the modelling task. Although data assets are static — they do not change after being recorded — they are still influenced by the external knowledge and beliefs of the recorders or by the recording mechanism itself. In the ransomware recovery model, the

data was comprised of: quantitative data about recovery timings and qualitative data from the participants in the co-design process.

Target system: conceptual representation of an area of study in which parts of the phenomena are manifested. Multiple ones can exist, but they must be clearly stated. For example, a model of organisational recovery under ransomware could have the following target systems: fleet of devices, network, external storage, ransomware behaviour, user movement.

Model scope: the list of target systems, their relevance to the studied phenomenon and the set of criteria that deem a representation of a target system as 'good enough'. These can be related to both structure and behaviour: for example, how close should be the implemented representation of a network to a real network — would routing and DNS resolution be implemented explicitly? — or, how close should be a pattern of network congestion obtained from the model execution to one recorded in real world data. In the spirit of co-design, all three are dynamic but should only be updated after an iteration through the translation zone or the complete cycle.

5.3.2 Candidate model construction

The main goal of the Model Construction phase is to produce a functional heterogeneous model and to use it to obtain results via a form of execution. The forms of execution are different based on the nature of the component: conceptual and formal components can produce consequences directly by deduction or results via exemplification; executable components can only produce results, computationally by code execution or physically by performing operations on the physical model. It is worth emphasising here that the configuration of the model regarding the metric detailed in Section 3.3.3 can change during cycle iterations. For example, conceptual model components can be implemented or formalised. Formal model components can be implemented or conceptualised. Executable model components can be formalised or conceptualised. The decision-making process used for changing the nature of the component is usually influenced by the modelling goals and the environment in which the model will be further deployed and used.

After the construction of the model, three phase transitions are possible: towards domain exploration via the second area of the translation zone, towards model use via deployment or towards model consequences derivation. Moving back towards domain exploration indicates that during the construction of the model, a need for an update in the model's scope has been identified. Moving towards model use carries the implication that all the representational criteria have been met and that the model is considered ready for use. Lastly, a move towards model consequences derivation requires different operations based on the nature of the model component: consequences can be drawn directly from formal or conceptual components via deduction, or indirectly from any kind of component that has produced results via analysis, interpretation and then reasoning about such results.

Conceptual model representation: conceptual model including all target systems, structured by the distributed systems metaphor, agreed upon by the participants and clearly expressed. A simplified conceptual representation of the ransomware recovery model can be seen in Figure 6.8. A more detailed one is available in Figure 8 of [151].

Formal model components: the model components expressed using formal constructs such as systems of equations or logical formulae. Neither of the 3 presented models include explicit formal components except for the probability distributions used in the sampling. However, a formal representation of the network sub-model is available in 6.3.4, when discussing the composition of the device and network models.

Executable model components: the implemented model components which manifest themselves in either a physical or computational environment. For all of the presented models, this is the Julia code used for execution.

Results: the outcomes produced by a form of execution of the model. For executable components, the results of code execution or performing actions on the physical model. For formal or conceptual components, the traditional meaning of execution does not have a direct correspondence. However, formal and conceptual model components can be used to produce results directly — not consequences in this case — via a process of exemplification. For all of the presented models, these are the numerical outputs produced by executing the model. For graphical representations of the results in the case of the ransomware recovery model see Appendix C.

5.3.3 Model consequences derivation

As the name suggests it, the purpose of this phase is to obtain a set of consequences about the model in a consistent format. For ease of understandability, this should be natural language, structured by the distributed systems metaphor. The consequences can be obtained via deduction from formal or conceptual components, or the analysis, interpretation and reasoning about the results or of the process of execution itself. To obtain the consistent format, the executable or formal consequences must be translated into natural language. Afterwards, a verification process is used to ensure no contradictions or incompatibilities can be found between the consequences. We note here that since up to this point, the consequences have not been reflected back into the domain, the verification process cannot ensure that all the issues have been determined. Only that there is no mismatch between the model components. Therefore, after multiple iterations through the verification process, the model might contain no mismatch between components but still not behave as expected.

Two cycle transitions are possible in this state: backwards toward candidate model construction or forward towards domain consequences translation. For the first one, if contradictions or incompatibilities are found during the verification process, the cycle backtracks to the previous phase and the model components are updated accordingly. If the verification process terminates successfully the model can transition towards the domain consequences translation phase by interpreting the consequences in the context provided by the domain.

Formal model consequences: the set of consequences obtained from the formal model components by using a form of deductive reasoning. These are usually expressed using a formal or semi-formal language. An example here, although in natural language would be that resources such as USB drives should not be found outside locations.

Executable model consequences: the set of consequences obtained from the analysis and interpretation of results or of the process of execution itself. For example, during an initial execution of the ransomware model, allocating two admin resources to help-desks was producing better results than allocating three admin resources.

Conceptual consequences representation: the set of all model consequences, in a clear natural language form. These can be obtained either by direct interpretation of the conceptual model components, or via translation from executable or formal consequences. It is of relevance here that the conceptual model consequences are not yet reflected back to the domain, but maintained at at the level of the model. For example, the idea that allocating more admins resources should reduce recovery timings.

Verification: the process of determining whether or not contradictions or incompatibilities are present between the consequences. To ensure consistency, all the consequences must be expressed in the same language. For the procedure to be understandable to all the modelling participants without additional explanation, the consequences should be expressed using natural language. However, in the case of extremely large models, additional formalisation, implementation and execution procedures might be used for automatic formal verification. Needles to say, in such cases, the identified contradictions or incompatibilities should be translated back to natural language and further analysed. As seen above, the verification process in the case of the ransomware model identified a contradiction between an executable and a conceptual consequence regarding admin allocation. More about this in Section 6.3.4.1.

5.3.4 Domain consequences translation

The main aims of this phase are the interpretation of model consequences — in our case the conceptual consequences representation in natural language — to domain consequences and the validation of such consequences with respect to the domain, via the model scope. Model consequences are interpreted in the domain usually by contextualisation with the environment: for example, an executable consequence such as the model cars moving at an average speed of 30 mph is now interpreted as real cars moving on real roads at 30mph. A formal consequence such as elements of type A not being included in set T of regular traffic participants can now be understood as ambulances not having to stop at traffic lights — this example is trivialised, since we do not fully explore all the implications of being part of set T. After this contextualisation step, validation procedures are used to determine whether or not the quality criteria from the model scope have been achieved.

A single cycle transition is possible in this state: towards the domain exploration phase. The result of the validation process is important in determining if subsequent cycle iterations are required: if mismatches between the translated consequences and the model scope are identified, updates will be required in the next iteration through the translation zone, which will then be further propagated at the level of the constructed model components; if not, the next iteration through the translation zone can be used for optimisation purposes or simply for the participants to decide if the model is considered ready to be used. However, we must also note that the modelling participants can decide that the model has reached a ‘good enough’ state and is ready for use even if the validation process is not completely successful. Similarly to the case of the classical cycle, the termination criteria are determined on a case-by-case basis, but should respect a few key general considerations relative to which the notion of accuracy with respect to the scope must be calibrated: remembering that ‘the map is not the territory’ [174]; appropriate level of detail; timeliness; and cost-effectiveness.

Domain consequences: formal model consequences, executable model consequences and conceptual consequences derived directly from the conceptual model representation translated at the level of the domain. Continuing the admin example from above, we exemplify two contradictory consequences. Firstly, that two admins were producing better results than three admins. Secondly, that more admins should produce better results than fewer admins.

Validation: the process of comparing the consequences of the translated domain with the scope of the model, and ensuring that the representation fidelity criteria have been met. In the above case, the modelling participants raised the concern that the model scope was underdeveloped. In the following cycle iteration, they suggested extensions for the admin deployment policies that were further introduced in the scope and then implemented.

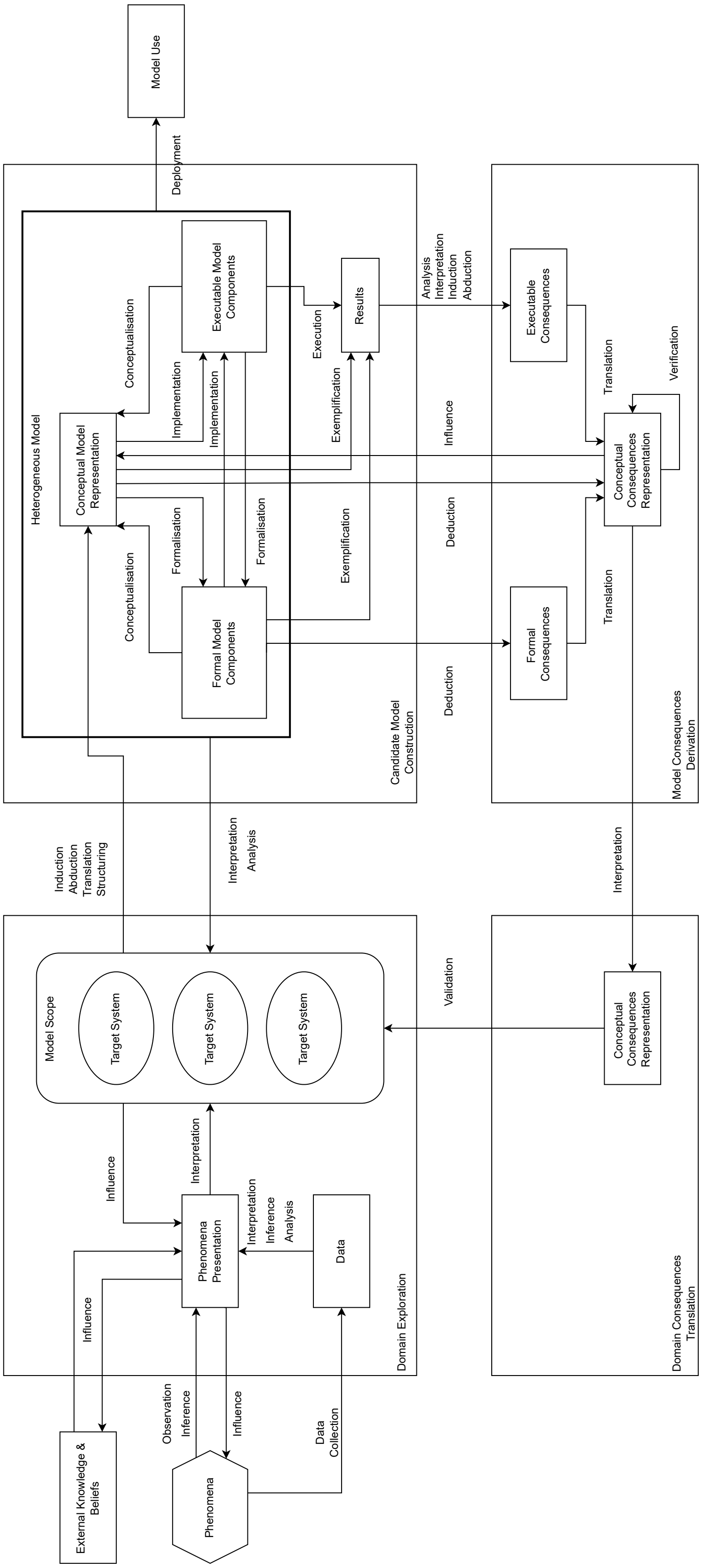


Figure 5.3: The extended co-design cycle [152]

5.4 Translation Zone

In the previous chapter, we have illustrated the compatibility between the distributed systems and trading zone metaphors, and some of the benefits of employing the distributed systems conceptualisation in a modelling context. Further, we shall explicitly describe how these ideas can be integrated with the extended co-design cycle from above, at the level of a translation zone.

First of all, the translation zone represents the area of the co-design cycle that links the domain exploration and model construction phases. More precisely, it can be viewed as a bi-directional set of processes, whose purpose is to align the different goals, epistemic beliefs, practical interests and expertise of the modelling participants, with the help of the distributed systems metaphor, in an attempt to produce a co-evolution of model scope — target systems, representation criteria, relevance — and constructed model and, facilitate knowledge sharing and the development of interactional expertise, by design.

As show in Figure 5.4, the processes of the Translation Zone can be separated into three different areas:

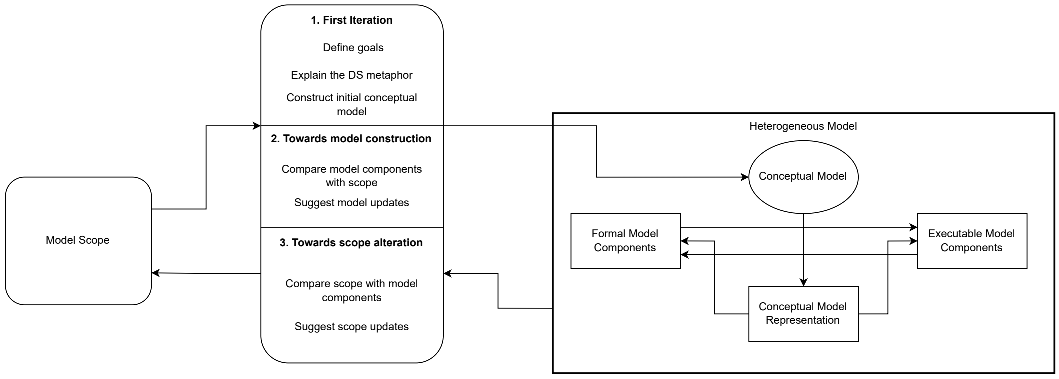

1. First Iteration
Define goals
Explain the DS metaphor
Construct initial conceptual model
2. Towards model construction
Compare model components with scope
Suggest model updates
3. Towards scope alteration
Compare scope with model components
Suggest scope updates
Model Scope
Heterogeneous Model
Conceptual Model
Formal Model Components
Executable Model Components
Conceptual Model Representation

Figure 5.4: The translation zone [152]

5.4.1 First iteration

The first iteration through the Translation Zone, from the Domain Exploration towards the Model Construction phase can be seen as an initialisation procedure. Starting from a relatively generic initial model scope, the participants aim to construct an initial conceptual representation of the heterogeneous model that includes all the relevant target systems. Abstractly, this can be seen as a dual process of reasoning and translation: given the nature of the available information about the systems, the reasoning process can be deductive, inductive or abductive followed by a translation towards the distributed systems conceptualisation. Furthermore, we detail the specific steps to be undertaken:

> **Define goals**: The overall model goal and the goals of each participant should be clearly defined, noted and agreed upon.
>
> **Explain the metaphor**: The responsibility of ensuring that the participants understand and are familiarised with the concepts employed in the distributed systems metaphor lies with the modeller. To facilitate this, the modeller should consider the background, knowledge, and expertise of each participant and tailor the language used and level of detail accordingly. Furthermore, the use of analogies, real-world examples and a positive, open atmosphere where questions and feedback are encouraged can greatly improve the process.
>
> **Construct initial conceptual model**: For each target system included in the model scope, a conceptual representation must be constructed. For that to be possible, the participants must first identify what elements will be directly expressed and at what level of detail and note down justifications for the omitted ones and for any underlying assumptions. Then, they must decide how to map each element to the distributed system concepts — what is a resource, what is a process, what is a part of environment, etc. We note here that multiple possible mappings can exist: for example, a building could be considered both a resource and a location, or an employee could be seen as a resource or an implied entity that starts a series of different processes — nevertheless, the decision is highly situational. Lastly, the participants suggest a direction for the construction of each model component — conceptual, formal or executable — by explicitly taking into account the model goals and the future deployment environment.

5.4.2 Towards model construction

Similarly to the initial iteration, the goal of this phase is to produce a conceptual model representation according to the criteria in the model scope. However, the main difference is that model components have already been constructed in some form, so the stage can be understood overall as aligning the components with the scope. In practice, the following steps should be considered:

> **Compare model components with scope**: To ensure that the model is being developed in the collectively agreed upon direction, each model component must be compared with its associated scope element. That means checking that both the conceptual representation of the component and its practical implementation — if it exists — are adhering to the representation quality criteria. For example, if the target system is a network, the quality criteria is based on recorded real-world data about the target and an architectural diagram and, a previous decision has been made to construct this component as an executable simulation, then the participants must ensure that the results of the simulation illustrate a network behaviour closely resembling the real-world target — the degree of closeness should be specified — and that the distributed systems conceptualisation of the network matches the architecture of the real network.
>
> **Suggest model updates**: Based on the above comparisons, areas of improvement can be identified and corrective actions are proposed, in line with the scope. For example, the simulated network's behaviour might not closely resemble the real network, but the verification processes might not identify any issues. In such a case, additional domain understanding would be required: interaction with engineers and network administrators could perhaps reveal that the network performance is influenced by additional constraints — perhaps backups and maintenance are conducted during certain hours, therefore reducing the available bandwidth, but this was not detailed in the original recorded dataset. In light of the discovery, the participants must now decide whether to explicitly structure the backup and maintenance processes

and then implement them, or perhaps implicitly reproduce the behaviour by altering network control parameters.

5.4.3 Towards scope alteration

As the name suggests, processes in this phase deal with the other aspect of the co-evolution, namely ensuring that the quality criteria and model scope are still relevant to the model.

> **Compare scope with model components**: Similarly to the first step above, participants must first analyse and compare the model components and scope. However, this time, their purpose is focused on identifying possible areas of improvement in the scope. For example, in the above case, the network target system can be seen as an area of improvement at the scope level.

> **Suggest scope updates**: As expected, the identification of areas of improvement is followed by actions that alter the scope in that direction. For example, if the participants decide to explicitly implement the additional network processes, additional structural quality criteria must be added to the model scope. However, this is not the only case when the scope could require updates: perhaps due to performance constraints, a model component shall be represented at a higher level of detail, to reduce the number of instantiated entities; furthermore, another component might require a translation from formal to explicit implementation, in an attempt to increase the model understandability. Needless to say, any such changes should be collectively agreed upon, documented and reflected in the updated scope.

5.5 Conclusion

In this chapter, we have attempted to present our modelling process explicitly, by following a similar type of compositional reasoning to the one employed in our modelling approach. This should not strike anyone as peculiar, since we are constructing, in a sense, a conceptual model of modelling.

To prioritise model integration, and inherently understanding, we have integrated the following components: the classical mathematical modelling cycle, co-design principles, the model conceptualisation provided by the triangle framework, the distributed systems metaphor — supported by the similarities with the trading zone metaphor –, and the philosophical commitments made in Chapter 2 — the acceptance of systems' heterogeneity and model multi-methodology, conceptualisation of models as collections of sub-models, and the reduction of scientific realist commitments.

At the level of the extended co-design cycle, including the translation zone, the above described components can be identified in the following areas:

> **Co-design principles**: The influence of co-design ideas is most visible in areas of the cycle where decision making processes take place, usually in the form of natural language debates such as: debating and updating the model scope in the domain exploration phase and after the validation process, the bi-directional processes of the translation zone, including deciding on the model configuration before the candidate model construction phase, interpreting and verifying the occurrence of contradictions at the level of model consequences, or interpreting domain consequences.
>
> **Classical mathematical modelling cycle**: This was the structural inspiration for the methodology. The four primary stages, and the nature of the transition processes between them represent extensions to the original versions present in the classical cycle.
>
> **Triangle framework**: The constitutive qualities of the triangle framework — conceptuality, executability, and formality — are directly considered in the translation zone, candidate model construction, and model consequences derivation phases, but may influence the model scope as well, particularly in relationship with model deployment — because not every construction technique is suitable for all deployment environments.
>
> **Distributed systems metaphor**: The conceptualisation provided by the dis-

tributed systems metaphor is employed during all the stages of the co-design cycle, except the first iteration through the domain exploration phase, which ends with the establishment of the first model scope.

Philosophical commitments: The acceptance of systems heterogeneity and model multi-methodology are explicitly accounted for, with the distributed systems metaphor and triangle framework serving as means of representing the diversity of model targets. Furthermore, the the ability to conceptualise and construct representations of models as collections of sub-models is supported for each type of model component as given by the triangle framework — composition can occur and is well specified for conceptual, formal, or executable model components. The scientific realist commitments have been relaxed, and the deflationary nature of the modelling account is observable especially at the level of the debates occurring between modelling participants, where a precise criterion for what is considered acceptable knowledge is not provided beforehand, but must be agreed upon during the modelling process as a result of structured debates.

Finally, we emphasise the practical value of this chapter for modellers, as it translates abstract principles into a structured, actionable methodology for building integrated models of heterogeneous systems. By extending the classical mathematical modelling cycle with co-design principles, the distributed systems metaphor, and the Triangle Framework, this chapter provides a systematic process for managing complexity in real-world modelling projects. The proposed co-design cycle, supported by the translation zone, enables iterative refinement of model scope and structure while fostering collaboration among diverse stakeholders. This approach not only improves model interpretability and alignment with stakeholder goals but also supports modularity, scalability, and adaptability across conceptual, formal, and executable dimensions. In practice, these features reduce development risks, enhance validation strategies, and ensure that models remain both analytically rigorous and operationally relevant. As such, the methodology presented here serves as a practical blueprint for constructing models that can evolve alongside dynamic systems and decision-making needs.

Chapter 6

Model Case Studies

The measure of the value of an experience lies in the perception of relationships or continuities to which it leads up.

John Dewey

In the previous chapter, we have explicitly described our modelling methodology and explained how the philosophical commitments from Chapter 2, the model conceptualisation based on means of construction from Chapter 3, the distributed systems metaphor and its correspondence with the trading zone metaphor from Chapter 4, and the classical modelling cycle and co-design considerations are all integrated at the level of the extended co-design cycle.

In this context, the goal of this chapter is to present the application of our method at the level of three security related models: a physical data-loss model, a trauma unit surge capacity model, and an organisational recovery under ransomware model. We describe their goals, internal structure and representation choices and illustrate how each of the models focuses on different aspects of the metaphor: the data loss model heavily focuses on physical locations, the ransomware recovery one on processes and the trauma unit one on resources. Subsequently, we explain the iteration of the co-design cycle steps within these three specific contexts and discuss the challenges encountered. We note here that the organisational recovery model is in a significantly more mature development state, being the only one including an explicit analysis of results, parameters or sensitivity analysis.

Sections 6.1 and. 6.2 are focused on the physical data-loss, and trauma unit surge capacity models respectively, and explore the main representation choices and co-design aspects involved.

In Section 6.3, we introduce the organisational ransomware recovery model developed in partnership with HP Security Lab Bristol. Precisely, this includes a description of the organisational ransomware recovery problem, two literature surveys on the nature of ransomware and recovery techniques, the model architecture, representation choices, co-design aspects employed, the experimental space — parameters, explicit simulation scenarios, overall behaviour, sensitivity analysis, verification, validation —, and implications for the use of the model as management tool.

As noted in the research paper declaration form, this chapter contains elements of the author's previously published work. More specifically, sections 6.1 and 6.2 are edited versions of sections 6.1 and 6.3 from [152]. Section 6.3 includes elements from section 6.2 from [152] and [151].

6.1 Physical data-loss model:

The first model we introduce here was developed in [64] and aims at assessing the impact of different physical security policies with respect to data loss at the level of a small-sized organisation. This is a simple model that illustrates the use of the distributed system metaphor approach and the role of the co-design cycle in a relatively uncomplicated setting.

We must note that this is not the first model ever constructed using a version of the distributed systems metaphor. For example, the executable models of [37, 38] are based on a system representation including processes and resources, but without an explicit characterisation of the nature of systems such as that provided by the distributed systems metaphor, and without explicit conceptualisations of locations, environments, or interfaces. Furthermore, the authors have continued to develop the conceptualisation in a series of works such as [75, 76, 74, 10], and consider applications in, for example [27, 60, 249, 20]. A previous implementation of these ideas, Gnosis [74], has been used in significant commercial applications [19, 34, 35] derived from an industry-based research project [140].

6.1.1 Structure & representation choices

As previously stated, this model attempts to provide a better understanding of the possible impact of physical security policies in relation with data loss. In order to capture the main elements related to these phenomena, we focus on representing the locations where physical data loss can occur: in the office, if external access is possible, or in transit. Structurally, this leads to three different sub-models, depicted in Figure 6.1.

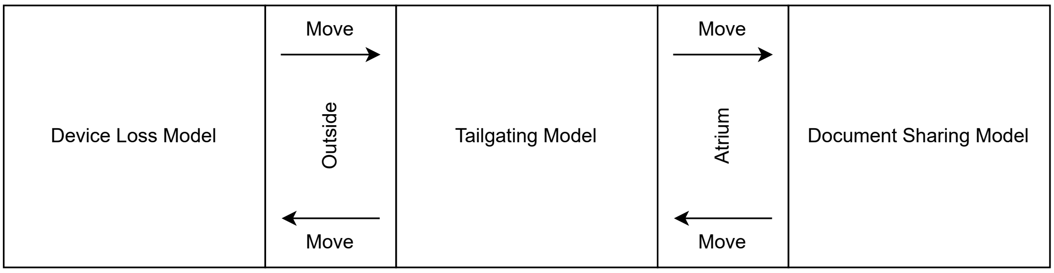


Figure 6.1: The simplified physical data loss model [152]

The device loss model, situated on the left side of the diagram, acts as an abstract representation of the areas outside an office building where data assets can be lost by employees in transit. Its operation can be described succinctly: while commuting to or from work, employees face the risk of losing the devices they carry, potentially containing confidential data. The quantity of confidential documents stored on these devices is influenced by the behaviour of employees in the document-sharing model.

The employees are being represented using a bundle of a resource, a process and multiple locations: the resource signifies the physical position of the agent within the model. The process is used for relocating this resource within various model locations, while also engaging with other resources as required. The positions associated with the agent are utilised to represent concepts such as possession or memory. For instance, to simulate an agent acquiring another resource, the agent's operation would relocate that resource to the position representing items being carried by the agent; conversely, releasing the resource would return it to a physical position within the model.

The tailgating model is used to illustrate the physical boundary of the organisation and assess the ability of an external attacker to traverse it to gain access to internal data assets. In practice, employees are required to present an ID card resource to gain access to the office via a security door location where the ID verification process takes place. When employees forget their cards, they are faced with a decision: either wait in line at the reception desk to obtain a temporary ID card for the day, or attempt to tailgate through the door. Employees who have successfully passed through the security door then observe if others attempt to tailgate behind them. In this scenario, the employee has two options: either ignore the tailgater and proceed directly to the office, or confront the tailgater and redirect them back to the reception area.

These decisions are represented as stochastic processes with varying associated probabilities controlled via external parameters. Additionally, there are attackers who consistently attempt to tailgate without making any decisions. These attackers can be intercepted by security guards or challenged by employees. The security guards and attackers are being represented in a similar fashion to the regular employees.

The document-sharing model examines how employees behave within the office environment when faced with decisions about sharing confidential documents among themselves. Typically, employees share documents through a shared drive that limits access to authorised personnel. However, this system often experiences downtime, requiring employees to resort to alternative methods for document sharing. Within the model, three options are available. Firstly, employees can utilise a global share accessible to all personnel. Secondly, they can opt to email the documents directly to recipients. Lastly, employees may choose to use portable media like CDs or USB sticks to share data. Each option presents its own drawback: documents on the global share are accessible to all employees, emailed documents end up on devices carried to and from work, and portable media left lying around the office poses a risk. Furthermore, attackers roam the office, collecting any abandoned portable media resource they encounter.

Given the above setting, we can now present the explicit mapping between the desired entities and phenomena to be modelled and the concepts of the distributed systems metaphor from Chapter 4, at the level of the composed model.

In terms of *locations*, the model is relatively simple: most locations are physical and are used to represent areas such as the employees' home, public transport, private transport such as cars, an area outside the main building's lobby, and internal areas such as the lobby, entryway, atrium, or actual office.

At the level of *resources*, the situation is similar: the concept of resource is being used to represent physical items where confidential information might be stored, including devices such as mobile phones or laptops, cds, USBs, or paper documents. Furthermore, the ID cards required by employees to enter the building are also resources.

With respect to *processes* the model can be viewed as slightly more complex. Processes are being used to represent a wide array of activities including travelling, queuing for a temporary ID badge, working, observing, challenging, ignoring, or performing tailgating, deciding how to send confidential documents, and then sending them, or searching for, or loosing data assets. Additionally, each process is associated with an agent — represented using a bundle of a resource, a process and multiple locations — symbolising a regular employee, a security guard, or an attacker.

Last but not least, *environments* are used to represent areas of the model that are not conceptualised or constructed in detail. For instance, given the model goal, the explicit activities performed by an employee at home outside office hours or by an attacker outside an attack timeline are not relevant. Because of that, environments initialised with probability distributions are being used to start the processes of employees travelling to work or for attackers arriving in the main building.

6.1.2 Co-design

Having briefly explored the structure of each underlying sub-model and the associated representation choices, we now turn to illustrating the co-design steps involved. However, we first again emphasise that this initial example is quite simple and should be viewed as an educational illustration for employing the distributed systems metaphor, rather than a detailed explanation of the use of explicit co-design. It is noteworthy to mention that the development of a co-design theory for heterogeneous modelling, at least in the sense shown in Chapters 4 and 5, was preceded by the construction of the physical data-loss model.

Nevertheless, the setting of the model still provides us with the opportunity to illustrate how the co-design cycle might have been used in this case, as a thought experiment. We limit the description to the domain exploration phase of the co-design cycle due to the reduced scope. The other two models to be presented in the following subsections will be used to detail aspects of the co-design process not present in the physical data-loss model.

In this context, the domain exploration phase can be seen as heavily influenced by the composition of the modelling team which only included two modellers with a great amount of expertise in information security, but no other stakeholder or user.

The phenomena to be studied are all relevant to the concept of physical data loss: physical data loss can only occur if an attacker can physically obtain access to a data asset. Assets can only be found in the office or in transit if lost — we do not consider scenarios involving theft from residential areas — and if they are in the office, the attacker must also be there. Therefore, it is reasonable to assume that the phenomena to be studied include transiting from home to the office, possibly losing data assets on the way, gaining access to the building, performing work, deciding how to share assets, deciding whether or not to tailgate into the building or to confront an attacker and so on.

Similarly, the entities involved in these phenomena are the agents — regular employees, security guards, attackers — the resources themselves — data assets, ID badges — and the locations where may be found — home, transit, office, lobby, etc.

With respect to data, the only source used was probabilistic: a negative exponential distribution was used to model the arrival rate of employees at the entrance to the office building. Other aspects related to the organisational security posture or attacker behaviour such as the probability of an employee or security guard to challenge an attacker, the probability of losing a device in transit, the attackers' arrival rate, or probability to discover a data asset were controlled through model parameters and used to generate different scenarios.

In light of this, we can view the model scope as containing 3 main target systems, closely matching the sub-models in Figure 6.1: a transit system, an office entrance system and an office internal space system. Their relevance is clear: they represent the areas in which physical data loss can occur. Given the lack of physical deployment and the exploratory rather than practical nature of the model, the representation criteria for the target systems are very simple: the employee arrival rates, device loss rates, successful tailgating, successfully discovery of assets, challenging an attacker and so on must resemble possible real-world occurrences reasonably close. In a practical, organisational case, this would have been very different: quantitative historical data about arrival rates, previous incurred attacks, office document disposal and storage policies, and past stakeholder experience can be used for a more detailed representation involving stricter quality criteria.

The establishment of the model scope marks the completion of the domain exploration phase, at least for the current iteration of the cycle. Given specific goal of experimentation with a simulation model, there is no surprise that the translation zone is minimal — conceptualising the target systems using the distributed system metaphor did not require extended multi-disciplinary debates — and the resulting model is heavily simulational.

6.2 Trauma unit surge capacity model:

The second model we introduce was used to explore the surge capacity of a trauma unit within a generic hospital emergency department. In comparison to the model described above, the trauma unit surge capacity model can be seen as simpler at the level of structure, but relevant for illustrating how the domain exploration and translation towards a candidate model can be performed in a context in which the modellers do not possess a high amount of experience or expertise. The relevance of this model for cybersecurity is explored below:

Firstly, surge capacity is related to organisational resilience, even though traditionally seen as outside cybersecurity. Combined with resource mismanagement, it can lead to situations in which patients no longer have access to medical care or in which the quality of care decreases substantially. These situations are conceptually similar to denial of service attacks — or to attacks affecting the integrity of a service, depending on severity — and, therefore, would be of relevance to security, incident or crisis management personnel.

Secondly, hospitals have become an increasingly prominent target for ransomware attacks, with works such as [168, 223, 296] further exploring the topic. Given this, we can envision a situation in which the ransomware recovery and surge capacity models could be composed to produce more traditional cybersecurity insights about the trauma room or even the entire hospital.

A trauma unit stands apart from regular medical services by specialising in the immediate and critical care of patients who have sustained severe injuries, typically due to accidents, violence, or other traumatic events. These units are specifically equipped and staffed to handle complex and life-threatening cases, often involving multiple injuries affecting different parts of the body. However, this is exactly why under crisis conditions, trauma units may become flooded with an influx of patients that can overwhelm their operational capacity.

For instance, after a major incident, such as a train crash or a terrorist attack, a large number of critically injured patients may arrive at the hospital in a short amount of time. Surge capacity refers to the number of patients that can be treated before the quality of care declines to unacceptable levels. The model is used to explore how different factors, such as staffing levels, staff skills and experience, or available equipment, affect surge capacity and provide insight into how capacity can be increased.

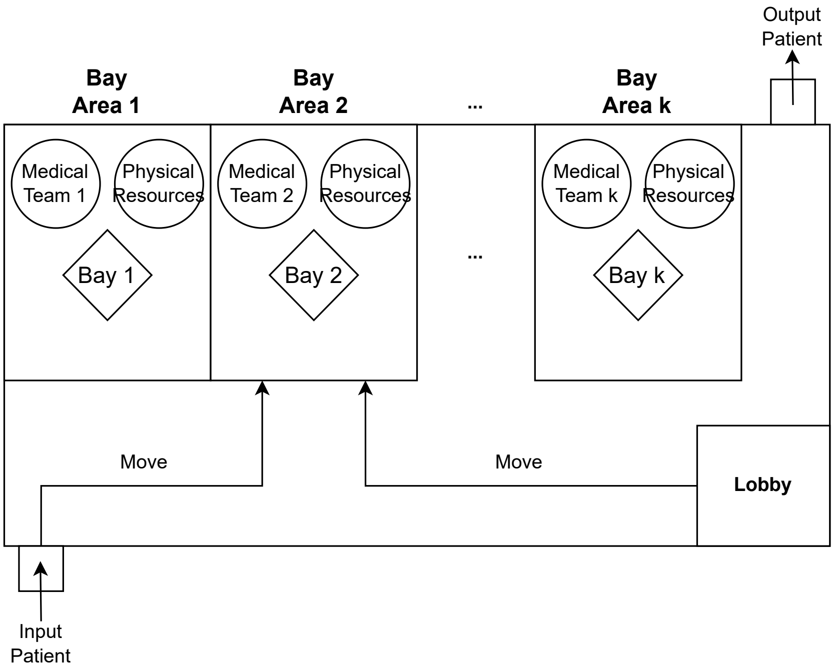


Figure 6.2: The trauma unit surge capacity model [152]

6.2.1 Structure & representation choices

As previously stated, structurally, the model is relatively simple, in the sense that no sub-model compositions are being performed. This can be easily observed in Figure 6.2, where the model representation focuses on the physical and operational characteristics of a trauma room, but without conceptualising other hospital areas that might be connected to it, such as perhaps a triage area, or the medical bays where the patients might be transferred to after no longer requiring emergency care.

