INTELLIGENT ARCHITECTURE:
USER INTERFACE DESIGN TO ELICIT KNOWLEDGE MODELS

*Alan Penn, *Ruth Conroy, *Nick Dalton, †Laura Dekker, *Chiron Mottram, †Alasdair Turner

*The Bartlett School of Architecture and Planning

†Intelligent Systems Lab, Department of Computer Science

University College London

Abstract

Much of the difficulty in architectural design is in integrating and making explicit the knowledge of the many converging disciplines (engineering, sociology, ergonomics and psychology, to name a few), the building requirements from many viewpoints, and to model the complex system interactions. The many rôles of the architect simply compound this. This paper describes a system currently under development—a 3D design medium and intelligent analysis tool, to help elicit and make explicit these requirements. The building model is used to encapsulate information throughout the building lifecycle, from inception and master planning to construction and 'lived-in' use. From the tight relationship between material behaviour of the model, functional analysis and visual feedback, the aim is to help in the resolution of functional needs, so that the building meets not only the aims of the architect, but the needs of the inhabitants, users and environment.

The Problem of Designing the Built Environment

It is often said that architecture is the mother of the arts since it embodies all the techniques of painting: line, colour, texture and tone, as well as those of sculpture: shape, volume, light and shadow, and the changing relative position of the viewer, and adds to these the way that people inhabit and move through its space to produce—at its best—a spectacle reminiscent of choreography or theatre. As with all the arts, architecture is subject to personal critical taste and yet architecture is also a public art, in that people are constrained to use it. In this it goes beyond the other arts and is called on to function, to modify the climate, provide shelter, and to subdivide and structure space into a pattern that somehow fits the needs of social groups or organisations and cultures. Whilst architecture may be commissioned in part as a cultural or aesthetic expression, it is almost always required to fulfil a comprehensive programme of social and environmental needs.

This requirement to function gives rise to three related problems that characterise the design and use of the built environment. The first depends on the difference between explicit knowledge—that of which we are at least conscious and may even have a scientific or principled understanding—and implicit knowledge, which, like knowing your mother tongue, can be applied without thinking. The functional programmes buildings are required to fulfil are largely social, and are based on implicit rather than explicit bodies of knowledge. The knowledge we exploit when we use the built environment is almost entirely applied unconsciously. We don't have to think about buildings or cities to use them; in fact, when we become aware of it the built environment is often held to have failed. Think of the need for yellow lines to help people find their way around the Barbican complex in the City of London, or the calls from tenants to 'string up the architects' when housing estates turn out to be social disasters.

INTELLIGENT ARCHITECTURE:
USER INTERFACE DESIGN TO ELICIT KNOWLEDGE MODELS

*Alan Penn, *Ruth Conroy, *Nick Dalton, †Laura Dekker, *Chiron Mottram, †Alasdair Turner

*The Bartlett School of Architecture and Planning

†Intelligent Systems Lab, Department of Computer Science

University College London

Abstract

Much of the difficulty in architectural design is in integrating and making explicit the knowledge of the many converging disciplines (engineering, sociology, ergonomics and psychology, to name a few), the building requirements from many viewpoints, and to model the complex system interactions. The many rôles of the architect simply compound this. This paper describes a system currently under development—a 3D design medium and intelligent analysis tool, to help elicit and make explicit these requirements. The building model is used to encapsulate information throughout the building lifecycle, from inception and master planning to construction and 'lived-in' use. From the tight relationship between material behaviour of the model, functional analysis and visual feedback, the aim is to help in the resolution of functional needs, so that the building meets not only the aims of the architect, but the needs of the inhabitants, users and environment.

