An Agent Approach to Supporting Collaborative Design in 3D Virtual Worlds

Mary Lou Maher, Pak-San Liew, Ning Gu and Lan Ding

Key Centre of Design Computing and Cognition, University of Sydney
http://www.arch.usyd.edu.au/~mary

Abstract. 3D Virtual worlds facilitate a level of communication and collaboration not readily available in conventional CAD systems. The integration of virtual worlds and CAD systems using a common data model can make a significant impact on synchronous collaboration and real time multi-user multi-disciplinary modification of building data. By using agents, the integration of 3D virtual worlds and CAD systems can go beyond that of passive data transfer. With sensors and effectors, each agent can interact with its environment by responding to changes in the CAD system or 3D virtual world, which can take the form of an update to the geometry, or as a recommendation to change non geometric information or to propagate changes to other parts of the design. The reasoning process for each agent can vary from a reflexive behaviour in which the agent responds directly to the sensor data to a reflective behaviour in which the agents reasons about its goals and alternatives before making a change to the environment. We demonstrate this approach using ArchiCAD and Active Worlds as the CAD system and the virtual world platform. An EDM database is used as the central repository for storing the representation of the relevant data model. A multi-agent system is developed to connect the virtual world to this database to allow active data sharing. This agent approach can be extended to the integration of other applications and data models.

Keywords. Design Collaboration, Virtual World, Agent and CAD.

Introduction

The construction industry has long recognised the need for specialised software for specific needs, even though this variation in software has limited the ability for information to be seamlessly shared among different disciplines or software applications. In addition to the need for data sharing, there is also a need for data visualisation that supports communication and collaboration. With recent developments in multimedia and the WWW, we are seeing project management and document sharing have an impact on design and construction processes. These developments have made it easier to transfer files and to interact with the visualisation data. However, they have not yet made an impact on synchronous collaboration and real time multi-user sharing of building data.

An alternative environment for virtual design studios uses the concept of “designing within the design” (Maher and Simoff, 2000). We follow through with this approach by integrating CAD systems, which can support asynchronous team design through file sharing, and 3D virtual worlds, which can support synchronous team design within a multi-user 3D model. A key issue to establish seamless integration and support for managing information flows in 3D virtual worlds is the support of tracking and updating changes in the CAD and virtual world models.

The conventional approach for controlling information flows between CAD systems and 3D virtual worlds is essentially an inefficient process of manual conversion. The modifications made during the collaborative session in the 3D virtual worlds could not be easily reflected in the CAD files. Models in the 3D virtual worlds include geometrical information and interactive content from external sources. We propose and demonstrate an architecture for using agents to monitor changes in a common data model so that a 3D virtual world model can be extracted for synchronous collaborative sessions. Agents are used to monitor changes made in the 3D virtual world in order to update the common data model. Each agent can also store a representation of geometrical and non-geometrical information of the design and has reasoning algorithms to operate on the information.

3D Virtual Worlds

A virtual world is a multi-user networked 3D virtual environment. Most virtual worlds have been developed for the entertainment industry, but we are beginning to see virtual worlds in architectural design (Moloney, Amor, Furness, and Moores, 2003; Shiratuddin and Thabet, 2003; Maher, Liew and Gero, 2003). 3D virtual worlds support communication and collaboration in a place-like context. The virtual worlds that we are considering are object-oriented systems that associate a 3D model and a behaviour with each element of the world. Examples of such worlds include: Active Worlds and VirTools . For the remainder of this paper, we will describe our agent approach using the Active Worlds platform.

In addition to the virtual world providing a shared environment, the people in the world are represented as avatars that can walk around the world and make modifications to the world. For example, new 3D objects are added to the world by an avatar through copying an existing object, moving it as required, and editing a dialog box to configure it, as shown in Figure 1. The dialog box allows the world builder to specify a 3D model and a script that describes the behaviour of the object. The 3D models can be taken from a standard library provided by Active Worlds, or can be generated in a 3D modelling package and added to the library.

