Artificial Intelligence for Engineering Design, Analysis and Manufacturing (2007), *21*, 267–277. Printed in the USA.

DOI: 10.1017/S0890060407000273

Agents for multidisciplinary design in virtual worlds

MARY LOU MAHER, MICHAEL ROSENMAN, AND KATHRYN MERRICK
Key Centre for Design Computing and Cognition, University of Sydney, Sydney, Australia

(RECEIVED June 5, 2006; ACCEPTED October 05, 2006)

Abstract

Agent models provide a generalized approach to the design of systems that autonomously monitor and affect an environment. Societies of agents that reason and communicate about an environment can achieve complex emergent behavior to facilitate and augment human activities. This paper introduces artificial agent technologies as a means by which the behavioral complexity of virtual worlds can be extended to provide the functionality needed to support collaboration in multidisciplinary design teams. Three key roles are identified that artificial agents can play to provide this functionality: support for multiple views of designed objects, support for the expression of relationships between designed objects, and compatibility with existing design tools. The implementation of a society of agents fulfilling these roles within a virtual world based, multidisciplinary design tool called DesignWorld is described. The increased behavioral complexity and functionality of DesignWorld's underlying virtual world is demonstrated using the results of multidisciplinary design experiments with DesignWorld.

Keywords: Artificial Agents; Collaborative Design; Multidisciplinary Design; Virtual Worlds

1. INTRODUCTION

Large design projects, such as those in the architecture, engineering, and construction (AEC) domain, involve collaboration between designers from many different design disciplines in varying locations. However, current software tools for documenting and developing models of buildings focus on supporting a single user at a time who is a specialist in the specific software used within his or her own discipline. Extensions to these tools for use by teams have tended to maintain the single discipline view and the features to support teams focus on issues such as version and file management. There is a perceived need in the AEC domain to have tools that specifically support collaboration among individuals from multiple disciplines with both a graphical representation of the design and a persistent data model. Using such a tool, individuals are able to work on their own part of the design using models appropriate to their discipline while communicating and collaborating with individuals from other disciplines.

Multiuser three-dimensional (3-D) virtual worlds are an appropriate software base for the development of a collaborative design tool to support multidisciplinary design. Online virtual worlds are inherently multiuser, and therefore directly support collaboration through a sense of awareness of others in the virtual world and their location within the world, and by providing various channels for direct and indirect communication. Some multiuser virtual world platforms also provide a 3-D building and modeling environment that can be adapted to the needs of the building and construction industry.

This paper describes how artificial agent technologies can be used to extend the behavioral complexity of virtual world platforms to provide the functionality required to support multidisciplinary design. Section 2 begins by reviewing the concept of artificial agents and motivating their use as a means of extending the functionality and behavioral complexity of virtual worlds. Section 3 identifies three key functional areas in which virtual worlds can be extended to support multidisciplinary design. Section 4 introduces DesignWorld, a design tool prototype in which artificial agents playing these roles extend the functionality of a virtual world to provide support for multidisciplinary design teams. A description of the implementation of a society of agents within DesignWorld is given in Sections 5 and 6. Finally, Section 7 demonstrates the new behavioral complexity and functionality of DesignWorld's underlying virtual world using a scenario from multidisciplinary design experiments with DesignWorld. The paper concludes with a discussion of the strengths and weaknesses of the agent based approach and future directions for this research.

Reprint requests to: Mary Lou Maher, Key Centre for Design Computing and Cognition, Building GO4, University of Sydney, Sydney, NSW 2006, Australia. E-mail: mary@arch.usyd.edu.au

2. AGENT MODELS FOR EXTENDING THE FUNCTIONALITY OF 3-D VIRTUAL WORLDS

A virtual world is a distributed, persistent, virtual space, created and evolved by its users with built-in content creation tools. In virtual worlds, people can interact with other people or the objects that comprise the world using an avatar controlled by the mouse and keyboard. Recent virtual worlds such as Second Life (www.secondlife.com) allow collaborative manipulation and visualization of shared objects, both synchronously and asynchronously. Users can select from a range of primitive objects with which to build, which can then be further modified using a number of built-in tools to achieve more complex objects.

The focus of many virtual world design platforms has been on the visual and interactive aspects of the world. Dynamic behavior of the 3-D objects in virtual worlds is typically achieved with simple, scripted behaviors triggered by events. The effort required to implement these simple behaviors over wide areas of a virtual world is high, as each object must be scripted individually. A major issue in the development of virtual worlds is the behavioral complexity of the objects in the world and the complexity of the implementation that is required to achieve the behaviors.

Maher and Gero (2002) proposed a way to increase the behavioral complexity of dynamic virtual worlds by giving agency to each persistent 3-D object in the virtual world. Agents are systems capable of perceiving their environment through sensors, reasoning about their sensory input using some characteristic reasoning process and acting in the world using their effectors as shown in Figure 1. The key requirement of agent reasoning processes is the ability to make an acceptable decision about what action to perform in time for this action to be useful (Wooldridge, 1997). This may be achieved by reasoning processes as simple as rules defining reflexes to predefined states of the environment or as complex as machine learning (Nilsson, 1996) and planning processes (Nilsson & Fikes, 1971). Maher, Gero, and Smith achieved an increase in behavioral complexity by using a cognitive agent model (Maher & Gero, 2002; Maher et al., 2003; Smith et al., 2003) for the behavior of the 3-D objects in a virtual meeting room. Their cognitive model includes processes for reasoning about the world at different levels of abstraction and a direct communication structure.

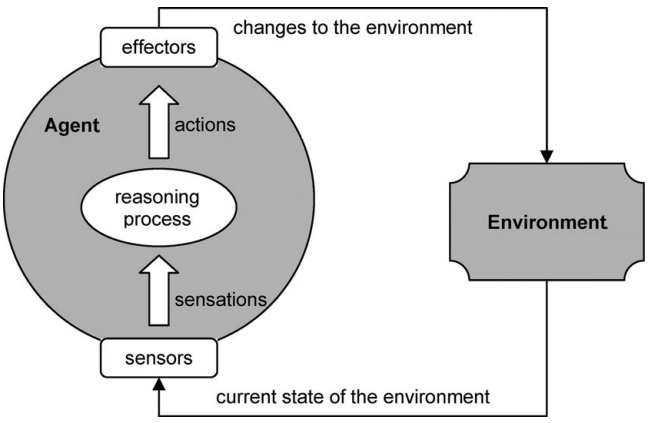

Fig. 1. The general agent model.