The Problem of Designing the Built Environment

It is often said that architecture is the mother of the arts since it embodies all the techniques of painting: line, colour, texture and tone, as well as those of sculpture: shape, volume, light and shadow, and the changing relative position of the viewer, and adds to these the way that people inhabit and move through its space to produce—at its best—a spectacle reminiscent of choreography or theatre. As with all the arts, architecture is subject to personal critical taste and yet architecture is also a public art, in that people are constrained to use it. In this it goes beyond the other arts and is called on to function, to modify the climate, provide shelter, and to subdivide and structure space into a pattern that somehow fits the needs of social groups or organisations and cultures. Whilst architecture may be commissioned in part as a cultural or aesthetic expression, it is almost always required to fulfil a comprehensive programme of social and environmental needs.

This requirement to function gives rise to three related problems that characterise the design and use of the built environment. The first depends on the difference between explicit knowledge—that of which we are at least conscious and may even have a scientific or principled understanding—and implicit knowledge, which, like knowing your mother tongue, can be applied without thinking. The functional programmes buildings are required to fulfil are largely social, and are based on implicit rather than explicit bodies of knowledge. The knowledge we exploit when we use the built environment is almost entirely applied unconsciously. We don't have to think about buildings or cities to use them; in fact, when we become aware of it the built environment is often held to have failed. Think of the need for yellow lines to help people find their way around the Barbican complex in the City of London, or the calls from tenants to 'string up the architects' when housing estates turn out to be social disasters.

The second is a problem of complexity. The problem is that buildings need to function in so many different ways. They are spatial and social, they function in terms of thermal environment, light and acoustics, they use energy and affect people's health, they need to be constructed and are made of physical components that can degrade and need to be maintained. On top of all this they have an aesthetic and cultural rôle, as well as being financial investments and playing an important rôle in the economy. Almost all of these factors are interactive—decisions taken for structural reasons have impacts on environment or cost—but are often relatively independent in terms of the domains of knowledge that need to be applied. This gives rise to a complex design problem in which everything knocks on to everything else, and in which no single person has a grasp of all the domains of knowledge required for its resolution. Even when the knowledge that needs to be applied is relatively explicit—as for instance in structural calculations, or those concerning thermal performance—the complex interactive nature of buildings creates a situation in which it is only through a team approach that design can be carried out, with all that this entails for problems of information transfer and breakdowns in understanding.

The third is the problem of 'briefing'. It is a characteristic of building projects that buildings tend not to be something that people buy 'off-the-shelf'. Often the functional programme is not even explicit at the outset. One might characterise the process that actually takes place by saying that the design and the brief 'co-evolve'. As a project moves from inception to full specification both the requirements and the design become more and more concrete through an iterative process in which design of the physical form and the requirements that it is expected to fulfil both develop at once. Feasible designs are evaluated according to what they provide, and designers try to develop a design that matches the client's requirements. Eventually, it is to be hoped, the two meet with the textual description of what is required and the physical description of the building that will provide it more or less tying together as the brief becomes a part of the contractual documentation that the client signs up to.