Benefits of using virtual worlds for sharing a visualization of the building model include the ease of the use of the interface and the inherent capability as a multi-user environment. CAD systems have complex interfaces that support the development and maintenance of the complex geometric models of the design. Virtual worlds are intended as multi-user collaborative systems and

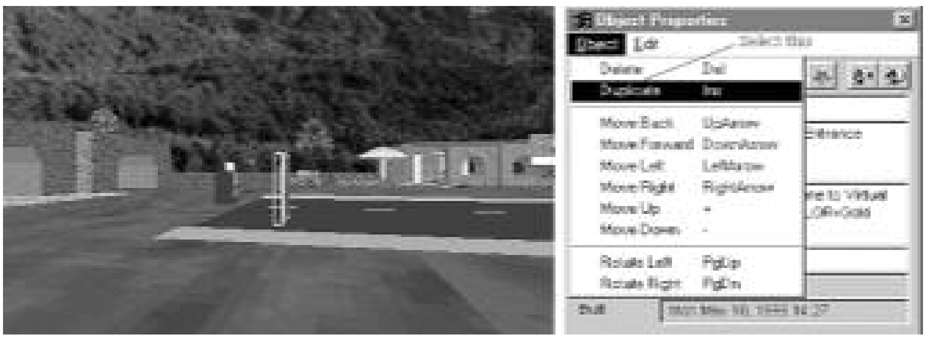

Figure 1. Inserting or modifying an object in a virtual world.

therefore provide a simpler interface that allows communication and limited modifications to the geometry of the model.

IFC and EDM Database

The Industry Foundation Classes (IAI, 2000) provide a standard object-oriented representation for life-cycle information in the AEC industry. This standardization allows the exchange and sharing of information across different applications based on a common object model. To facilitate data sharing, the information created by different applications according to IFC is stored within a central database where it is accessed and modified as required. The Express Data Manager (EPM, 2002) database is an object-oriented database that provides direct support for modelling, application development and database management of data defined according to the IFCs. It uses EXPRESS as a data description language to define the representation of a model. This paper uses an EDM database as a central database for the integration of CAD models and virtual environment models using IFCs as the common model.

An Agent Paradigm

In this paper, agents (Figure 2) are treated as intentional systems that operate independently and rationally, seeking to achieve their goals by interacting with their environment through the use of sensors and effectors (Wooldridge and Jennings, 1995).

We have developed a multi-agent system that interacts with and within a virtual world. A set of agents connects the virtual world to a common data model and each virtual object in the world is represented as an agent. The agent model provides a common vocabulary for describing, representing, and implementing agent knowledge and communication. This model (Figure 3) has sensors, effectors, and operates according to the following processes: interpretation, hypothesizer, and action.

Interpretation is a process that finds grounded patterns of invariance in the agent's representation of the sense data and associates concepts

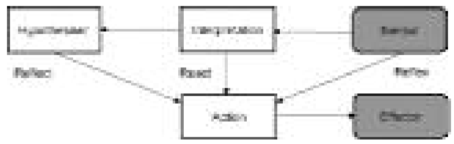

Figure 3. An agent model with 5 processes (after Maher and Gero, 2002).

with these sense data or patterns of these sense data. The hypothesiser identifies mismatches between the current and desired situation, which goals are relevant to the current state of the environment and reasons about which goal should be achieved in order to reduce or eliminate that mismatch. It identifies possible actions which when executed will change the environment to meet those goals. Action reasons about which sequence of operations on the building model, when executed, can achieve a specific goal.

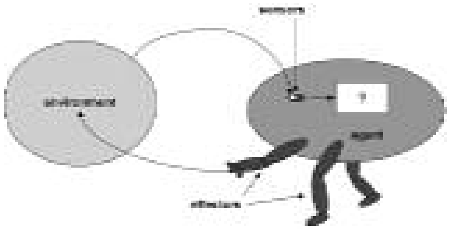

Figure 2. An agent interacts with its environment through its sensors and effectors.