Maher and Merrick (2005) analyzed a number of agent models for use in dynamic environments and concluded that the benefit of using an agent model is that the agent model provides a generalized approach that needs only be implemented once then applied, possibly with small modifications to a number of agents, controling multiple objects. This provides coherence in the implementation of object behaviors and means that dynamic behavior, and thus additional functionality, can be achieved over large areas of 3-D virtual worlds more easily than using scripted behaviors.

3. VIRTUAL WORLD REQUIREMENTS FOR SUPPORTING MULTIDISCIPLINARY DESIGN

The complexity of building design leads to two conflicting requirements for multidisciplinary modeling systems: the ability of the different disciplines to work on their part of the project using their own specific models, and the ability to communicate and negotiate with the other disciplines on the synthesis and integration of the different design models. Two approaches for addressing the need for a virtual environment in which designers can coordinate domain-specific and integrated models are a multiuser computer-aided design (CAD) system and a multiuser virtual world. Although the CAD system approach uses a familiar modeling environment, CAD was not designed to be a multiuser environment, and the models tend to be specific to one discipline. In contrast, a virtual world approach has more potential in providing a flexible approach for modeling and communication that is not discipline specific.

The creation of different discipline models and the creation of relationships between the objects in the different models are central to the maintenance of consistency between the models. Creating these relationships requires communication between the different disciplines that can be facilitated with shared 3-D visualization, walkthroughs, and rendering of the various views of the design as modeled by the different disciplines. This is of special importance at the conceptual stage of the design because much of the early collaborative decision making is carried out at this stage. A virtual world environment based on an underlying object-oriented representation of the design specifically supports synchronous collaboration for multiple disciplines in the design of buildings. Lee et al. (2003) used a commercial CAD system for visualization. However, one of the advantages of virtual world environments is that they allow users to be immersed in the 3-D model, allowing for real-time walkthroughs and collaboration (Conti et al., 2003; Savioja et al., 2003). Moreover, CAD models contain a great deal of detail that affects the performance of real-time interactions.

https://doi.org/10.1017/S0890060407000273 Published online by Cambridge University Press

Although virtual environments such as Second Life support collaborative virtual design with built-in design tools, they do not incorporate the multidisciplinary tools required by "real-world" designers such as architects and engineers. Different disciplines have different views of a design object (building) according to their functional concerns, and hence create different representations or models of that object to suit their purpose. For example, a building may be viewed as a set of activities that take place in it, a set of spaces, a sculptural form, an environment modifier or shelter provider, a set of force resisting elements or as a configuration of physical elements. This leads to a number of functional requirements for a multidisciplinary design tool that are not supported by existing virtual environments:

- support for multiple views of designed objects,
- support for the expression of relationships between designed objects, and
- compatibility with existing design tools.

We discuss each of these requirements in the following sections.

3.1. Multiple views of designed objects

Depending on the view taken, certain objects in a model and the properties of those objects become relevant. For architects, for example, floors, walls, doors, and windows are associated with spatial and environmental functions, whereas structural engineers see the walls and floors as elements capable of bearing loads and resisting forces and moments. Hence, each will create a different model incorporating the objects and properties relevant to them. Both models must coexist because the two designers will have different uses for their models. According to Bucciarelli (2003), "There is one object of design, but different object worlds. . . . No participant has a 'god's eye view' of the design." Previous approaches to multidisciplinary design have used a single shared data model (Wong & Sriram, 1993; Krishnamurthy & Lay, 1997). However, a single-model approach to representing a design object is insufficient for modeling the different views of the different disciplines (Rosenman & Gero, 1996, 1998). Each viewer may represent an object with different elements and different composition hierarchies. Although architects may model walls on different floors as separate elements, the structural engineers may model only a single shear wall encompassing several architects' walls. Each discipline model must, however, be consistent with the objects described. Although Nederveen (1993), Pierra (1993), Pierra et al. (1998), and Naja (1999) use the concept of common models to communicate between the discipline models, it is unclear who creates the common models and maintains the consistency between them and the discipline models. We take the approach that members of each discipline will create their own views of the design, with members of each discipline assigning properties to their objects according to their needs. Consistency is provided by interrelationships between the various objects in different disciplines, modeled by explicit (bidirectional) links from one object to another, as discussed in the next section.

3.2. Relationships between designed objects

When several views of a design exist, relationships provide consistency between the elements that compose one view and the elements that compose another.

Figure 2 shows and example of this approach, with members of each discipline labeling its objects according to their needs and corresponding objects associated with *correspondsTo* relationships. Relationships may be intra- or interrelationships, that is, they may exist within a single discipline or between disciplines. The *correspondsTo* relationship is an example of an interrelationship, whereas the *partOf* relationship may be both an intra- and an interrelationship. Although this approach may have the disadvantage of replicating information about the same object in two object models, it saves the complexities of creating the common concepts and allows members of each discipline to have greater flexibility in creating their own models. The discipline models allow members of each discipline to work according to their own concepts and representations. The whole model may be seen as the union of the different models.

3.3. Compatibility with existing design tools

With the extensive range of existing design tools for members of different design disciplines comes the requirement for new

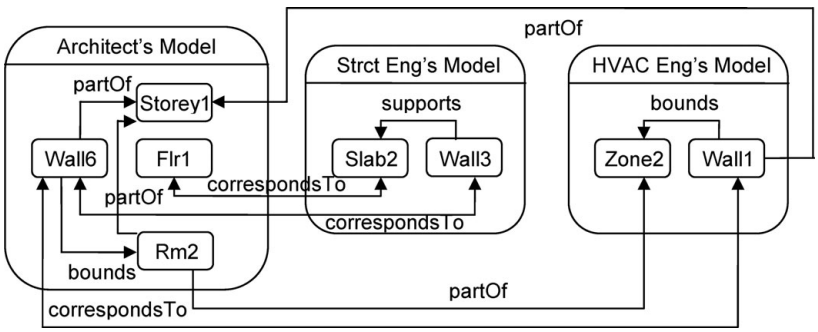

Fig. 2. Discipline models and relationships.

https://doi.org/10.1017/S0890060407000273 Published online by Cambridge University Press

collaborative design tools to provide some level of interoperability with existing data formats. For example, compatibility with industry foundation classes (IFCs; IAI, 2000) provides the potential for models to be uploaded from IFC-compatible applications such as ArchiCad for use in collaborative sessions. Likewise, models developed during collaborative sessions should be compatible with IFCs to make them accessible using other tools.