In the model, patients, staff, equipment, and supplies are viewed as resources. Depending on a patient's injuries, different treatments, modelled as processes, need to be provided. These treatment processes require various resources, such as staff with particular skills, in order to execute. These resources are the limiting factor for surge capacity. For example, when there is a shortage of staff with a particular skill patients' treatment might be delayed, leading to undesirable outcomes.

In terms of *locations*, the structure for this model is quite straightforward.

There are locations for each treatment bay area, a lobby where medical personnel are awaiting patients, the emergency department, its entrance, and one representing the rest of the hospital, where patients move after treatment.

Similarly, *resources* are intuitively simple. There are resources representing patients, hospital staff, and available treatment bays. Each patient resource has an associated set of procedures that are required for treatment; each staff resource has a set of skills — with an associated skill level — which define their capabilities in terms of treatment.

With respect to *environments*, they are again being used for starting the patient resource arrival process according to different probability distributions.

Processes are slightly more complicated. Each patient has an associated process that moves the patient resource from outside the hospital, into the trauma room, and into a treatment bay when one is available.

Another process assigns the medical teams to the treatment bays. This process encodes the decision-making around team formation as learned from the hospital staff during the iteration through the modelling cycle.

When the staff resources are moved to a treatment bay area, treatment can begin. As mentioned above, each patient resource has a set of procedures that describe their treatment. Each of these procedures is modelled using a different process. Each of these processes requires staff resources with different skills. For example, the team leader process requires someone with a high level of the 'team leader' skill — experience in leading trauma teams; or, the intubation procedure (called 'advanced airway' in the model) requires someone with the 'advanced airway' skill at a suitable level. The procedures have dependencies. For example, the 'advanced airway' procedure can only begin after the 'patient assessment', 'airway assessment', and 'IV' procedures have completed.

Procedures can execute in parallel as long as their dependencies have been met and sufficient staff resources with appropriate skills are present. When team sizes are reduced because of higher patient numbers, some procedures might have to wait until a staff member becomes free, which extends the duration of the patient's treatment.

This representation of procedures, although complex, was chosen as it allows the effects of staff limitations to be captured by the model.

6.2.2 Co-design

After a brief examination of the model's structure and associated representation choices, we redirect our focus towards illustrating some of the co-design elements at play. Given the less familiar nature of the topic area for modellers, we describe two approaches employed to achieve a better understanding both during domain exploration and within the translation zone: a practice dummy exercise and a tabletop exercise.

From the very beginning, we can clearly state that co-design featured prominently in this work, even at goal level. For example, the initial question about surge capacity was brought to the modelling team — who at that time knew nothing about hospital operation and management — by a consultant anaesthetist from the hospital, who then worked closely with the modelling team throughout the process. Additionally, other medical personnel were involved in the process as part of the modelling team, on a case-by-case basis, but mediated with the help of the consultant anaesthetist.

As part of the domain exploration phase of the co-design cycle, the modelling team visited the hospital on many occasions, observing the operation of the trauma unit as well as other aspects of hospital operation. To better understand the decision-making process behind the formation of treatment teams, and how various treatment procedures are performed on patients, the modelling team observed a simulated treatment of a trauma patient on a training dummy — which is standard practice in hospitals to maintain skills. This revealed an important aspect regarding the treatment processes: different treatment procedures are temporally subordinated to other treatment procedures. In other words, for certain treatment procedures to be performed, some other treatment procedures must be first completed — to perform the 'advanced airway' procedure on a patient requires the 'patient assessment', 'airway assessment', and 'IV' procedures to be completed first.

In terms of data, the primary source for the modelling team was the consultant anaesthetist working closely with them. This was further supplemented or verified by discussion with more hospital staff members, and ranged from informal discussions with hospital colleagues about various aspects of the model — duration of procedures, accuracy of the flow of patients as described by the model — to a more formal tabletop exercise.

The tabletop exercise was designed to gain insight into how teams of staff are formed to treat patients, and how team formation and re-allocation changes when additional patients arrive. The exercise involved sticker notes representing members of staff and patients. We would begin the exercise with a single patient and asking the subject to form a team using the cards. We then introduced additional patients and asked how they would form new teams, and whether they would remove staff members from existing teams. From a co-design perspective, this can be understood as an attempt to infer domain behaviour from an artificial version of the domain. Nonetheless, this did not represent an issue, because the present medical staff explained the possible limitations of this environment, and how conditions could change during real treatment.

With respect to translation, it is important to note that the consultant anaesthetist learned how to express ideas in terms of the distributed system metaphor — for example, by thinking in terms of the process of patient treatment, and the resources — staff, skills, equipment — required for different procedures. Furthermore, he ensured the communication between modellers and other medical staff led to a translation of concepts towards the metaphor, without relevant information being lost.

In this context, the most relevant quantitative data used in the simulations was related to the medical procedure timings. In the current version of the model, the timings are quantified using ranges estimated by trauma unit staff. Future work will employ a nurse or other staff member to record accurate timings for different procedures to be used in the model. To ensure the validation procedure is understood and can be performed with the help of medical staff, a graphical visualisation of the model was produced, showing patients moving into treatment bays, the formation of teams, and the progress of the various treatment procedures. This was then shown to hospital staff to get feedback and to validate the behaviour of the model. We note here that the model did not attempt to construct a generalised representation of any trauma room, but rather focused on producing targeted insights regarding the specific procedures observed in the specific trauma room available. Therefore, during validation, the medical staff working in that trauma room assessed whether or not the model was producing 'believable' outputs, similar to those they encountered on a daily basis.

Lastly, after the model was constructed, it was used to simulate the trauma unit under real-world conditions as exhibited during the 2017 London Bridge terror attack — this included data regarding patient load, injury type, and arrival rate. In this setting, the model produced similar behaviour to the actual trauma room — validated by staff — and even produced some insights about the expected difference in surge capacity for daytime and nighttime staffing levels.

6.3 HP Organisational Recovery Model

The process of organisational recovery can be defined as the totality of actions an organisation can undertake to restore its functions after a traumatic event. From an information security perspective, this can be specified as the ability to restore the integrity and availability of data and services. Therefore, organisational recovery represents the main facilitator of organisational resilience. As described by Gibson & Tarrant in [129] and formalised by Ioannidis et al. in [155], organisational resilience should be conceptualised as an ability to adapt to changing organisational circumstances. From a security perspective these changes arise from differing IT usage patterns, the changing landscape of malware attacks as well as failures in IT processes. A wider view would include other phenomena like natural disasters, poor financial and human resource management or the inability to comply with new legislative initiatives.

In this model, we focus on studying endpoint device recovery mechanisms that enable recovery after failures and attacks. As it might be expected, multiple approaches to recovery exist and we give an overview of some of these in Section 6.3.2. The decision problem here is what mix of mechanisms and supporting IT processes needs to be put in place to have a robust and timely recovery strategy. The effectiveness of these approaches depends on the nature of the attack, spread, size, required recovery time, cost of deployment, existing policies, organisational structure, and employee knowledge and behaviour. Our modelling approach leads to the construction of balanced models that include a simulational component, allowing for different choices and outcomes, such as time to recover, to be explored.

We motivate our focus on the organisational recovery problem by first exploring the nature of ransomware attacks. As shown in Richardson & North [259], O'Kane et al. [231] and Oz et al. [233], ransomware on its own represents a constantly evolving threat with serious implications for organisations today. Furthermore, threat assessment reports from government and law enforcement agencies such as [107, 110, 134] conclude that ransomware is one of the most relevant threats for organisations; Tweneboah et al. [291] gives a more nuanced position on the variations across different industries. In their empirical ransomware research study, Connolly et al. [313] show that out of 55 different ransomware attacks on organisations of different sizes and industries, 21% have managed to fully disrupt business continuity for two weeks or more and in 19% of the cases, organisational recovery took several months if at all. Ransomware provides a rich attack space as it is distributed through a wide range of mechanisms and from a modelling methodology perspective, we show how a varied ransomware attack space can be abstracted and captured in different attack models that can be composed into our overall organisational representation.

In addition to different preferences, different organisations will have different requirements, structure, and architectures. The number of employees, their travel patterns, the size and number of offices, the devices used, the value of the information on different devices, and the network structure will all vary between organisations. We use a compositional modelling approach to provide the flexibility required to create models that can be adapted to capture all these differences between organisations. We discuss how the model and modelling approach allows us to adapt the parameters of the model to fit different these different organisational characteristics. We then show how decision makers can try different recovery and IT process resourcing choices through the model to explore how the choices could affect key operational criteria such as speed to recover. In doing so, we demonstrate the practicality and usefulness of the approach for strategic decision making.

6.3.1 Ransomware

Ransomware, as the name suggests, represents a subdivision of malware that is primarily being used to obtain benefits from a target by limiting the target's access and control over information and/or essential operational infrastructure. Such benefits can vary, from traditional cyber-crime motivators like economic revenue — often in the form of cryptocurrencies given the complexity involved in tracing and identifying beneficiaries of such transactions [181] — to more obscure ones, such as gaining competitive advantage economically, politically or militarily by crippling the operational capabilities of the target.

Since ransomware observed in the wild is constantly evolving, we attempt to identify a subset of characteristics that can enable thinking and reasoning about ransomware at a more general level. This represents an initial stage in the conceptualisation of the phenomenon and environments under study — and inherently of domain inference —, and will further influence both representation and parameter selection procedures at the level of the model presented further in this section.

6.3.1.1 Lockers & Crypto-lockers

As previously stated, ransomware can be viewed as the instantiation of a coercive action with visible effects at the level of information systems. As described by Schelling [265] and Pape [234], a coercive action compels a party to act involuntarily through means of either threats or force. Most ransomware adheres to this definition because once a target system is infected, either its operability is drastically reduced via actions such as overlaying various windows over visual interfaces, disabling I/O devices or simply interrupting the operating system booting process by displaying a notification, or, the information on the system is directly encrypted to a certain extent. Although targeting different assets, these two different types of ransomware which are sometimes called lockers and crypto-lockers essentially fall into the category of coercive actions, because their disseminators [175, 48] expect to obtain benefits from their victims by the use of force.

Information located on systems affected by these types of ransomware can usually be restored by either removing the processes causing the reduced operability — by using an anti-malware tool or reinstalling the operating system —, using a decryption key, exploiting certain implementation vulnerabilities in the ransomware code or simply paying the ransom while accepting the risks of back-doors and fraud. However, in the case of crypto-lockers, recovering the information can be non-trivial, particularly in the case of ransomware based on hybrid encryption with large keys.

A particularly relevant example for both the locker and crypto-locker categories is the Reveton Ransomware. In its original form, this ransomware removes the user's ability to perform operations in the standard boot mode of a device by interrupting the booting process, displaying an immovable warning notice in full screen and disabling keyboard shortcuts for minimising screens or opening the task manager [207], but without encrypting any system or user files. The ransomware achieved persistence by masquerading as a .dll file and having a shortcut of itself in the main Windows directory to ensure it would be run every time the operating system is launched. However, it is not the locking procedure that makes this ransomware particularly relevant, but the fact that the warning notice contained a customised direct threat to the users. The warning notice is presented as a fine coming from a government agency — FBI, Australian Federal Police, Metropolitan Police, etc. — based on the user's location and on 'evidence' that the user has breached either copyright, child-pornography or illegal access to information systems law. Furthermore, in an attempt to increase the coercive force, the notice contains the user's IP, location and IPS and has an embedded recording capability that is being reflected back to the user, giving the impression of real-time surveillance. With respect to spreading and distribution, Reveton has been observed as having diverse mechanisms, but none of them were particularly targeting a specific group of users. For example, the most observed distribution method was through exploit kits such as BlackHole [287] or Cool [145] that used the CVE-2012-1723 Java vulnerability, but distribution through phishing or application downloads on mobile devices [108] was also possible. In addition to Reveton, such exploit kits usually dropped additional malicious software such as the Citadel [207] or Zeus [305] Trojans that harvested credentials, monitored web traffic, altered HTML code displayed in browser and introduced the infected device into botnets. Although in theory, these operation were not part of the ransomware itself when Reveton appeared around 2012, some were imported in later versions [288] that included BitCoin mining, file encryption or credential stealing capabilities through the Pony Stealer module [106]. As shown in [177, 213], even though Reveton was not based on hybrid encryption, its financial impact was as high as $50,000 a day or $400,000 per month during its peak period around 2012-2014.

6.3.1.2 Leakware

Based on strategies to increase the coercive force of the attack, a third category of ransomware can be specified: leakware. Generally, malware in this category may or may not affect the ability of its victims to access relevant information or use the affected information system. The primary goal of such attacks is to obtain sensitive information — common targets include intellectual propriety, third-party information or information that might be deemed as embarrassing — and then threaten the victims with publication in case the attacker's demands are not met. In more abstract terms, this strain of ransomware affects the degree of control that a user has over information, shifting the primary focus of the coercive action from usability or ability to use systems to restraining the possible spread of sensitive data. However, compared to lockers, ransomware in this category are harder to detect because they do not perform operations that reduce the operational capacity of the target. Although they might present automatic spreading capabilities, the actual data exfiltration operation might be performed manually, similarly to the Grozio Chirurgija [138] cosmetic surgery data breach.

The 2017 data breach that had occurred at the Grozio Chirurgija [138] cosmetic surgery clinic in Lithuania can be considered, from our perspective, as a leakware attack by the APT28 group, or FancyBear. Even though the precise infection vector has not been made public, roughly 25000 private photos and other personal patient data such as passport scans or national insurance numbers have been published by the threat actor in an attempt to obtain ransom from both the clinic and its patients to stop further publishing. The ransom demanded varied between €50 and €2000, based on the sensitivity of the information stolen and the price of the complete database was 50 bitcoins. No data was encrypted on the clinic's machines and the attack was discovered only after the publishing of the data.

Nevertheless, pure leakware attacks did not reach the popularity of lockers, mostly because the extortion tactic they employ can also be used in composition with the forceful coercion of lockers for a greater impact on the victims. This type of double extortion approach can be seen in newer ransomware strains such as Maze [169], Conti [294] or DarkSide[226] which have been the cause of a series of targeted attacks in the near past: the Maze infection of Allied Universal [3] in 2019, the Conti attacks on JVCKenwood [2] in 2019, Ireland's Health Service [218, 146] in 2021 or the Darkside attack on the US Colonial Pipeline [8, 143] in 2021, with the interesting aspect that in the Colonial Pipeline incident, the Darkside group attempted a triple extortion tactic by threatening with additional denial-of-service attacks in case the ransom was not paid.

6.3.1.3 Destructive ransomware

A fourth category of ransomware has become more and more prevalent in recent years: destructive ransomware. When compared to the other three categories, this type of malware no longer focuses on obtaining benefits from victims via coercive actions. Destructive ransomware directly inflicts irreparable damage to an information system by deleting, overwriting or encrypting both user and system files and memory regions. Therefore, instead of coercing victims, disseminators of this malware type usually conceptualise the loss sustained by the victim as an actual gain and are driven by political motivations, so a direct way of recovering the information is usually not integrated in the malware design process.

The Shamoon [92, 50] infections of Saudi Arabia and Qatar's national oil companies in 2012 can be seen as a relevant example of this type of attack. The primary characteristics of the attack were the presence of a 'logic bomb' that activated the wiping and overwriting of the hard drive data in the infected machines at 11:08 am on Wednesday, August 15, in an attempt to inflict maximum damage and reduce the chance of discovery — because it was believed that a majority of staff were on holiday —, the creation of a service enabling the malware to persist on an infected network of devices and the ability to automatically spread via network shares. The infection vector for the attack is believed to be a malicious email, as described by Chris Kubecka, a former security advisor to Saudi Aramco [158].

To accentuate the political motivations, similar types of malware have been attributed to the current Russian invasion of Ukraine [206, 71], but only a single variant — Hermetic Wiper [136] — was distributed using worm-like spread capabilities.

6.3.1.4 Ransomware dissemination

Furthermore, we describe two additional mechanisms for disseminating ransomware: via human operation and ransomware-as-a-service. As illustrated in Microsoft's security best practices report [88] from June 2022, human-operated ransomware represents 'an active attack by cybercriminals that infiltrate an organisation's on-premises or cloud IT infrastructure, elevate their privileges, and deploy ransomware to critical data.' and directly focuses organisations rather than singular devices. These types of attacks behave as shown in the Mitre ATT&CK [15] matrix model. For an example of threat conceptualisation using this approach, see Xiong et. al. [308] Nonetheless, the 2021 attack on the information technology infrastructure company Kaseya [67, 111] by the REvil group can be considered an example of human-operated ransomware: the malicious actor managed to leverage a vulnerability in the proprietary remote monitoring and management software for the VSA

Cloud and SaaS servers — which shows reconnaissance has been performed — and disseminated a ransomware payload to both Kaseya and a subset of its clients. The impact of the incident varies across sources, with REvil claiming to have encrypted more than one million systems [67] and Kaseya declaring between 800 and 1500 businesses as being affected [253].

However, the latest years have not only brought specialisation efforts in terms of more targeted and sophisticated strains of ransomware, but also an increase in the accessibility of deployment for non-technical users. As described in [102] ransomware-as-a-service represents a specialisation of the software-as-a-service model: skilled malware writers produce high-quality samples which are then employed by less skilled attackers either via a one-time payment or subscription method to be deployed against certain targets, and everything is done via easy to used web interfaces which sometimes even have user reviews, scoring systems and catchy marketing phrases. For extended reviews, see Keijzer [166], Alwashali et al. [7] or Meland et al. [198] for en economic perspective. To grasp the current degree of evolution, Karapapas et al. [163] even describe a proof-of-concept ransomware-as-a-service model based on IPFS file system and Ethereum blockchain. Nevertheless, from a behavioural perspective, the actual ransomware strains used via ransomware-as-a-service do not differ from other strains. In the future, we expect this model of operation to continue developing and provide users with even more dangerous types that might employ direct handler operation or zero-day exploits.

6.3.1.5 Reflections

Although ransomware in the above categories is being used to achieve different objectives, the technological means of achievement are similar and have undergone formalisation and generalisation attempts over time. For example, Young & Yung describe ransomware in their seminal 1996 article as a ‘cryptovirology attack’ [312] in which cryptographic approaches are used offensively to ‘mount extortion based attacks that cause loss of access to information, loss of confidentiality, and information leakage, tasks which cryptography typically prevents’ [312]. Their attack follows a hybrid encryption scheme and resembles the formalisation of a digital envelope. Its main steps are presented below, at a reasonably high level of abstraction.

- The attacker generates a public/private pair of encryption keys, constructs the ransomware code and places the public key inside it;

- The attacker releases the ransomware, which then finds and infects a victim based on its spreading and infection mechanisms;

- The malware generates a symmetric encryption key using a randomness source and then encrypts its target files with it. To generate the key, the ransomware either contains a random number generator or accesses one that is available on the victim machine in the form of an encryption library, API or simply uses an OS internal process as randomness source;

- The ransomware encrypts the symmetric key with the public key and overwrites any remaining plain-text and the symmetric key;

- The ransomware leaves a note with the ransom and means of contacting the attackers.

Using this approach and assuming that the ransomware implementation does not suffer from trivial vulnerabilities such as weak random number generators, failure to remove the symmetric key from the victim's machine or leaving residual information in the RAM memory, the complexity of retrieving the data is equivalent to the complexity of reversing the encryption schemes used. Assuming that the encryption key sizes are large and the encryption algorithms are deemed as secure, breaking the encryption represents a computationally hard problem, for which no algorithm can produce solutions in polynomial or lower time. Examples of computationally hard problems used in the space of cryptography include the Integer Factorisation Problem [46], Discrete Logarithm Problem [197], Elliptic Curve Discrete Logarithm Problem [123], Lattice Shortest Vector Problem [5] or Lattice Closest Vector Problem [199]. For an in-depth review, see Salem et. al. [263].

For the purpose of our model, the above described behavioural schema might be seen as too focused on the internal operations of the ransomware. However, modelling ransomware does not require understanding only the inner workings of the ransomware code. For example, the identification of a victim could be an automated process of scanning IP ranges, or it might involve a carefully planned reconnaissance procedure. Once identified, the victim could be infected through a series of means: phishing, spear-phishing, physical social engineering to provide access or introduce malicious hardware components into the system, file downloads, Trojans and others. Furthermore, after the initial infection, additional worm-like network spreading [153] might occur via misconfigured network shares — with or without the need for zero day exploits such as EternalBlue [43] — automated phishing based on gathered information from the already infected devices or direct credential harvesting and dropping if escalation of privileges is successful [66, 65]. For a complete list of ransomware strains that contributed to the above behavioural classification, see Appendix A, Table A.1. Given the evolution of ransomware and the development of modern strains such as Maze or Conti which combine characteristics from multiple categories to increase the coercion on users, a similar compositional capability at the level of modelled ransomware behaviour is required and provided by our co-design approach.

6.3.2 Recovery Techniques

As described at the start of the section, the goal of organisational recovery is to bring an organisation into a business-as-usual state after a traumatic event. In the context of this model, this traumatic event is represented by a ransomware outbreak, although the model we present could be used with other parameter configurations to simulate other types of events, such as a flooding denial-of-service attack, for example.

For a better understanding of the possible recovery mechanism choices, we firstly draw attention to a few important aspects about the nature of the threat. Modern ransomware strains are harder to fully erase at an organisational level because of a series of spreading and persistence mechanisms: as seen in the previous paragraphs, ransomware such as WannaCry, NotPetya or BadRabbit extend the class of crypto-lockers with powerful worm-like spreading mechanisms constructed on top of zero day exploits, greatly increasing their dissemination speed. However, this evolution cannot be considered as a statistical indicator of a higher severity attack on its own yet: as shown in Connolly et al. [313], the hypothesis that 'the crypto-ransomware propagation class influences the impact severity of a ransomware attack' is rejected in their study with the comments that a combination of factors such as the nature of business, availability of resources to recover data or pay the ransom, the type of systems affected and level of preparedness should be further analysed. An insight we consider relevant in their study is that the overall severity of 'generation two crypto-ransomware', which maps onto our crypto-locker category without worm like spreading achieved a score of 0.32, whereas the 'generation three crypto-ransomware' which manifests worm-like spreading only achieved 0.23. We argue that this should not be interpreted as the spreading factor not being relevant to the severity of the infection, but rather that the security posture and type of organisation play a more important role than purely the technical advancements of the ransomware. Furthermore, one of the most severe attack they identified was a worm-based crypto-locker that targeted a large public organisation — GovSecA as named in the study — and managed to encrypt close to 100 servers. At the time of their study, the organisation still had not fully completed their recovery processes, almost 8 months later. The example does not give details about why this recovery period was so significant, but we allow ourselves to speculate that reinfection played a big part. For more information about reinfection, including a stochastic epidemics model that simulates parts of the behaviour of Viking.gt malware on Norwegian Bank, see Hole [144].

Nevertheless, it is not only the spreading speed that has evolved over time, but also the persistence mechanisms employed — with techniques such as modifying registry keys, altering run once keys, the bootexecute key, boot helper objects, keys used by the WinLogon process, startup keys, launching of additional services to facilitate reinfection in the case of recovery, maintaining a command and control structure or avoid detection methods during data exfiltration or altering the DLL search order mechanism itself as being only a handful of approaches. For additional information and platform specific approaches, see [133, 303, 47]. In addition to these purely persistence oriented mechanisms, modern ransomware has also exhibited a series of destructive elements such as overwriting the MBR (Master Boot Record) [272] in the case of NotPetya and Shamoon, which make backups and re-imaging of devices mandatory to ensure business as usual can be re-established. The details of the process and how organisations often end up paying ransoms rather than rebuilding their systems are described in [302].

However, it is not just ransomware that spreads over networks that necessitates recovery at scale. Malware families, such as Emotet [70], also have mechanisms to spread rapidly through email. Spreading patterns have evolved to include WiFi [36], and can make it hard to clean corporate systems without taking everything offline. Some malware can be cleaned by anti-virus systems, but it can be hard to guarantee and trust that systems are clean; hence, easing the re-imaging process can become an essential part of a company's response to malware attacks. For example, the SANS incident-responder's handbook recommends re-imaging of systems' hard drives to ensure malware is eradicated [176], with recent surveys showing incident response processes often leading to re-imaging [49].

Companies are becoming aware that they must start planning for both large-scale and smaller-scale outages in order to get their systems and staff back up and functioning as soon as possible [105]. There are, of course, various products and approaches to backup, re-imaging and restoration. Yet, there is a lack of tools to help IT decision-makers decide on the most appropriate strategy and assess whether they have the necessary tools and infrastructure in place. This is the problem that we look at with the help of this model: we demonstrate how modelling and simulation can be used to aid the decision-makers in the choices they make. A good example of an executive level document which details the Microsoft strategic and operational approach against ransomware can be found in [88] and focuses at least on secure backup, privileged access plan, data protection plan and security posture and governance, all provided with clear accountability paths and KPIs.

There are several approaches that organisations might employ in order to maintain the operation and re-imaging of client systems. Underlying these approaches there are three basic choices: full system backups, re-imaging to a corporate image, and modern management systems.

6.3.2.1 Full system backups:

Some companies will have backup systems that keep a full system backup of each client. Restoration will then happen by reinstalling this full backup. The backup vendor would support this restoration process with a typical reinstall process involving the download of a Windows PE agent along with the full system backup, placing this onto a bootable USB stick, and then, through the BIOS menus, booting into this cut-down version of Windows, which will reinstall from the full backup — for an extensive survey of classical recovery methods, see Chervenak et al. [69]. For an updated version, including information regarding cooperative approaches, see Killijian et al. [170]. Taking full backups is becoming less common, particularly in the hardware sense, as it means keeping copies of many standard system files and there are advantages to re-imaging to a clean up-to-date OS image.

6.3.2.2 Re-imaging to a corporate image:

A more common scenario is for companies to have a standard corporate image along with a data backup strategy. For example, a company will often create a Windows image containing corporate management tools — such as a management agent for a system such as Microsoft's Endpoint Configuration Manager — and its security software, both AV and EDR systems, such that when a client image installation happens the system is secure and manageable [257]. Microsoft provide a management deployment toolkit [204] that describes and supports this overall deployment process. The management tools will then typically help install other applications as required. Such images will be updated regularly — quarterly or half yearly, for instance — to include the latest version of Windows, patches, and software.

Data backup may be integrated through backup software to an enterprise server or the cloud, although companies are increasingly using synchronised cloud-based storage such as OneDrive, where data is stored on the cloud with local copies cached on the endpoint.

From a recovery perspective, as a user decides they need to re-image a system they get hold of a bootable image on a USB stick (or occasionally a DVD) and boot into this to re-image the system. In an office environment, where there is IT support, the IT engineers will maintain a set of current OS images on bootable media. In smaller offices, or where there are home workers, the OS image can be downloaded and there will be instructions for the user to create the bootable media and reinstall. Such instructions can be complex for a typical user and require access to a USB stick that can be wiped and reformatted. If a user is at home they would need a functioning computer to use to download the image.

IT support labs will also often have a PXE (Preboot Execution Environment) boot set-up to make re-imaging easier [203]. They have an image hosted on a local server and use PXE boot to point the system to that image to install. This can ease the problem of setting up larger numbers of client systems, although it requires staff and infrastructure.

6.3.2.3 Modern management systems:

There is an increasing move towards the use of modern management systems (Uniform Endpoint Management), such as Microsoft's InTune System [202]. This approach allows the use of a standard Windows image, such as that initially placed on the computer, rather than a specially maintained corporate image. During the install process, the management infrastructure will push critical security patches, AV signatures, Windows domain policies (Group policy objects), and necessary software. This produces a similar effect to having a corporate image, but removes the need to maintain custom images.

Typically, after a new image has been installed, an out-of-the-box experience (OOBE) process runs, the user will be led through configuration screens, and will login using their corporate email. The login directs the system to a cloud-based management server, so that the enterprise configurations can be found and installed, and the computer added to the enterprise domain. This process can be simplified further using Microsoft's AutoPilot [205], where a computer is preregistered as belonging to a company, and user interactions and configurations can be simplified and reduced.

The re-imaging process still requires that the user can get hold of a clean Windows install image. However, the company does not need to maintain and host its own Windows image. Instead, a user can download the latest OS copy from either the PC manufacturer or from Microsoft. Companies using these mechanism will typically use cloud-synced storage, discussed above, to provide data resilience and, as the system is re-imaged and added to the corporate domain, data will gradually be synced back to the client.

6.3.2.4 Re-imaging Mechanisms

Re-imaging will typically involve booting from an ISO image and installing this onto a drive, or via a reduced version of Windows, such as WinRE or WinPE, which can install Windows from WIM files. Windows itself also includes a number of repair processes [201] — for example, allowing rollbacks to previous snapshots using Shadow Volume Copy. However, ransomware often disables volume shadow copies and deletes snapshots, making recovery and the retrieval of older files hard. Incident responders often recommend a clean install to ensure malware is eradicated.

Re-imaging processes require a boot into a system rather than the normal OS. The boot process is controlled by the BIOS, which will have a defined boot order and set of devices that can be used to boot the system. Many systems will boot from an attached USB device or PXE boot before the main disk, making re-imaging easy — but with no controls. Early in the boot cycle, users can get into the BIOS menu and boot to an alternative device. Some enterprises will lock down the BIOS with passwords and ensure the system boots only from the internal disk and, in this case, re-imaging will require an IT support engineer who knows the BIOS password.

In this context, HP has built a bare metal recovery system — HP 'Sure Recover' [149] — into the BIOS in order to simplify the re-imaging process. The Endpoint Security Controller — EpSC — holds a configuration containing the location of image-servers, which may be either HP's servers for standard Windows images or other servers specified by the enterprise. The configuration also contains public keys of the authority allowed to sign the Windows image to be installed and, in this way, an enterprise can guarantee the image being installed has integrity and has been approved. Recovery can be triggered by the user at boot time through the BIOS recovery option or it can trigger automatically when the system fails to boot — such as with NotPetya. When triggered, the BIOS gets this configuration information and uses it to download a recovery agent, which then downloads the full OS image and re-images the system. Both recovery agents and the full image are signed and the signature is validated as part of the recovery process ensuring authenticity of the recovered image. The process simplifies recovery for the user as they no longer need to be able to find where to obtain the OS image and do not need an available USB stick. From the enterprise perspective, it allows the enterprise to lock the BIOS without support engineers doing rebuilds, as well as guaranteeing that the image installed is correct.

An additional option is available, HP 'Sure Recover' Embedded, which adds additional storage onto the endpoint device that is used to keep a local up to date copy of the recovery image. It reduces network download times and means recovery can happen when no network is available, or when networking is limited or metered.

6.3.2.5 Reflections: enterprise recovery choices

The descriptions above show that there is a wide range of choices available to the enterprise as it looks to implement an image management and recovery strategy; for example:

- Maintain a corporate image or use a standard image. There is a choice as to how often the image is updated. After recovery, patches will need installing and updating images more often will reduce the need and time taken for post re-image patching. However, this option requires additional resources to manage the actual image, in the sense that its actual content might require frequent updates. When a company maintains its own OS images, they must maintain servers to support the download of images. Download speeds may depend on the location of these servers, or how they are distributed over the world, the location of users, and the network bandwidth available in an office or to a home or travelling user. The volume of traffic to these servers will depend on the recovery scenario — in terms of the numbers of users likely to be recovering at a given time and on the state of the underlying infrastructure.

- How much IT support is needed and how much in each office? Many companies are looking to reduce IT costs, and this often creates pressure to centralise help-desks and remove support from offices. However, a lack of local support and locally kept OS images can delay recovery times for multiple reasons: some users might require physical support in starting the recovery procedure and some might simply needed to be guided remotely, but a bottleneck in the actual support stream can lead to a reduced stream of recovery even if the infrastructure is extremely capable. At the same time, the enterprise will need to plan for remote workers either working from home or as they travel; such support needs have increased dramatically with Covid 19, for example, since the actual work location of employees is less geographically bound to an office.

- Control over the re-imaging process can bring various choices with which the enterprise may wish to lock down its client platforms, but this adds a considerable burden in recovery.

- There are many different data backup strategies, from the use of cloud synced drives through to full system backups. Each will have an impact on the ease of recovery and potential user data loss.

- The choice of employing different re-imaging techniques, such as using a USB stick or PXE boot, in comparison to having recovery mechanisms such as HP ‘Sure Recover’ built into the system — whether with an image stored locally or downloaded.

6.3.3 The Recovery Model

Having described the relevant domain inference information regarding the nature and behaviour of ransomware and recovery mechanisms in the previous subsection, it is now time to focus on the actual model. Since the main goal of this model is to help organisations in deciding what allocation of recovery technologies best suits them based on an understanding of the consequences involved, we concentrate on a partially-generic model that allows for further customisation of both the structure, because of the modularity provided by the underlying distributed systems metaphor, and parameters.

Making use of the compositionality provided by the method, we construct our organisational recovery model as a composition of four different sub-models: a device model, a network model, a server model and a malware model. The high-level architecture of this composed model can be observed in Fig. 6.3.