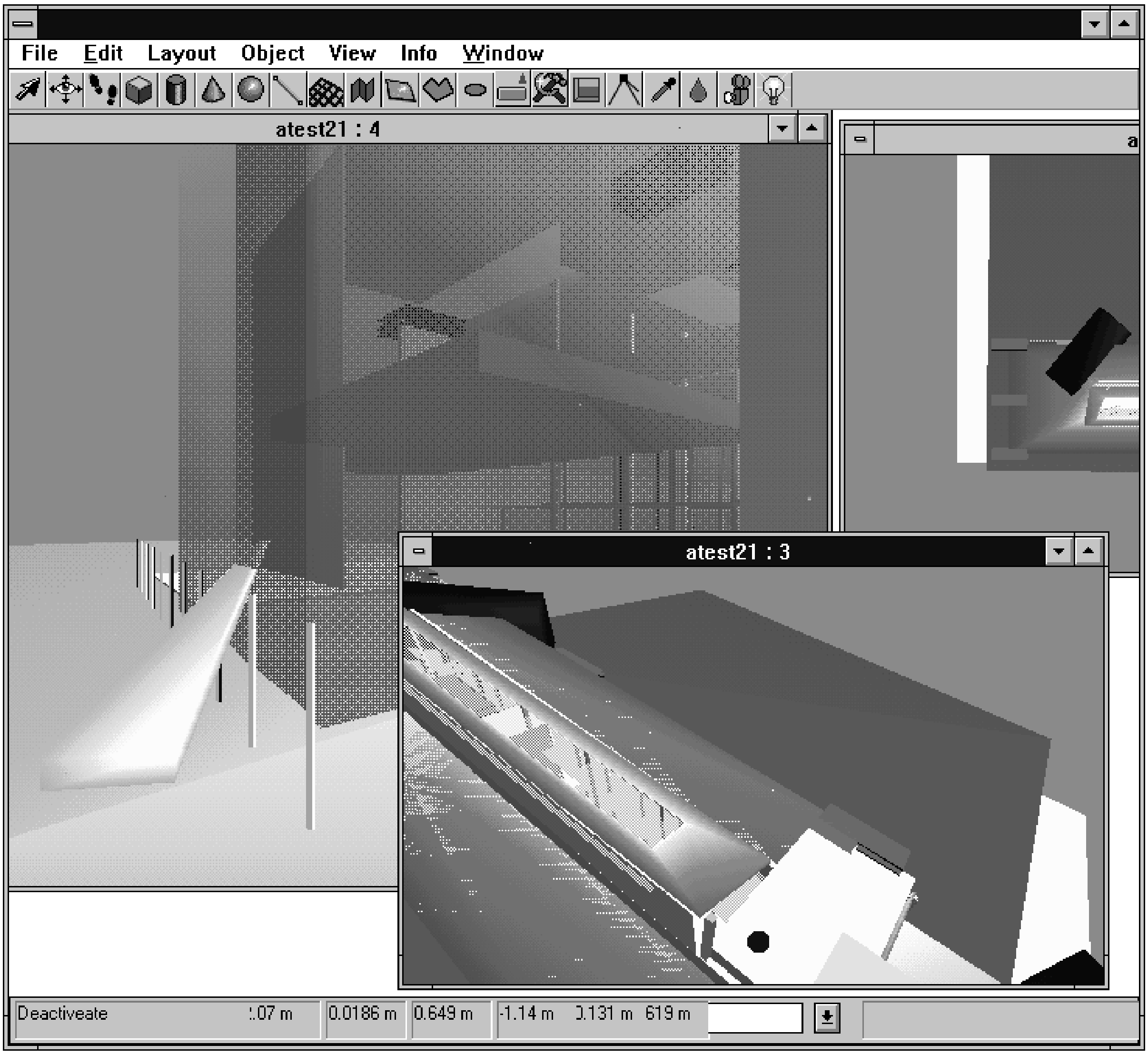
These three problems compound themselves in a number of ways. Since many of the core objectives of a client organisation rest on implicit knowledge—the need for a building to foster communication and innovation amongst its workers for instance—it is all too easy for them to be lost to sight against the more explicitly stated requirements such as those concerned with cost, environmental performance or statutory regulations. The result is that some of the more important aspects of the functional programme can lose out to less important but better understood issues. This can be compounded by the approach that designers take in order to control the complexity of projects. All too often the temptation is to wait until the general layout of a building is 'fixed' before calling in the domain experts. The result is that functional design has to resort to retrofitting to resolve problems caused by the strategic plan.

The Intelligent Architecture project is investigating the use of a single unified digital model of the building to help resolve these problems by bringing greater intelligence to bear at the earliest 'form generating' phase of the design process when the client's requirements are still being specified and when both physical design and client expectations are most easily modified. The aim is to help narrow the gap between what clients hope to obtain and what they eventually receive from a building project.