The additional functional requirements of virtual worlds necessary to support multidisciplinary design are particularly suited to an agent solution. As discussed in the preceding sections, the additional functionality requires modifications to be made to the designed environment to display different views of a design, change the properties of objects in a design to represent relationships, or construct new models from external data. These functions require the program implementing them to be aware of existing objects in the 3-D virtual world and be able to modify the environment according to certain rules or input from designers. This process corresponds directly with the sense–reason–act architecture of artificial agents. In addition, use of an agent model provides a standard interface with the virtual world via sensors and effectors. Different reasoning processes can be implemented within this structure to provide different functionality as required.

4. DesignWorld

DesignWorld is a prototype system for enabling collaboration between designers from different disciplines who may be in different physical locations. DesignWorld consists of a 3-D virtual world for creating and visualizing designs, augmented with a number of Web-based tools for communication and design as shown in Figure 3. Agent technology is used to maintain different views of a single design, support relationships between objects, and implement compatibility with existing design tools. Maher et al. (2006) introduce the DesignWorld environment and Rosenman et al. (2006) describe the implementation and integration of agents with the SecondLife platform. The following sections focus on how we achieved the different roles of agents in supporting multidisciplinary design.

DesignWorld is a conceptual design tool, aimed at the early phases of design in which the scope of a problem is determined by exploring a range of alternative solutions to a brief or set of requirements. Conceptual design is characterized by

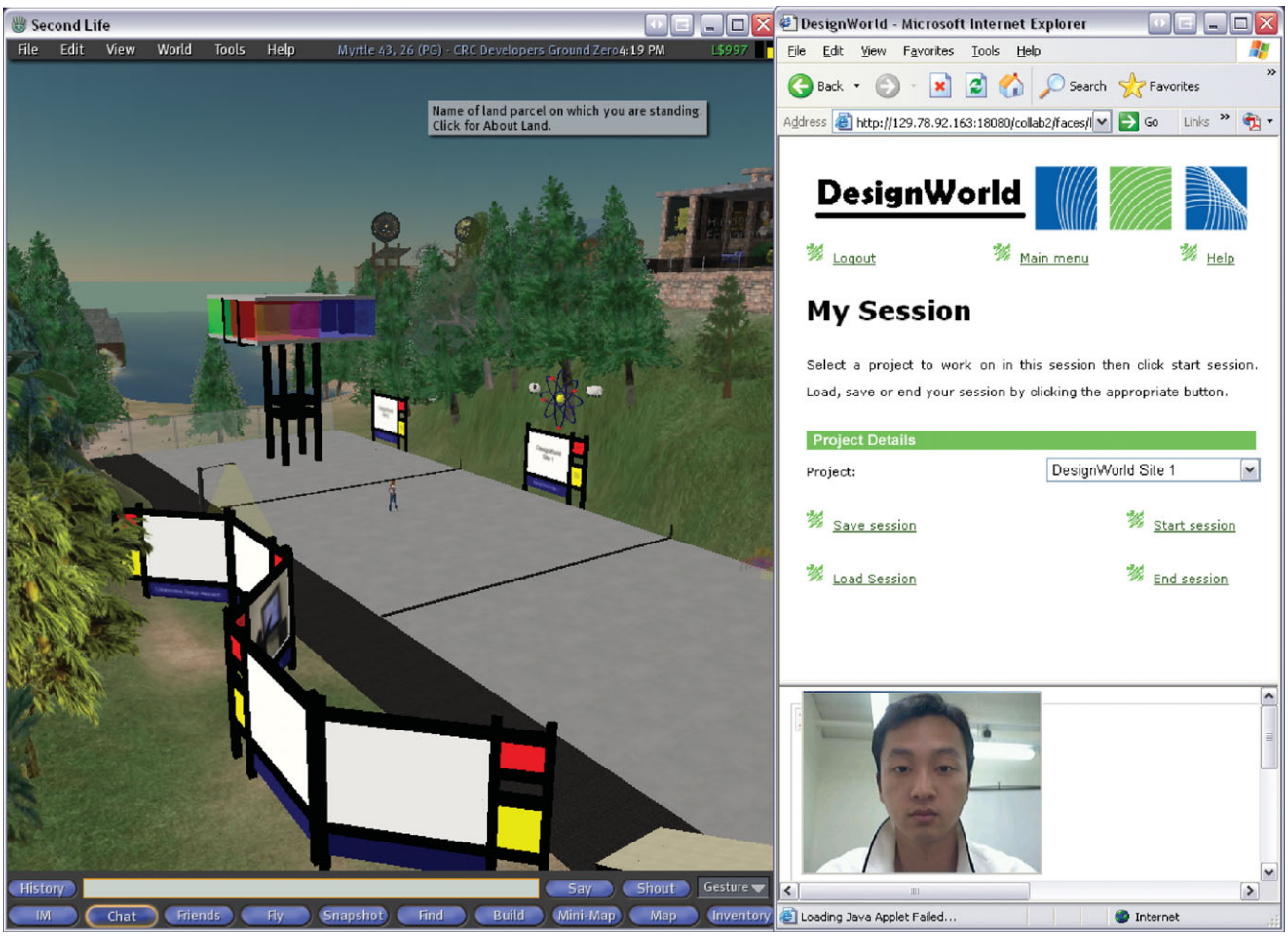

Fig. 3. DesignWorld consists of a 3-D virtual environment (left) augmented with Web-based communication and design tools (right). [A color version of this figure can be viewed online at www.journals.cambridge.org]

https://doi.org/10.1017/S0890060407000273 Published online by Cambridge University Press

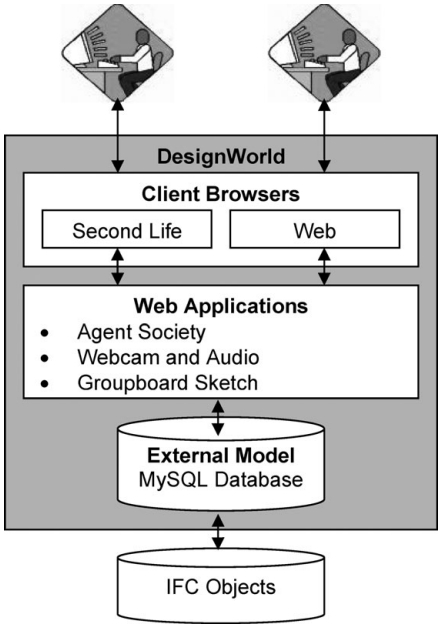

Fig. 4. DesignWorld system architecture. Client browsers (shown in Fig. 3) are supported by Web applications and a MySQL database.

a high degree of uncertainty and fluid design ideas. To address these issues, DesignWorld provides a sketching tool with which designers can rapidly produce design alternatives, while maintaining semantic density and ambiguous representation, and uses the Second Life 3-D virtual world to provide a direct-manipulation style building system that avoids the complexity and rigor required for a CAD system.