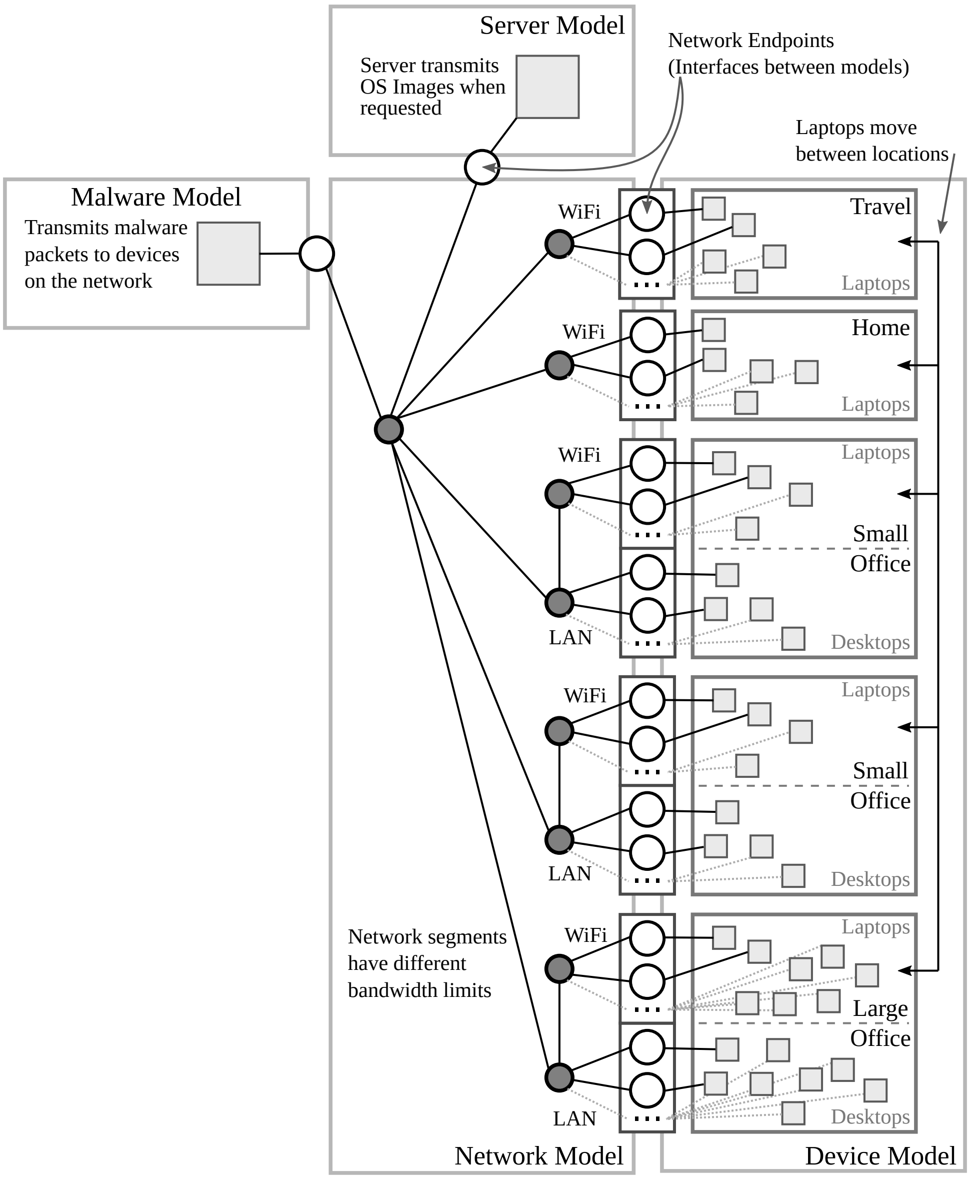

Server Model
Server transmits OS Images when requested
Network Endpoints (Interfaces between models)
Laptops move between locations
Malware Model
Transmits malware packets to devices on the network
WiFi
Travel
Laptops
Home
Small Office
Desktops
LAN
Large Office
Network segments have different bandwidth limits
Network Model
Device Model

Figure 6.3: Organisational Recovery Model [151]

6.3.3.1 The device model

The device model stands out as the most intricate among the four, and is used to represent the physical structure of a medium-sized organisation. This is comprised of a series of physical locations — offices of varying sizes, home, hotels, coffee shops — where employees can work on their devices if a network connection is available. Naturally, employees can move between locations, but the specific location they are present in influences their available resources and restricts their possible recovery choices: locations have different network bandwidth allocation and non-office locations do not have access to a physical help desk. An iteration through the model should be viewed as a temporal sequence of movement between locations, working, getting infected by ransomware and then performing recovery actions — USB recovery, network recovery or embedded recovery or combinations of them. A conceptual diagram of the processes present at the level of the device model is presented in Figure 6.4. The explicit list of model components is presented below:

- **Resources** — devices, blank usbs, usb images, os images, recovery agent, image requests, image responses, network data, helpdesk admins;
- **Locations** — work locations (home, offices, travel locations), network endpoint locations;
- **Processes** — installation of images from usb, embedded, network-based or mixed, usb recovery, embedded recovery, network-based recovery, mixed recovery, fetching image from server, fetching recovery agent from server, admin delay, device movement;
- **Interfaces** — network endpoints;

Following the distributed systems metaphor, we conceptualise devices as resources and both the work and abstract locations as locations: work locations could include small or large offices, travel locations such as airports, coffee shops or the home; abstract locations exist mostly in the form of endpoint locations, an abstraction for the network switches or routers that connect devices to a network. As an implicit organisational policy, each large office is considered as hosting a help desk with a variable size that can help users perform a recovery process for their device if their level of expertise does not allow them to perform it on their own. The helpdesk employees are conceptualised as resources associated with the helpdesk. Furthermore, the employees are not being modelled explicitly, but rather as devices that move between locations and can perform work related activities or recover a device. The actual recovery actions performed by the devices or helpdesk employees are modelled as processes.

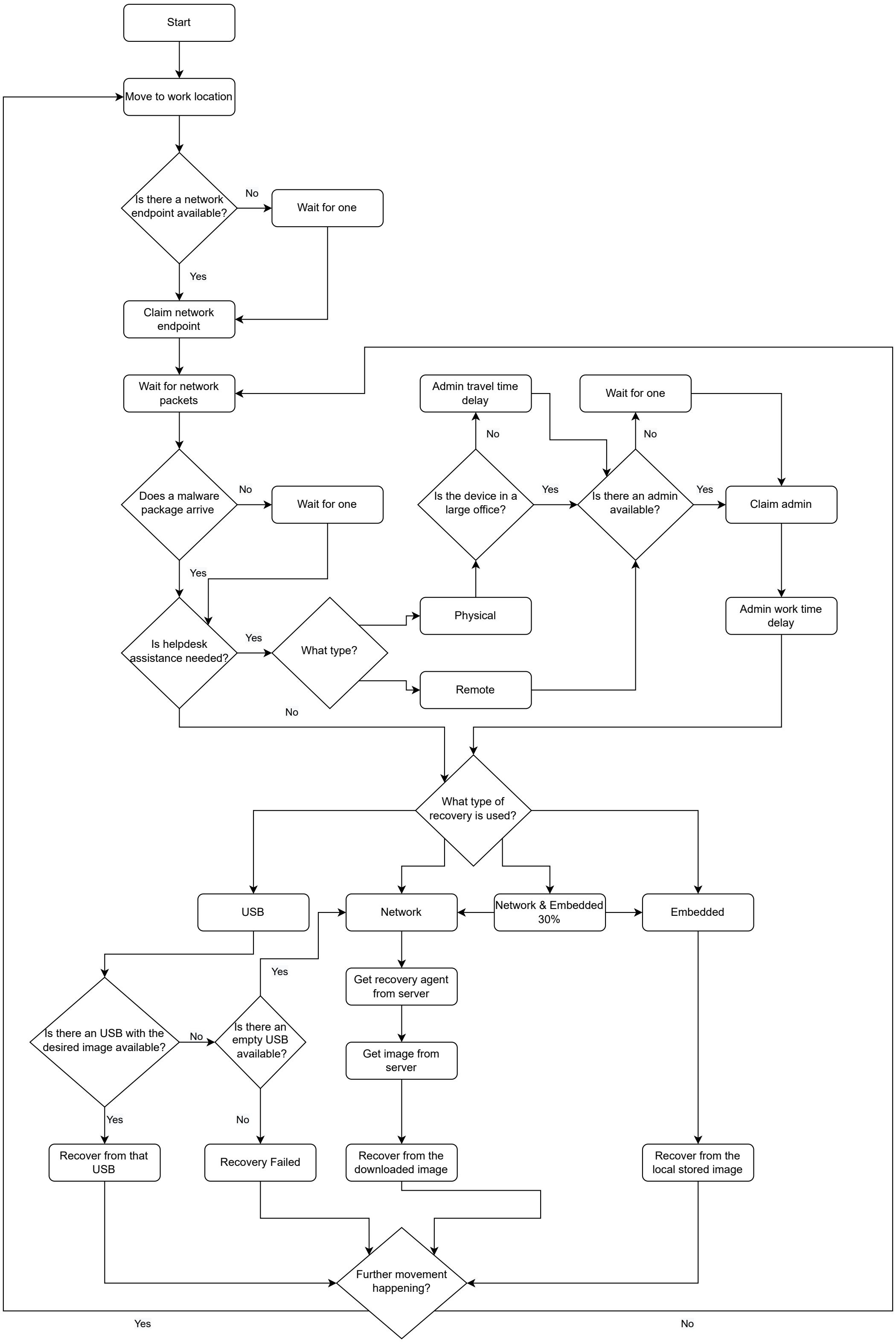

Start
Move to work location
Is there a network endpoint available?
No
Wait for one
Yes
Claim network endpoint
Wait for network packets
Does a malware package arrive
No
Wait for one
Yes
Is helpdesk assistance needed?
Yes
What type?
Physical
Remote
No
Admin travel time delay
No
Is the device in a large office?
Yes
Wait for one
No
Is there an admin available?
Yes
Claim admin
Admin work time delay
What type of recovery is used?
USB
Network
Network & Embedded 30%
Embedded
Is there an USB with the desired image available?
No
Is there an empty USB available?
Yes
Yes
No
Get recovery agent from server
Get image from server
Recover from that USB
Recovery Failed
Recover from the downloaded image
Recover from the local stored image
Further movement happening?
Yes
No

Figure 6.4: Device Model Process Diagram. Rectangles show significant points in the behaviour of the process. Diamonds show where the process branches. [151]

For example, when a device moves to a location, it obtains use of a network endpoint so it can send and receive data on the network by claiming one of these availability resources; when it leaves, it releases that availability resource so another device may use that endpoint later. Each device in the model has its own process. This process is responsible for all the device's behaviours, from movement, to recovery, to sending and receiving data on the network. As part of the configuration of the model, each device is set up with: a movement pattern (the sequence of physical locations to move to, and probability distributions determining the length of time it stays in each one) and a method of recovery to use. Furthermore, separate processes determine whether or not a specific device can recover on its own and if not, what type of helpdesk assistance it requires. With these parameters, the device process executes. It moves the device resource from physical location to physical location according to the sequence, remaining in each one for a certain amount of time. If a particular device should recover, action indicated by the arrival of an infection package to the device, the device process initiates this.

As it was briefly explained in Section 6.3.2, at the level of the model, we look at four recovery methods.

- **USB Recovery** — A fresh OS is installed on devices from a USB stick. The device process tries to claim a USB stick resource with the OS image on it; if none are available, it tries to claim a blank USB; if no USBs of either type are available, and none *become* available, the recovery process fails. If a blank USB is obtained, the process must download the OS image by sending a request and waiting for the response, and writing it to the USB. This destroys a blank USB resource and creates a new USB stick resource with the image on it.

- **Network Recovery** — Devices request and receive an OS image over the network from an image server. The process starts by creating a request to download the recovery agent and moving it to the network endpoint so it can be sent to the server storage sub-model; it then waits for the response by claiming a response resource at the network endpoint. After receiving this, the process creates a request for the OS image, moves it to the endpoint, and waits for the response. For clarity purposes, we restate here that the network recovery process is behaviourally similar to, and uses the HP ‘Sure Recover’ [149] — Section 6.3.2.4 — with network re-imaging as a primary reference point. This differs from PXE re-imaging used in IT labs and for servers that would work on a LAN and not be available wherever the user is located.

- **Embedded Recovery** — Devices have a built-in storage capability that is used to hold an OS image for recovery. To model embedded recovery, the device process simply waits for the amount of time (as measured on real-world devices) it takes to restore from the embedded storage. This is based on the HP ‘Sure Recover’ [149] with embedded storage.
- **Mixed Recovery** — Combines the embedded and network-based recoveries. An allocation strategy is required. For example, devices in small offices could be allocated embedded recovery capabilities whereas those in big offices might rely on network recovery.

The above points, supported by Figure 6.4, describe the recovery choices and underlying actions that have been modelled. However, these do not encompass the stochastic, temporal nature of the model. Throughout all these steps in the process, there are time delays modelling the length of time it takes to, for example, verify an image after download, copy an image to disk, or run the installer. Additional delays are introduced if helpdesk assistance is required: helpdesks have a finite number of admins who can offer assistance remotely or in person — therefore, devices end up queuing for the helpdesk resource — and, if in-person assistance is needed, the admin might have to travel to a different location if the device in need is not located in the same large office as the admin.

6.3.3.2 The network model

The network model acts as the central representation for the organisation’s communication network and facilitates the interaction between all the models via the transfer of network packets. Functionally, its goal is to ensure network packets arrive at their correct destinations after an amount of time influenced by the network congestion. Structurally, it contains abstract locations symbolising network endpoints that devices can access to connect to the network. These locations are interlinked, representing the network segments that actual network packets would traverse. Briefly, the model operates as follows: network packets arrive at endpoints, are transferred to an abstract transit location, and after an appropriate delay aligned with data size, network segment speed and congestion, dispatched to their respective destination endpoints. A conceptual diagram of the processes present at the level of the network model is presented in Figure 6.5. The components of the sub-model are shown below:

- **Resources** — network data;
- **Locations** — network endpoint locations, transit location;
- **Processes** — transfer data;
- **Interfaces** — network endpoints;

Additionally, it includes a separate location representing data in transit. Structurally, all these locations are connected in the form of a graph of network segments representing the actual network routes packages would be routed through. This sub-model has one process, which claims resources that arrive at the endpoints, moves them to the transit location, and, after a delay suitable for the size of the data and the speed of the network segments it would traverse, moves them to the destination endpoint and releases them. If transfers are already ongoing when more resources are claimed or released, the process recalculates the time when the transfers will finish based on how throttled the network segments are.

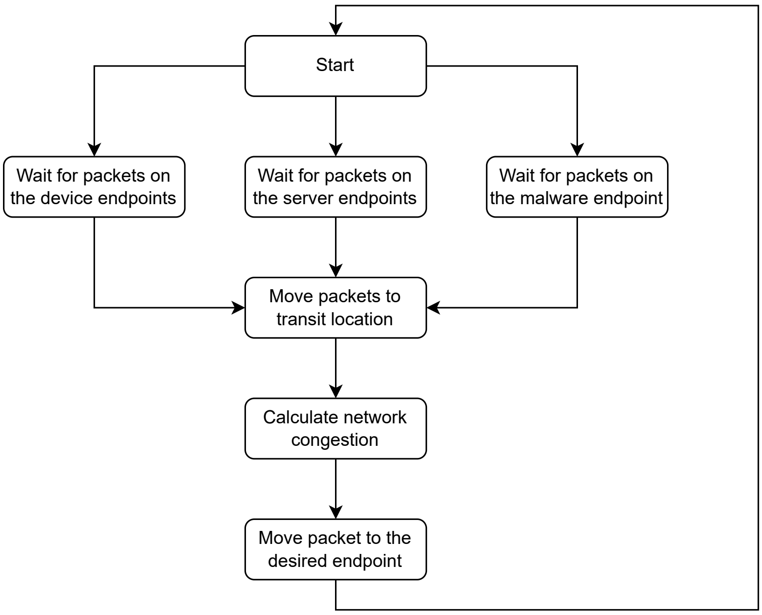

Start
Wait for packets on the device endpoints
Wait for packets on the server endpoints
Wait for packets on the malware endpoint
Move packets to transit location
Calculate network congestion
Move packet to the desired endpoint

Figure 6.5: Network Model Process Diagram [151]

6.3.3.3 The server model

The server model essentially depicts the storage area for recovery images which are requested by devices over the network, depending on the recovery type chosen in the device model. Its behaviour closely resembles the real interaction between a user and a server: it is used to determine if network requests contain valid recovery image requests and if so, to send the required images to their corresponding devices via the network. Its components are showcased below:

- **Resources** — network data, os images, recovery agent, image requests, image responses;
- **Locations** — server network endpoint location, storage location;
- **Processes** — process messages, move network data to the endpoint;
- **Interfaces** — network endpoint;

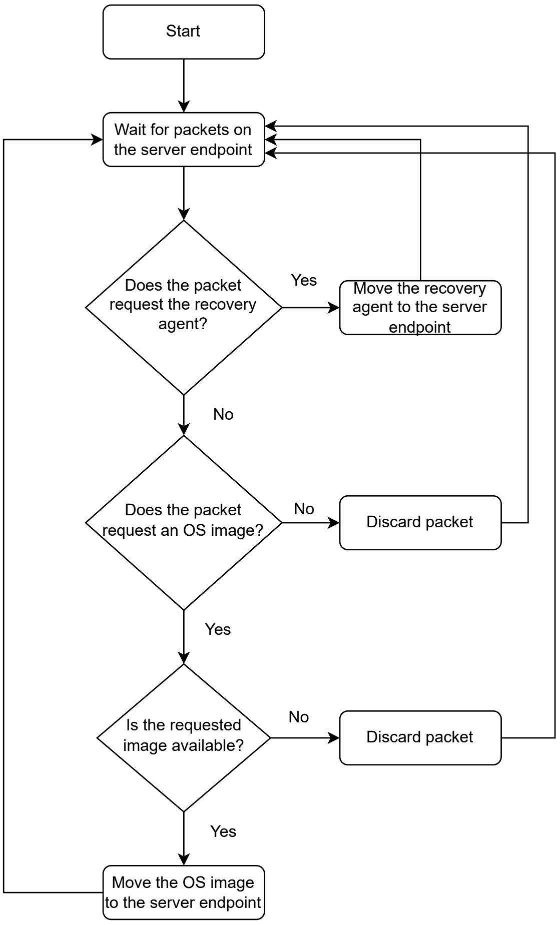

Start
Wait for packets on the server endpoint
Does the packet request the recovery agent?
Yes
Move the recovery agent to the server endpoint
No
Does the packet request an OS image?
No
Discard packet
Yes
Is the requested image available?
No
Discard packet
Yes
Move the OS image to the server endpoint

Figure 6.6: Server Model Process Diagram [151]

At a high level, it is comprised of a network endpoint and a storage abstract locations. The storage location contains clean operating system images and the recovery agent necessary for the network based recovery. Behaviourally, a process awaits requests from the devices on the network endpoint and then delivers the operating system image requested back to the network model. A process diagram detailing the operations can be seen in Figure 6.6.

6.3.3.4 The ransomware model

The ransomware model encapsulates the main aspects of the ransomware behaviour: the targets, the spread pattern and the infection timings. Structurally, it contains a single network endpoint location and three different processes: one that determines ransomware targets, one producing the timings when the infection packets are being injected into the network based on different probability distributions, and another one which determines if the targeted device actually gets infected and formats and sends the actual malicious packets. After the injection procedure, the packets are handled by the network model, reach their targets, infect them and then the device model triggers the recovery processes. The process diagram can be seen in Figure 6.7.

- **Resources** — network data;
- **Locations** — malware network endpoint location;
- **Processes** — choose targets, determine timings, infect targets;
- **Interfaces** — network endpoint;

The combination of targeting, infection probability, timing distributions and duration of attack allows the modelling of ransomware behaviour as described in Section 6.3.1 and Table A.1 in Appendix A. Concerning the first process, we note that both specific location targeting and reinfection of devices is possible, even at the level of a single model iteration.

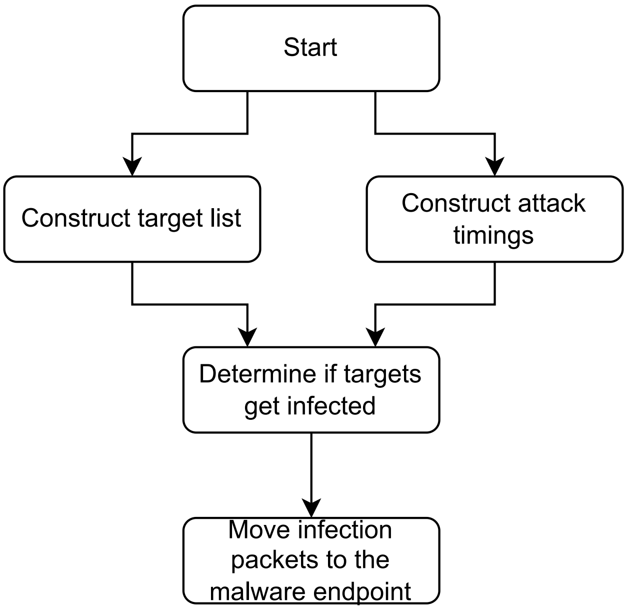

Start
Construct target list
Construct attack timings
Determine if targets get infected
Move infection packets to the malware endpoint

Figure 6.7: Ransomware Model Process Diagram [151]

6.3.3.5 The composed model

As highlighted in Chapter 4, the main abstraction that practically facilitates the abilities to compose, substitute or local reason about the model using this approach is the interface. In the case of this model, the interfaces are defined at the network endpoints, essentially allowing the flow of network packets from a device to the storage server and backwards or from the malware endpoint to a device. The network model therefore becomes the glue that sticks the server model, device and malware models together. The server model composes with the network model at an endpoint; the malware model composes with the network model at an endpoint; the device model composes with the network model at *many* endpoints. After this, a request moved into the endpoint by the device model will be sent over the network by the network model, received by the server model, and the response sent back over the network model to the device model. However, such a request would only be transmitted if a malware packet was moved into the network endpoint by the malware model, routed by it to reach the endpoint of an actual device and the infection would be successful. The operation of an iteration of the composed recovery model would be comprised of the following steps:

Firstly, the four separated models would be initialised with their specific parameter set, which will be discussed in Section 6.3.5. This would construct the organisational structure and recovery policies of the modelled organisation, define the capacity of the network, the available images for recovery, etc. At this moment, each model except the malware one has at least one process awaiting for network packets — devices awaiting a ransomware packet or a response from the storage server, the network process awaiting to route packets or the server awaiting requests — without knowing about the other models. The external processes that would bring the packets at the interface level are considered environmental processes.

Secondly, once the parametrisation of the isolated models is completed, the composition of the models can be performed if pairs of interfaces exist at each model level. For example, in the case of the storage model composing with the network one, a similar interface object must exist in both of them. In the case of the network and device model, the same number of interface objects must exist in both. The composition of models actively transforms the environmental processes described above to internal model ones: if at the previous stage, a device would await a malware packet from the environment, now it would await it from the network model, but without knowing if other models were involved in the construction of the package along the way.

Thirdly, a simulation duration should be chosen, and then the execution of the model could commence. At the level of the device model, this would start the movement in between locations and the waiting for packets. Regarding the malware module, the complete list of devices and their distribution sampled timing of injection, including possible reinfections would be computed and then added to the network to be routed. The arrival of one malicious packet at the level of a device would trigger the associated recovery option on that device. Additional delays might happen here based on the user's ability to recover on its own and the need of helpdesk staff. Once the helpdesk interaction is determined an the timings applied, the actual recovery process between the device and storage server can be performed. We note here that the arrival of a malicious packet at device level during the recovery steps does not restart the recovery process: our conceptualisation of both network and embedded enterprise recovery, as shown in Section 6.3.2.5, implies that recovery in a modern environment should be atomic. Reinfections can only occur after the recovery process has been successfully completed.

6.3.4 Methodological Observations

Having described the model architecture, construction and operation in previous sections, we now focus on describing the implications of using the chosen methodological approach in our concrete setting. Starting with our conceptualisation choice, we note that organisational modelling has proven to be a complex task even without considering recovery. This is because organisations themselves have been a difficult candidate for abstraction and, thus, multiple conflicting interpretations exist. For a comprehensive review of organisational metaphors, see [128]. For a meta-classification of organisational metaphors, see [285]. Nevertheless, we choose to think about organisations as being inherently compositional at the level of functionality, so we employ an interpretation based on different sub-components such as organisational goals, people, processes and technology [219], which construct an organisational boundary that effectively determines the organisation.

In practice, information about these sub-components, or what could be considered the formal side of the organisation, is provided by organisational structure and business processes documents, and further complemented with insights from stakeholders, expert knowledge, actual employees that take part in the modelled operations or KPIs. The knowledge co-creation process undertaken by these different parties — including modellers — in an attempt to construct a model representation that facilitates the achievement of the model goal is described in Chapter 5 and represents the updated co-design process.

Furthermore, it is important to note that our model is a partially generic prototype. Using the classification by Weisberg [304], we can view our model as 'modelling a generalised target'. The distinction is significant, since we are not producing a model for a precise client or company from an actual organisational structure, but rather attempt to derive a subset of elements relevant to an organisation from a recovery perspective and then construct a semi-generic prototype of model that can be parametrised on a case by case basis. In Weisberg's terms, the target of our model is the subset of features relevant to recovery and common to all organisational instances at a certain level of abstraction, but of course limited by the project scope. Two criteria, necessarily but not sufficiently enough to ensure correctness must be satisfied in this set-up: firstly, to ensure that 'the relevant set of specific targets actually share the relevant features, such that an intersection of their sets of features is an informative generalised target' [304], and secondly, that 'a model can be constructed at the appropriate level of abstraction so that just those features can be modelled' [304]. A longer discussion regarding the meaning of simulation correctness is outside the scope of this paper.

At an abstract level, the first criterion is tackled by the co-design process and the second one by the distributed systems metaphor and the theoretical considerations behind it. On one hand, the co-design process ensures that the interaction of modellers, stakeholders and other experts produces a more suitable model representation for the model goal, precisely because the information about what constitutes relevant targets and features is not fully known by the modeller on its own and, multiple iterations of the design and construction cycle lead to a co-constructed ontology and representation of phenomena. On the other hand, the distributed systems metaphor can be used to construct models at essentially any level of abstraction that is decomposable into its basic notions of process, resource, location, interface and environment. In addition to that, the phenomena that can introduce uncertainty or do not have a generally accepted scientific knowledge base yet constructed are treated stochastically, as explained in Section 4.1.3.2.

In practice, we have chosen a high level of abstraction, focused on a small subset of relevant features and used the distributed systems metaphor described in Chapter 4 as underlying structural element. In this case, an organisation is seen as a collection of locations where employees produce generic work over certain periods of time and can travel in between the locations. A communication network connects the locations and its characteristics such as speed, bandwidth or throttling factor have an impact on the work produced and timings for recovery. In other words, since the quality of work or its value for the company are highly subjective, we maintain the model focus on time, in the hope of using the timings to perform financial analysis when an actual concrete target is established and additional information is available. Furthermore, we justify our network based organisational conceptualisation with two arguments. Firstly, the works Burns & Stalker [57], Mintzberg [215], Crainer [80], Eccles & Crane [100], Gulati et al. [137] or Baker [18] which are extremely relevant in the field of organisational theory all argue for different forms of network-based organisations, with Baker even arguing that ‘the network form can be designed to handle product development tasks and market environments that demand flexibility and adaptability’ [18]. Secondly, given the focus on recovery and the fact that many technologies that are recovery-related are network-based, direct modelling at the network level was a natural choice because it offered the ability to translate from model consequences to domain consequences without having to unpack additional layers of conceptual complexity.

Moreover, the advantages of the methodology extend beyond the conceptualisation of the system. The compositional approach to the design of these models facilitates the ability to reason locally about the underlying components, providing two primary advantages: modularity and the ability to focus the analysis on a singular model component without the need to reason about its relationships with other components. The modularity aspect complements the generality of the distributed systems metaphor and translates to scale-free modelling at any level of abstraction or representation.

To see how our approach to compositionality and local reasoning can be applied to such a setting, let’s consider — following [62] — a stripped down, somewhat abstracted, version of the composite organisational recovery model. Here, for simplicity, we assume that composed models — Server–Network and Network–Device — have interfaces that are identical; that is, in terms of our definition in Section 4.1, this amounts to the interfaces from each of the models that are used

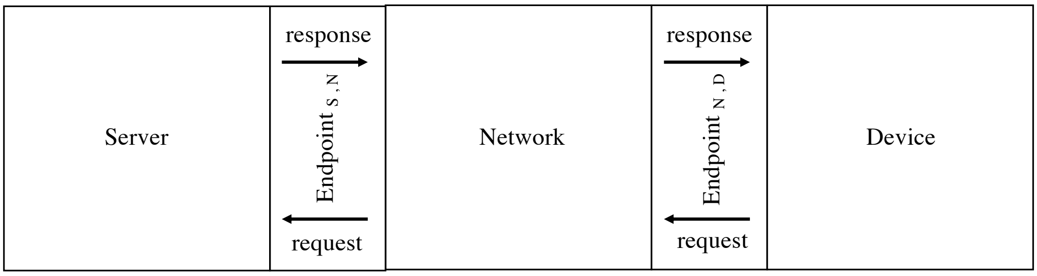


Figure 6.8: A Simplified Recovery Model [62]

in a composition being identical in each model. The simplified composite model is depicted in Figure 6.8 .

Now consider the composition of the device model and the network model. A device may request an image from the server by sending a request from the endpoint interface for transmission over the network to the server. The server's response, including the image, is transmitted over the network and received at the Endpoint interface, which now holds the image for receipt by the device:

$$\text{Endpoint}_{N,D} \overset{\textit{response}}{\longrightarrow} \text{Endpoint}'_{N,D}$$

The availability of the image that is appropriate for the device can be expressed by a logical assertion such as

$$\text{Endpoint}'_{N,D} \models \text{Image}_X \wedge \text{Device}_X$$

where X denotes the required OS, so that Image_X denotes a proposition asserting that an X image is available and Device_X denotes that the device requires the X image.

Note that the separation condition, as defined above,

$$\textit{response}\,\#\,\text{Device}_X \backslash \text{Endpoint}_{N,D}$$

holds. Consequently, applying the frame rule, we can substitute a different device model, Device'_X, provided

$$\text{Endpoint}'_{N,D'} \models \text{Image}_X \wedge \text{Device}'_X$$

can be verified.

This modularity of reasoning brings benefits similar to agile software design methods, such as reducing development time and increase focus, since both the stakeholders and modeller have a common way of understanding how the model evolves and can offer feedback or directions. Using the interpretation by Galison [126, 125, 124], the distributed systems metaphor can be considered to act as a 'pidgin language' between stakeholders and modellers, and improves the quality of the co-created knowledge. Additionally, given the security context, the ability to reason locally about sub-models at formal level increases the level of assurance the model provides while at the same time reducing reasoning time.

6.3.4.1 Co-design

As it can be easily observed, the ransomware recovery model is significantly more complex than both the physical data-loss, and the surge capacity ones in terms of both structure and co-design: the represented entities are more varied, and multiple stakeholders have been involved in the process. Because of that, we will use this model for exemplifying areas of the co-design cycle such as model consequence derivation, domain consequences translation, and a translation zone iteration that altered the candidate model.

The construction of the organisational ransomware recovery model was the result of a collaborative research project between the authors of this paper and HP Security Lab Bristol which remains as of now yet unpublished, but available in [151]. Specifically, here, we will focus on illustrating a translation zone iteration that altered the candidate model, and a derived model consequence, its interpretation in the domain, and the validation procedure which led to the discovery of an undesirable, possible real-world situation.

The motivation behind it was twofold. Firstly, to better understand how different recovery techniques can be allocated at an organisational level, in an attempt to reduce the impact of ransomware attacks of varying severity. Secondly, to explore the suitability of the distributed metaphor and co-design approach in a technical, real-world security organisational setting.

The modelling team included the following participants: a senior security research manager with a particular interest in ransomware who acted as main stakeholder and had extensive knowledge in both the security research and organisational areas, three modellers, each focused on formal, conceptual and simulation models, a security architect who was involved in the design and construction of the recovery technologies represented in the model and a help desk specialist. The interaction between members was realised through a series of meetings with different configurations. For instance, the security research manager and the three modellers met regularly — on average, once a week — and were involved in all aspects of the co-design cycle, including explicit analysis of the software code. We note here that during these meetings, the security research manager learned and started actively using the distributed systems conceptualisation. However, the security research manager also conducted separate meetings with the security architect, and help-desk specialist in the interest of data collection, and better understanding of the recovery technologies — some of which were constructed by the architect's development team — help-desk employee behaviour and timings. The direct participation of the security architect and help-desk specialist in meetings involving all the participants was not possible due to time constraints and other organisational commitments. However, in later stages of the model development, one of the modellers was deployed in the organisational setting for six months, and interacted with them directly — with the research manager, on a daily basis, and with the others, on a case-by-case basis — for the purpose of validation and possible identification of phenomena not accounted for.

That being said, we now turn our attention towards an example of translation zone iteration that led to an update at the level of the candidate model construction. To understand that, we must first note the fact that the device model presented in the above subsection did not always have this configuration. Originally, a much higher focus has been placed on modelling the internal hardware components of devices, in an attempt to construct a representation that would explicitly provide more details about the underlying recovery methods used. From the perspective of the translation zone, we can argue that the first iteration led to a conceptual candidate model representation in which each device was seen as a separate model, with distinct memory areas as locations, various software operations such as initialising the BIOS or verifying a security certificate as processes, and different software resources such as encryption keys, log files and so on. However, an analysis of this representation has shown that this high level of detail would not produce significant behaviour for the overall organisational recovery target, because an extended set of timing measurements for this highly specialised hardware process were unavailable, and very hard to measure — some of the operations were taking place before the operation system started. This was accounted for in the scope alteration phase of the translation zone, and the new model scope was updated to reflect the new conceptualisation for the device model, as presented above. In turn, this resulted in a re-implementation of the device model in the next cycle iteration.

The resulting simulation model was executed over 9000 different parameter configurations, totalling an amount of 450000 iterations — we describe this in more detail in Section 6.3.5. In this context, we describe one of the model consequences resulting from the model execution: under the exponential category of attacks with a high infection probability, allocating two admins to help-desks in large offices produced better recovery results than allocating three admins — Figure C.13. This seemed contradictory, given the fact that admins are directly involved in the recovery procedures, and waiting for an admin to become available can be one of the reasons for high recovery timings. Therefore, the verification process at the level of the model produced a contradiction, but the analysis of input data and code did not reveal any issues. During domain consequences translation, this was also discussed by the modellers and research manager, then validated with the help of the security architect and help-desk specialist at the level of the domain. Direct analysis of the results showed something interesting: in the case of three admins being present in a large office, there were moments when two of them were being deployed to smaller locations to help with recovery and only a single one remained in the initial location. However, when only two were present, the workload was high enough so they were not able to move. Discussions with the help-desk specialist revealed the absence of a policy for this specific case, which in turn indicated that the situation was possible in practice, yet unaccounted for. Therefore, the analysis of outputs produced by a model constructed using the distributed systems metaphor and co-design approach — namely the organisational recovery model — was used to determine a real possible area of improvement which might be relevant for any organisation that lacks a detailed admin deployment policy under ransomware attacks.