The strategy is simple. By capturing representations of the building as a physical and spatial system, and using these to bring domain knowledge to bear on a design at its earliest stages, it is hoped that some of the main conflicts that lead to sub-optimal designs can be avoided. By linking between textual schedules of requirements and the physical/spatial model it is intended to ease the reconciliation of the brief and the design, and help the two to co-evolve. By making available some of the latest 'intelligent' techniques for modelling spatial systems in the built environment, it is hoped to help put more of the implicit knowledge on an equal footing with explicit knowledge, and by using graphical feedback about functional outcomes where explicit knowledge exists, to bring these within the realm of intuitive application by designers.

The Workbench

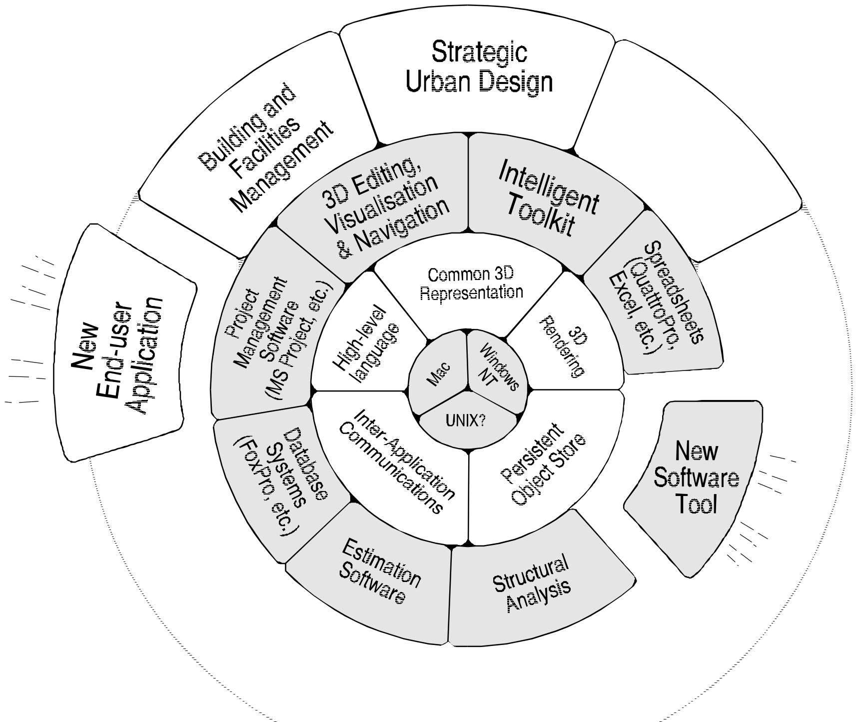
In order to do this, Intelligent Architecture has developed *Pangea*. Pangea has been designed as a *general-purpose* environment for intelligent 3D modelling—it does not pre-suppose a particular way of working, a particular design solution, or even a particular application domain. Several features make this possible.



The New Powergen Headquarters in Coventry, Modelled with Pangea

Worlds can be constructed from 3D and 2D primitives (including blocks, spheres, irregular prisms and deformable surfaces), which can represent real-world physical objects, or encapsulate some kind of abstract behaviour. The 3D editor provides a direct and simple interface for manipulating objects—to position, reshape, rotate and rework. All objects, both physical and abstract, have an internal state (defined by attributes), and behaviour, rules and constraints (in terms of a high-level-language ‘script’). Attributes can be added dynamically, making it possible for objects to change in nature, in response to new knowledge about them, or to a changing environment. Scripts are triggered by events, so that objects can respond and interact, as in the built environment, molecular systems, or fabric falling into folds on an irregular surface.

Dynamic linking allows Pangea’s functionality to be extended to include standard ‘off-the-peg’ software tools — spreadsheets, statistical analysis applications, graphing packages and domain-specific analysis software, such as finite element analysis for air-flow modelling.