DesignWorld uses a client–server architecture, shown in Figure 4 to provide design and collaboration tools. Designers interact with DesignWorld using a client browser. The client browser, also depicted in Figure 3, has two components, a 3-D world window and a Web window. The 3-D world window is the primary interface through which designers can build representations of design artifacts. The tools displayed in the Web window include interfaces for viewing the nonspatial properties of a design, creating and managing relationships, sketching, and audiovisual communication. The viewing and relationship management tools use forms to gather information about requests from designers for new views or relationships. These requests are carried out on behalf of the designer by agents, as discussed in the next section.

The DesignWorld external model is a MySQL database. The database schema is compatible with IFCs (IAI, 2000) so a model can be uploaded from IFC-compatible applications such as ArchiCad for use in collaborative sessions. The external model contains project information for a group of objects, and for each object there is discipline, versioning, and relationship information.

5. THE ARCHITECTURE OF DesignWorld AGENTS

Agents are software systems that can sense their environment using sensors, reason about sensory input using some characteristic reasoning process, and act in their environment using effectors. The general function of an agent is to act for or represent another entity. The general agent model is shown in Figure 1. This model does not specify a particular reasoning process, but rather provides a framework into which any reasoning process can be embedded that is capable of reactive, real-time decision making. The environment for DesignWorld agents comprises three subenvironments as shown in Figure 5: a MySQL database, the Second Life virtual environment, and the Web-browser (HTTP) environment. These environments represent the three key components of the DesignWorld system architecture. Although various approaches exist for the design of agent reasoning processes, including learning and planning, we begin with a simple reflex agent approach. This allows us to explore the agent roles required to support collaborative, multidisciplinary design in a simple reasoning setting. Reflexive reasoning processes (Maher & Gero, 2002) select actions based on sensory input. They use two reasoning processes: the sensation process, and the

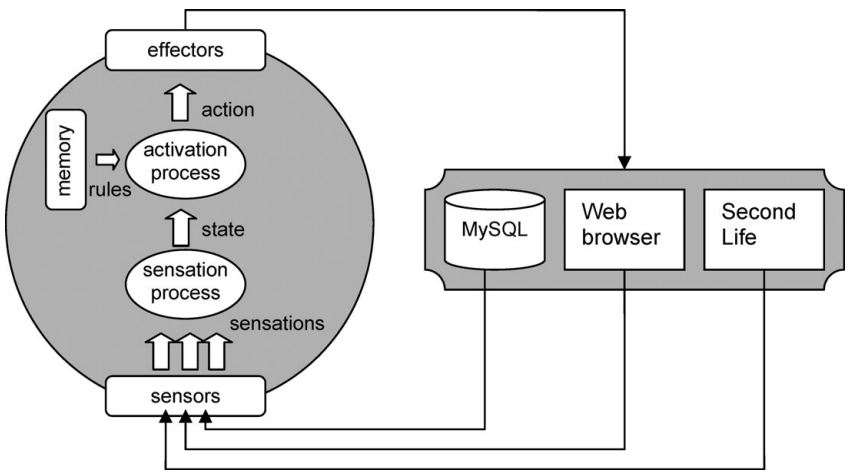

Fig. 5. A reflex agent model for DesignWorld agents.

https://doi.org/10.1017/S0890060407000273 Published online by Cambridge University Press

activation process as shown in Figure 5. The sensation process groups data sensed from the agent's subenvironments into a single description of the current state of the environment. The activation process uses preprogrammed rules to select an action in response to the state of the agent's environment.

In the remainder of this section we describe the general sensor, effector, and social structures for agents in DesignWorld. These provide the basis on which a reflexive reasoning process, or indeed other kinds of reasoning processes, can be built to facilitate multidisciplinary, collaborative design.

5.1. Sensors and effectors

Agents receive information about the state of their environment using sensors, whereas effectors are the means by which actions are achieved in the environment. DesignWorld agents have three types of sensors and effectors to enable them to sense or affect the Second Life, HTTP, and SQL environments.

5.1.1. Second Life sensors and effectors

The sensors and effectors that allow a DesignWorld agent to sense or affect the Second Life environment have two parts: a Java component and a Linden Scripting Language (LSL) component. These components communicate via XML-RPC. In addition to the LSL component of each sensor and effector, each object in a design also contains an LSL script that contains rules defining how the object responds to messages from sensors and effectors. Objects communicate with the LSL components of sensors and effectors by broadcasting chat on hidden channels that cannot be heard by human inhabitants of the virtual environment.

Second Life sensors are active sensors. The agent controlling them must actively send a message to its sensor that then scans the Second Life environment and returns data describing it. Only small amounts of data can be transferred between the LSL and Java sensor components after any scan. Thus individual sensations contain only a partial description of the environment. The sensation process is responsible for combining individual sensations into more complete, higher level, descriptions of the current state of the world. DesignWorld agents may make use of any of the Second Life sensors and effectors shown in Table 1.

The objects with which users construct their designs in DesignWorld are called DWObjects, and contain an LSL script that defines how they should respond to messages from sensors or commands from effectors. The object script listens for messages from effectors on a hidden chat channel. The channel on which the message occurs indicates the rule that should be fired. The message itself contains the parameter values required for the rule to be executed. The object script can also fire rules in response to user actions in the Second Life environment to register information with the LSL component of a sensor in preparation for messages from the Java component of that sensor.