6.3.5 The Experimental Space

After focusing on the explicit architecture and model building steps in the previous section, it is now the time to explore the experimental space that the model constructs and inherently operates in. Firstly, we describe the overall parameter space by taking into account the range of possible parameter values, their meaning and relevance with respect to the modelled phenomena and, practical considerations related to the execution of the model over the parameter space. Secondly, we explain how meaningful organisational scenarios based on specific parameter choices can be constructed, exemplify a few such scenarios and justify those choices. Thirdly, we describe the validation procedures employed. Lastly, we present the results obtained from the execution of the model and discuss how real world organisation can employ them to support security decision-making.

6.3.5.1 Parameters

As previously explained in Section 4.1.3, the presented organisational recovery model contains significant stochastic aspects and can be considered as behaving closely to a Monte Carlo simulation. Given the nature of the recovery problem and organisational environment, it is no surprise that such a simulation requires a relatively large number of parameters to describe and conceptualise recovery at a reasonable level of detail,.

Tables B.3 and B.4, in Appendix B, contain information regarding parameter names, types, values and meaning. Building on the stochastic aspects, the model employs two different types of parameters, fixed and variable. The fixed ones do not vary across simulation iterations and have a singular value related to the target organisational posture, whereas the variable ones do and, can either be represented as ranges of values or sets. Nevertheless, we firstly focus on the variable ones:

- *device_scenario* — represents the recovery technique that devices will attempt to use in the case of being hit by ransomware. As already discussed in Section 6.3.3, four types of recovery are available: USB recovery, full network recovery, embedded recovery and a combination of 30% embedded recovery at the level of laptops and the rest network based.

- *attack_scenario* — describes the part of ransomware behaviour related to how the timings for the ransomware packages are being calculated. In more detail, five different probability distributions can be chosen from, for the timings sampling procedure: an uniform, exponential, F, uniform combined with exponential or uniform combined with the F distribution. By sampling from these distribution, we obtain the actual model time of a possible device infection. The shapes of these distributions are relevant in this context: the uniform distribution is used to represent attacks with a relatively stable infection rate — for example phishing attacks distributed by email at a slow rate to maintain a lower chance of detection —, the exponential and F distributions are used to describe the behaviour of fast spreading ransomware, with the main difference between them being that the F distribution increases slightly slower at the start and then decreases smoother — showing a more persistent attack that is harder to remove from systems — and the two combined distributions being used to showcase attacks that start in an uniform manner until they reach a certain infection threshold and then expand faster — such as email spam combined with internal spread capabilities. These behaviour examples are consistent with those described in Section 6.3.1. Furthermore, we note that the attack construction processes described at the level of the malware model in Section 6.3.3 are inherently compositional and additional ransomware behaviour can be introduced in the model by composing already implemented behaviour both at the level of a single malware model with multiple processes, or by composing multiple malware models at the level of the network.

- *infection_probability* — is being used to decide if a device targeted by a ransomware package actually gets infected or not. The value of this parameter can imply the simulated organisation has a strong or weak security posture at the level of employee training or deployed countermeasures, or that the ransomware infection mechanism is rather novel or well-known. To explore the behaviour of the recovery techniques under attacks with varying success rates, we have chosen the following infection probabilities: 10%, 30%, 50%, 70% and 95%.

- *attack_duration* — represents the period of time that the ransomware attack takes to infect the targeted devices. We are using values of two, four and eight hours to describe attacks of different intensity. The number of attack packets used in a two and an eight hours attack is the same — given by the nr_of_samples parameter –, but distributing them during a smaller time period is bound to have a higher impact on the network.

- *admins_nr* — is used to determine the number of administrative staff deployed in a single help-desk. A help-desk is placed at the level of each big office location. Varying this parameter can imply different organisational policies about the number of available staff that can help users perform the recovery procedures if they are lacking the adequate skills to do so.

- *admins_need* — describes probability of a users' need of help to perform the recovery operations. We use different values in the range of [0.0, 1.0] in an attempt to illustrate varying skill levels at the level of employees. In case a user needs help, a time delay on the overall recovery process is introduced.

This time delay varies with the type of help the employee needs: physical or remote. In the physical case, if the device is located in an office that has a help-desk, an admin resource can be claimed if available and the time delay specified by the parameter physical_admin_time is applied. If the device does not have access to a help-desk in its current location, an additional time delay specified by the admin_movement_time parameter is also applied to account for the moving time of the admin. In the case of remote help being required, a single time delay is applied, specified by the admin_remote_time parameter.

However, if the variable parameters are being used to describe variations in the behaviour of ransomware, recovery techniques, organisational policies and staff training levels, then the fixed ones contain information about more stable aspects such as: the organisational structure, size of offices, number of devices, targets, movement patterns of devices, scaling factors for the time distributions, network speed, etc.

- *num_iterations & proc_num* — these two parameters are related to the parallel execution of the model. *num_iterations* determines how many times a set of parameters should be executed and *proc_num* shows how many different processes should be used. For example, in a case of forty iterations and four processes, each process will be used to execute ten iterations of the given parameter set.

- *nr_of_samples* — describes the actual number of ransomware packets used in the attack. In our simulation, we have opted for an attack size of three hundred packets, which is a value high enough so that all the devices have a chance to be hit and even reinfections occurring, but small enough so that the attack still resembles a ransomware infection and not a full scale, denial of service flood, given the size of the organisation.

- *attack_targets* — contains the list of locations to be targeted by the ransomware packets. In the scenarios to be further presented, we have chosen to target all the possible locations where devices can be placed in.

- *physical_admin_time & admin_movement_time & admin_remote_time* — represent the additional time delays to be used in case an employee requires help with the recovery procedures. The values are being added over the actual timing of recovery, so if a user requires five minutes of additional help remotely and the actual recovery time for that device is 60 minutes, the overall recovery duration will be 65 minutes.

- *os_images* — contains the list of operating systems image resources available on the storage server. In the case of USB recovery, a windows10iso image is being used of 5GB. In the others, the devices first request a recovery_agent of size 350Mb and then an actual windows10wim file of size 4.7 GB.

- *max_office_devices & max_home_devices & max_coffee_devices & max_travel_devices* — represent the maximum number of devices that can be present in a type of location at any given time. For example, for a *max_home_devices* value of 30, no more than 30 users can work from home at the same time.

- *scale_uni & scale_dst* — are scaling factors used to ensure that sampling from the attack timing distributions do not yield results outside the desired limits given by the *attack_duration parameter*. For example, this ensures that sampling from the F distribution in the case of a two hours attack will not produce an attack timing of four hours.

- *num_office_desktops & num_office_laptops & num_small_office_desktops & num_small_office_laptops & num_travel_laptops* — represent the actual number of devices to be found in each defined location at the start of the simulation.

- *_speed* — in Table B.4, Appendix B, parameters starting from *server_speed* and ending with *travel_link_speed* are being used to set the downloadupload characteristics of the network, based on the type of network endpoint used. Naturally, wired connections will be faster than wireless ones. Parameters containing ‘link’ in their description refer to the download or upload speeds present at the level of network endpoints.

- *office1_usb_images & office2_usb_images & small_office1_usb_images & small_office2_usb_images & home_usb_images & coffee_usb_images & travel_usb_images* — represent the available images already present on USB sticks in a certain location. If a device is using USB recovery and the location where the device is present does not contain USB sticks with that image preinstalled, additional time delays for downloading the image and flashing an USB stick, if available, will be introduced in a similar fashion to the network based recovery.

- *office1_usb_blanks & office2_usb_blanks & small_office1_usb_blanks & small_office2_usb_blanks & home_usb_blanks & coffee_usb_blanks & travel_usb_blanks* — are being used to set the number of available blank USBs for each location. In our scenario, these resources are only being distributed to offices, implying a certain organisational policy.

- *_movement* — in Table B.4, Appendix B, parameters starting from travel_movement and ending with small_office_movement are being used to define the movement patterns of devices across locations. For example, in the case of office laptops, they can be placed in between zero and three hours at home — in the case of working from home for a limited amount of time — in between five and eight hours at the office and in between twenty minutes and two hours in a coffee shop. The order and precise duration of these movements is decided using Julia’s shuffle function, which produces pseudo-random permutations of a given collection.

6.3.5.2 Scenarios

As it can be clearly seen from the above, the available array of parameters allows an actual organisation willing to use our method to tailor its representation in detail and according to their own needs. This set-up allows the enterprise to compare different recovery options and parameters in a range of conditions and then combine this with other information about recovery, such as a security or financial analysis, to aid decision-making. When used in an actual real-world context, the parametrisation of the model must be performed using both KPIs and employee knowledge, according to the principles of co-design. Nevertheless, to explore the descriptive power and usefulness of our model and method, we construct a basic enterprise scenario and detail the specific parameter choices from the above that cloud be considered relevant from the perspective of an organisation.

We conceptualise our target organisation as medium-sized, having an organisational structure comprised of two large offices, two small offices and additional adjacent locations such as a coffee shop, the home or travel locations such as airports,hotels, etc. The large offices host 40 employees, the small ones 20 employees, with each being equally split in using either laptop or desktop as work devices. A main difference between employees using desktops and laptops is that the ones using desktops are bound to their start location when working, whereas the laptop using ones are mobile and can move in between offices, the coffee shop, travel locations or home. Furthermore, the locations differ at the level of network connection speeds: we assumed that larger offices have better connections than smaller ones and that corporate locations such as the offices have better connections than general purpose locations such as a coffee shop or the home. The actual data about the network speeds can be consulted in Appendix B.

In addition to the different connection speeds, locations differ at the level of the available support resources. Each large office contains a help desk which provides users with help to perform recovery procedures. Nonetheless, support is not necessarily bound to large offices, since the administrative staff can also move to other locations or provide users with advice remotely — as explained in Section 6.3.3. In addition to that, we assume that large offices have access to USB sticks with preloaded operating system images, the small offices have access to USB sticks, but without images and the other locations do not have access to USBs.

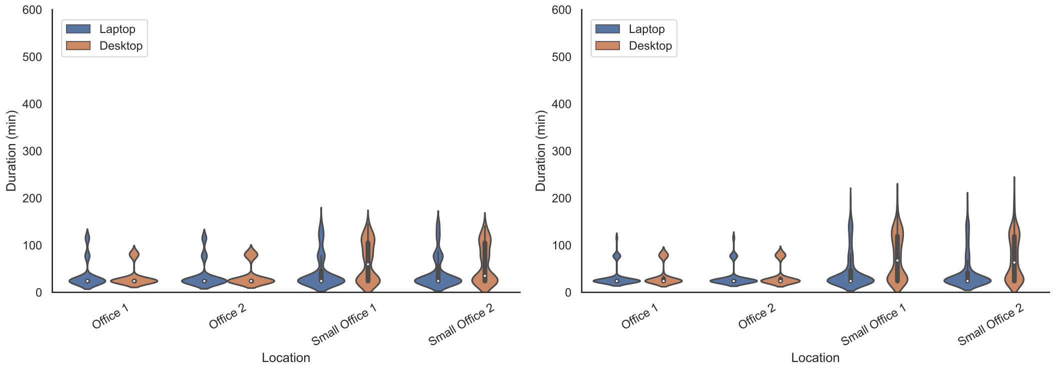
In this organisational setting, we look at four different recovery techniques: USB, network based, embedded and a mix of network based and 30% embedded —

as explained in Sections 6.3.3 & 6.3.5.1. It is worth noting that the model captures the active elements of the different approaches but does not represent the security of the different approaches. Here, we should consider the ‘Sure Recover’-based mechanisms as secure in that they check the signature of the image, thus validating that the image is correct and as intended by the enterprise — for example, with the chosen enterprise AV systems installed. USB-based recovery has no validation of the image and either requires administrators who know BIOS admin passwords to initiate the installation or requires the computer to be kept in an insecure state — allowing USB boot or boot with no BIOS password. Furthermore, we looked at five different ransomware spread behaviours, by sampling attack timings from probabilistic distributions such as uniform, exponential, F and, combinations between uniform and exponential and uniform and F. Justifications for choosing these distributions can be found in Section 6.3.5.1, when describing the *attack_scenario* parameter. In addition to the temporal aspect, we test the network under difficult conditions that can resemble a fast-spreading ransomware by varying the *infection_probability* on a range of 15% to 95%, *attack_duration* between two, four or eight hours and, by setting the size of the attack, *nr_of_samples*, at three hundred packets.

Below, we present three different recovery scenarios in an attempt to illustrate how specific parameter choices can lead to meaningful organisational scenarios. These should not be interpreted as a complete exploration of the model results — in Subsection 6.3.5.3 we look at the implications of altering each input parameter in relation to the output produced. Rather, they illustrate examples of relevant insights that the model can offer to organisational stakeholders in three specific cases. For example, the first scenario shows that USB based recovery is not to be fully discarded as a countermeasure against uniform attacks if certain policies regarding USB vetting are included. The second scenario illustrates the poor performance of network-based recovery under a high-severity attack. The third scenario complements the second, showcasing the utility of a mixed recovery method in the face of an even higher severity attack.

Scenario 1: USB Recovery under uniform attack distribution

In the first recovery scenario we aim to determine the feasibility of using only USB recovery in the case of a temporally, uniformly distributed attack of varying severity, but a short time period of two hours. This can be interpreted as an intensive spear phishing attack that leads to ransomware being dropped on devices without an internal network spreading mechanism. Furthermore, we assume that users have the knowledge of performing an USB recovery process, so only one in thirty might require help-desk assistance, and that help-desks contain 10 administrative staff each. Therefore, the employees have a good knowledge of performing recovery and organisational policies ensures that the help-desks are well staffed. However, we must take into account that large offices have access to seven USBs pre-loaded with the right recovery image each, small offices have access to three empty USBs each and, the other locations do not have access to USBs. This implies a security policy of only using security vetted USBs, which are not available in travel locations.



(a) Probability of infection 30%

(b) Probability of infection 95%

Figure 6.9: USB Recovery under uniform attack distribution (Scenario 1) [151]

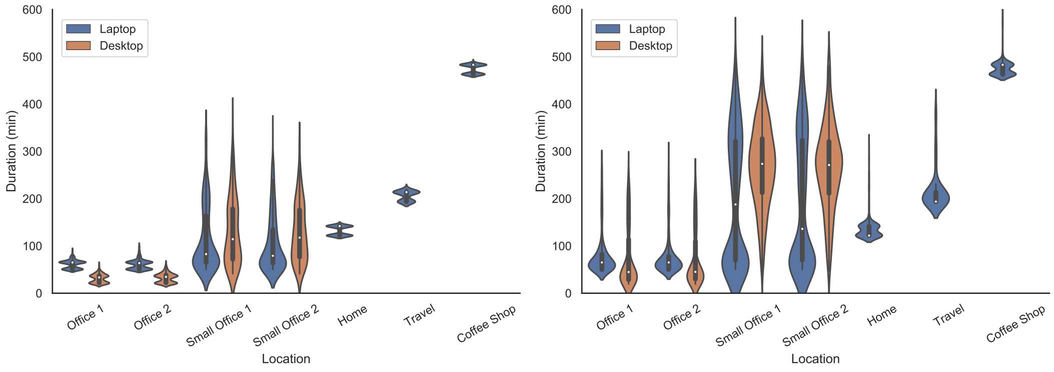
Figure 6.9 shows the average recovery duration across locations, under a probability of infection of 30% and 95% respectively. These violin plots show the summary statistics and distributions of timings for devices across all the simulation runs [191]. This variation in infection rate represents the level of preparedness the organisation possesses in regard to phishing, both at the level of employee training and active countermeasures such as spam filters. As it can easily be observed, the large offices have a similar performance, regardless of the infection probability, with the highest values for desktops and laptops, close to 100 and 140 minutes respectively. This is due to two reasons: the uniform attack is unbiased at the level of targeting, so the ransomware packets do not flood a singular location and, the pre-loaded USBs do not require interaction with the network. Furthermore, we must note that in the 30% probability of infection case, an average of 58 devices recovered, whereas in the 90% case, there were 129.

However, in the case of small offices, the situation changes. For a 30% infection probability, both laptops and desktops have similar recovery timings of 186 and 176 minutes on the extreme end. Therefore, although they are using blank USBs which require network downloads, the severity of the attack is not enough to produce a greater delay. Nevertheless, the difference in number of devices, need of network interaction and, to a smaller extent even the need of physical help-desk assistance lead to a 33% recovery time increase in the case of laptops and a 76% increase in the case of desktops, on the extreme case. For the 95% infection probability, the recovery timings for laptops reach 225 minutes, whereas desktops go up to as much as 251 minutes. Interestingly, there is a small difference in between the performances of the two small offices that we attribute to the movement of laptops in between locations.

Therefore, this first scenario reveals some relevant facts for organisations that wish to employ USB based recovery. Firstly, even with a well equipped help-desk and users that know how to flash an USB, pre-loading the USBs with recovery images leads to an almost twice as faster recovery rate in the case of a 95% successful spear phishing attack without automatic network spread. Secondly, under such conditions, a number of USBs representing almost a sixth of the number of devices (5.714) in the case of big offices is enough to prevent a steep increase in the duration of recovery. Thirdly, alterations to the security policy about the vetting of USBs must be made, because under the current one, all the devices in external locations — non-offices — failed the recovery process.

Scenario 2: Network Recovery under F attack distribution

This recovery scenario depicts an organisation with a weaker security posture than the above scenario, both at the level of user training — one in ten employees requiring admin assistance to perform device recovery — and at the level of help-desk staffing — each help-desk having 5 staff members. This organisation is being targeted by a more virulent ransomware strain, modelled using the F distribution, but over a longer duration of four hours. This resembles the behaviour of a fast spreading ransomware, as presented in Section 6.3.1, particularly because when initialised with the values 50 and 8 as degrees of freedom, the probability density function of the F distribution increases steeply at the start and then decreases more gradually — in [164, 216, 306], the authors describe propagation graphs for worm-like spreading malware using similar types of distributions.



(a) Probability of infection 30% **(b)** Probability of infection 95%

Figure 6.10: Network Recovery under F attack distribution (Scenario 2) [151]

Figure 6.10 shows the average recovery duration across locations, again, under a probability of infection of 30% and 95% respectively. Compared with the previous scenario, we can observe differences in recovery time at the levels of desktops and laptops in both small and large offices. In the case of the 30% infection probability in large offices, the laptops recovery time varies between 50 and 98 minutes — with a median of 51 minutes —, whereas the desktops recover between 21 and 48 minutes — with a median of 23 minutes. In the case of 95% infection probability, laptops recover between 48 and 309 minutes — and a median of 71 minutes — and desktops between 23 and 300 minutes — with a median of 41 minutes. Furthermore, we must note that in the 30% probability of infection case, an average of 61 devices recovered, whereas in the 90% case, there were 141.

At the level of small offices, the increase in recovery time is steeper. For example, in the 30% case, laptops' recovery time varies between 54 and 385 minutes — with a median of 83 minutes — and, desktops' recovery time varies between 43 and 415 minutes — and a median of 125 minutes. In the 95% case, the laptops' recovery time varies between 53 and 588 minutes — with a median of 173 minutes — and the desktops recover between 23 and 557 minutes — with a median of 275 minutes. The median recovery timings for laptops at home are 150 and 125 minutes for the 30% and 95% infection probability. For the travel location, they are 225 and 200 minutes and, for the coffee shop, 487 and 486 minutes, respectively. However, since the attack duration is only 4 hours and the simulation time is 24 hours, a very small number of devices manage to both perform movement between locations and finish the recovery process.

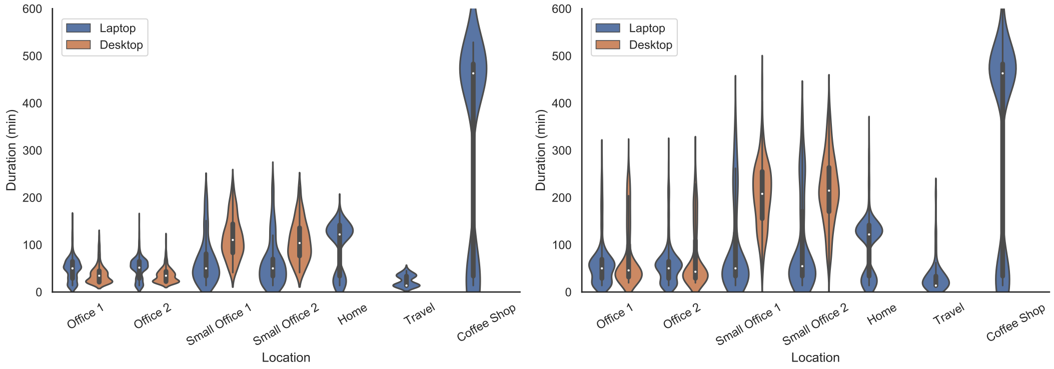
A few observations can be drawn. Firstly, the big offices are more resilient to this type of attack, given the easier access to help-desks and the better link speed. The attack size of 300 network packets spread among 4 hours does not manage to produce a severe impact at the level of big offices in the 30% infection case. The 95% case is more interesting, with the median of desktops increasing with 56%, and the one for laptops with 71%.

Secondly, the desktops perform worse than laptops in the small office locations. Although peculiar at first sight, this represents an effect of laptops being able to move between locations, combined with the 24 hours simulation time: the number of laptops in small offices decreases because of movements, and those that are targeted early in the attack manage to complete the recovery, so the overall recovery time of successful laptops decreases.

Thirdly, the extreme values are significantly higher for the small offices. For example, even though on average, in the 95% case, a desktop manages to recover in around four and a half hours — which is less than the overall attack duration and means the device can get reinfected — some could take even more than nine hours, which is the equivalent of more than an entire day of work. Because of this, organisations might seek alternative recovery processes, especially if a priority of recovery between employees exists.

Scenario 3: Mixed Recovery under Exponential attack distribution

In this scenario, we maintain the same set of parameters for the organisation — one in ten employees requiring admin assistance to perform device recovery and, each help-desk having five staff members — and the attack duration of four hours, but we change the recovery method and the attack distribution. We use an exponential distribution, which has a higher potential of increasing the network throttling because the distribution of ransomware packets is steep from the beginning, compared to the F distribution. However, the mixed recovery method used means that 30% of the laptops use embedded recovery in all the locations except the travel one. All the laptops in the travel location use embedded recovery.



(a) Probability of infection 30% **(b)** Probability of infection 95%

Figure 6.11: Mixed Recovery under Exponential attack distribution (Scenario 3) [151]

Figure 6.11 shows the average recovery duration across locations, again, under a probability of infection of 30% and 95% respectively. We must note that in the 30% probability of infection case, an average of 63 devices recovered, whereas in the 90% case, there were 150. In the case of big offices, we observe that laptops' recovery time varies between 13 and 175 minutes — with a median of 49 minutes — in the 30% case and 20 and 325 minutes — with a median of 53 minutes — in the 95% case. The desktops' recovery time varies between 15 and 135 minutes — with a median of 30 minutes — in the 30% case and, between 25 and 328 minutes

— with a median of 51 minutes — in the 95% case.

Furthermore, analysing the performance of small offices yields the following results. The laptops' recovery time varies between 13 and 265 minutes — with a median of 49 — for the 30% infection probability and 13 and 450 minutes — with a median of 54 — for the 95% infection probability. The desktops' recovery time varies between 38 and 265 minutes — with a median of 108 — for the 30% infection probability and, between 42 and 375 minutes — with a median of 210 — for the 95% infection probability.

In addition to that, we can easily observe that a significantly higher amount of laptops manage to complete the recovery process in a non-office environment. In the case of home, laptops manage to recover for both infection probability cases on average around 125 minutes. When considering the coffee shop, the median for laptop recovery is around 460 minutes for both infection probability cases. In the travel location, considering that the laptops use only embedded recovery, the median recovery time is 13 minutes.

This allows us to draw some relevant observations about the allocation of this type of recovery technique. Firstly, using embedded recovery on 30% of the laptops has reduced the medians of device recovery in both in the 30% and 95% infection probability cases, regardless of device type, when compared to the previous scenario. Particularly in the case of small offices under 95% infection probability, the median of recovery time has decreased from 173 to 54 minutes in the case of laptops — a 68.78% decrease — and from 275 to 210 minutes in the case of desktops — a 23.63% decrease. This reveals an interesting insight for organisations: deploying a faster recovery method across a subset of all the devices in an organisation can have a positive impact on all the devices if the organisation was mainly relying on network-based recovery previously. Of course, this is not a 'silver-bullet' type of solution, because the extreme values for laptops and desktops are still 450 and 375 minutes in small offices, but the reduction at the level of medians shows that this type of approach is viable. Secondly, recovery in remote locations becomes more feasible, since more employees now have the chance to complete recovery before moving to another location — the coffee shop and home locations in Fig 6.11 show this clearly when compared with the same locations in Fig 6.10, because although the medians are similar, the number of points differs greatly. At the level of the travel location, even the medians differ because embedded recovery is used on all the laptops.

6.3.5.3 Overall Behaviour and Model Sensitivity

The previous subsections were focused on providing detailed information about the model parameters and, exemplifying the model utility by showing three scenarios that could be useful for organisational decision-making with respect to security; we now focus on more general characteristics of the model.

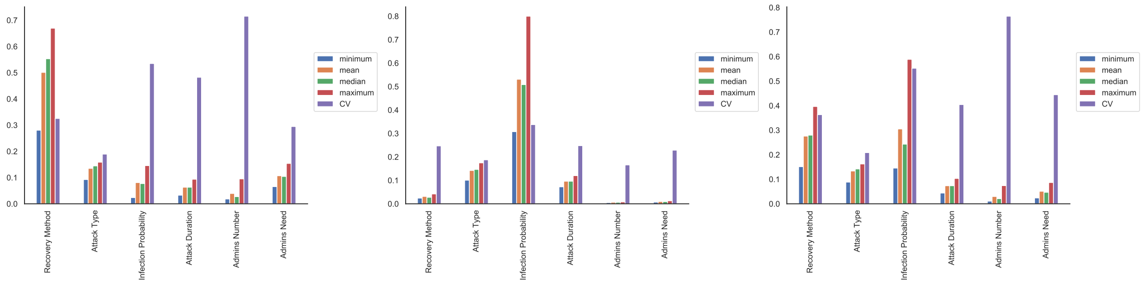
Firstly, 9000 configurations of variable parameters — as explained in Section 6.3.5.1 and shown in Table B.3 — have been used for the model execution. Each configuration has been run 50 times, so in total, the model execution is comprised of 450000 different runs. Each run employed both fixed and variable parameters as detailed in Section 6.3.5.1 and, elements such as when a ransomware packet is being sent on the network, to what target or to which location a device might move are being sampled for stochastic variables that represent the environment in which the model operates. The execution of the model was carried out in a distributed computing environment, across multiple cluster nodes with an 8GB memory limit. The size of the produced model output files was 4.42GB.

6.3.5.4 PAWN Sensitivity Analysis

Given the complexity of the phenomena under study and the stochastic nature of the modelling approach, we employ sensitivity analysis in an attempt to better understand the relationships between input and output variables and increase confidence in the overall model behaviour. We have chosen to use the PAWN sensitivity analysis method as described by Pianosi & Wagener [239, 240] and further developed by Baroni & Francke [25] for a series of reasons: allowing both categorical and numerical data to be used as input, offering information about the relevance of input variables with respect to output variables and at the same time describing the variability that changes in the input variables produces at the level of output, good performance with respect to multi-modal data — especially because of the ransomware behaviour mechanism, but also given the composition of multiple distributions.

In very brief terms, PAWN sensitivity analysis focuses on two metrics: the PAWN sensitivity index which describes the relevance of an input variable with respect to an output variable and, the coefficient of variation which is a measure of general variability produced during the variation of the input parameters. The key aspect of the method is the usage of cumulative distribution functions for output quantification instead of probability density functions or other traditional variance-based methods. The PAWN sensitivity index of an input variable with respect to an output variable is computed as the statistical difference between the unconditional CDF of the output variable — obtained by varying the input factors at the same time — and the conditional CDF of all the other input variables except the one under study. This measure is known as the Kolmogorov-Smirnov statistic [173, 274], a well-established method for calculating the distance between multiple cumulative distribution functions. The coefficient of variation is calculated as the division of standard deviation by mean, over the previously computed indexes. In layman's terms, a high sensitivity index implies that the specific input variable has a high impact on the value of the output variable, whereas a high value for the coefficient of variation translates to the input variable producing high amounts of variation on the output, but not necessarily producing severe output changes.

We employ the above described method to analyse the relationships between the six variable input parameters described in Section 6.3.5.1 and three output variables: the average recovery duration, total recovery duration and the average number of devices recovering. The results are shown in Figure 6.12 below, and Figure C.1, Figure C.2, Figure C.3, Table C.1, Table C.2 and Table C.3 in Appendix C. The reason for choosing this specific output variable configuration is twofold: firstly, these three output variables represent relevant metrics for guiding organisational recovery on their own and, secondly, the behaviour shown with respect to the total recovery duration acts as a useful verification checkpoint for both the average recovery duration and the average number of devices recovering.



(a) Average recovery duration. **(b)** Average number of devices. **(c)** Total recovery duration.

Figure 6.12: Sensitivity Analysis [151]

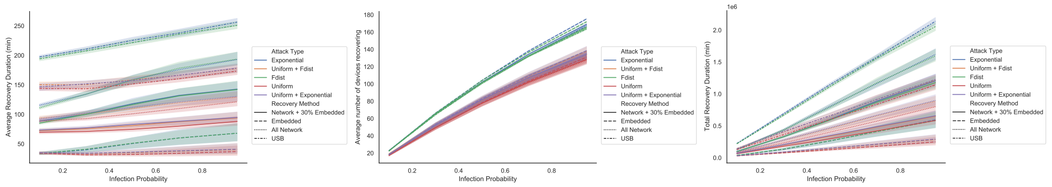
Starting with Figure C.1 and Table C.1, we can observe that the recovery method, type of attack and users' need for admins have the biggest impact on the average device duration of recovery. The extremely influential, maximum sensitivity index value of 0.669556 for the recovery method can be attributed to the fully embedded recovery mechanism and, reveals the fact that deployment of such a mechanism at the level of all devices in an organisation can drastically reduce the impact of other factors on the average recovery time. With respect to variation, the highest CV scores were obtained for the number of admins, infection probability and attack duration. However, this shows a complete separation between the high-impact-producing input variables and the ones producing low-scale variations, which has led us to further analyse the total recovery duration.

In Figure C.2 and Table C.2, we observe a different variable allocation with respect to the average number of devices recovering. The infection probability, attack type and attack duration are the most influential and, the infection probability, attack duration and recovery method produce the highest amount of variation. Although one might expect a higher sensitivity index value for the recovery method, this is not the case since the individual simulation time of five days allows all infected devices to recover, in the end. Thus, the impacts of the recovery method, number of admins and users' need for admins are drastically reduced.

Now, since the total recovery duration time is dependent on both the recovery time of singular devices and on the actual number of devices that manage to completely perform the recovery procedure, we would expect that the sensitivity analysis for the total recovery duration time to maintain the trends from the analysis of average recovery duration and the total number of devices. Figure C.3 and Table C.3 reveal the following: the most impactful variables were the recovery method, infection probability and attack type and, the ones producing the most variation were the number of admins, infection probability and the users' need for admins. As it can easily be observed, the recovery method was the most influential input variable in C.1, the infection probability in C.2 and the attack type was the second most influential in both cases. The difference between the index values of the recovery method and infection probability was only 0.036667, so their impact can be regarded as similar. With respect to variation, the number of admins and infection probability maintained their leading trends from Figure C.1 and C.2. Therefore, our expectation was confirmed, since the sensitivity analysis of the total recovery duration did conserve the primary impact insights from above.

Although insightful about the general behaviour of the model, the sensitivity analysis above does not reveal enough information about the types of relationships between the variables, because it simply does not take into account their structure, but only the input/output changes. Consequently, we proceed to analyse Figure C.4 to Figure C.15 in an attempt to better understand how the structure of the input variables affects the value of the output ones.

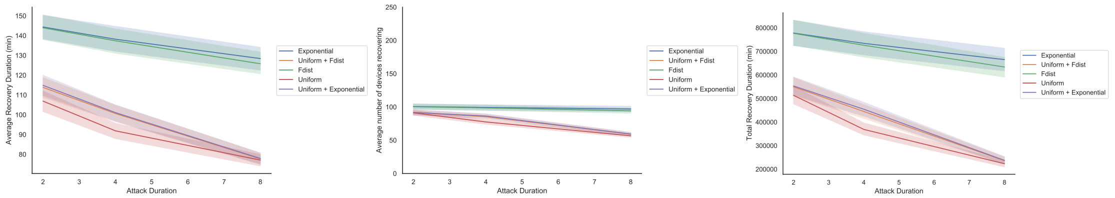
- **Infection Probability:** Figure 6.13, and C.4, C.5, C.6 in Appendix C are focused on the impacts of the infection probability, which was deemed the second most influential variable. A separation of classes of attacks based on the type of distribution used is immediately visible, but most clearly in Figure C.5. A similar observation can be done about the recovery techniques. Given the stochastic aspects of the model, the produced output is surprisingly structured: for example, USB-only recovery under exponential types of attacks has the worst performance across all the diagrams and the full embedded recovery has the best performance regardless of the attack type. However, some additional insights can be gathered from the intermediary classes: in Figure C.4, we can observe that at around 0.5 infection probability, the performance of USB-only recovery under uniform and uniform-exponential attack classes becomes better than the performance of network-only recovery under exponential types of attacks. Since in the real world, uniform-exponential attacks happen more often than purely exponential ones — the uniform part of the attack can be viewed as reconnaissance actions or an initial phishing campaign —, we can argue that for organisations with a low-security posture, USB recovery could be a viable option, especially if the only other one is purely network based. Furthermore, Figure C.5 shows the benefits of a composite allocation of recovery techniques — 30% embedded recovery and 70% network-based: the distance between a full network approach and a composite one is clearly visible at the level of uniform attacks, but the separation zone between the distributions increases even more at the level of exponential attacks. In other words, the more virulent an attack is and the lower the security posture, the better will our composite approach behave when compared to a fully network-based one.



(a) Average recovery duration. **(b)** Average number of devices. **(c)** Total recovery duration.