A Conceptual View of Pangea

The ‘intelligent toolkit’ includes neural networks [Koho89] [Wass89], genetic algorithms [Gold89] [Holl75] and other stochastic search techniques [KiDe95], together with a rule-based and fuzzy logic system [Zade84]. The intelligent tools are objects, just like the normal 3D primitives: they have 3D presence and can interact with other 3D objects. A natural consequence of this design is easy ‘hybridisability’ of techniques, widely considered as vital to the success of intelligent techniques in solving realistically complex problems [GoKh95]. This infrastructure of primitive forms, intelligent techniques and high-level language makes it possible to build applications to deal with a broad range of problems, from the generation of architectural form, spatial optimisation, object recognition and clustering, and inducing rules and patterns from raw data.

Embedding Intelligence

Many consider that there is an inevitable trade-off between computers as a pure design medium, and computers with intelligence, ‘as a thinking machine’ [Rich94]. We propose here that it is possible to provide both these types of support, and allow the user to choose how best to use each, or not, according to the situation.

It is essential that the creative rôle of the architect is preserved as he or she uses the workbench, that the architect as artist may draw manipulate the world as seen through the workbench as freely as they would when using a sheet of paper. Much of the knowledge entered into the workbench in this way is unexpressed: an architect may draw a block, but intend that the block to be a stair or a door or a room. By using a clustering algorithm we have tried to capture some of the knowledge that the architect as artist does not express. In this manner, the architect as engineer may then pick up the sketch and continue to work with a technical drawing. Once we have identified the components, we can also apply rule-based systems to make recommendations for the design. Thus, by using a clustering algorithm, we are crossing the bridge from implicit knowledge of the architect as artist to the explicit knowledge of architect as engineer.

The object-oriented nature of the workbench allows a common interface to the clusterer, while implementing the clustering engine itself by a choice of simple linkage clustering algorithm or neural network (either a back-propagation multi-layer network or Kohonen). The architect specifies the attributes of an object in the world that are considered important to the definition of a set of objects, and we make a normalised vector representation of those attributes. Attributes chosen might be volume or the greenness or an arbitrary combination of attributes such as these. The clustering algorithm is applied to these vectors, and places each cluster of objects into a collection. After clustering, the architect can name the set of objects retrieved (for example, ‘doors’ or ‘stairs’); the workbench can make sensible guesses as to which new objects belong to those clusters, and the workbench or architect can use the information to reason about the clusters.

Since each object can contain inside itself the rules relating to it, these can be triggered into operation when a particular event occurs. For example, when a building core (the area containing lifts, stairwells, ventilation shafts, and so on) is moved, an event is sent to it, and it can automatically go through its set of rules to determine if it still meets regulations for access and fire safety. This is particularly valuable when many rules need to be processed simultaneously, making it possible to investigate various designs, with rapid feedback. This can be taken further, by encapsulating rules in ‘expert’ or ‘critic’ objects [FNOS93], whose job it is to oversee some kind of design evaluation.

For example, the Pangea thermal modelling object would contain the rules and other information relating to the thermal environment. When invoked (by clicking with the mouse on it) it would iterate through the 3D objects in the building to determine the expected thermal performance. When dragged and dropped into another area of the building it would carry out the same function there. The metaphor of cameras and filters providing new ‘views’ on the world has been found to be very effective. In the case of thermal modelling, the same knowledge would be built into a thermal camera that could be pointed at a scene, overlaying it with contours to indicate the hotspots, as if being viewed through an infra-red filter. These cameras can then be copied from one application to another, as re-usable components encapsulating different functionality.

This mechanism of component re-use is exploited in the Intelligent Architecture component libraries, being built for specific domains, including intelligent housing types and commercial building components. These parameterised components, where a component can even be a whole building, contain rules about sizing, placement and their relationship with other objects. The intelligent office building component, for example, can be resized, and automatically determines the correct number of floors for the given scaling, the approximate size of core necessary, and so on. The aim is to embed useful expert knowledge at the early stages of design, when such influences can have a major effect on the final outcome.