The agent's sensors recognize two kinds of data: sense_data in which the agent identifies the relevant data, and receive_data in which the agent receives a message from another agent. The agent's effectors are the means by which actions are achieved. An effector can either causes a direct change to the environment or sends a message to another agent.

Agents can function in three modes: reflexive, reactive, or reflective. Each mode requires increasingly sophisticated reasoning, where reflexive is the simplest. These modes are indicated in Figure 3 by the labels on the paths through specific agent processes. In the reflexive mode, the agent responds to sense data from the environment with a pre-programmed response (a reflex) without any reasoning. Actions are a direct consequence of sense data. In the reactive mode, the agent interprets the sense data, and this interpretation triggers the relevant actions. In the reflective mode, the agent reasons about its expectations and the interpreted sense data before effecting an action into the environment.

An Agent Approach to Supporting Collaborative Design

Figure 4 illustrates an agent-based approach for supporting collaborative design in a virtual world. A common data model resides within an EDM object database shared by different applications. Data from ArchiCAD is translated to a neural file format according to IFCs and stored into the EDM database to form the common data model. Each graphical object in the database is associated with a 3D model in the virtual world and linked to an agent (not shown in Figure 4) for intelligent behaviour within the world.

An interface agent encapsulates the connection between the virtual world and the object database. This agent acts autonomously to monitor any changes made to the object represented in the virtual world and to update the common data model accordingly. There is an interface agent for each object type in the EDM database. For example, an interface agent that deals with all the walls in EDM is labeled as “walls” agent. This agent coordinates the data about all the walls in EDM and Active Worlds. Although the interface agent for coordinating the EDM and Active Worlds is illustrated, any two applications that need to share the same data model synchronously can use the same structure.

The interface agent is made up of a reasoning component, a set of sensors and effectors related to the database and another set of sensors and effectors related to a specific application, that is,

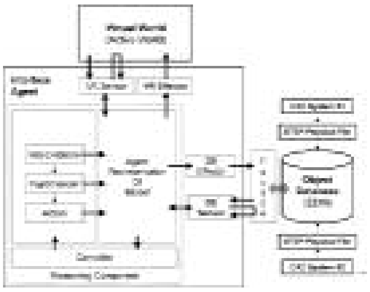

Figure 4. Framework for an agent-centric approach to data sharing.

the virtual world in this case. The reasoning component provides the knowledge for perception, conception, hypothesizer and action.

The database sensor (DB Sensor) has two functions: it senses the database for any modifications and looks out for specific data within the database as directed by the model of reasoning. As new data is added or old data is modified, the database will trigger the sensor by pushing data into the agent. This data is structured by interpretation into a form suitable for reasoning within the agent. When the sensor is triggered by the reasoning model to look for specific data within the database, it pulls the required data from the database.

The virtual world sensor (VW Sensor) also has two functions: it senses the virtual world for any modifications to objects within it and it looks out for specific data within the world as directed by the model of reasoning. As new entities are added or old entities are modified, the virtual world will trigger the sensor by pushing data into the agent. This data is structured by interpretation into a form suitable for reasoning within the system. When the sensor is triggered by the reasoning component, it looks for specific data within the world and pulls the data into the agent for processing.

The reasoning component deals with the consistency of the data within a specific application and data shared across different applications. In the case of the virtual environment, any modifications and creation of relevant data in the virtual world or within the database are propagated to each other.

Implementation

The agent framework has been implemented in Java with the knowledge core of each agent coded in Jess , a rule-based production system language. The following scenario illustrates the implementation:

• An ArchiCAD file includes a model of a simple building that consists of four walls. These walls are exported from ArchiCAD as a neutral file based on IFC's and this file is imported into an EDM database.

• The walls agent senses data from the EDM database and constructs a group of wall agents, one for each wall in the ArchiCAD model.