Figure 6.13: Impact of Infection Probability on the total, average duration and the number of devices recovering. [151]

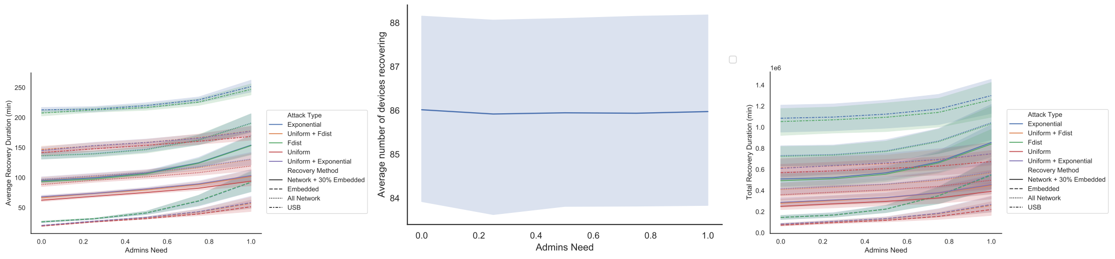
- **Attack Duration:** Figure 6.14, and C.7, C.8 and C.9 in Appendix C are concerned with the attack duration. As expected, since the total amount of network packets allocated to an attack is fixed, the more the attack takes, the less network throttling manifests, so the relationship is inversely proportional. Furthermore, because the individual simulation time is five days, almost all the devices manage to recover regardless of the attack type, which is exactly what can be observed in Figure C.9. Interestingly, in such a case, more devices manage to recover in the case of exponential attacks because they get infected earlier and the throttling spread over five days is not influential enough, whereas, under an uniform attack, the last devices to get infected might not manage to complete recovery.



(a) Average recovery duration. (b) Average number of devices. (c) Total recovery duration.

Figure 6.14: Impact of Attack Duration on the total, average duration and the number of devices recovering. [151]

- **Need for Admins:** Figures 6.15, C.10, C.11 and C.12 focus on the users' need for admin help while performing the recovery procedures. Particularly in figures C.10 and C.11, we can observe that the attack type and recovery technique generate similar separation classes to Figures C.4 and C.5. This confirms the insights of the sensitivity analysis, which described the infection probability as a more influential input variable than the need for admins, especially since in Figure C.11, the highest values of the need parameter lead to lower values of recovery time than in the case of infection probability. Similarly to Figure C.9, Figure C.12 depicts the low influence the input variable has on the total number of devices managing to completely recover, which is an effect of the five-day simulation time.

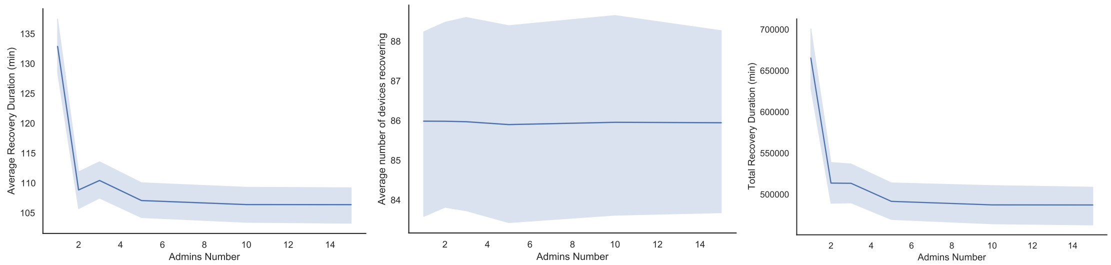


(a) Average recovery duration. (b) Average number of devices. (c) Total recovery duration.

Figure 6.15: Impact of Admins Need on the total, average duration and the number of devices recovering. [151]

- **Number of Admins:** Last but not least, Figures 6.16, C.13, C.14 and C.15

target the number of admins allocated to each help-desk location. As we can easily observe in Figures C.13 and C.14, given the model configuration, an optimal value for this parameter can be found in the interval [2, 5]. The small recovery duration increase that can be observed in Figure C.13 around the value of three for the number of admins is a consequence of admin behaviour: if three admins are allocated to a help desk in a large office and then some other small offices are hit by ransomware, up to two admins could get moved there to assist with recovery procedures. This can lead to a situation where more admins end up being present in small offices rather than large ones, which can in turn increase the waiting time for users who require physical admin assistance. Furthermore, this particular result suggests a possible need for improvement in admin workflow policies: if the model under analysis would have been built for an actual organisation, this type of anomaly would reveal an extreme case where the admin policies would not work as previously expected.



(a) Average recovery duration. **(b)** Average number of devices. **(c)** Total recovery duration.

Figure 6.16: Impact of Admin Number on the total, average duration and the number of devices recovering. [151]

6.3.5.5 Verification & Validation

Given our intention of showing that the above-described model represents an instantiation of a modelling methodology able to produce useful models for supporting security decision-making, we must now ensure that the model's conceptual reality is representative of the depicted scenarios. In the following lines, we describe a series of verification and validation procedures employed to do so.

- **Model conceptualisation** — As a particularly relevant aspect of the Co-design cycle described in Chapter 5, we ensured a consensus was reached between a literature-only representation and expert knowledge. The ransomware, network, organisation and recovery methods behaviour were constructed based on an extensive literature review for ransomware — which in turn led to a selection of features that included a targeting distribution, a timings distribution, infection probability and attack duration —, a well-known conceptual representation in the organisational literature for the organisation — the network organisation —, a simplified version of actual internet routing for the network and technical documentation for the recovery methods. Furthermore, additional improvements were brought in by expert knowledge, until a state of considered model utility was reached. As described in the Co-design methodology, this conceptualisation was not fixed in an initial design phase but continued changing throughout development.

- **Antibugging** — Workflow checks were introduced at the level of the code to check whether quantitative or structural constraints were violated during execution, as a part of the verification process shown at the level of the extended co-design cycle. Examples could include the maximum number of users in a type of office, the users' movement patterns or the location they end up in, the minimum and maximum network speeds available or if ransomware packets are being transmitted by the ransomware model after the given attack duration.

- **Prototyping & Walk-through iterations** — The model construction process started with basic, compositional models for the devices, network, storage server and ransomware that were further developed and continuously validated by expert knowledge. To ensure this was possible, multiple walk-through iterations were carried out for both separated and composed models. Furthermore, some runs made use of placeholders for an easier structural understanding, whereas others included stochastic input for the assessment of behaviour.

- **Real-World data** — The recovery timings for the USB recovery method and the network speed constraints were manually gathered and then validated by expert knowledge — particularly the office distribution changes — and comparison with UK network speed statistics.

- **Event tracing** — Based on the nature of the model, we were able to use a dual system of event tracing at the level of locations and timings. This helped with debugging once unexpected behaviour was manifested, but at the same time simplified analysing two different instances of the same model, in different stages of development. For example, a previous iteration of our recovery model included only an uniform attack spread of a constant rate. The current one allows for much greater customisation by allowing different input distributions for spread and timings. Nevertheless, setting the right parameters for the current iteration with respect to rate and speed leads to similar behaviour to the previous iteration, and therefore conserving consistency.

- **Sensitivity analysis** — Sensitivity analysis was, again, used with a dual purpose. First of all, it confirmed that the choice of input parameters had an influence on the outputs — which was expected, but useful. Secondly, the general-level insights produced by it were used as verification checks for the examples and the structural analysis of the features. For example, the sensitivity analysis in Figure C.3 reveals the infection probability as more influential to the total recovery duration than the attack duration. Figures C.5 and C.8 confirm it at the level of the Y-axis scale.

Nevertheless, we end this sub-section on validation with the idea that especially in an organisational context, the verification and validation procedures must be the result of an alignment between modellers, experts and stakeholders. For sure, traditional approaches such as antibugging and structured walk-throughs are still useful for ensuring the agreed-upon model conceptualisation is implemented and the actual model iteration is understood. However, we must take into account that the organisational modelling goal is not only to produce an accurate representation of phenomena — accurate to what criteria could be a legitimate question — but an accurate representation useful for a practical outcome. This is precisely why stakeholders have an increased presence during model design and development through co-design: on one hand, they can introduce additional environment-specific insights that can act as conceptual optimisations for the model — the scope of an organisational model is smaller than the one of a purely scientific one — and on the other hand, they gain an increased understanding and belief in the usefulness of the model that can lead to more real-world outcomes.

6.3.6 The Model as a Management Tool

As stated at the end of the previous paragraph, one of the innate goals of an organisational model is to produce some form of real-world outcomes either directly — if the model is being used without human intervention — or indirectly, by supporting decision making. Our focus is on the latter. To support such decision-making, the model must be parametrised and validated accordingly. However, since the presented model acts as a methodological example, we assume with the help of expert knowledge that the modelled organisation acts as a reasonable surrogate for a real one. Given this, we describe in the following lines, a list of possible actions that decision-making factors might implement at the organisational level, based on the model.

Firstly, precise policy changes could be enacted at the level of help desks. For example, the policy regarding admin movement between offices could be altered such that no more than half the initial number of admins in one help desk is allowed to be out of the initial location at once. Furthermore, the number of admins in a help desk could be capped at a value in the interval [2, 5]: the model reveals this interval as producing the highest rate of reduction in total recovery time. A fixed value can then be decided upon based on additional hiring or economic policies that the organisation might have interacting with its security posture.

Secondly, organisations can use the model to test different configurations of recovery techniques and then analyse their performance and benefits. For example, we have observed that the recovery techniques employed form classes of recovery behaviour that are reducible: on average, the organisation displayed a similar recovery duration when under an uniform rate attack but using USB based recovery and, when under an exponential rate attack but using only network recovery. Based on historical data and risk appetite, organisations might consider one class of attacks as a priority, and therefore base their recovery technology choice on that. In our execution, a combination of 30% embedded recovery and 70% network-based recovery mechanisms on laptops was shown to produce a significant decrease in recovery time across different locations, particularly when compared with the more rigid but often used, complete network approach. This led to an increase in the overall number of devices that managed to perform recovery at all, even in locations where the embedded recovery was not deployed, due to a reduction in network throttling. Therefore, organisations might choose to experiment with the exact percentage value and the locations in which different recovery technologies are deployed. Possible deployment strategies could be focused on the criticality of the employee tasks to the organisation, departmental budgets or organisational hierarchy.

Thirdly, the choice of inputs and their sensitivity with respect to the outputs can be used to highlight possible areas of further development, if their degree of control is being taken into account. In our case, although the attack type and duration were seen as having a somewhat medium impact on the recovery duration, they are not controllable by the organisation in any way. Differently, the infection probability can be controlled with improvements at the level of technical defence mechanisms, employee training or changes in policy. Furthermore, additional insights can be drawn by performing the same sensitivity analysis procedure but fixating some of the variable inputs — particularly the ones considered uncontrollable — to average values observed historically in that specific organisational context.

To summarise, the modelling is intended to aid decision-makers in exploring the consequences of potential decisions. Here we have explored a model with some specific scenarios and, for decision-making, a company would need to parametrise the model with its situation in terms of office set-up, staff skill-levels, and help-desk support levels and policies. However, even in the scenarios we have run, there are some general insights:

- Embedded recovery is valuable both for travellers with poor connectivity and in reducing recovery network loads within the office.
- Help-desk and support policies can be critical in the smooth running of mass recovery, but need tailoring according to an organisation's set-up and staff skill-levels.
- The model shows sensitivities to factors outside of the direct recovery solution, such as the probability of infection. Thus, when making recovery decisions, it is important to examine the wider enterprise systems such as AV, phishing protections, etc.

6.3.7 Reflections

With an estimated global cost of $20 billion in 2021 alone [45] that might or might not have been reduced during the Covid lockdown period, and an ever increasing list of high profile victims in both private and public sector organisations, ransomware is and will continue to be a problem in the future. That is why most organisations will have to understand and manage the risk of ransomware infection, or else face severe operational problems that could even turn to existential crisis.

However, the tools available to support decision makers in better understanding the ransomware phenomena at the level of their own organisation and guide the recovery technology selection, allocation and policy adaptation procedures, are not that many. Furthermore, most of the times, such tools exist in the form of general guidelines that do not offer enough practical understanding to decision makers which are less security inclined. Given this, we have considered organisational recovery under ransomware as suitable for explicit, simulation modelling, due to descriptive power, modularity and possibility for extension, and ability to produce quantitative measures for further analysis.

Here, we have described our structured modelling conceptualisation based on the so called 'distributed systems metaphor' and explained how it supports the model co-design methodology. We have exemplified the suitability of our approach in a complex, yet technology focused organisational environment with the implementation of the recovery model. We then used the model to generate several recovery scenarios corresponding to the behaviour manifested by ransomware strains and tested the impact of recovery techniques on organisational recovery time under such conditions. Furthermore, we have detailed the main verification and validation procedures employed, and how the model could be practically used in a decision-making context.

However, as with any model, the decisions regarding what not to model explicitly are just as relevant. Organisational resilience is a complex phenomenon: device recovery, the nature of attacks, technical security mechanisms, policy, budgets and user training and awareness and possibly even other ‘unknowns’ play important parts in overall resilience. Our models focus primarily on the performance and behaviour of device recovery mechanisms under different ransomware attacks. Although we could argue that elements of technical security mechanisms, policy, user training and awareness are present in the models at a very low level of detail — in the infection probability and need for admins parameters — a separate analysis and model would be required to reason about security in this context. For example, this could include an analysis of how images are protected; is there a mechanism for ensuring only approved images are installed and that their integrity can be validated — Section 6.3.2.4 — and where images are maintained on the device, can they be protected from malware. Such security analysis and decisions should go alongside the organisational model when considering a robust recovery strategy.

Nevertheless, the current model configuration is well-suited for further development. On the practical side, model extensions could include explicit conceptualisation and implementation of employee work — for example in the form of processes generating revenue —, data loss, technical security mechanisms and policies around data storage or increasing the level of detail for users and devices interacting with one another. From a theoretical perspective, the modelling formalism could be further extended at the level of executable supporting tools: for example, to support local reasoning about the compositional model structure or to provide dynamic model-checking capabilities. Last but not least, additional research can be done in the area of empirical case studies: to include models constructed using this approach and deployed in varied organisational environments, in an attempt for further methodological refinement.

6.4 Conclusion

In this chapter, we have attempted to illustrate how the distributed systems conceptualisation and co-design methodology were used to practically design and construct the three presented models. Here, we draw considerations by reasoning about how the concepts of the metaphor and co-design steps were employed in the models.

We start by analysing how the properties of the metaphor, as shown in Section 4.1.4 manifested at the level of the models.

In terms of *generality*, it can be claimed without a doubt that the distributed systems concepts were suitable for constructing working representations of the systems under study. Irrespective of the characteristics of the target system, whether it was abstract and technical as seen in the ransomware recovery model or more centred on human factors as observed in the trauma unit surge capacity model, the metaphor successfully encapsulated all pertinent entities and relationships according to the model goal. Furthermore, a high degree of flexibility can be observed: for instance, in the data-loss model, resources are mostly physical — laptops, cds, USBs —, whereas in the recovery one, they can be physical — devices, USBs —, abstract — recovery images, network data —, or even people — help-desk employees. Additionally, hierarchies of relationships can be constructed between concepts of the same type. In the recovery model, both recovery images and recovery agents resources are encoded in network data resources to be then routed through the network model. Similarly, in the trauma unit model, starting the various treatment processes may depend on the completion of other treatment processes beforehand. Finally, as observed in the data-loss model, the concepts themselves can be bundled to produce new abstractions, such as the one for agents, involving a resource, a process and multiple abstract locations.

With respect to *recognisability*, the interaction between modellers, stakeholders and practitioners in both the ransomware recovery and trauma unit models has shown that the elements of the distributed systems metaphor are relatively easy to grasp, especially since they are general, and versions of their definitions already exist in the scientific areas that the stakeholders and practitioners were familiar with. For example, at the level of the recovery model, the research manager did not have any issues in conceptualising the physical and abstract resources immediately, but a longer discussion was needed for help-desk employees.

Similarly, we have observed that the *identity conservation* criterion was not only satisfied, but that the constructed identity for the model components, based on the distributed systems metaphor, facilitated focused and clear discussions between modelling participants during the entire co-design process. This was showcased at the level of the recovery model, where originally devices were conceptualised as models with a complex internal structure, then simply as resources. Although the original representation was more detailed, that level of detail did not provide relevant information from an organisational recovery perspective. Furthermore, because the available information sources did not explicitly specify the precise effects of various ransomware strains at the level of individual hardware components, the first representation ended up distracting the modellers from the most relevant recovery aspects: the time of device infection and the recovery duration. However, once the device conceptualisation was changed in a subsequent cycle iteration, and the modelling team members became familiar with the new device identity, the rest of the translation zone processes became simpler, because they did not require integration with the removed aspects of the device representation.

Regarding the *scale-freeness* property, we can argue that it was manifested in a set of different ways. Firstly, at the level of the concepts: both resources and locations were treated similarly, regardless of their physical or abstract nature or real-world scale. Secondly, all the models' scale can be adjusted with the help of model parameters. For example, in the trauma unit model, the number of available medical personnel, the number of treatment bays or the patient arrival rate can be set through external parameters. This can lead to simulations representing very small trauma units, or large trauma departments by essentially using the same structure. Lastly, *scale-freeness* is also supported via the formal composition operation. For instance, we can easily envision the emergency trauma unit model as being composed with a triage model on the patient input side, and with other hospital departments on the output side, in an attempt to explicitly represent an entire hospital.

Turning our attention to co-design aspects, we further describe some of the observed benefits, but also difficulties encountered.

Firstly, in both the cases of the ransomware recovery and trauma unit models, the object of inquiry was not decided by modellers alone, but rather was provided by organisational stakeholders like the security research manager and consultant anaesthetist, and then aligned with the research goals of the modellers during the construction of the model scope. Because of that, the stakeholders had a direct interest in the success of the model development, and helped the modelling team not only with domain knowledge, but also by facilitating meetings with other experts and stakeholders. This explicitly improved the domain exploration and validation steps, by allowing the modellers to have access to more information sources and providing experience-based opinions where necessary. Also, it mitigated one of the primary difficulties encountered, namely the impossibility of all the stakeholders to be continuously present in the modelling process. Although this was not the case in the showcased model examples, our belief is that the overall model quality might have been reduced if these stakeholders would not have been interested and actively engaged in the modelling task.

Secondly, both the ransomware recovery and trauma unit models have produced representations, behaviours, and results comparable to their real-world targets. In both the case of ransomware recovery and trauma unit models, this has been shown via comparisons with the Sophos 2024 ransomware report [276], and with the recorded data about the exhibited behaviour in the trauma unit during the 2017

London Bridge terror attack, as described in the above subsection. Additionally, the results produced by the ransomware recovery model were used to determine a possible admin policy issue that was not even originally considered or the main goal of the model. Therefore, this showcases the ability of the modelling methodology to produce models that can facilitate the discovery of phenomena outside of the explicit model scope, but with a causal influence on it. The absence of a limitation in admin behaviour was noted precisely because it had an impact on the overall recovery duration, but since the stakeholders — and specifically the help-desk specialist — knew that such a policy did not exist, the model did not accounted for it either.

Last but not least, it is important to acknowledge the occurrence of mutual learning throughout the co-design process. For example, during the development of the ransomware recovery model, the research manager installed, ran, and analysed model outputs — with the help of the modellers — in a programming language with which he was not familiar with, and the modellers learned about state-of-the-art hardware recovery mechanisms. Similarly, in the trauma unit case, the modellers learned about healthcare facility operation and treatment procedures, whereas the consultant anaesthetist became familiarised with the design, construction and interpretation of models under our presented approach.

Chapter 7

General Conclusions

In this thesis, we have attempted to construct, illustrate, and justify the utility of an approach to modelling heterogeneous systems focused on pragmatic multi-methodological integration, inspired by a metaphor of distributed systems from computer science. The argument justifying this decision can be summarised as follows:

If the objects of modelling continue increasing in complexity and heterogeneous systems maintain their importance for the average persons' life, then modelling approaches will have to adapt to these new targets as well, or risk losing their relevance. Given the heterogeneity aspect, this will most likely require integration between modelling traditions of various scientific disciplines. For that to be possible, the exploration of the domain in which the object of modelling is manifested and the relationship between model and goals must take into account notions that span across multiple disciplines. We argue that such candidates are the means of construction of both the observed phenomena and model — hence the multi-methodological commitment —, which can provide a neutral interpretation basis when underlined by a pragmatic inferentialist perspective. Furthermore, the principles of co-design describe ways in which such integration can be achieved, but require the participants to have an interest in pluri-perspectivist understanding of the phenomena under study and to partake in knowledge sharing during the process. We cannot assume a priori that modelling participants posses these interests, but we can attempt to introduce them during the modelling process and justify their utility towards the participants. In an attempt to do so, we conceptualise the translation zone of our extended co-design cycle as a trading zone and then employ the concepts of the distributed systems metaphor as a cultural, in-between language tool — this would not be possible if the trading zone and distributed systems metaphor would not be compatible — to facilitate knowledge sharing and interactional expertise development, therefore driving the evolution of the trading zone from initially fractionated, towards inter-language.

In the following sections, we summarise the key points illustrated at the level of each chapter of the thesis, relate them to the research questions proposed initially, discuss advantages, limitations and possibilities for further work and conclude.

7.1 Chapters Summary

In Chapter 2 we briefly analysed referentialist, inferentialist and more pragmatic, engineering-focused philosophic positions regarding the nature of models. In doing so, we identified a research gap regarding the conceptualisation of heterogeneous systems: even if most modelling accounts included subtle hints about accepting multi-methodological approaches and claimed an ability to represent heterogeneous systems, this was usually not explicitly reflected in the conceptual frameworks or abstractions involved — the representation of heterogeneous model targets was conducted in a somewhat reductionist manner at best. Our analysis has led us to the idea that in order to acknowledge the possibility of multiple valid interpretations, but also meaningfully consider their implications towards model outcomes, a sceptical, pragmatic, inferentialist modelling account is required. As an initial step in the construction of the account, we made some necessary commitments: to accept heterogeneity and multi-methodology, to reduce commitments to scientific realism — epistemically, ontologically, and metaphysically —, but to conserve a series of useful properties such as composition, substitution and local reasoning which can be associated to scientific realism and direct reference theory — but have been constructed in the field of logics.

In Chapter 3, following the above commitments, we introduced our modelling account. We started by constructing a definition of heterogeneous system — the explicit target of our modelling formalism — as a synthesis of multiple definitions of the notions of system and heterogeneity used across sciences. Then, we defined our notion of model and explained the philosophical aspects of the account, centred on models as collections of sub-models epistemically, ontologically, and metaphysically. Based on that, we constructed a qualitative metric for describing models grounded in inferentialism and specifically focused on the means of construction of both observed phenomena and constructed models. We then employed it at the level of a security models case study attempting to provide a general view of the nature of models used in the security research community in 2020. Lastly, we reflected on the implications of heterogeneity and model diversity towards validation in the same security context, by looking at three research streams — economics, management science and systems dynamics — where this issue has been previously encountered, and which have been commonly imported to the field of information security.

In Chapter 4, we described the conceptual framework employed by our modelling methodology explicitly — namely the distributed systems metaphor — in terms of conceptual, formal and executable aspects — the elements of the 'triangle framework' from Chapter 3. In order to ensure that the resulting models have a higher chance of achieving their pre-stated goals, we introduced a second metaphor — namely the trading zone [126, 125, 124] — compared it with the distributed systems metaphor in terms of entities, interactions, language, methods, practices and goals, and determined them as compatible. This choice was in no way arbitrary. The trading zone offered two primary advantages: a criterion for functioning based on identity conservation [278] — which we showed the distributed systems metaphor conserves —, and an evolution pattern from fractionated towards inter-language [72] that facilitates an increase in the development of interactional expertise and knowledge sharing, which are relevant factors for better representational quality but also inherent motivators for the modelling participants. Finally, we determined that an evolution trajectory from fractionated towards inter-language can be achieved in a modelling context with the help of an in-between language cultural tool, represented by the distributed systems conceptualisation.

In Chapter 5 we showcased the actual methodology, integrating all the above described factors. Firstly, we clarified the research context by comparing it to co-design and co-creation approaches. Then, we described and identified caveats in the classical, mathematical modelling cycle and constructed our own version of co-design cycle in an attempt to solve them for the case of heterogeneous models.

In Chapter 6, we illustrated the flexibility and usability of the overall approach at the level of three different security oriented models: a physical data-loss model, a trauma unit surge capacity model, and an organisational recovery under ransomware model. We described their goals, internal structure and representation choices and illustrated how each of the models focused on different aspects of the metaphor. Again, we note that these three models have different maturity levels, and the organisational ransomware recovery one is significantly more complex than the others.

Lastly, appendices A, B, and C contain additional information related to the organisational recovery model: the analysis of ransomware strains, the recovery timings used for the network, embedded and usb recovery techniques, the types, values and meanings of both the static and variable parameters, the numerical and graphical representations of the PAWN sensitivity analysis, and the impact of the variable input parameters on the model outputs, respectively.

7.2 Research questions

As stated initially in Section 1.1, this thesis addresses the following open research question in its entirety:

How can models better describe the world and produce better results in the world if the underlying realities and systems to be modelled become more complex, the number of parties involved in the modelling process increases, and their assumptions about such realities are not always in agreement?

Under our subjective interpretation, we have achieved a reasonable degree of success in answering that question. Below, we present our succinct answers to each of the component questions derived from the above one:

Q1: *What philosophical attitude towards modelling would be most suitable to underline a modelling account focused on integration?*

A1: Sceptical inferentialism with deflationary notes, and an explicit acceptance of multi-methodology. This has been discussed in Chapter 2.

Q2: *How should we conceptualise models of heterogeneous systems?*

A2: As collections of sub-models — metaphysically, epistemically, and ontologically —, by initially focusing on the nature of the techniques involved in their construction, as described by the Triangle Framework, and then by following the distributed systems conceptualisation at the level of the actual representation. This question has been addressed in Chapters 3 and 4.

Q3: *How can we ensure the consistency of a heterogeneous model?*

A3: Philosophically, the concept of multi-methodological consistency is almost paradoxical if understood in the classical sense: multi-methodological consistency is about the consistency of the process of integrating the different perspectives, rather than a singular ontology or common epistemic criteria. At the level of sub-models, the techniques used in each sub-model must be consistent with those of the research or practice communities — including the expertise of participants — associated with the domain of inquiry. When considering the whole model, consistency is achieved representationally, via the distributed systems metaphor. Furthermore, the iterations through the extended co-design cycle ensure the resulting model achieves a degree of consistency with respect to the interpretation of phenomena behaviour in the domain, as possessed by the participants in the act of modelling. For more details on this, see Chapters 4 and 5.

Q4: *How can we integrate the knowledge and beliefs of the modelling participants at the level of the model?*

A4: By constructing models using the process described by the extended co-design cycle. Multiple steps in the cycle explicitly require debates between participants: the interpretation of available domain information, the establishment and update of model scope, the translation zone processes, the translation and interpretation of model consequences, and the interpretation of verification and validation results. Their purpose is precisely to ensure that the different perspectives of the participants are incorporated in the model. This question is explored in Chapters 5

Q5: *How can we increase the probability of a model achieving its goals?*

A5: By using the distributed systems metaphor — and the triangle configuration — as in-between language cultural tool in the extended co-design cycle. This works because the distributed systems metaphor satisfies the identity conservation criterion for the functioning of a trading zone, and the trading zone and distributed systems metaphors are compatible. This ensures that the resulting modelling team culture facilitates the development of interactional expertise and knowledge sharing, which are relevant factors for both high representation quality and motivators for the participants. This point was described in Chapters 4 and 5.

7.3 Advantages

In this thesis, we have presented a modelling theory comprised of three different components: a methodology — the extended co-design cycle from Chapter 4 —, a modelling account — the philosophical commitments from Chapter 2, reflected in the definitions and epistemic, metaphysic, and ontological positions of Chapter 3 and supported by the qualities described in the triangle framework —, and a structuring metaphor — or meta-language represented by the distributed systems metaphor.

Below, we summarise the proposed advantages of this construction:

- The ability to construct pluri-perspectivist representations of phenomena — extended to heterogeneous systems —, by explicitly integrating the multiple views of the modelling participants at the level of the model — model scope, translation zone, interpretation of verification and validation processes.

- A descriptive metric — the triangle framework — assisting the modelling participants in determining the nature of model components, for the purpose of increasing representation accuracy, in alignment with the modelling goals, and therefore reducing the probability of creating a model impractical for use.

- A general and recognizable meta-framework — the distributed systems metaphor — to serve as a common base for argumentation, and therefore language for both structuring the process of design and construction and, formulation of requirements.

- The ability to construct scale-free representations which conserve their identity, as provided by the distributed systems metaphor.

- A way of systematically analysing a model through all the design and construction stages — and possibly deployment in the future — rather than just at the end, regardless of possible conflicting or unstated assumptions;

- A reduction in development time as a result of the ability to determine and decide upon the nature of model components during co-design iterations, rather than requiring an analysis of the emergent model, post construction.

- The conservation of a set of formal properties — composition, substitution, and local reasoning — associated with the modularity, ease of use, but also rigorous, structured reasoning about model components. Local reasoning, in this context, can be viewed as an operation which reduces the time of model analysis, by only considering the relevant aspects of sub-models involved in a composition.

- An extensive toolset for the practical construction of executable models — the Julia SysModels [61] package.

- A way of dealing with uncertainty at the level of executable model simulations, due to the stochastic treatment.

- A method facilitating the development of inherent motivators at the level of the modelling team, such as a culture fostering knowledge sharing and interactional expertise development.

- A set of promising real-world results achieved in security oriented contexts: both the descriptive power showcased in the security models case study in Chapter 3, and the representation quality of the three models in Chapter 6 — as assessed by the medical practitioners in the case of the trauma room surge capacity model, and as illustrated by the comparison between the results of the ransomware recovery model, and the Sophos 2024 ransomware report [276].

- An ability to uncover new factors influencing the model outcomes, as shown by the discovery of the relevance of admin deployment strategies in the ransomware recovery model.

We are aware that exemplification alone cannot be seen as a complete proof for quality, but we believe that the theoretical argument, corroborated with the useful properties introduced and the promising results obtained up to this point should at least make our methodological approach to heterogeneous modelling a solid candidate for further experimentation.

7.4 Limitations & further work

Given the complexity of the approach proposed in this thesis, there should be no surprise that multiple areas of future work can be identified. Below, we illustrate a few directions that we consider worth pursuing:

Firstly, although the distributed systems metaphor is general and recognizable enough as to be used to represent any kind of system, this does not mean it is the most suitable meta-framework to underline representation for all models across all domains, at least in the most basic form. However, as shown in Section 6.1.1, the metaphor allows for further conceptual refinement via the construction of new concepts — agents as a bundle of a resource, a process and multiple locations. An in-depth analysis of the most used metaphors across scientific fields could be used to determine new concepts to be included in the metaphor when modelling phenomena specific to that area of inquiry.

Secondly, as previously stated, our approach is centred around deflationary, sceptical inferentialism, leaving open the possibility of inference from beliefs — because explicit criteria for what can be considered knowledge are not provided. Yet, to satisfy the adherence to a from of scepticism, the introduced beliefs cannot be aleatory: they ought to be at least justifiable — with respect to replicable empiric observation, well established criteria in a scientific field or profession, or produce useful inferences — and accepted by the modelling participants. The proposed version of our approach places the decision of what is considered justified belief with the participants, but additional research could be carried out in an attempt to determine if specific types of 'justified beliefs' can be associated with higher model success post deployment.

Thirdly, the extended co-design cycle could be used as a tool for identifying areas where bias was introduced during model design and construction, even in the cases of model constructed under a different formalism. However, this requires access to extended information about the data, involved team, assumptions, structural updates, post deployment behaviour which are more often than not unavailable.

Fourthly, the provided software modelling package could be further enhanced by an addition of dynamic model checking and explicit local reasoning capabilities.

Fifthly, as with any emerging methodology, additional empirical research towards refinement is desirable. This could be directed towards model comparisons to assess quality — over time, in competitive environment — studying the relationship between the triangle configuration, goal achievement and the extended co-design cycle's exit criteria, or attempting to predict model evolution based on goals, team composition and available data.

Sixthly, and related to the above point, additional experimentation can be done with respect to the philosophy that underlines the methodology. For the author, pragmatic inferentialism with deflationary and sceptical notes is seen as a viable direction, given the focus on integration. However, the methodological steps are valid even when underlined by another philosophical position — for instance, as shown in Chapter 2, Frigg's DEKI account manifests similarities when compared to the Co-design Cycle, but is underlined by scientific realism — so other directions can be pursued, and perhaps even mapped to models from different scientific disciplines.

Last but not least, we must acknowledge possible interactions between our proposed theory, method, account, and AI models — under our definition, an AI model is still a model. Therefore, it should be theoretically possible to translate an AI model to one underlined by the distributed systems metaphor by analysing, interpreting and conceptualising it from a large enough sample of the AI behaviour — in this way obtaining a higher degree of understanding about its workings. Additionally, this process might be susceptible to automation: a possible direction might be represented by the integration of a query based learning algorithm and a large language model. In [41], the authors managed to infer the symbolic representation of a neural network by using a modified version of Angluin's L* algorithm [11]. If meaningful translation between such a symbolic representation and the distributed systems one is possible, with the help of behaviour confirming queries and automated interpretation, then perhaps automated model construction while ensuring the understanding of inner model workings is not impossible. If that is the case, we can speculate about significant impacts, for instance in the development of defensive solutions for gatekeeping model behaviour — AI safety — or offensive applications like model reversion and extraction attacks.

7.5 Conclusion

As concluding remark, we can only hope that the people exposed to this work will treat models carefully. While they are essential tools for thinking, our belief is that they cannot capture the entirety of what we seek, when we seek it and perhaps not even in the form that we seek. Whether or not this is due to limitations of language, knowledge, human cognition, or more empiric concerns still remains open. However, renouncing the use of models will never be an option, at least not until the concept of cognition as scientifically known today will drastically change. Therefore, we suggest the reader adopts a humble approach to interacting with and constructing models, acknowledging that certain dimensions of human existence may forever lie beyond our capacity to define or capture.

In a certain sense, humans are still living in the same cave, chained with models while at the same time looking through models to unchain themselves.