Table 1. *Sensors and effectors for DesignWorld agents' environment*

Environ.	Sensor/Effector	Functionality
Second Life	Object sensor	Senses objects in the Second Life environment
	Selected object sensor	Senses object "touches" by avatars in the Second Life environment
	Chat sensor	Senses chat in the Second Life environment
	Chat effector	Broadcasts text on a specified channel
	Unselect object effector	Unselects all objects
	Relate effector	Relates objects
	Add object effector	Adds an object to the 3-D world
	Change object effector	Changes the properties of an object in the 3-D world
HTTP	Parameter sensor	Senses parameters with a label and value as follows: param(label, value)
	Property effector	Displays the nongeometric properties of an object on a Web page
SQL	Query sensor	Senses entries in an SQL database defined by a SELECT query
	Update effector	Modifies SQL database using ADD, DELETE, or UPDATE queries

5.1.2. HTTP sensors and effectors

The sensors that allow a DesignWorld agent to sense the HTTP environment are written in Java. In contrast to Second Life sensors, HTTP sensors are passive sensors. This means that the agent passively waits for a message to be sent to these sensors rather than actively requesting data. DesignWorld agents may make use of any of the HTTP sensors and effectors shown in Table 1.

5.1.3. SQL sensors and effectors

Like HTTP sensors, the sensors and effectors that allow a DesignWorld agent to sense or affect the SQL environment are written in Java. SQL sensors are active sensors. DesignWorld agents may make use of any of the SQL sensors and effectors shown in Table 1.

5.2. Agent societies

DesignWorld agents are organized into societies corresponding to design projects. A project in DesignWorld refers to a piece of virtual land and the objects built on it. Each project is worked on by a team of human designers and a society of artificial agents that perform certain tasks on behalf of the human designers.

6. AGENT ROLES IN DesignWorld

The rules available to an agent define the role it plays in DesignWorld. There are five DesignWorld agents: the Modeler, the Relationship Manager, the Discipline Viewer, the Object–

https://doi.org/10.1017/S0890060407000273 Published online by Cambridge University Press

Property Viewer, and the Builder agents. These agents combine as a society to fulfill the three functionality requirements identified in Section 3. The following sections describe the high-level roles of each of the DesignWorld agents. In this paper, a reflex agent approach is taken in which reflexes are implemented as rules about the agent's environment. A detailed specification of the rules that define these roles can be found in Table 2.

6.1. Modeler agent

The Modeler agent facilitates the presentation of different views of a design by constructing and maintaining a data model of the design artifacts from the 3-D world in the SQL external model. This persistent model is capable of describing more properties of an object than can be represented in the 3-D environment. For example, in Second Life an object may have an owner but the SQL external model might additionally specify a project and a design discipline to which the owner and the object belong.

The model created and maintained by the Modeler agent is used by the Discipline Viewer and Object–Property viewer agents to construct different views of the unified model when requested by designers.

6.2. Discipline viewer agent

Designers will produce their model according to the needs of their particular discipline. The Discipline Viewer agent presents different models of a design relevant to designers from different disciplines by retrieving relevant information from the SQL external model and modifying the design displayed in the 3-D virtual environment window.

6.3. Object–Property Viewer agent

The DesignWorld external model is capable of representing more information about a design object than it is possible to display in the 3-D virtual environment window. For example, DesignWorld associates disciplines, projects, and relationships with objects. The Object–Property Viewer agent displays nongeometric information about design objects by retrieving relevant information from the SQL external model and displaying it in tabular form in a Web browser. At present, the nongeometric properties that can be attached are the discipline to which the object belongs and the relationships associated with that object. These properties are attached by DesignWorld. At present, properties are not imported from the IFC model but could be in future.

6.4. Relationship Manager agent

When several views of a design exist, there may be relationships between the elements that compose one view and the elements that compose another. There may also be relationships between the elements that compose a single view. Relationships express the associations between objects in a design. For example, one object may bound another object, one object may correspond to another object, or one object

Table 2. *DesignWorld agent reasoning rules*

Agent	State	Rule
Modeler	Message from HTTP environment requesting model of project P	Sense state of SL to sense objects O
	Objects O	Sense nongeometric properties of objects O in external model to construct external objects O′
	Objects O′	Affect SQL environment by inserting objects O′ as a new version
Discipline Viewer	Message from HTTP environment requesting view V of project P	Sense objects O from view V of project P in the external model
	Objects O	Affect the SL environment by making objects O visible and objects ∼O invisible
Object Property Viewer	Message from HTTP environment requesting view V of project P	Sense objects O' from view V of project P in the external model
	Objects O′	Affect the HTTP environment by displaying nongeometric properties of objects O′ in a browser window
Relationship Manager	Message from HTTP environment requesting unselect on project P	Affect SL environment by unselecting all objects on project P
	Message from HTTP environment requesting relationship type T	Sense all selected objects in the SL environment on project P, then affect SL environment by relating all selected objects on project P, then affect SL environment by unselecting all objects on project P
	Message from HTTP environment requesting relate on project P	Affect SL environment by relating all selected objects on project P, then affect the SQL environment by inserting relationship records
	Message from HTTP environment requesting delete relationship R on project P	Sense SQL environment to retrieve objects O in relationship R, then affect the SQL environment by deleting R, then affect the SL environment by unrelating objects O
Builder	Message from HTTP environment requesting IFC to be built	Sense objects O′ of project P in the external model
	Objects O′	Affect the SL environment by building objects O′

https://doi.org/10.1017/S0890060407000273 Published online by Cambridge University Press

may decompose to several other objects. The Relationship Manager agent takes information specified by a system user and uses it to create a persistent record of design relationships in the SQL database.

DesignWorld supports a number of different relationships. The *correspondsTo* relationship allows the association of objects in different discipline models so as to say that they are the same object but may have different nongeometric and nonphysical properties. For example, a wall in the architect's model may be the same as a wall in the structural engineer's model. The wall has the same shape, dimensions, location, and materials, but its function for the architect may be to provide privacy to a space, whereas its function for the structural engineer may be to support a slab. The *decomposesTo* relationship provides an association between a complex object and its components. This may also exist between objects in different disciplines. For example, a single wall object in the structural engineer's model may be associated with three walls (one above each other) in the architect's model. The *bounds* relationship provides for bounding associations between objects. For example, in the early conceptual design stages, an architect may only create spatial objects, whereas a structural engineer may create wall and slab objects. The relationship between the structural engineer's objects and the architect's object will be through a *bounds* relationship, for example, Wall1(engineer object) bounds Space1(architect object). The *supports* relationship allows the association of inter- or intradisciplinary objects where one object is providing physical support for the other. The *adjacentTo* relationship allows the association of components that are required to be close to each other. For example, in the architects' model of a restaurant, it may be a requirement for the kitchen to be adjacent to the dining area.