During the design process a building tends to be evaluated using many orthogonal or conflicting criteria, such as cost of construction, cost of running, environmental impact, social factors and comfort levels. Since these tend to be measured in different ways, it is often not possible to resolve them into a *single* value to indicate the quality of the design. Where these criteria can be expressed explicitly and quantified, Pangea can use colour, graphs and gauges to indicate the quality of the design from these multiple viewpoints. As these values move up and down, this can help to give a feel of how each of these evaluations are affected as the design changes, and how the various factors interact. Often, however, these quantities cannot be expressed in such an explicit manner. From a group of candidate designs that perform well over the range of criteria, the designer will have to make the final decision based on factors less easy to make explicit, or factors that only become consciously apparent from manipulating the designs and internalising the behaviour in terms of the multiple evaluations. Genetic algorithms and other intelligent optimisation techniques can be exploited to generate this set of candidate designs, leaving the designer to make the final decisions. In this way intelligent techniques act in the more subtle rôles of *decision support*, rather than attempting to *replace* the expert.

Some particularly effective and novel optimisation techniques incorporated into Pangea are *dynamic representation* hill-climbers and genetic algorithms. If the search space for an optimisation problem is viewed as a landscape, with the aim of locating the single highest point (the global optimum), standard hill-climbers, and even genetic algorithms often get stuck in the foothills at local optima. By dynamically remapping the search space throughout the search, the space becomes 'rearranged' so that a difficult and bumpy terrain can become smoothed out for the search to traverse more easily. This technique has proven to be very effective at tackling a range of difficult optimisation problems [KiDe95]. Within Intelligent Architecture they are being used on many space management problems, from site planning to open-plan office layouts.

Conclusions

Most of the knowledge that is exploited in design is implicit knowledge. This creates a problem for the knowledge engineer, who requires an explicit statement of the criteria and rules to be used in making a decision.

In the Intelligent Architecture project we are working on the basis that if a tool is actually going to be *used* by creative designers, it must appeal to their implicit understanding of problem domains. In order to do this we are trying to make a new creative medium in which one of the material properties is an ability to give graphical feedback about functional performance. We hope in this way that designers will be able to internalise the dynamics of functional performance as they model the form of the design. In this sense we are aiming to help people to become more 'intelligent'.

One side-effect of this approach has been the discovery that in designing the user interface to give feedback on particular problem domains—for instance, the requirements of housing development layouts, where over-looking, over-shadowing, density and car parking criteria must all be squared with each other—the boundary conditions are clarified. It is here that the process of integrating essentially manual systems helps the knowledge engineer to define the relationships between the many interacting criteria which characterise the design of the built environment. The effect is to make implicit knowledge explicit enough to allow advanced optimisation techniques to be brought into use, to map out the fitness landscape and give guidance to designers on strategies that are likely to be fruitful.

References

[FNOS93] Fischer G, Nakakoji K, Ostwald J, Stahl G, Sumner T. "Embedding Critics in Design Environments", *The Knowledge Engineering Review*, Vol. 8:4 (1993).

[GoKh95] Goonatilake S., and Khebbal S (eds), *Intelligent Hybrid Systems*. John Wiley (1995).

[Gold89] Goldberg DE. *Genetic Algorithms In Search, Optimisation and Machine Learning*. Addison-Wesley, Reading, Mass (1989).

[Holl75] Holland JH. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor (1975).

[KiDe95] Kingdon J, Dekker L. "The Shape of Space", *Proceedings of the First IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*. IEE, London (1995).

[Koho89] Kohonen T. *Self-Organization and Associative Memory*, *3rd Ed*. Springer-Verlag, Berlin (1989).

[Rich94] Richens P. "Does Knowledge Really Help? CAD Research at the Martin Centre", *Automation in Construction 3.* Elsevier, Amsterdam (1994).

[Wass89] Wasserman PD. *Neural Computing: Theory and Practice*. Van Nostrand Reinhold, New York (1989).

[Zade84] Zadeh L. "Making Computers Think Like People", *IEEE Spectrum Vol. 21:8*. (1984).