Appendix A

Ransomware Strains

This section contains information regarding the ransomware strains analysed for the purpose of constructing the conceptual ransomware model from section 6.3. Based on the overall goals and techniques employed in the attacks, the strains are clustered into five categories — locker, crypto locker, leakware, destructive ransomware and ransomware as a service —, with the mention that some ransomware can fit into multiple categories at the same time. Therefore, these categories should be understood rather as operational capabilities of the ransomware, than completely separable classes. Furthermore, we took into account the entry point and type of network spreading once infection has been achieved.

Name	**Year**	**Locker**	**Crypto Locker**	Leakware	**Destructive**	**RaaS**	**Entry Point**	**Automated Spread**
PC Cyborg	1989	X					Floppy disks	None
Alien (Young & Yung)	1996		X				None	None
Young Leakware	2003			X			None	None
Krotten	2004	X					Website downloads	None
PGPCoder	2005		X				Website downloads Phishing emails	None
Archiveus	2006		X		X		Website downloads Phishing emailss	None
CryZip	2006		X				Website downloads	None
Reveton	2012	X	X				Website downloads Phishing emails Exploit Kits	None
Shamoon **StoneDrill**	2012				X		Phishing emails & Human operation	Network shares
CryptoLocker	2013		X				Phishing emails Botnet droppers	None
CryptoWall **TorrentLocker**	2014		X				Phishing emails Website downloads Exploit Kits	None
Linux.Encoder **KeRanger**	2015		X				Magento shopping cart vulnerability	None
RansomWeb	2015		X				Manual exploit	None
Fusob **Small**	2015	X	X				Website downloads Exploit kits	None
Erebus	2016		X				Website downloads	None
Petya	2016		X				Phishing emails MeDoc exploit	Credential harvesting EternalBlue Worm like
SamSam	2016		X				JexBoss Exploit Kit RDP with stolen or brute force credentials	None
WannaCry	2017		X				Phishing emails Scanning TCP port 445 — EternalBlue DoublePulsar backdoor	EternalBlue DoublePulsar Worm like
NotPetya	2017		X		X		MeDoc exploit, EternalBlue Eternal Romance, Phishing emails	Credential harvesting, token impersonation EternalBlue, EternalRomance Worm like
BadRabbit	2017		X				Website downloads	Credential harvesting, dictionary attacks, network shares, EternalSynergy, Worm Like
Maze	2019		X	X			Phishing emails Website downloads Exploit kits Citrix web gateway RDP	None (human operated, various exploits are used, but manually)
RobbinHood	2019		X				RDP (brute force) Exploit kits	None (human operated, various exploits are used, but manually)
Conti	2019		X	X	X	X	TrickBot Malware Spear Phishing Human Operation(Buying access RDP with stolen or brute force credentials)	None (human operated, various exploits are used, but manually)
Darkside	2020		X	X		X	Phishing emails Human operated (RDP with stolen or brute forced credentials Attacks on Virtual Desktop Infrastructure (VDI))	None (human operated, various exploits are used, but manually)
Netwalker	2020		X	X		X	Phishing emails Human operated (RDP with stolen or brute forced credentials Attacks on Virtual Desktop Infrastructure (VDI) Pulse Secure VPN exploit Telerik UI exploit)	None (human operated, various exploits are used, but manually)

Table A.1: Ransomware Strains [151]

Appendix B

Organisational Recovery Model Parameters

This section contains information regarding the array of parameters used in the initialisation and execution of our models. Tables B.1 and B.2 contain manually gathered recovery timings for the network, embedded and usb recovery. The same timing values have been also used in [20]. Furthermore, tables B.3 and B.4 describe in detail the types, values and meanings of both the static and variable parameters used in the model.

Recovery Steps	Sure Recover	Embedded
Initialise Recovery	40	30
Copy from embedded	N/A	20
Download and verify Recovery Agent	100	N/A
Boot to recovery agent	15	25
Initialise Drive	25	N/A
Download Imaged	1130	N/A
Verify Image	50	40
Extract Image	180	145
Install Drivers	60	80
Windows Installer to Config Screen	480	480

Table B.1: HP Sure Recover times for both the network-based recovery and embedded recovery. Times are given in seconds and are based on a number of recovery cycles. [151]

USB Step	Time
Create Bootable USB	
Download recovery tool	60
Run Tool	45
Partition USB	35
Download Imaged	1260
Extract Image to USB	3120
Install from USB	
Boot USB to Installer	120
Partition Disk	60
Install Windows and Drivers to disk	780
Windows Installer to Config Screen	480

Table B.2: USB Based Recovery times. The first part of the table shows the steps in using a recovery tool to create a bootable windows installer. The second section shows times for the install from the USB stick. Again, times are quoted in seconds. [151]

Parameter Name	**Type**	**Values**	**Meaning**
device_scenario	Categorical	1 - USB recovery 2 - Network recovery 3 - Embedded Recovery 4 - Network recovery with 30% embedded recovery	The type of recovery technique the devices will use.
attack_scenario	Categorical	1 - Uniform 2 - Exponential 3 - Fdist 4 - Uniform + Exponential 5 - Uniform + Fdist	The types of distributions used when sampling for the timings of the attack/malware packets.
infection_probability	Float	[0.1, 0.3, 0.5, 0.7, 0.95]	The probability a device will get infected when hit by a malware packet.
attack_duration	Int	1 - 2 hours 2 - 4 hours 3 - 8 hours	The duration in hours that the malware attack will take.
admins_nr	Int	[1, 2, 3, 5, 10, 15]	The number of admins that can be found in a big office.
admin_need	Float	[0.0, 0.25, 0.50, 0.75, 1.0]	The probability a user needs a physical admin, or to speak with an admin remotely to start the recovery process.

Table B.3: Model Variable Parameters [151]

Parameter Name	Type	Values	Meaning
num_iterations	Int	50	The number of times the model will be run with the same parameter configuration.
proc_num	Int	50	The process number running a certain model iteration. Used for running multiple iterations in parallel.
nr_of_samples	Float	300	The number of attack packets sent by the malware model in a specified attack_duration.
attack_targets	[String]	[”Office 1 LAN”, ”Office 2 LAN”, ”Travel WiFi”, ”Small Office 1 LAN”, ”Small Office 2 LAN”, ”Office 1 WiF”, ”Office 2 WiFi”, ”Small Office 1 WiFi”, ”Coffee WiFi”,”Home WiFi”,”Small Office 2 WiFi”]	The names of the locations where devices to be targeted by malware can be found.
phisical_admin_time	Float	15m	The time it takes for an admin to physically perform the needed operations to start a recovery process.
admin_movement_time	Float	120m	The time it takes for an admin to physically move to a location where a recovery process is needed.
admin_remote_time	Float	20m	The time it takes for an admin to guide a user remotely to start a recovery process..
os_images	[OSImage]	[windows10iso, windows10wim, recovery_agent]	The list of available recovery images on the recovery server.
max_office_devices	Int	30	The maximum number of devices that can connect to the network from an office.
max_home_devices	Int	65	The maximum number of devices that can connect to the network from home.
max_coffee_devices	Int	65	The maximum number of devices that can connect to the network from a coffee shop.
max_travel_devices	Int	30	The maximum number of devices available for travelling.
scale_uni	Float	[72, 144, 288]	Scaling factor used for interval boundaries movement for attack packets that do not use mixed distributions.
scale_dst	Float	[400, 750, 1500]	Scaling factor used for interval boundaries movement for attack packets that use mixed distributions.
num_office_desktops	Int	20	The actual number of desktops in a big office.
num_office_laptops	Int	20	The actual number of laptops in a big office.
num_small_office_desktops	Int	10	The actual number of desktops in a small office.
num_small_office_laptops	Int	10	The actual number of desktops in a small office.
num_travel_laptops	Int	5	The actual number of laptops that can be used for travelling from a big office.
server_speed	Float	10.0 * 1024ˆ4	The upload/download speed of the recovery server.
office_lan_speed	Float	1.0 * 1024ˆ3	LAN upload/download speed for a big office.
office_wifi_speed	Float	150.0 * 1024ˆ2	Wifi upload/download speed for a big office.
office_link_speed	Float	1.0 * 1024ˆ3	Switch upload/download speed for a big office.
small_office_lan_speed	Float	1.0 * 1024ˆ3	LAN upload/download speed for a small office.
small_office_wifi_speed	Float	150.0 * 1024ˆ2	Wifi upload/download speed for a small office.
small_office_link_speed	Float	200.0 * 1024ˆ2	Switch upload/download speed for a small office.
coffeeshop_download_speed	Float	12.0 * 1024ˆ2	Download speed for a coffee shop.
coffeeshop_upload_speed	Float	2.0 * 1024ˆ2	Upload speed for a coffee shop.
coffeeshop_link_speed	Float	100.0 * 1024ˆ4	Switch upload/download speed for a coffee shop.
home_download_speed	Float	50.0 * 1024ˆ2	Download speed for home.
home_upload_speed	Float	4.0 * 1024ˆ2	Upload speed for home.
home_link_speed	Float	100.0 * 1024ˆ4	Switch upload/download speed for home.
travel_download_speed	Float	30.0 * 1024ˆ2	Download speed for a travelling location.
travel_upload_speed	Float	5.0 * 1024ˆ2	Upload speed for a travelling location.
travel_link_speed	Float	100.0 * 1024ˆ4	Switch upload/download speed for a travelling location.
office1_usb_images	[OSImage]	[windows10iso]	The recovery images available on usb in office1.
office1_usb_blanks	Int	7	The number of blank usbs in office1.
office2_usb_images	[OSImage]	[windows10iso]	The recovery images available on usb in office2.
office2_usb_blanks	Int	7	The number of blank usbs in office2.
small_office1_usb_images	[OSImage]	Empty	The recovery images available on usb in small office1.
small_office1_usb_blanks	Int	3	The number of blank usbs in small office1.
small_office2_usb_images	[OSImage]	Empty	The recovery images available on usb in small office2.
small_office2_usb_blanks	Int	3	The number of blank usbs in small office2.
home_usb_images	[OSImage]	Empty	The recovery images available on usb at home.
home_usb_blanks	Int	0	The number of blank usbs at home.
coffee_usb_images	[OSImage]	Empty	The recovery images available on usb in a coffee shop.
coffee_usb_blanks	Int	0	The number of blank usbs in a coffee shop.
travel_usb_images	[OSImage]	Empty	The recovery images available on usb in a travelling location.
travel_usb_blanks	Int	0	The number of blank usbs in a travelling location.
travel_movement	[Movement]	[Movement(”Travel”, ”Travel WiFi”, Uniform(4hours, 5days)), Movement(”Office 1”, ”Office 1 WiFi”, Uniform(2hours,6hours)), Movement(”Office 2”, ”Office 2 WiFi”, Uniform(2hours,6hours)), Movement(”Small Office 1”, ”Small Office 1 WiFi”, Uniform(2hours,6hours)), Movement(”Small Office 2”, ”Small Office 2 WiFi”, Uniform(2hours,6hours))]	The list of possible movements from a travel location.
large_office_movement	[Movement]	[Movement(”Home”, ”Home WiFi”, Uniform(0hours, 3hours)), Movement(”Office ”, ”Office WiFi”, Uniform(5hours, 8hours)), Movement(”Coffee Shop”, ”Coffee WiFi”, Uniform(20minutes, 2hours))]	The list of possible movements from a large office location.
small_office_movement	[Movement]	[Movement(”Home”, ”Home WiFi”, Uniform(0hours, 3hours)), Movement(”Small Office ”, ”Small Office WiFi”, Uniform(5hours, 8hours)), Movement(”Coffee Shop”, ”Coffee WiFi”, Uniform(20minutes, 2hours))]	The list of possible movements from a small office location.

Table B.4: Model Static Parameters [151]

Appendix C

Sensitivity Analysis & Results

This appendix contains information regarding the overall model execution and sensitivity analysis. Tables C.1 to C.3 contain the numerical outputs of the PAWN Sensitivity Analysis. Figures C.1 to C.3 contain the graphical representation of the sensitivity analysis. Figures C.4 to C.15 show additional experimentation performed: we analysed the possible impact of the variable input parameters — infection probability, attack duration, need of admin help and number of admins — on the model outputs — average recovery duration, total recovery duration and average number of devices recovering. We note here that additional experimentation could have been performed at the level of the recovery method: for example, by fixing one recovery method and another variable and then performing sensitivity analysis again.

	Minimum	Mean	Median	Maximum	CV
Recovery Method	0.280556	0.501074	0.553111	0.669556	0.325332
Attack Type	0.092444	0.134972	0.144444	0.158556	0.189384
Infection Probability	0.023667	0.080861	0.077111	0.145556	0.534977
Attack Duration	0.032667	0.063111	0.063111	0.093556	0.482394
Admins Number	0.018333	0.039289	0.027222	0.095000	0.715312
Admins Need	0.065444	0.106944	0.104222	0.153889	0.294688

Table C.1: Sensitivity Analysis on the average recovery duration [151]

	Minimum	Mean	Median	Maximum	CV
Recovery Method	0.024222	0.031259	0.027556	0.042000	0.246833
Attack Type	0.100556	0.142000	0.146444	0.174556	0.187439
Infection Probability	0.307556	0.531000	0.508278	0.799889	0.337688
Attack Duration	0.072333	0.096167	0.096167	0.120000	0.247834
Admins Number	0.005111	0.006422	0.006333	0.007889	0.165729
Admins Need	0.007111	0.009639	0.009333	0.012778	0.228576

Table C.2: Sensitivity Analysis on the average number of devices recovering [151]

	Minimum	Mean	Median	Maximum	CV
Recovery Method	0.151222	0.275815	0.279667	0.396556	0.363265
Attack Type	0.088333	0.133667	0.142000	0.162333	0.208527
Infection Probability	0.145556	0.305056	0.243000	0.588667	0.552403
Attack Duration	0.043778	0.073500	0.073500	0.103222	0.404384
Admins Number	0.010667	0.029622	0.020889	0.074000	0.764506
Admins Need	0.023778	0.051000	0.046778	0.086667	0.444535

Table C.3: Sensitivity Analysis on the total recovery duration [151]

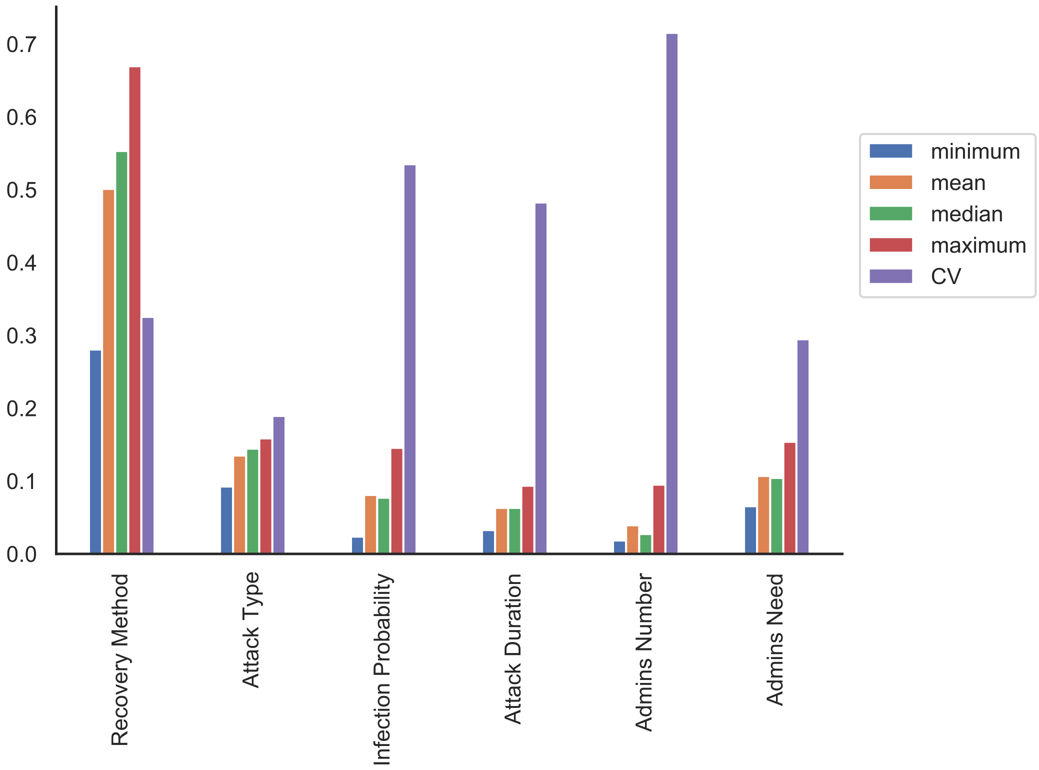


Figure C.1: Sensitivity Analysis on the average recovery duration of devices [151]

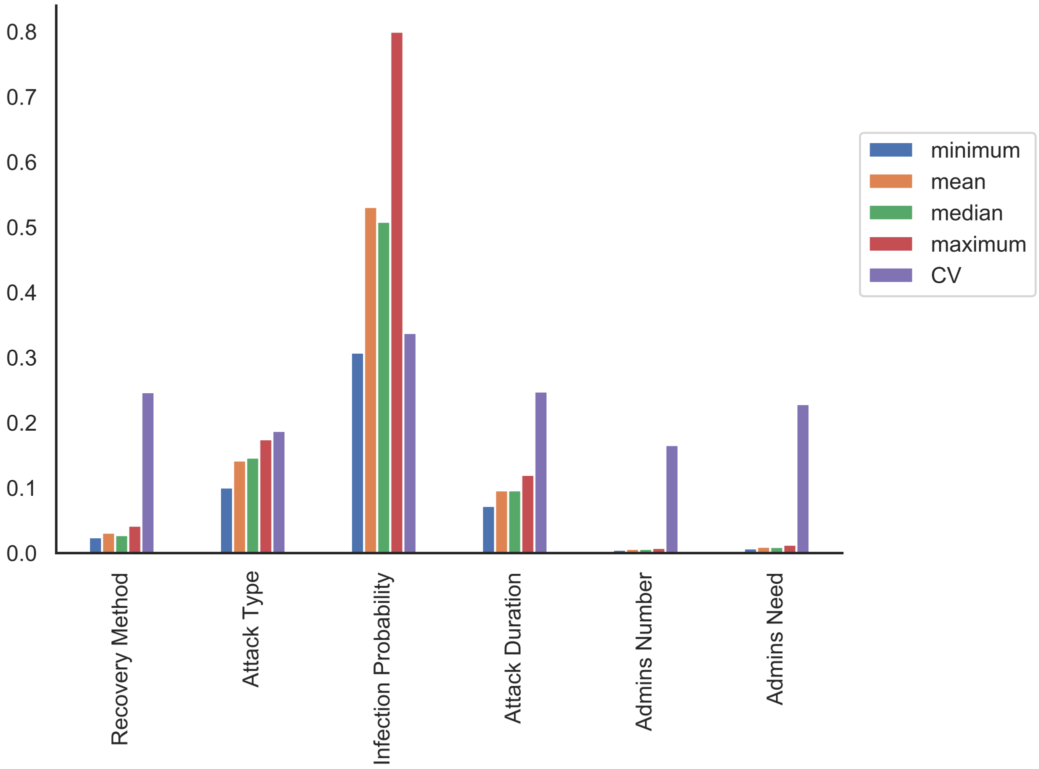


Figure C.2: Sensitivity Analysis on the average number of devices recovering [151]

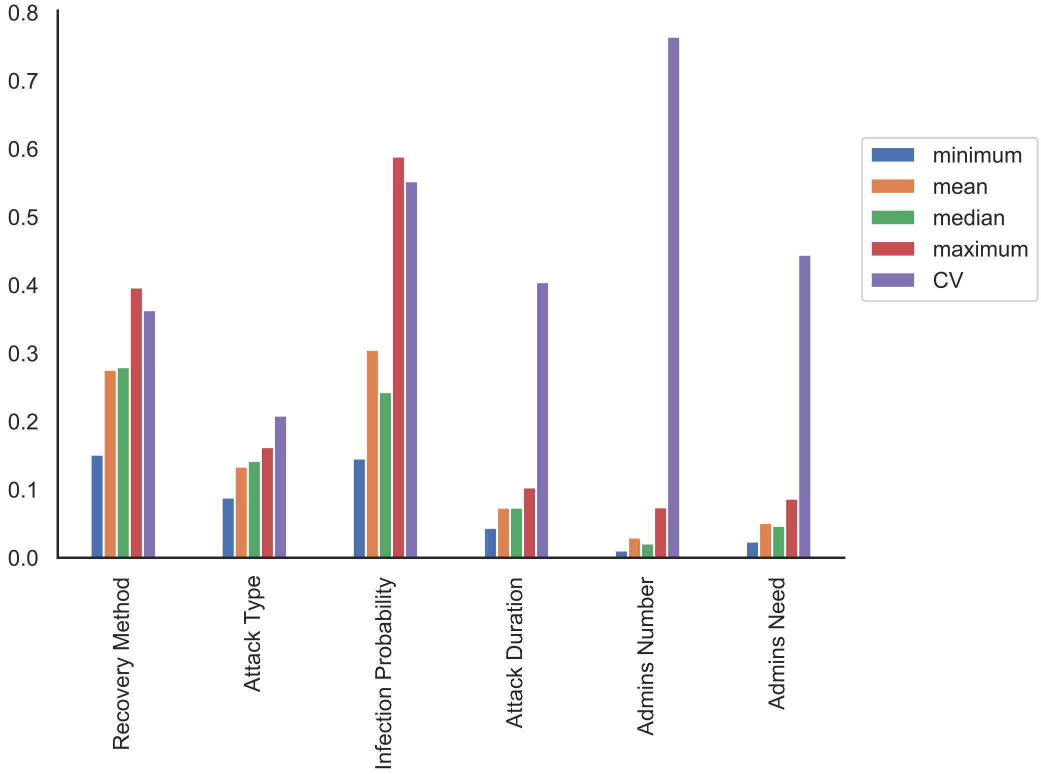


Figure C.3: Sensitivity Analysis on total recovery duration of devices [151]

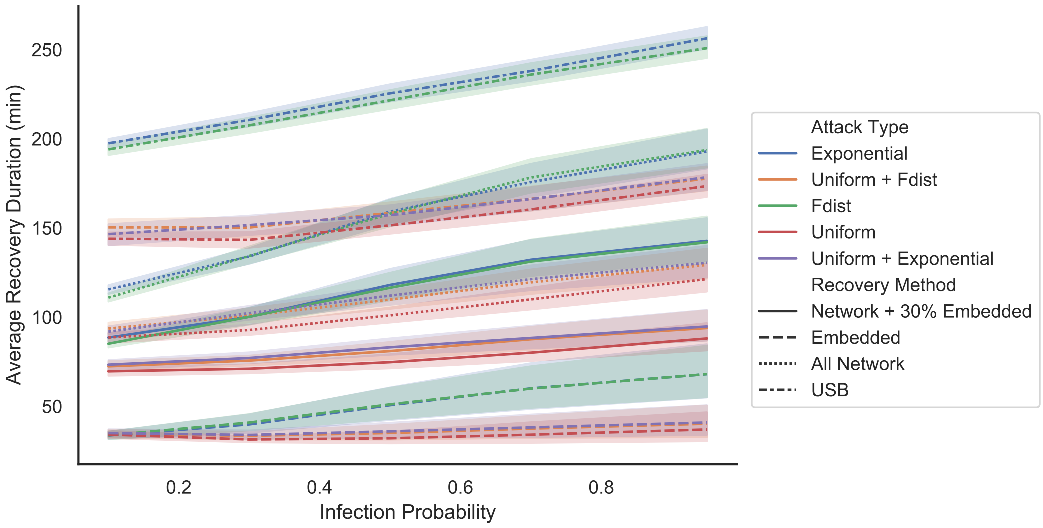


Figure C.4: Impact of infection probability on average recovery duration [151]

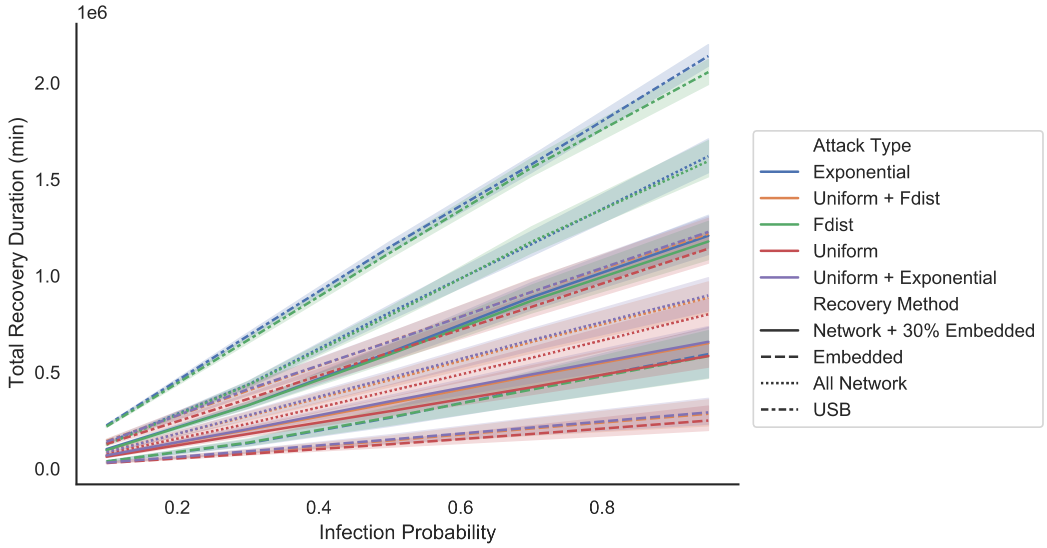


Figure C.5: Impact of infection probability on total recovery duration [151]

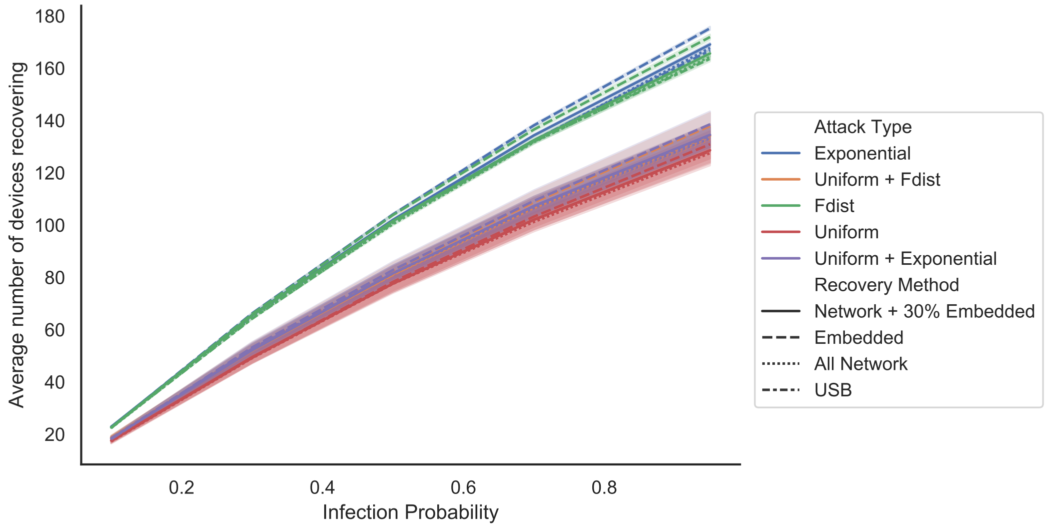


Figure C.6: Impact of infection probability on average number of devices recovering [151]

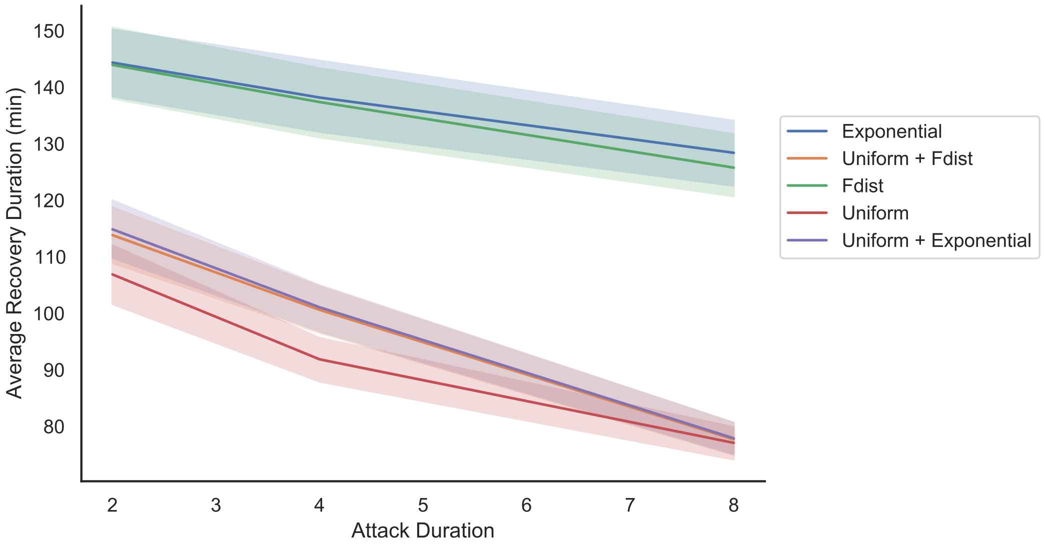


Figure C.7: Impact of attack duration on average recovery duration [151]

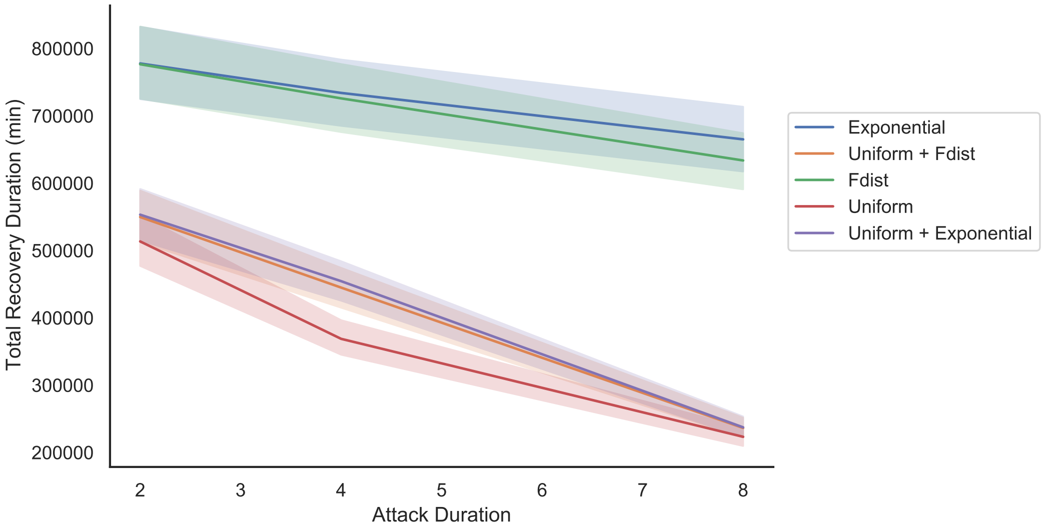


Figure C.8: Impact of attack duration on total recovery duration [151]

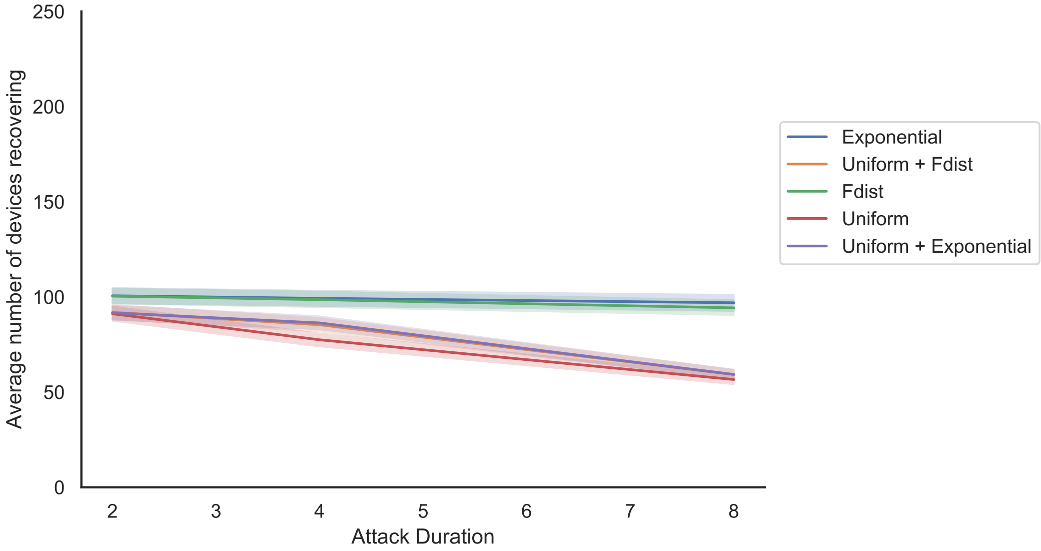


Figure C.9: Impact of attack duration on average number of devices recovering [151]

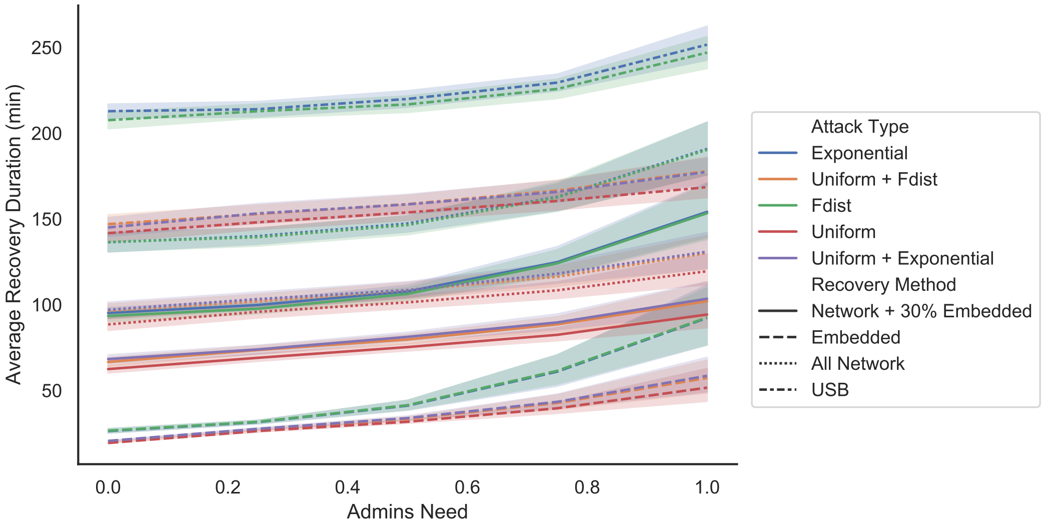


Figure C.10: Impact of Admins Need on average recovery duration [151]