6.5. Builder agent

The Builder agent makes it possible to import models built using other modeling tools into DesignWorld for use in a collaborative session. A building may be modeled in an external CAD tool such as ArchiCAD, exported to an IFC file, and then converted to the Second Life primitive model. The converter application was implemented in the Java programming language and resides on the Web application server. It imports an IFC file into a new model in an EDM database, iterates through each supported building element in the model, and creates entries the DesignWorld SQL external model with the required information to create a Second Life primitive for the element. The Builder agent is then invoked to read the entries from the external model and create the primitive shapes in Second Life. The Builder agent was implemented with limited functionality as a proof of concept.

7. DesignWorld AGENTS IN ACTION

We performed a number of user experiments comparing face-to-face designing with designing using remote sketching and 3-D virtual worlds to understand the different design process behaviors in different design environments (Gul & Maher, 2006; Maher et al., 2006). These experiments also provided insight into the requirements for multidisciplinary design in environments that combine 3-D virtual worlds and sketching tools. Although the detailed analysis of the experiments is beyond the scope of this paper, in this section we describe a scenario that is taken from one of the experiments in which designers used DesignWorld with the agents that support multidisciplinary design. Here we focus on the part played by the agent society in a design session that considered architectural and structural engineering considerations. In the experiments, designers worked in pairs to complete a design brief by using DesignWorld to construct a 3-D model to fulfill the brief.

At the start of the experiment, the designers are told that they are to collaborate using DesignWorld to design an exclusive restaurant tower for a developer in Sydney. The brief stipulates that the restaurant proper has to be positioned with harbor views. In the 3-D world, a new project has been created including a society of agents and a full-scale mockup of the real-life site, so that the designers can get a sense of the surrounding landscape and the height required for the harbor view. For the remainder of the scenario, we will refer to the designers by their screen names, Adel (the architect) and Mary (the structural engineer).

Adel and Mary fly up with their avatars to 60 m in the air as shown in Figure 6a and satisfy themselves that the harbor view from that height is acceptable. Having established the height that they want their restaurant tower to reach, Adel and Mary open the sketching tool as shown in Figure 6b to draw and discuss possible layouts for the restaurant, finally settling on a design with two elevators to access the restaurant. Mary then proposes a cross-sectional view of the tower using the sketching tool, which Adel agrees to after modifying the supporting struts to extend above the ceiling to produce the esthetic effect that he is looking for.

Keeping the sketching tool open as a reference, Mary begins to build the supporting beams, columns, and elevator shafts in the 3-D world. Adel simultaneously creates transparent blocks in the 3-D world to represent the foyer area and the restaurant proper as shown in Figure 6c. During this time, the Modeler agent for the project uses its Object–Sensor to monitor the objects being built in the 3-D world and keeps a record of these objects for further reasoning.

As Adel finishes placing the spaces and watches Mary building, he realizes that building regulations for the part of Sydney for which the building is being designed require emergency fire exists, which the design is lacking. He alerts Mary, and they go back to the sketching tool, adding two stairwells to access the restaurant (Fig. 6d). Mary duplicates the elevator shafts in the 3-D world and changes their description so that they represent the stairwells. She places them, working off the modified sketch.

Adel explains to Mary that he wants a series of columns around the windows of the restaurant to produce the esthetic

https://doi.org/10.1017/S0890060407000273 Published online by Cambridge University Press