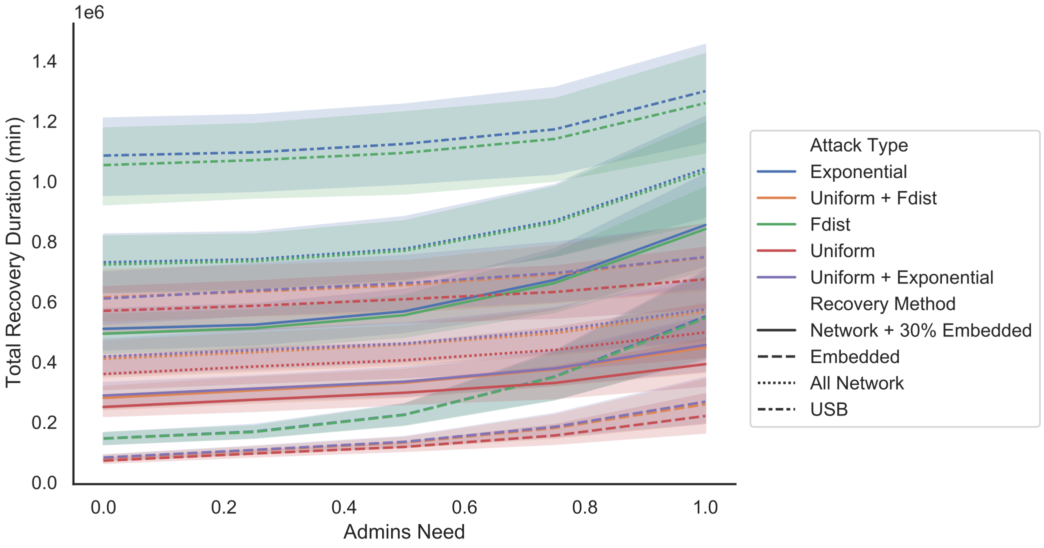


Figure C.11: Impact of Admins Need on total recovery duration [151]

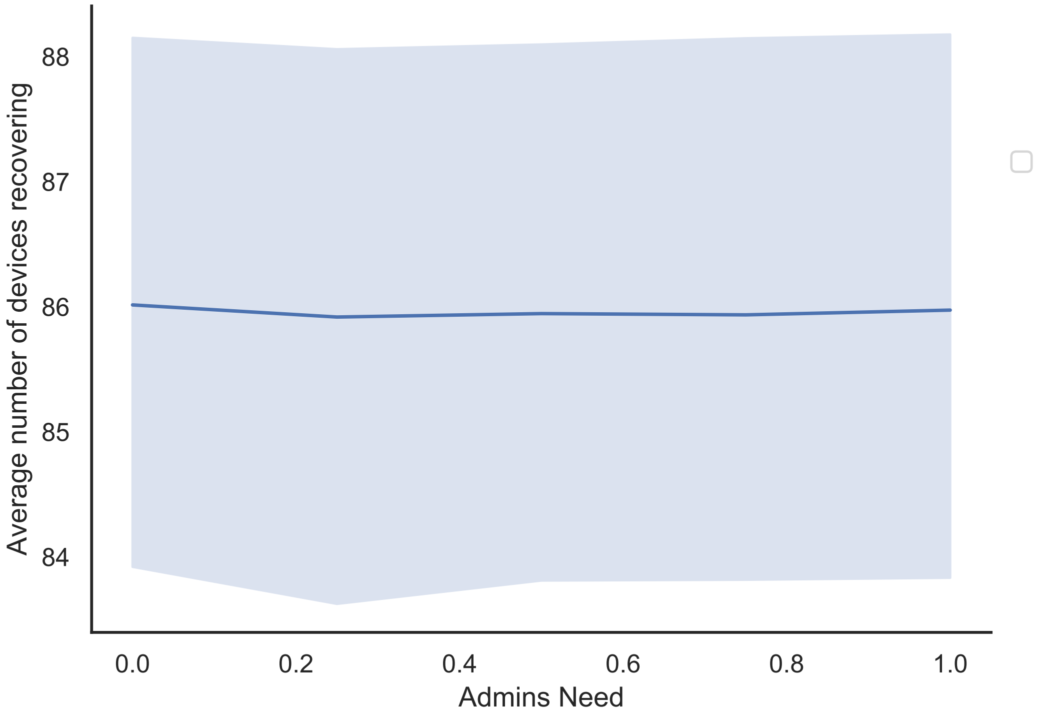


Figure C.12: Impact of Admins Need on the average number of devices recovering [151]

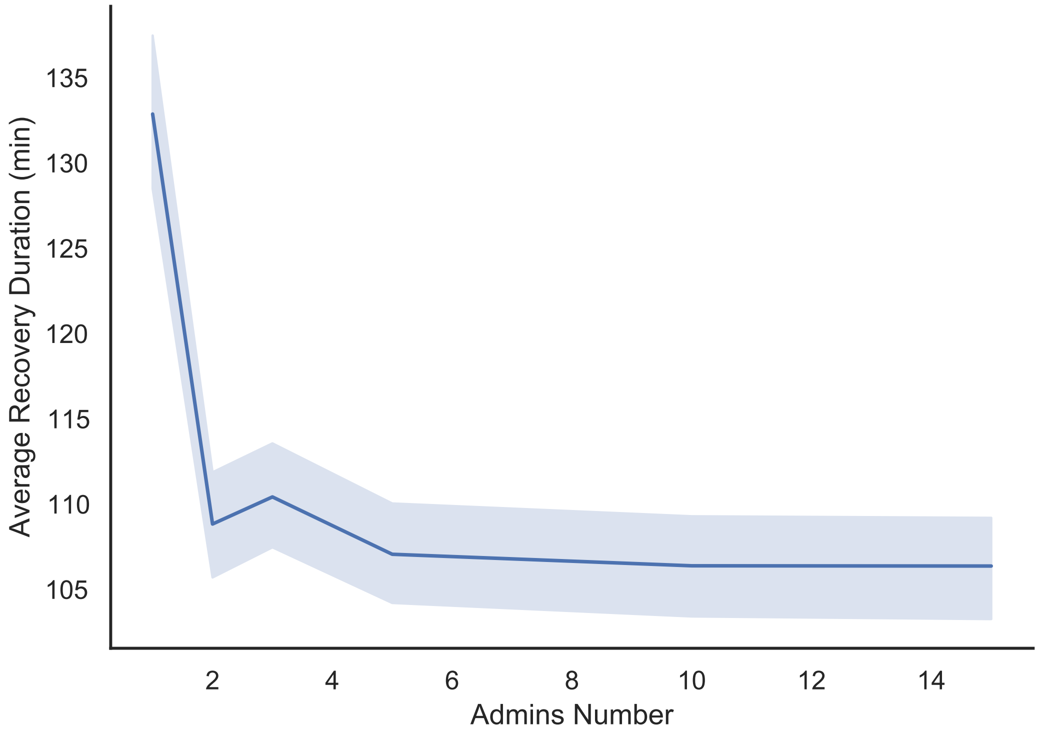


Figure C.13: Impact of Admins Number on average recovery duration [151]

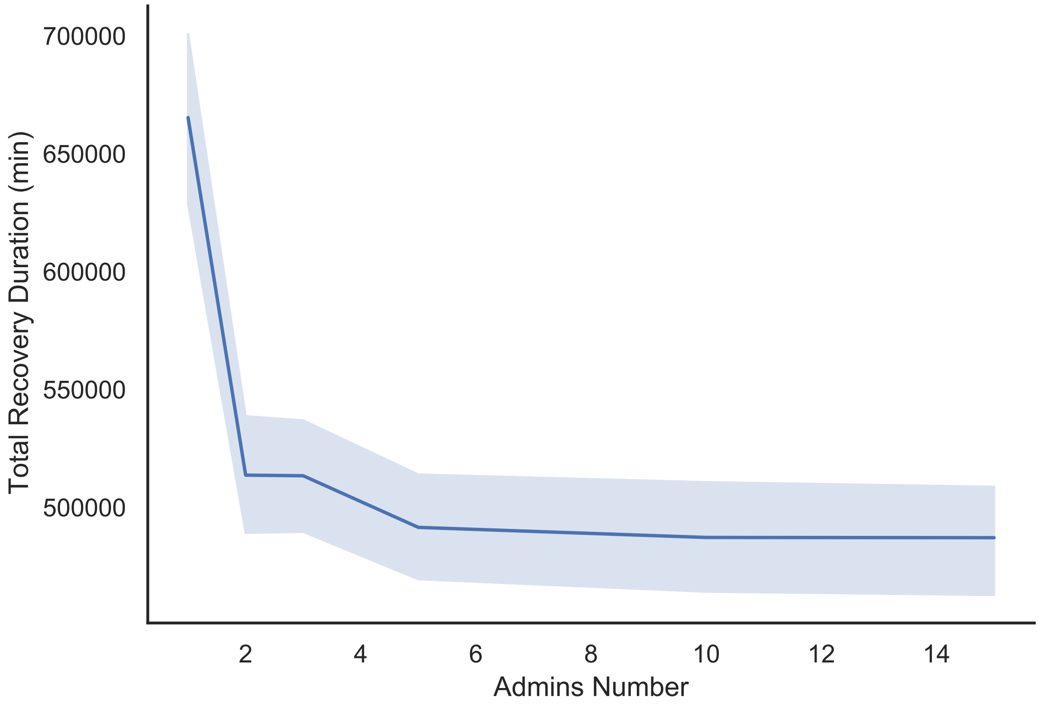


Figure C.14: Impact of Admins Number on total recovery duration [151]

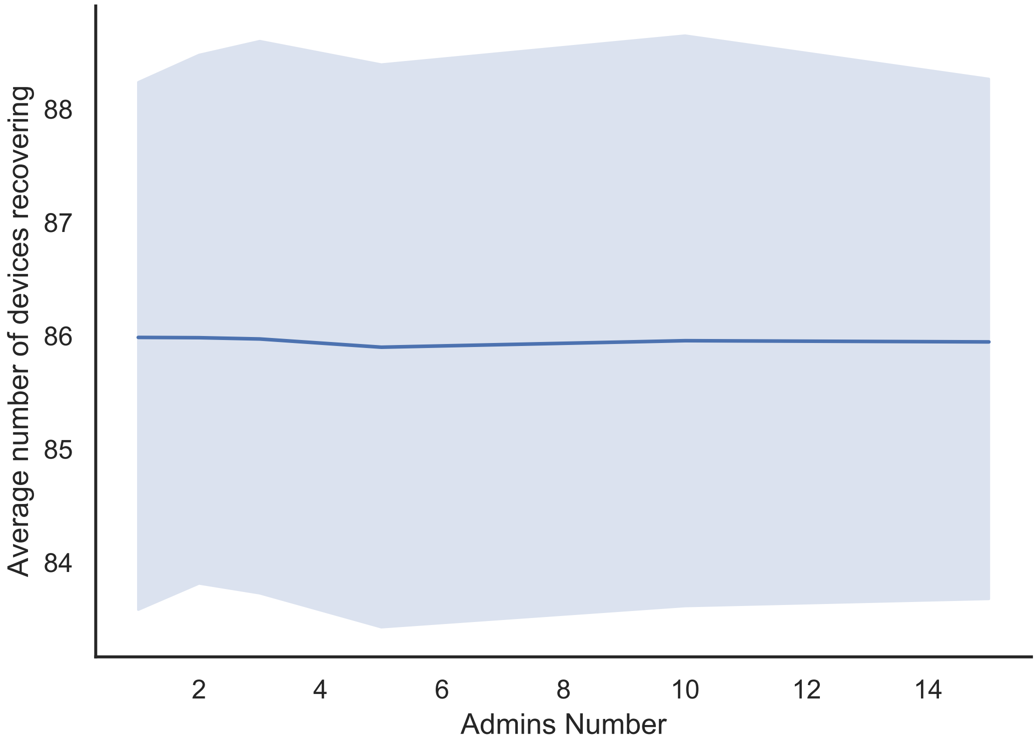


Figure C.15: Impact of Admins Number on average number of devices recovering [151]

Appendix D

Colophon

This document was set in the Times typeface using LaTeX and BibTeX, composed with Overleaf, and uses the *ucl_thesis.cls* created in 1996 by Russel Winder and maintained and distributed with permission, by Ian Kirker. The class can be publicly accessed at:

https://github.com/UCL/ucl-latex-thesis-templates/blob/master/ucl_thesis.cls

Bibliography

[1] Pekka Abrahamsson, Outi Salo, Jussi Ronkainen, and Juhani Warsta. Agile software development methods: Review and analysis. *arXiv preprint arXiv:1709.08439*, 2017.

[2] Lawrance Abrams. Jvckenwood hit by conti ransomware claiming theft of 1.5tb data., 2021. Available at: `https://www.bleepingcomputer.com/news/security/jvckenwood-hit-by-conti-ransomware-claiming-theft-of-15tb-data/`. Accessed 16/12/2024.

[3] Lawrence Abrams. Allied universal breached by maze ransomware, stolen data leaked, 2019. Available at: `https://www.bleepingcomputer.com/news/security/allied-universal-breached-by-maze-ransomware-stolen-data-leaked/`. Accessed 16/12/2024.

[4] Chadia Abras, Diane Maloney-Krichmar, Jenny Preece, et al. User-centered design. *Bainbridge, W. Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications*, 37(4):445–456, 2004.

[5] Dorit Aharonov and Oded Regev. Lattice problems in npconp. *Journal of the ACM (JACM)*, 52(5):749–765, 2005.

[6] Ian Alam. An exploratory investigation of user involvement in new service development. *Journal of the Academy of Marketing Science*, 30:250–261, 2002.

[7] Ali Ahmed Mohammed Ali Alwashali, Nor Azlina Abd Rahman, and Noris Ismail. A survey of ransomware as a service (raas) and methods to miti-

gate the attack. In *2021 14th International Conference on Developments in eSystems Engineering (DeSE)*, pages 92–96. IEEE, 2021.

[8] Oxford Analytica. Us pipeline hack to make ransomware risks a priority, 2021. Available at: `https://www.emerald.com/insight/content/doi/10.1108/oxan-ga261470/full/html`. Accessed 16/12/2024.

[9] David F Andersen. How differences in analytic paradigms can lead to differences in policy conclusions. *Elements of the system dynamics method*, pages 61–75, 1980.

[10] Gabrielle Anderson and David Pym. A calculus and logic of bunched resources and processes. *Theoretical Computer Science*, 614:63–96, 2016. `https://doi.org/10.1016/j.tcs.2015.11.035`.

[11] Dana Angluin. Queries and concept learning. *Machine learning*, 2:319–342, 1988. `https://doi.org/10.1023/A:1022821128753`.

[12] H Igor Ansoff and Dennis P Slevin. An appreciation of industrial dynamics. *Management science*, 14(7):383–397, 1968. `https://doi.org/10.1287/mnsc.14.7.383`.

[13] ARM. What is heterogenous compute?, 2024. Available at: `https://www.arm.com/glossary/heterogenous-compute`. Accessed 16/04/2024.

[14] William Ashby. *Design for a brain: The origin of adaptive behaviour*. Chapman & Hall, 1960. `https://doi.org/10.1037/11592-000`.

[15] MITRE ATT&CK. Mitre att&ck matrix for enterprise, 2015. Available at: `https://attack.mitre.org/matrices/enterprise/`. Accessed 16/12/2024.

[16] Petra Badke-Schaub and Eckart Frankenberger. Analysis of design projects. *Design Studies*, 20(5):465–480, 1999.

[17] Petra Badke-Schaub, Andre Neumann, Kristina Lauche, and Susan Mohammed. Mental models in design teams: a valid approach to performance in design collaboration? *CoDesign*, 3(1):5–20, 2007.

[18] Wayne Baker, N Nohria, and RG Eccles. The network organization in theory and practice. *Classics of Organization Theory*, 8:401, 1992.

[19] Adrian Baldwin, Yolanta Beres, Geoffrey B. Duggan, Marco Casassa Mont, Hilary Johnson, Chris Middup, and Simon Shiu. Economic methods and decision making by security professionals. In *Schneier B. (eds) Economics of Information Security and Privacy III*. Springer, New York, NY, 2012. 978-1-4614-1981-5.

[20] Adrian Baldwin, Tristan Caulfield, Marius-Constantin Ilau, and David Pym. Modelling organizational recovery. In *International Conference on Simulation Tools and Techniques*, pages 284–314. Springer, 2021.

[21] Y. Barlas. Formal aspects of model validity and validation in system dynamics. *System Dynamics Review*, 12:183–210, 1996.

[22] Yaman Barlas and Stanley Carpenter. Philosophical roots of model validation: two paradigms. *System Dynamics Review*, 6(2):148–166, 1990.

[23] Barry Barnes. *Scientific knowledge and sociological theory*. Routledge, 2013.

[24] Barry Barnes, David Bloor, and John Henry. *Scientific knowledge: A sociological analysis*. A&C Black, 1996.

[25] Gabriele Baroni and Till Francke. An effective strategy for combining variance- and distribution-based global sensitivity analysis. *Environmental Modelling & Software*, 134:104851, 2020.

[26] Laura Basco-Carrera, Andrew Warren, Eelco van Beek, Andreja Jonoski, and Alessio Giardino. Collaborative modelling or participatory modelling?

a framework for water resources management. *Environmental Modelling & Software*, 91:95–110, 2017.

[27] Adam Beautement, Robert Coles, Jonathan Griffin, Christos Ioannidis, Brian Monahan, David Pym, Angela Sasse, and Mike Wonham. Modelling the human and technological costs and benefits of usb memory stick security. *Managing information risk and the economics of security*, pages 141–163, 2009.

[28] Adam Beautement, M. Angela Sasse, and Mike Wonham. The compliance budget: Managing security behaviour in organisations. In *Proceedings of the 2008 New Security Paradigms Workshop*, NSPW '08, page 47–58, New York, NY, USA, 2008. Association for Computing Machinery.

[29] Claus Beisbart. How can computer simulations produce new knowledge? *European journal for philosophy of science*, 2(3):395–434, 2012.

[30] Claus Beisbart and John D Norton. Why monte carlo simulations are inferences and not experiments. *International Studies in the Philosophy of Science*, 26(4):403–422, 2012.

[31] James A Bell and James F Bell. System dynamics and scientific method. *Elements of the system dynamics method*, pages 3–22, 1980.

[32] James A Bell and Peter M Senge. Methods for enhancing refutability in system dynamics modeling. *TIMS Studies in the Management Sciences*, 14(1):61–73, 1980.

[33] Elena Bennett and Monika Zurek. Integrating epistemologies through scenarios. *Bridging scales and knowledge systems: Concepts and applications in ecosystem assessment*, pages 275–294, 2006.

[34] Y. Beres, Jonathan Griffin, S. Shiu, Max Heitman, David Markle, and Peter Ventura. Analysing the performance of security solutions to reduce vul-

nerability exposure window. *2008 Annual Computer Security Applications Conference (ACSAC)*, pages 33–42, 2008.

[35] Yolanta Beresnevichiene, D. Pym, and S. Shiu. Decision support for systems security investment. *2010 IEEE/IFIP Network Operations and Management Symposium Workshops*, pages 118–125, 2010.

[36] Binary Defense. Emotet Evolves With new Wi-Fi Spreader, 2020. Available at `https://www.binarydefense.com/emotet-evolves-with-new-wi-fi-spreader/`. Accessed 16/12/2024.

[37] G. Birtwistle. *Demos — discrete event modelling on Simula*. Macmillan, 1979. `https://doi.org/10.1007/978-1-4899-6685-8`.

[38] G. Birtwistle. Demos implementation guide and reference manual, 1981. Available at: `https://hdl.handle.net/1880/45701`. Accessed 17/12/2024.

[39] David Bloor. *Knowledge and social imagery*. University of Chicago press, 1991.

[40] David Bloor. Sociology of scientific knowledge. In *Handbook of epistemology*, pages 919–962. Springer, 2004.

[41] Sophie Blum, Raoul Koudijs, Ana Ozaki, and Samia Touileb. Learning horn envelopes via queries from language models. *International Journal of Approximate Reasoning*, 171:109026, 2024. Synergies between Machine Learning and Reasoning.

[42] Todd D Bowers. Towards a framework for multiparadigm multimethodologies. *Systems Research and Behavioral Science*, 28(5):537–552, 2011.

[43] Petar Boyanov. Educational exploiting the information resources and invading the security mechanisms of the operating system windows 7 with the exploit eternalblue and backdoor doublepulsar. *Association Scientific and Applied Research*, 14:34, 2018.

[44] Robert B Brandom. *Articulating reasons: An introduction to inferentialism*. Harvard University Press, 2001. ISBN: 9780674001589.

[45] David Braue. Global ransomware damage costs predicted to exceed $265 billion by 2031, 2022. Available at: `https://cybersecurityventures.com/global-ransomware-damage-costs-predicted-to-reach-250-billion-usd-by-2031/`. Accessed 17/12/2024.

[46] Richard P Brent. Recent progress and prospects for integer factorisation algorithms. In *International Computing and Combinatorics Conference*, pages 3–22. Springer, 2000.

[47] Calvin Brierley, Jamie Pont, Budi Arief, David J Barnes, and Julio Hernandez-Castro. Persistence in linux-based iot malware. In *Nordic Conference on Secure IT Systems*, pages 3–19. Springer, 2020.

[48] Stearns Broadhead. The contemporary cybercrime ecosystem: A multidisciplinary overview of the state of affairs and developments. *Computer Law & Security Review*, 34(6):1180–1196, 2018.

[49] Matt Bromiley. Sans 2019 incident response(ir) survey: It's time for a change, 2012. `https://www.sans.org/reading-room/whitepapers/analyst/2019-incident-response-ir-survey-time-change-39070`. Accessed 16/12/2024.

[50] Christopher Bronk and Eneken Tikk-Ringas. The cyber attack on saudi aramco. *Survival*, 55(2):81–96, 2013.

[51] Louis L Bucciarelli. An ethnographic perspective on engineering design. *Design studies*, 9(3):159–168, 1988.

[52] Louis L Bucciarelli. *Designing engineers*. MIT press, 1994. ISBN: 9780262522120.

[53] Louis L Bucciarelli. Between thought and object in engineering design. *Design studies*, 23(3):219–231, 2002.

[54] Otávio Bueno. Computer simulations: An inferential conception. *The Monist*, 97(3):378–398, 2014.

[55] Manuela Bujorianu, Tristan Caulfield, Marius-Constantin Ilau, and David Pym. Interfaces in ecosystems: concepts, form, and implementation. In *International Conference on Simulation Tools and Techniques*, pages 27–47. Springer, 2024.

[56] Hans Burkhardt, Barry Smith, et al. *Handbook of metaphysics and ontology*, volume 2. Philosophia Verlag Munich, 1991.

[57] Tom Burns and George M Stalker. Mechanistic and organic systems. *Classics of organizational theory*, pages 209–214, 1961.

[58] Michel Callon. Some elements of a sociology of translation: domestication of the scallops and the fishermen of st brieuc bay. *The sociological review*, 32(1_suppl):196–233, 1984.

[59] T. Caulfield and D. Pym. Improving security policy decisions with models. *IEEE Security and Privacy*, 13(5):34–41, Sept/Oct 2015.

[60] T. Caulfield, D. Pym, and J. Williams. Compositional security modelling: Structure, economics, and behaviour. *Lecture Notes in Computer Science*, 8533:233–245, 2014.

[61] Tristan Caulfield. Sysmodels julia package, 2021. Available at `https://github.com/tristanc/SysModels`. Accessed 16/12/2024.

[62] Tristan Caulfield, Marius-Constantin Ilau, and David Pym. Engineering ecosystem models: Semantics and pragmatics. In *International Conference on Simulation Tools and Techniques*, pages 236–258. Springer, 2021.

[63] Tristan Caulfield, Marius-Constantin Ilau, and David Pym. Meta-modelling for ecosystems security. In *International Conference on Simulation Tools and Techniques*, pages 259–283. Springer, 2021.

[64] Tristan Caulfield and David Pym. Modelling and simulating systems security policy. *EAI Endorsed Transactions on Security and Safety*, 3(8):e3–e3, 2016.

[65] Holmes Chad. Malware lateral movement: A primer, 2015. `https://www.mandiant.com/resources/malware-lateral-move`. Accessed 16/12/2024.

[66] S Sibi Chakkaravarthy, D Sangeetha, and V Vaidehi. A survey on malware analysis and mitigation techniques. *Computer Science Review*, 32:1–23, 2019.

[67] Osbourne Charlie. Updated kaseya ransomware attack faq: What we know now, 2021. `https://www.zdnet.com/article/updated-kaseya-ransomware-attack-faq-what-we-know-now/`. Accessed 16/12/2024.

[68] Peter Checkland and John Poulter. Soft systems methodology. *Systems approaches to making change: A practical guide*, pages 201–253, 2020.

[69] Ann Chervenak, Vivekenand Vellanki, and Zachary Kurmas. Protecting file systems: A survey of backup techniques. In *Joint NASA and IEEE Mass Storage Conference*, volume 99. Citeseer, 1998.

[70] CISA. Emotet Malware, 2017. Available at `https://us-cert.cisa.gov/ncas/alerts/aa20-280a`. Accessed 16/12/2024.

[71] CISA. Destructive malware targeting organizations in ukraine, 2022. Available at: `https://www.cisa.gov/uscert/ncas/alerts/aa22-057a`. Accessed 16/12/2024.

[72] Harry Collins, Robert Evans, and Mike Gorman. Trading zones and interactional expertise. *Studies in History and Philosophy of Science Part A*, 38(4):657–666, 2007.

[73] Harry M Collins and Robert Evans. The third wave of science studies: Studies of expertise and experience. *Social studies of science*, 32(2):235–296, 2002.

[74] M. Collinson, B. Monahan, and D. Pym. *A Discipline of Mathematical Systems Modelling*. College Publications, London, 2012. ISBN: 978-1-904987-50-5.

[75] M. Collinson and D. Pym. Algebra and logic for resource-based systems modelling. *Math. Structures in Comput. Sci.*, 19:959–1027, 2009.

[76] Matthew Collinson, Brian Monahan, and David Pym. Semantics for structured systems modelling and simulation. In *Proc. Simutools 2010*. ACM Digital Library, ISBN 78-963-9799-87-5, 2010.

[77] Jason A Colquitt, Jeffery A Lepine, and Michael J Wesson. Organizational behavior: Improving performance and commitment. *Organizational Behaviour. McGraw-Hill Education. www. mhhe. con*, 2015.

[78] Gabriele Contessa. Scientific representation, interpretation, and surrogative reasoning. *Philosophy of science*, 74(1):48–68, 2007.

[79] Byron Cook. Formal reasoning about the security of amazon web services. In *International Conference on Computer Aided Verification*, pages 38–47. Springer, 2018.

[80] S Crainer. Key management ideas: Thinkers that changed the management. *World Pretince Hall Books. New York*, 1993.

[81] Richard Creath. Every dogma has its day. *Erkenntnis*, 35(1):347–389, 1991.

[82] Richard Creath. The role of history in science. *Journal of the History of Biology*, 43(2):207–214, 2010.

[83] Nigel Cross and Anita Clayburn Cross. Observations of teamwork and social processes in design. *Design Studies*, 16(2):143–170, 1995. Analysing Design Activity.

[84] Richard M Cyert and E Grunberg. Assumption, prediction, and explanation in economics. *A behavioral theory of the firm*, 298:311, 1963.

[85] Newton CA Da Costa and Steven French. *Science and partial truth: A unitary approach to models and scientific reasoning*. Oxford University Press on Demand, 2003.

[86] Ole-Johan Dahl and Kristen Nygaard. Simula: an algol-based simulation language. *Communications of the ACM*, 9(9):671–678, 1966.

[87] Bruno Dallago. The organizational effect of the economic system. *Journal of Economic Issues*, 36(4):953–979, 2002.

[88] Dansimp, mjcaparas, vj̃mathew, MSFTTracyP, JoeDavies-MSFT, and alexbuckgit. What is ransomware, 2022. `https://docs.microsoft.com/en-us/security/compass/human-operated-ransomware`. Accessed 16/12/2024.

[89] A Philip Dawid. Bayes's theorem and weighing evidence by juries. In *PROCEEDINGS-BRITISH ACADEMY*, volume 113, pages 71–90. Citeseer, 2002.

[90] Xavier de Donato Rodriguez and Jesús Zamora Bonilla. Credibility, idealisation, and model building: An inferential approach. *Erkenntnis*, 70:101–118, 2009.

[91] R. de Simone. Higher-level synchronising devices in Meije-SCCS. *Theor. Comput. Sci.*, 37:245–267, 1985.

[92] Zakariya Dehlawi and Norah Abokhodair. Saudi arabia's response to cyber conflict: A case study of the shamoon malware incident. In *2013 IEEE International Conference on Intelligence and Security Informatics*, pages 73–75. IEEE, 2013.

[93] Albesë Demjaha. *Co-design and modelling of security policy for cultural and behavioural aspects of security in organisations*. PhD thesis, UCL (University College London), 2023. Available at: `https://discovery.ucl.ac.uk/id/eprint/10173397`. Accessed 16/12/2024.

[94] Albesë Demjaha, David Pym, and Tristan Caulfield. Found in translation: Co-design for security modelling. In *International Workshop on Socio-Technical Aspects in Security*, pages 108–128. Springer, 2021.

[95] Demos2k. `sourceforge.net`. The Demos2k distribution is part of the Seymour distribution, 2007.

[96] Phoebus J Dhrymes, E Philip Howrey, Saul H Hymans, Jan Kmenta, Edward E Leamer, Richard E Quandt, James B Ramsey, Harold T Shapiro, and Victor Zarnowitz. Criteria for evaluation of econometric models. In *Annals of Economic and Social Measurement, Volume 1, number 3*, pages 291–324. NBER, 1972.

[97] Oxford English Dictionary. Heterogeneous, 2024. Accessed 2 September 2024.

[98] Keith S Donnellan. Reference and definite descriptions. *The philosophical review*, 75(3):281–304, 1966.

[99] Juan Manuel Durán. What is a simulation model? *Minds and Machines*, pages 1–23, 2020.

[100] Robert G Eccles and Dwight B Crane. Managing through networks in investment banking. *California management review*, 30(1):176–195, 1987.

[101] Roger Eckhardt. Stan ulam, john von neumann, and the monte carlo method. *Los Alamos Science*, 15(131-136):30, 1987.

[102] Kost Edward. What is ransomware as a service (raas)? the dangerous threat to world security, 2022. `https://www.upguard.com/blog/what-is-ransomware-as-a-service`. Accessed 16/12/2024.

[103] Ernest Ehlers and Simeon Potter. Phase, 2024. Available at: `https://www.britannica.com/science/phase-state-of-matter`. Accessed 2 September 2024.

[104] Walter M Elsasser. Outline of a theory of cellular heterogeneity. *Proceedings of the National Academy of Sciences*, 81(16):5126–5129, 1984.

[105] Eric Parizo. Maersk CISO Says NotPeyta Devastated Several Unnamed US firms, 2019. Available at: https://www.darkreading.com/threat-intelligence/maersk-ciso-says-notpeyta-devastated-several-unnamed-us-firms/a/d-id/1336558?page_number=2. Accessed 16/12/2024.

[106] Jose Miguel Esparza. Understanding the credential theft lifecycle. *Computer Fraud and Security*, 2019(2):6–9, 2019.

[107] Europol. Internet organised crime threat assessment, 2020. https://www.europol.europa.eu/cms/sites/default/files/documents/internet_organised_crime_threat_assessment_iocta_2020.pdf. Accessed 16/12/2024.

[108] F-Secure. Trojan:androidkoler, 2013. https://www.f-secure.com/v-descs/trojan_android_koler.shtml. Accessed 16/12/2024.

[109] Facebook. Open-sourcing Facebook Infer, 2015. Available at: https://engineering.fb.com/2015/06/11/developer-tools/open-sourcing -facebook-infer-identify-bugs-before-you-ship/. Accessed 16/12/2024.

[110] FBI. 2019 internet crime report, 2020. https://www.fbi.gov/news/stories/2019-internet-crime-report-released-021120. Accessed 16/12/2024.

[111] FBI. Cisa-fbi guidance for msps and their customers affected by the kaseya vsa supply-chain ransomware, 2021. https://www.cisa.gov/uscert/ncas/current-activity/2021/07/04/cisa-fbi-guidance-msps-and-their-customers-affected-kaseya-vsa. Accessed 16/12/2024.

[112] Paul Feyerabend. Putnam on incommensurability. *The British Journal for the Philosophy of Science*, 38(1):75–81, 1987.

[113] Jay Wright Forrester. Industrial dynamics. *Journal of the Operational Research Society*, 48(10):1037–1041, 1997.

[114] Charles François. Systemics and cybernetics in a historical perspective. *Systems Research and Behavioral Science: The Official Journal of the International Federation for Systems Research*, 16(3):203–219, 1999.

[115] Gottlob Frege. Über sinn und bedeutung. *Zeitschrift für Philosophie und philosophische Kritik*, 100:25–50, 1892.

[116] Gottlob Frege. Der gedanke. eine logische untersuchung. *Wittgenstein Studien*, 4(2), 1997.

[117] Marjolaine Frésard and Jean Boncoeur. Controlling the biological invasion of a commercial fishery by a space competitor: a bioeconomic model with reference to the bay of st-brieuc scallop fishery. *Agricultural and Resource Economics Review*, 35(1):78–97, 2006.

[118] Michael Friedman. Explanation and scientific understanding. *the Journal of Philosophy*, 71(1):5–19, 1974.

[119] Michael Friedman. *Dynamics of reason*. Csli Publications Stanford, 2001.

[120] Milton Friedman. The methodology of positive economics. *Essays in Positive Economics*, 1953.

[121] Roman Frigg and James Nguyen. The fiction view of models reloaded. *The Monist*, 99(3):225–242, 2016.

[122] Roman Frigg and James Nguyen. Models and representation. *Springer handbook of model-based science*, pages 49–102, 2017.

[123] Steven D Galbraith and Pierrick Gaudry. Recent progress on the elliptic curve discrete logarithm problem. *Designs, Codes and Cryptography*, 78(1):51–72, 2016.

[124] Peter Galison. *Image and logic: A material culture of microphysics*. University of Chicago Press, 1997.

[125] Peter Galison. *Trading Zone: Coordinating Action and Belief (1998 abridgment)*, pages 137–160. Routledge, 1999.

[126] Peter Louis Galison and S D'Agostino. *How experiments end*, volume 88. University of Chicago Press Chicago, 1987.

[127] D. Galmiche, D. Méry, and D. Pym. The Semantics of BI and Resource Tableaux. *Math. Structures in Comput. Sci.*, 15:1033–1088, 2005.

[128] Morgan Gareth. *Images of Organisation*. Sage Publications London, 2019. ISBN: 9781412939799.

[129] Carl A Gibson and Michael Tarrant. A'conceptual models' approach to organisational resilience. *Australian Journal of Emergency Management, The*, 25(2):6–12, 2010.

[130] Ronald N Giere. How models are used to represent reality. *Philosophy of science*, 71(5):742–752, 2004.

[131] Ronald N Giere. *Scientific perspectivism*. University of Chicago press, 2019.

[132] S. Gilmore and J. Hillston. The PEPA Workbench: A Tool to Support a Process Algebra-based Approach to Performance Modelling. In *Proceedings of the Seventh International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, number 794 in Lecture Notes in Computer Science, pages 352–368. Springer-Verlag, 1994.

[133] Zane Gittins and Michael Soltys. Malware persistence mechanisms. *Procedia Computer Science*, 176:88–97, 2020.

[134] UK Government. Cyber security breaches survey 2020, 2020. https://www.gov.uk/government/statistics/cyber-security-breaches-survey-2020/cyber-security-breaches-survey-2020. Accessed 16/12/2024.

[135] Patrick Grim and Nicholas Rescher. How modeling can go wrong: Some cautions and caveats on the use of models. *Philosophy and Technology*, 26(1):75–80, 2013.

[136] INSIKT GROUP. Overview of the 9 distinct data wipers used in the ukraine war, 2022. https://www.recordedfuture.com/overview-9-district-data-wipers-ukraine-war. Accessed 16/12/2024.

[137] Ranjay Gulati, Nitin Nohria, and Akbar Zaheer. Strategic networks. *Strategic management journal*, 21(3):203–215, 2000.

[138] Alex Hearn. Hackers publish private photos from cosmetic surgery clinic, 2017. https://www.theguardian.com/technology/2017/may/31/hackers-publish-private-photos-cosmetic-surgery-clinic-bitcoin-ransom-payments. Accessed 16/12/2024.

[139] M. Hennessy and G. Plotkin. On observing nondeterminism and concurrency. In *Proceedings of the 7th ICALP*, volume 85 of *Lecture Notes in Computer Science*, pages 299–309. Springer-Verlag, 1980.

[140] Hewlett-Packard Laboratories. Security Analytics, 2011. Available at https://www.hpl.hp.com/news/2011/oct-dec/security_analytics.html. Accessed 16/12/2024.

[141] Derek K Hitchins. *Putting systems to work*, volume 325. Wiley Chichester, 1992. ISBN: 9780598030337.

[142] C. A. R. Hoare. *Communicating Sequential Processes*. Prentice-Hall International, London, 1985. ISBN: 9780131532717.

[143] Allegra Hobbs. The colonial pipeline hack: Exposing vulnerabilities in us cybersecurity. In *SAGE Business Cases*. SAGE Publications: SAGE Business Cases Originals, 2021.

[144] Kjell Jørgen Hole. Robustness to malware reinfections. In *Anti-fragile ICT Systems*, pages 93–98. Springer, 2016.

[145] Michael Hopkins and Ali Dehghantanha. Exploit kits: The production line of the cybercrime economy? In *2015 second international conference on Information Security and Cyber Forensics (InfoSec)*, pages 23–27. IEEE, 2015.

[146] Frank Houghton. Cybersecurity, ransomware attacks and health: Exploring the public health implications of the recent cyberattack on ireland's health service. *Medicina Internacia Revuo*, 29(116):160–163, 2021.

[147] Peter William House, John McLeod, and John McLeod. *Large-scale models for policy evaluation*. Wiley New York, 1977. ISBN: 9780471415558.

[148] Paul Hoyningen-Huene. *Reconstructing scientific revolutions: Thomas S. Kuhn's philosophy of science*. University of Chicago Press, 1993. ISBN: 9780226355511.

[149] HP. Hp surerecover, 2021. Available at: `https://www8.hp.com/h20195/v2/GetPDF.aspx/4AA7-4556ENW.pdf`. Accessed 16/12/2024.

[150] Peter J Hugill. *Global communications since 1844: Geopolitics and technology*. JHU Press, 1999. ISBN: 9780801860393.

[151] Marius-Constantin Ilau, Tristan Caulfield, and David Pym. Modelling and simulating organizational ransomware recovery: structure, methodology, and decisions. Under Review. Available at: `http://www0.cs.ucl.ac.uk/staff/D.Pym/HP_Security_Recovery_Journal.pdf`, 2023.

[152] Marius-Constantin Ilau, Tristan Caulfield, and David Pym. Co-designing heterogeneous models: a distributed systems approach. *arXiv preprint arXiv:2407.07656*, 2024.

[153] ANTON Iliev, NIKOLAY Kyurkchiev, ASEN Rahnev, and TODORKA Terzieva. Some new approaches for modelling large-scale worm spreading on the internet. ii. *Neural, Parallel, and Scientific Computations*, 27(1):23–34, 2019.

[154] Philip G. Inglesant and M. Angela Sasse. The true cost of unusable password policies: Password use in the wild. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, page 383–392, New York, NY, USA, 2010. Association for Computing Machinery.

[155] Christos Ioannidis, David Pym, Julian Williams, and Iffat Gheyas. Resilience in information stewardship. *European journal of operational research*, 274(2):638–653, 2019.

[156] S.S. Ishtiaq and P. O'Hearn. BI as an assertion language for mutable data structures. In *Proc. POPL*, 2001.

[157] Peter Jachim, Filipo Sharevski, and Paige Treebridge. Trollhunter [evader]: Automated detection [evasion] of twitter trolls during the covid-19 pandemic. In *Proceedings of the 2008 New Security Paradigms Workshop*, NSPW '20, page 59–75, New York, NY, USA, 2020. Association for Computing Machinery.

[158] Pagliery Jose. The inside story of the biggest hack in history, 2015. Available at: `https://money.cnn.com/2015/08/05/technology/aramco-hack/index.html`. Accessed 16/12/2024.

[159] Fabian Kaczmarczyck, Bernhard Grill, Luca Invernizzi, Jennifer Pullman, Cecilia M. Procopiuc, David Tao, Borbala Benko, and Elie Bursztein. Spotlight: Malware lead generation at scale. In *Annual Computer Security Applications Conference*, ACSAC '20, page 17–27, New York, NY, USA, 2020. Association for Computing Machinery.

[160] David Kaplan. Afterthoughts. In Joseph Almog, John Perry, and Howard

Wettstein, editors, *Themes From Kaplan*, pages 565–614. Oxford University Press, 1989.

[161] David Kaplan. An essay on the semantics, logic, metaphysics, and epistemology of. *Themes from kaplan*, page 481, 1989.

[162] David Kaplan. Words. *Proceedings of the Aristotelian society, supplementary volumes*, 64:93–119, 1990.

[163] Christos Karapapas, Iakovos Pittaras, Nikos Fotiou, and George C Polyzos. Ransomware as a service using smart contracts and ipfs. *arXiv preprint arXiv:2003.04426*, 2020.

[164] Vasileios Karyotis and M.H.R. Khouzani. Chapter 3 - early malware diffusion modeling methodologies. In Vasileios Karyotis and M.H.R. Khouzani, editors, *Malware Diffusion Models for Wireless Complex Networks*, pages 39–60. Morgan Kaufmann, Boston, 2016.

[165] Terrence P Kee and Pierre-Alain Monnard. Chemical systems, chemical contiguity and the emergence of life. *Beilstein Journal of Organic Chemistry*, 13(1):1551–1563, 2017.

[166] Noël Keijzer. The new generation of ransomware: an in depth study of ransomware-as-a-service. Master's thesis, University of Twente, 2020.

[167] Udo Kelle. " emergence" vs." forcing" of empirical data? a crucial problem of" grounded theory" reconsidered. *Historical Social Research/Historische Sozialforschung. Supplement*, pages 133–156, 2007.

[168] Bruno Kelpsas and Adam Nelson. Ransomware in hospitals: what providers will inevitably face when attacked. *The Journal of Medical Practice Management*, 32(1):67–70, 2016.

[169] Quintin Kerns, Bryson Payne, and Tamirat Abegaz. Double-extortion ransomware: A technical analysis of maze ransomware. In *Proceedings of the Future Technologies Conference*, pages 82–94. Springer, 2021.

[170] Marc-Olivier Killijian, Ludovic Courtès, and David Powell. A survey of cooperative backup mechanisms, 2006.

[171] Charles Kittel and Paul McEuen. *Introduction to solid state physics*. John Wiley & Sons, 2018. ISBN: 9780471415268.

[172] Maaike Kleinsmann and Rianne Valkenburg. Barriers and enablers for creating shared understanding in co-design projects. *Design studies*, 29(4):369–386, 2008.

[173] Andrey Kolmogorov. Sulla determinazione empirica di una lgge di distribuzione. *Inst. Ital. Attuari, Giorn.*, 4:83–91, 1933.

[174] Alfred Korzybski. Science and sanity: An introduction to non-aristotelian systems and general semantics, 1958. ISBN: 9780937298015.

[175] Erika Kraemer-Mbula, Puay Tang, and Howard Rush. The cybercrime ecosystem: Online innovation in the shadows? *Technological Forecasting and Social Change*, 80(3):541–555, 2013.

[176] Patric Kral. The incident handlers handbook, 2012. Available at: `https://www.sans.org/reading-room/whitepapers/incident/incident-handlers-handbook-33901`. Accessed 17/12/2024.

[177] Brian Krebs. Inside a reveton ransomware operation, 2012. Available at: `https://krebsonsecurity.com/2012/08/inside-a-reveton-ransomware-operation/`. Accessed 17/12/2024.

[178] Saul Kripke. Speaker's reference and semantic reference, ludlow p.(red.). *Reading in the Philosophy of Language*, pages 383–414, 1977.

[179] Saul A Kripke. A puzzle about belief. In *Meaning and Use: Papers Presented at the Second Jerusalem Philosophical Encounter April 1976*, pages 239–283. Springer, 1979.

[180] Saul A Kripke et al. *Naming and necessity*, volume 217. Springer, 1980. ISBN: 9780631128014.

[181] Nir Kshetri and Jeffrey Voas. Do crypto-currencies fuel ransomware? *IT professional*, 19(5):11–15, 2017.

[182] Thomas S Kuhn. *The structure of scientific revolutions*. University of Chicago press, 2012. ISBN: 9780226458083.

[183] Sari Kujala. User involvement: a review of the benefits and challenges. *Behaviour & information technology*, 22(1):1–16, 2003.

[184] Jaakko Kuorikoski. Factivity, pluralism, and the inferential account of scientific understanding. In *Scientific understanding and representation*, pages 217–233. Routledge, 2022.

[185] Jaakko Kuorikoski and Samuli Reijula. Making it count. an inferentialist account of computer simulation, 2019. Available at: `https://api.semanticscholar.org/CorpusID:241488900`.

[186] Jaakko Kuorikoski and Petri Ylikoski. External representations and scientific understanding. *Synthese*, 192:3817–3837, 2015.

[187] Martin Kusch. *Knowledge by agreement: The programme of communitarian epistemology*. OUP Oxford, 2002.

[188] Martin Kusch. *Psychologism: The sociology of philosophical knowledge*. Routledge, 2005.

[189] Karina Kwapiszewska. Physicochemical perspective of biological heterogeneity. *ACS Physical Chemistry Au*, 4(4):314–321, 2024.

[190] Tony Lawson and Joan Garrod. *Dictionary of sociology*. Routledge, 2012. ISBN: 9781579582913.

[191] Eryk Lewinson. Violin plots explained, 2019. Available at: `https://towardsdatascience.com/violin-plots-explained-fb1d115e023d`. Accessed 17/12/2024.

[192] Elisabeth A Lloyd. *The structure and confirmation of evolutionary theory*. Princeton University Press, 2021. ISBN: 9780313255632.

[193] Nancy A. Lynch and Mark R. Tuttle. An introduction to input/output automata. *CWI Quarterly*, 2:219–246, 1989.

[194] Morvin Savio Martis. Validation of simulation based models: a theoretical outlook. *The electronic journal of business research methods*, 4(1):39–46, 2006.

[195] Andreu Mas-Colell, Michael Dennis Whinston, Jerry R Green, et al. *Microeconomic theory*, volume 1. Oxford university press New York, 1995.

[196] J.H. McColl. *Probability*. Elsevier: Butterworth–Heinemann, 1995. ISBN: 9780340614266.

[197] Kevin S McCurley. The discrete logarithm problem. In *Proc. of Symp. in Applied Math*, volume 42, pages 49–74. USA, 1990.

[198] Per Håkon Meland, Yara Fareed Fahmy Bayoumy, and Guttorm Sindre. The ransomware-as-a-service economy within the darknet. *Computers & Security*, 92:101762, 2020.

[199] Daniele Micciancio. The hardness of the closest vector problem with preprocessing. *IEEE Transactions on Information Theory*, 47(3):1212–1215, 2001.

[200] Noor Michael, Jaron Mink, Jason Liu, Sneha Gaur, Wajih Ul Hassan, and Adam Bates. On the forensic validity of approximated audit logs. In *Annual Computer Security Applications Conference*, ACSAC ’20, page 189–202, New York, NY, USA, 2020. Association for Computing Machinery.

[201] Microsoft. Windows recovery environment, 2017. Available at: https://docs.microsoft.com/en-us/windows-hardware/manufacture/desktop/windows-recovery-environment--windows-re--technical-reference. Accessed 17/12/2024.

[202] Microsoft. *Apply features and settings on your devices using device profiles in Microsoft Intune*. Microsoft, 2020. Available at: https://docs.microsoft.com/en-us/mem/intune/configuration/device-profiles. Accessed 17/12/2024.

[203] Microsoft. *Deploy Windows 10 using PXE and Configuration Manager*. Microsoft, 2021. Available at: https://docs.microsoft.com/en-us/windows/deployment/deploy-windows-cm/deploy-windows-10-using-pxe-and-configuration-manager. Accessed 17/12/2024.

[204] Microsoft. *Microsoft Deployment Toolkit*. Microsoft, 2021. Available at: https://docs.microsoft.com/en-us/windows/deployment/deploy-windows-mdt/get-started-with-the-microsoft-deployment-toolkit. Accessed 17/12/2024.

[205] Microsoft. *Overview of Windows Autopilot*. Microsoft, 2021. Available at: https://docs.microsoft.com/en-us/mem/autopilot/windows-autopilot. Accessed 17/12/2024.

[206] Microsoft. Destructive malware targeting ukrainian organizations, 2022. Available at: https://www.microsoft.com/security/blog/2022/01/15/destructive-malware-targeting-ukrainian-organizations/. Accessed 16/12/2024.

[207] Jason Milletary. Citadel trojan malware analysis, 2012. Available at: https://botnetlegalnotice.com/citadel/files/patel_decl_ex20.pdf. Accessed 16/12/2024.

[208] R. Milner. *A Calculus of Communicating Systems*, volume 92 of *LNCS*. Springer Verlag, 1980.

[209] R. Milner. *Communication and Concurrency*. Prentice Hall, New York, 1989. ISBN: 9780131150072.

[210] R. Milner. *Communicating and mobile systems: the π-calculus*. Cambridge University Press, 1999. ISBN: 9780521658690.

[211] Robin Milner. Calculi for synchrony and asynchrony. *Theoretical computer science*, 25(3):267–310, 1983.

[212] Robin Milner. Bigraphs as a model for mobile interaction (invited paper). In *ICGT 2002, First International Conference on Graph Transformation*, volume 2505 of *LNCS*, pages 8–13. Springer, 2002.

[213] Michael Mimoso. Europol takes down ransomware gang in spain, uae, 2013. Available at: `https://threatpost.com/europol-takes-down-ransomware-gang-spain-uae-021413/77529/`. Accessed 16/12/2024.

[214] John Mingers and John Brocklesby. Multimethodology: Towards a framework for mixing methodologies. *Omega*, 25(5):489–509, 1997.

[215] Henry Mintzberg. The structuring of organizations. *Englewood Cliffs*, 330, 1979.

[216] Bimal Kumar Mishra and Navnit Jha. Seiqrs model for the transmission of malicious objects in computer network. *Applied Mathematical Modelling*, 34(3):710–715, 2010.

[217] Ian I Mitroff. Fundamental issues in the simulation of human behavior: a case in the strategy of behavioral science. *Management Science*, 15(12):B–635, 1969.

[218] M Moran Stritch, M Winterburn, and F Houghton. The conti ransomware attack on healthcare in ireland: Exploring the impacts of a cybersecurity breach from a nursing perspective. *Canadian Journal of Nursing Informatics*, 16(3-4), 2021.

[219] James M Morgan and Jeffrey K Liker. *The Toyota product development system: integrating people, process, and technology*. Productivity press, 2020. ISBN: 9781563272820.

[220] Flavia Muggianu, Alfredo Benso, Roberta Bardini, E Hu, Gianfranco Politano, and Stefano Di Carlo. Modeling biological complexity using biology system description language (bisdl). In *2018 IEEE international conference on bioinformatics and biomedicine (BIBM)*, pages 713–717. IEEE, 2018.

[221] Michael J Muller and Allison Druin. Participatory design: The third space in human–computer interaction. In *Human Computer Interaction Handbook*, pages 1125–1153. CRC Press, 2012.

[222] Thomas H. Naylor, J. M. Finger, James L. McKenney, William E. Schrank, and Charles C. Holt. Verification of computer simulation models. *Management Science*, 14(2):B92–B106, 1967.

[223] Hannah T Neprash, Claire C McGlave, Dori A Cross, Beth A Virnig, Michael A Puskarich, Jared D Huling, Alan Z Rozenshtein, and Sayeh S Nikpay. Trends in ransomware attacks on us hospitals, clinics, and other health care delivery organizations, 2016-2021. In *JAMA Health Forum*, volume 3, pages e224873–e224873. American Medical Association, 2022.

[224] James Nguyen and Roman Frigg. *Scientific Representation*. Cambridge University Press, 2022. ISBN: 9781009009157.

[225] William D Nordhaus. World dynamics: measurement without data. *The Economic Journal*, 83(332):1156–1183, 1973.

[226] Jordan Nuce, Jeremy Kennelly, Kimberly Goody, Andrew Moore, Alyssa Rahman, Matt Williams, Brendan McKeague, and Jared Wilson. Shining a light on darkside ransomware operations, 2021. Available at: `https://cloud.google.com/blog/topics/threat-intelligence/shining-a-light-on-darkside-ransomware-operations/`. Accessed 16/12/2024.

[227] P. O'Hearn. Resources, concurrency, and local reasoning. *Theor. Comput. Sci.*, 375(1–3):271–307, May 2007.

[228] Peter W O'Hearn and David J Pym. The logic of bunched implications. *Bulletin of Symbolic Logic*, 5(2):215–244, 1999.

[229] Peter W. O'Hearn, John C. Reynolds, and Hongseok Yang. Local reasoning about programs that alter data structures. In *Proceedings of the 15th International Workshop on Computer Science Logic*, CSL '01, page 1–19, Berlin, Heidelberg, 2001. Springer-Verlag.

[230] P.W. O'Hearn and D.J. Pym. The logic of bunched implications. *Bulletin of Symbolic Logic*, 5(2):215–244, 1999.

[231] Philip O'Kane, Sakir Sezer, and Domhnall Carlin. Evolution of ransomware. *Iet Networks*, 7(5):321–327, 2018.

[232] Alessandro Oltramari. Representation, presentation and conceptual schemas. In *XIII Trieste Symposium on Perception and Cognition, Trieste Italy*, 2005.

[233] Harun Oz, Ahmet Aris, Albert Levi, and A Selcuk Uluagac. A survey on ransomware: Evolution, taxonomy, and defense solutions. *ACM Computing Surveys (CSUR)*, 2021.

[234] Robert A Pape. *Bombing to win.* Cornell University Press, 2014. ISBN: 9780801483110.

[235] Talcott Parsons. *The social system.* Routledge, 2013. ISBN: 9780710019318.

[236] Barbara Partee et al. Compositionality. *Varieties of formal semantics*, 3:281–311, 1984.

[237] Charles Sanders Peirce. The fixation of belief. *Writings of Charles S. Peirce: A chronological edition*, 3:242–257, 1986.

[238] Charles Sanders Peirce. How to make our ideas clear. *The nature of truth: Classic and contemporary perspectives*, pages 193–209, 2001.

[239] Francesca Pianosi and Thorsten Wagener. A simple and efficient method for global sensitivity analysis based on cumulative distribution functions. *Environmental Modelling & Software*, 67:1–11, 2015.

[240] Francesca Pianosi and Thorsten Wagener. Distribution-based sensitivity analysis from a generic input-output sample. *Environmental Modelling & Software*, 108:197–207, 2018.

[241] Gabriele Piccoli and Federico Pigni. *Information systems for managers: with cases*. Prospect Press, Inc., 2019. ISBN: 9781943153053.

[242] Andrew Pickering. *Science as practice and culture*. University of Chicago press, 2010.

[243] Andrew Pickering et al. From science as knowledge to science as practice. *Science as practice and culture*, 4:9780226668208–002, 1992.

[244] G. D. Plotkin. A structural approach to operational semantics. Technical Report DAIMI FN-19, Computer Science Dept., Aarhus University, Aarhus, Denmark, 1981.

[245] Karl Popper. *The logic of scientific discovery*. Routledge, 2005. ISBN: 9780091117214.

[246] Roger S Pressman. *Software engineering: a practitioner's approach*. Palgrave macmillan, 2005. ISBN: 9789339212087.

[247] Hilary Putnam. The meaning of" meaning". *Minnesota Studies in the Philosophy of Science*, 7, 1975.

[248] David Pym. Resource semantics: Logic as a modelling technology. *ACM SIGLOG News*, 6(2):5–41, April 2019.

[249] David Pym and Simon Shiu. Security analytics: Bringing science to security management. *IISP Pulse*, 4(Summer):12–13, 2010.

[250] D.J. Pym, P.W. O'Hearn, and H. Yang. Possible Worlds and Resources: The Semantics of BI. *Theor. Comput. Sci.*, 315(1):257–305, 2004.

[251] Michael J Radzicki. Institutional dynamics: An extension of the institutionalist approach to socioeconomic analysis. *Journal of Economic Issues*, 22(3):633–665, 1988.

[252] Nicholas Ralph, Melanie Birks, and Ysanne Chapman. The methodological dynamism of grounded theory. *International Journal of Qualitative Methods*, 14(4):1609406915611576, 2015.

[253] Satter Raphael. Up to 1,500 businesses affected by ransomware attack, u.s. firm's ceo says, 2021. Available at: `https://www.reuters.com/technology/hackers-demand-70-million-liberate-data-held-by-companies-hit-mass-cyberattack-2021-07-05/`. Accessed 16/12/2024.

[254] Barbara J Regeer and Joske FG Bunders. Knowledge co-creation: Interaction between science and society. *A Transdisciplinary Approach to Complex Societal Issues. Den Haag: Advisory Council for Research on Spatial Planning, Nature and the Environment/Consultative Committee of Sector Councils in the Netherlands [RMNO/COS]*, 2009.

[255] John Reynolds. Separation logic: A logic for shared mutable data structures. In *Proc. LICS*, 2002.

[256] John C. Reynolds. Separation logic: A logic for shared mutable data structures. In *Proceedings of the 17th Annual IEEE Symposium on Logic in Computer Science*, LICS '02, pages 55–74, Washington, DC, USA, 2002. IEEE Computer Society.

[257] Chris Rhodes and Andrew Bettany. *Windows Installation and Update Troubleshooting*. Apress, 2016. ISBN: 9781484218266.

[258] George P Richardson and Alexander L Pugh III. Introduction to system dynamics modeling with dynamo. *Journal of the Operational Research Society*, 48(11):1146–1146, 1997.

[259] Ronny Richardson and Max M North. Ransomware: Evolution, mitigation and prevention. *International Management Review*, 13(1):10, 2017.

[260] Richard Rorty et al. *Contingency, irony, and solidarity*, volume 5. Cambridge university press Cambridge, 1989.

[261] JS Sagoo and JT Boardman. Towards the formalisation of soft systems models using petri net theory. *IEE Proceedings-Control Theory and Applications*, 145(5):463–471, 1998.

[262] Muhammad Sahimi. *Heterogeneous Materials I: Linear transport and optical properties*, volume 22. Springer Science & Business Media, 2003.

[263] Ameera Salem Abdouli, Joonsang Baek, and Chan Yeob Yeun. Survey on computationally hard problems and their applications to cryptography. In *2011 International Conference for Internet Technology and Secured Transactions*, pages 46–52, 2011.

[264] Elizabeth B-N Sanders and Pieter Jan Stappers. Co-creation and the new landscapes of design. *Co-design*, 4(1):5–18, 2008.

[265] Thomas C Schelling. *Arms and influence*. Yale University Press, 2008. ISBN: 9780300008821.

[266] Klaus-Dieter Schewe and Bernhard Thalheim. The co-design approach to web information systems development. *International Journal of Web Information Systems*, 1(1):5–14, 2005.

[267] Klaus-Dieter Schewe, Bernhard Thalheim, Klaus-Dieter Schewe, and Bernhard Thalheim. The co-design framework. *Design and Development of Web Information Systems*, pages 3–30, 2019.

[268] Klaus-Dieter Schewe, Bernhard Thalheim, Klaus-Dieter Schewe, and Bernhard Thalheim. The co-design methodology. *Design and Development of Web Information Systems*, pages 377–427, 2019.

[269] Wilfrid Sellars. Empiricism and the philosophy of mind, 1956.

[270] Wilfrid Sellars. Philosophy and the scientific image of man. *Frontiers of science and philosophy*, 1:1–40, 1962.

[271] Peter M Senge and Jay W Forrester. Tests for building confidence in system dynamics models. *System dynamics, TIMS studies in management sciences*, 14:209–228, 1980.

[272] Shaun Hurley and Karan Sood. NotPetya Technical Analysis Part II: Further Findings and Potential for MBR Recovery, 2017. Available at: `https://www.crowdstrike.com/blog/petrwrap-technical-analysis-part-2-further-findings-and-potential-for-mbr-recovery/`. Accessed 16/12/2024.

[273] Lars Skyttner. *General systems theory: Problems, perspectives, practice*. World scientific, 2005. ISBN: 9789812564672.

[274] Nikolai V Smirnov. On the estimation of the discrepancy between empirical curves of distribution for two independent samples. *Bull. Math. Univ. Moscou*, 2(2):3–14, 1939.

[275] Joseph D Sneed. Philosophical problems in the empirical science of science: A formal approach. *Erkenntnis*, pages 115–146, 1976.

[276] Sophos. The state of ransomware 2024, 2024. Avaialble at: `https://assets.sophos.com/X24WTUEQ/at/9brgj5n44hqvgsp5f5bqcps/sophos-state-of-ransomware-2024-wp.pdf`. Accessed 17/12/2024.

[277] Clay Spinuzzi. The methodology of participatory design. *Technical communication*, 52(2):163–174, 2005.

[278] Susan Leigh Star and James R Griesemer. Institutional ecology,translations' and boundary objects: Amateurs and professionals in berkeley's museum of vertebrate zoology, 1907-39. *Social studies of science*, 19(3):387–420, 1989.

[279] Marc Steen. Co-design as a process of joint inquiry and imagination. *Design issues*, 29(2):16–28, 2013.

[280] John D Sterman. The growth of knowledge: Testing a theory of scientific revolutions with a formal model. *Technological Forecasting and Social Change*, 28(2):93–122, 1985.

[281] Colin Stirling. *Modal and Temporal Properties of Processes*. Springer Verlag, 2001. ISBN: 9781441931535.

[282] Mauricio Suárez. An inferential conception of scientific representation. *Philosophy of science*, 71(5):767–779, 2004.

[283] Mauricio Suárez. Scientific representation. *Philosophy Compass*, 5(1):91–101, 2010.

[284] Mauricio Suárez. Deflationary representation, inference, and practice. *Studies in History and Philosophy of Science Part A*, 49:36–47, 2015.

[285] Łukasz Sułkowski et al. Types of metaphors of organisation. *Journal of Intercultural Management*, 3(2):221–227, 2011.

[286] Patrick Suppes. Models of data. In *Studies in logic and the foundations of mathematics*, volume 44, pages 252–261. Elsevier, 1966.

[287] Gabor Szappanos. Inside the blackhole, 2012. Available at: `https://www.virusbulletin.com/virusbulletin/2012/10/inside-black-hole-part-1`. Accessed 17/12/2024.

[288] Threat Intelligence Team. Reveton ransomware has dangerously evolved, 2014. Available at: `https://blog.avast.com/2014/08/19/reveton-ransomware-has-dangerously-evolved/`. Accessed 17/12/2024.

[289] Bernhard Thalheim. Model suites for multi-layered database modelling. In *Information Modelling and Knowledge Bases XXI*, pages 116–134. IOS Press, 2010.

[290] Bernhard Thalheim. Towards a theory of conceptual modelling. *J. Univers. Comput. Sci.*, 16(20):3102–3137, 2010.

[291] Samuel Tweneboah-Kodua, Francis Atsu, and William Buchanan. Impact of cyberattacks on stock performance: a comparative study. *Information & Computer Security*, 2018.

[292] Thomas Uebel. *Empiricism at the Crossroads: The Vienna Circle's Protocol-Sentence Debate Revisited*, volume 4. Open Court, 2015.

[293] Thomas E Uebel. *Overcoming logical positivism from within: The emergence of Neurath's naturalism in the Vienna Circle's protocol sentence debate*, volume 17. Brill, 2021.

[294] Rusydi Umar, Imam Riadi, and Ridho Surya Kusuma. Analysis of conti ransomware attack on computer network with live forensic method. *IJID (International Journal on Informatics for Development)*, 10(1):53–61, 2021.

[295] Johan van Benthem. *Logical Dynamics of Information and Interaction*. Cambridge University Press, 2011.

[296] Liselotte S van Boven, Renske WJ Kusters, Derrick Tin, Frits HM van Osch, Harald De Cauwer, Linsay Ketelings, Madhura Rao, Christian Dameff, and Dennis G Barten. Hacking acute care: a qualitative study on the health care impacts of ransomware attacks against hospitals. *Annals of emergency medicine*, 83(1):46–56, 2024.

[297] D. van Dalen. *Logic and Structure*. Springer, Berlin, third edition, 1997. ISBN: 9780521873970.

[298] Bas C Van Fraassen. *The scientific image*. Oxford University Press, 1980. ISBN: 9780198244271.

[299] Alexey Voinov, Karen Jenni, Steven Gray, Nagesh Kolagani, Pierre D Glynn, Pierre Bommel, Christina Prell, Moira Zellner, Michael Paolisso, and Rebecca Jordan. Tools and methods in participatory modeling: Selecting the right tool for the job. *Environmental Modelling & Software*, 109:232–255, 2018.

[300] Alexey Voinov, Nagesh Kolagani, Michael K McCall, Pierre D Glynn, Marit E Kragt, Frank O Ostermann, Suzanne A Pierce, and Palaniappan Ramu. Modelling with stakeholders–next generation. *Environmental Modelling & Software*, 77:196–220, 2016.

[301] Ludwig Von Bertalanffy. General systems theory. *The science of synthesis: exploring the social implications of general systems theory*, 103, 2010.

[302] Gerrit De Vynck, Rachel Lerman, Ellen Nakashima, and Chris Alcantara. *The anatomy of a ransomware attack.* Washington Post, 2021. Available at: `https://www.washingtonpost.com/technology/2021/07/09/how-ransomware-attack-works/?itid=mr_innovations_1`. Accessed 16/12/2024.

[303] Patrick Wardle. Methods of malware persistence on mac os x. In *Proceedings of the virus bulletin conference*, 2014.

[304] Michael Weisberg. *Simulation and similarity: Using models to understand the world.* Oxford University Press, 2012. ISBN: 9780190265120.

[305] James Wyke. What is zeus?, 2011. Available at: `https://www.mercurymagazines.com/pdf/GNSOPHOS1.pdf`. Accessed 17/12/2024.

[306] Xi Xiao, Peng Fu, Changsheng Dou, Qing Li, Guangwu Hu, and Shutao Xia. Design and analysis of seiqr worm propagation model in mobile internet. *Communications in Nonlinear Science and Numerical Simulation*, 43:341–350, 2017.

[307] Yang Xiao, Shanghao Shi, Ning Zhang, Wenjing Lou, and Y. Thomas Hou. Session key distribution made practical for can and can-fd message authentication. In *Annual Computer Security Applications Conference*, ACSAC '20, page 681–693, New York, NY, USA, 2020. Association for Computing Machinery.

[308] Wenjun Xiong, Emeline Legrand, Oscar Åberg, and Robert Lagerström. Cyber security threat modeling based on the mitre enterprise att&ck matrix. *Software and Systems Modeling*, 21(1):157–177, 2022.

[309] Hongseok Yang and Peter O'Hearn. A semantic basis for local reasoning. In Mogens Nielsen and Uffe Engberg, editors, *Foundations of Software Science and Computation Structures*, pages 402–416, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[310] Hongseok Yang and Peter O'Hearn. A semantic basis for local reasoning. In Mogens Nielsen and Uffe Engberg, editors, *Foundations of Software Science and Computation Structures*, pages 402–416. Springer Berlin Heidelberg, 2002.

[311] Gi-Tae Yeo, Ji-Yeong Pak, and Zaili Yang. Analysis of dynamic effects on seaports adopting port security policy. *Transportation Research Part A: Policy and Practice*, 49:285–301, 2013.

[312] Adam Young and Moti Yung. Cryptovirology: Extortion-based security threats and countermeasures. In *Proceedings of the 1996 IEEE Conference on Security and Privacy*, SP'96, page 129–140, USA, 1996. IEEE Computer Society.

[313] Lena Yuryna Connolly, David S Wall, Michael Lang, and Bruce Oddson. An empirical study of ransomware attacks on organizations: an assessment of severity and salient factors affecting vulnerability. *Journal of Cybersecurity*, 6(1), 12 2020.

[314] Enrico Zio. Monte carlo simulation: The method. In *The Monte Carlo simulation method for system reliability and risk analysis*, pages 19–58. Springer, 2013.