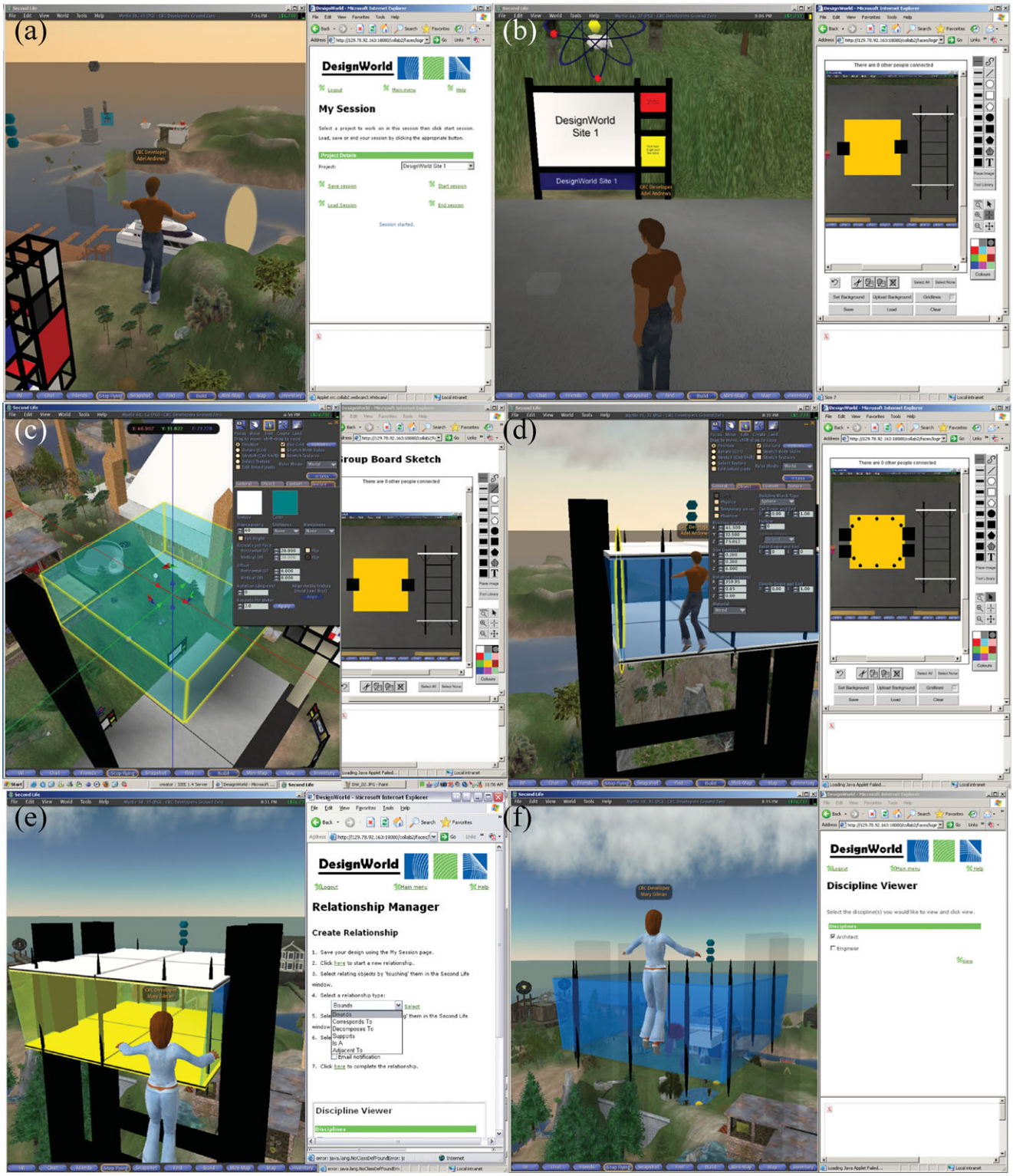

Fig. 6. (a) Adel inspects the view of the harbor. (b) Mary and Adel complete their sketch. (c) The Modeler agent monitors the design as Mary starts work on the columns and Adel creates the restaurant space. (d) Adel adds the columns following the new sketch. (e) The Relationship Manager agent assists Mary to create a bounding relationship between the floor and the restaurant. (f) The Discipline Viewer agent creates the architect view of the completed tower for Mary. [A color version of this figure can be viewed online at www.journals.cambridge.org]

https://doi.org/10.1017/S0890060407000273 Published online by Cambridge University Press

that he has in mind, marking up the sketch to show where he envisages them as shown in Figure 6d. Mary says that those columns would not be load bearing, so Adel builds them as part of the architect model while Mary adds ceilings and floors to the spaces built by Adel.

At this point, Mary uses the Web pages to send a message to the Modeler agent to save the current version of the 3-D model in the SQL database. The Modeler agent uses its SQL-sensor to retrieve information from the external model to associate nongeometric information, such as the discipline of the object owners, with every object in the current Second Life environment. It then uses its SQL-Effector to affect the external model by writing records describing the state of the model in Second Life.

Mary also uses the Web pages to request the Relationship agent to define relationships between the elements of the model that bound the spaces in the model as shown in Figure 6e. Mary selects the relationship type and the objects to be related and the Relationship agent uses its Selected-Object-Sensor to sense the objects Mary selected and its SQL-Effector to make a corresponding entry in the relationship table in the SQL database. The Relationship agent then uses its Change-Object-Effector to set a flag in the related objects so that if Adel or Mary modify some part the model in the future they will be alerted by a dialog box that their change impacts on other parts of the model.

Before finishing the design session Mary inspects the architect's model using the discipline viewer Web page. She requests the Discipline Viewer agent to show the architect's view so she can check that it matches the sketch and that each part is owned by the correct person. Upon receiving the request, the Discipline Viewer agent uses its SQL sensor to retrieve the discipline properties associated with the objects in the design from the SQL database. It then uses its Change-Object-Effector to make the objects from the architect's model visible and the objects from the engineer's model invisible, as shown in Figure 6f. Mary then sets the model back to the default view, and Mary and Adel log out of the 3-D world and end their DesignWorld session.

This scenario illustrates a number of key differences between DesignWorld and CAD systems. In particular, DesignWorld creates a sense of place by defining a world beyond the current model on which designers are working. The idea of designing from within the design provides the designer with new perspectives on their design, allowing them to move within and experience the model they are building rather than just manipulating the model. In DesignWorld, designers can talk in real time and modify different parts of the model simultaneously, rather than needing to work asynchronously on shared files.

8. DISCUSSION AND FUTURE WORK

This paper describes an implementation of DesignWorld agents as reflexive agents that sense their environment and requests from designers and respond by modifying their environment according to predefined rules. However, use of an agent architecture makes it possible for more complex reasoning processes to be implemented, to further extend the functionality of virtual worlds as design environments. For example, the Builder agent could be augmented with grammar based reasoning processes such as those proposed by Gu and Maher (2004) to enable it to suggest initial designs for new projects, as well as importing human made designs. Use of an agent architecture also makes it possible to embed unsupervised (Zoubin, 2003) learning models to create models of human system usage from which automated responses can be learned.

The key limitations of DesignWorld agents are caused by the underlying virtual world technology. The first issue noted by designers in the user experiments was the speed of the system. Although the agent programs are capable of fast reasoning, their responses are slowed by built in delays in Second Life, which limit the agent to just a three or four world modifications per second. This is compounded by the fact that only small amounts of data (256 characters) can be transmitted to and from the agents' sensors and effectors at a time. This impacts on all of the DesignWorld agents as they all need to modify large numbers of objects to respond to individual requests. The second key issue is caused by Second Life's underlying primitive object model that limits agents such as the Builder agent to only a few basic shapes. This means that currently only simple models can be imported from CAD systems. Despite these limitations with respect to CAD, however, DesignWorld does offer advantages with respect to collaboration and multidisciplinary design. Furthermore, in the future, we foresee that the development of virtual world platforms will overcome many of the current limitations.

9. CONCLUSION

This paper has described how artificial agent technologies can be used to extend the behavioral complexity of virtual world platforms to provide the functionality required to support multidisciplinary design. We identify three key roles that artificial agents can play to provide this functionality: support for visualizing different views of the designed objects, support for the expression of relationships between designed objects, and support for translation to and from other design tools. DesignWorld includes an implementation of a society of agents fulfilling these roles as an augmented virtual world specifically for multidisciplinary design. The increased behavioral complexity and functionality of DesignWorld's underlying virtual world was demonstrated using a scenario from one of the multidisciplinary design experiments with DesignWorld.

The primary contributions of this research are the definition and implementation of different roles of agents in multidisciplinary design as an extension of virtual world platforms. This allows the development and advancement of virtual world technologies to occur independently of the requirements of

https://doi.org/10.1017/S0890060407000273 Published online by Cambridge University Press

design environments, as commercial products that are suitable for other purposes. Our further development of design agents that augment such platforms can be connected to any suitable virtual world, and therefore can focus specifically on design research and the needs of teams of designers.

REFERENCES

Bucciarelli, L.L. (2003). Design and learning: a disjunction in contexts. *Design Studies 24(3)*, 295–311.

Conti, G., Ucelli, G., & De Amicis, R. (2003). JCAD-VR—a multi-user virtual reality design system for conceptual design. TOPICS. *Reports of the INI-GraphicsNet 15*, 7–9.

Gu, N., & Maher, M.-L. (2004). A grammar for the dynamic design of virtual architecture using rational agents. *International Journal of Architectural Computing 4(1)*, 489–501.

Gul, F., & Maher, M.-L. (2006). Studying design collaboration in DesignWorld: an augmented 3D virtual world. *Proc. 3rd Int. Conf. Computer Graphics, Imaging and Visualisation*. New York: IEEE.

IAI. (2000). *Industry Foundation Classes—Release 2x, IFC Technical Guide*. Accessed November 10, 2004 at http://www.iai-international.org/iai_international/technical_documents/documentation/IFC_2x_technical_guide.pdf

Krishnamurthy, K., & Law, K.H. (1997). A data management model for collaborative design in a CAD environment. *Engineering with Computers 13(2)*, 65–86.

Lee, K., Chin, S., & Kim, J. (2003). A core system for design information management using industry foundation classes. *Computer-Aided Civil and Infrastructure Engineering 18*, 286–298.

Maher, M.-L., Bilda, Z., Gul, F., Huang, Y., & Marchant, D. (2006). Comparing distance collaborative designing using digital ink sketching and 3D models in virtual environments, Clients Driving Innovation. *Proc. 2nd Int. Conf. Construction Innovation*.

Maher, M.-L., & Gero, J.S. (2002). Agent models of 3D virtual worlds. In *ACADIA 2002: Thresholds*, pp. 127–138. Pomona, CA: California State Polytechnic University.

Maher, M.-L., & Merrick, K. (2005). Agent models for dynamic 3D virtual worlds. *The 2005 Int. Conf. Cyberworlds*, Singapore, pp. 27–34.

Maher, M.-L., Smith, G.J., & Gero, J.S. (2003). Design agents in 3D virtual worlds. In *IJCAI Workshop on Cognitive Modelling of Agents and Multi-Agent Interactions*, pp. 92–100. Acapulco, Mexico.

Naja, H. (1999). Multiview databases for building modelling. *Automation in Construction 8*, 567–579.

Nederveen, S.V. (1993). View integration in building design. In *Management of Information Technology for Construction* (Mathur, K.S., Betts, M.P., & Tham, K.W., Eds.), pp. 209–221. Singapore: World Scientific.

Nilsson, N.J. (1996). *Introduction to Machine Learning*. Accessed January 2006 at http://ai.stanford.edu/people/nilsson/mlbook.html

Nilsson, N.J., & Fikes, R.E. (1971). STRIPS: a new approach to the application of theorem proving to problem solving. *Artificial Intelligence 2(4)*, 189–208.

Pierra, G. (1993). A multiple perspective object oriented model for engineering design. In *New Advances in Computer Aided Design and Computer Graphics* (Zhang, X., Ed.), pp. 368–373. Beijing: International Academic Publishers.

Pierra, G., Sardet, E., Potier, J.C., Battier, G., Derouet, J.C., Willmann, N., & Mahir, A. (1998). Exchange of component data: the PLIB (ISO 13584) model, standard and tools. *Proc. CALS EUROPE '98 Conf.*, pp. 160–176, Paris, September 16–18.

Rosenman, M., & Gero, J.S. (1996). Modelling multiple views of design objects in a collaborative CAD environment. *AI in Design 28(3)*, 207–216.

Rosenman, M., & Gero, J.S. (1998). CAD modelling in multidisciplinary design domains. In *Artificial Intelligence in Structural Engineering* (Smith, E., Ed.), pp. 335–347. New York: Springer.

Rosenman, M., Merrick, K., Maher, M.L., & Marchant, D. (2006). DESIGNWORLD: a multidisciplinary collaborative design environment using agents in a virtual world. In *Design Computing and Cognition 2006* (Gero, J.S., Ed.), pp. 695–710. Dordrecht: Springer.

Savioja, L., Mantere, M., Olli, I., Äyräväinen, S., Gröhn, M., & Iso-Aho, J. (2003). Utilising virtual environments in construction projects. *ITCon 8*, 85–99.

Smith, G.J., Maher, M.-L., & Gero, J.S. (2003). Designing 3D virtual worlds as a society of agents. In *Digital Design: Research and Practice—Proc. 10th Int. Conf. Computer Aided Architectural Design Futures* (Chiu, M.-L., Tsou, J.-Y., Kvan, T., Morozumi, M., & Jeng, T.-S., Eds.), pp. 105–114.

Wong, A., & Sriram, D. (1993). SHARED an information model for cooperative product development. *Research in Engineering Designs 5*, 21–39.

Wooldridge, M. (1997). Agent based software engineering. *IEEE Proceedings on Software Engineering 144(1)*, 26–37.

Zoubin, G. (2003). Unsupervised learning. In *Advanced Lectures on Machine Learning* (Bousquet, O., Raetsch, G., & von Luxburt, U., Eds.), pp. 72–112. Berlin: Springer–Verlag.

Mary Lou Maher is a Professor of design computing at the University of Sydney. She is currently on leave at the NSF in the United States, developing a funding emphasis on creativity and information technology. She received her PhD from Carnegie-Mellon University. Dr. Maher's research interests span different areas of design computing, specifically designing and collaborating in virtual environments, representation of design knowledge, design cognition, and the role and impact of new technologies on designers' cognition and design behavior.

Michael Rosenman is a Senior Lecturer at the Key Centre of Design Computing and Cognition, Faculty of Architecture, Design and Planning, University of Sydney. He is an architect with over 40 years of experience. He has carried out research into design, optimization, artificial intelligence, knowledge-based design, conceptual modeling, and evolutionary design. Dr. Rosenman is the author of over 90 publications in books, journals, and conference proceedings, and he has presented at numerous conferences all over the world. He is the regional editor for the Far East and Australasia for the journal *Design Studies.*

Kathryn Merrick is a PhD student in the School of Information Technologies at the University of Sydney. She completed a bachelor of computer science and technology (Advanced) with first class honors at the University of Sydney. Kathryn is currently working as a Research Assistant in the Key Centre for Design Computing and Cognition at the University of Sydney. Her research interests lie in the areas of agent architectures, virtual worlds, computational models of motivation, and machine learning.