• Each wall agent converts the wall geometry from the IFC format to the rwx format used in Active Worlds, and then constructs a 3D wall object in Active Worlds that corresponds to one of the walls in the four-wall assembly in ArchiCAD. The walls agent monitors changes in ArchiCAD via the EDM database using the DB sensors, and updates these changes in Active Worlds using the VW effectors.

• The wall agents monitor changes in Active Worlds via the VW sensors and passes the interpreted sense data on to the walls agent. The walls agent updates these changes in the EDM database via the DB effectors, and consequently updates the ArchiCAD model.

Roles of Walls Agent and Wall Agents

The walls agent creates a wall agent for each wall in the building. The walls agent communicates with both the EDM database and Active Worlds while wall agents focus on supporting intelligent behaviours within Active Worlds. The walls agent maintains the consistency of geometrical data in the EDM database and Active Worlds by controlling two reversible processes of information flows and creates the required wall agents (Figure 5). The following are the two reversible information flows controlled by the walls agent:

• From the EDM database: the walls agent senses the wall assembly of four separated walls, and creates four separate wall agents and passes the interpreted sense data from the EDM database on to the wall agents

• From Active Worlds: the walls agent senses the changing location of different 3D wall objects during a collaborative session via the wall agents, and updates the EDM database.

Wall agents build 3D wall objects in Active Worlds and reason about the objects in Active Worlds. They assist the walls agent in completing the information flow from the EDM database to Active Worlds by each receiving the EDM sense data of a specific wall, and create the respective 3D wall object in the Active Worlds.

Conclusion and Future Work

Figure 5. Roles of walls agent and wall agents.

The agent approach to supporting collaborative design described in this paper facilitates synchronous collaboration and real time multi-user multi-disciplinary modification of shared geometrical and non-geometrical data during a collaborative design session. This is implemented as an agent framework that integrates the functionalities of different sensors, effectors and reasoning components. The agent framework serves as a foundation upon which subsequent reasoning components can be added for further reasoning. The association of an agent with each object component of the 3D virtual world integrates design knowledge with 3D models, which allows the designers to interact with objects while moving around and within the design.

Future work includes expansion of current demonstration to consider agent models for more building elements besides walls, and further development of the reasoning component to allow more reflective behaviours, such as reasoning about the fire risk of a building. Constraints contained within the EDM database will drive the agent behaviour by indicating any invalid data modifications. The hypothesizer will consider alternative solutions in addition to indicating invalid modifications when the agent operates in a reflective mode.

References

Smythe, J. S. (ed.): 1990, Applications of Artificial Intelligence to Communication, CMP and Springer-Verlag, Berlin.

EPM Technology.: 2002, EDMAssist4.5, vols 1-5, Norway: EPM Technology.

Gamma, E., Helm, R., Johnson, R. and Vlissides, J.: 1995, Design Patterns, Addison-Wesley, MA.

Gero, J. S.: 1998, Conceptual Designing as a Sequence of Situated Acts, in Smith I. (ed.), Artificial Intelligence in Structural Engineering, Springer, Berlin, pp. 165-177.

Gero, J. S.: 1990, Design Prototypes: a Knowledge Representation Schema for Design, AI Magazine, 11(4), pp. 26-36.

Huhns, M. N. and Stephen, L.: 1999, Multiagent Systems and Society of Agents, in Weiss, G. (ed.), Multiagent Systems, A Modern Approach to Distributed Artificial Intelligence, MIT Press, MA, pp.79-120.

IAI.:2000, IFC Specifications, www.iai-international.org.

Maher, M. L. and Gero, J. S.: 2002, Agent Models of Virtual Worlds, ACADIA 2002: Thresholds, CA: California State Polytechnic University, Pomona, pp. 127-138.

Maher, M.L., Liew, P., and Gero, J. S.: 2003, An Agent Approach to Data Sharing in Virtual Worlds and CAD, Proceedings of the CIB W78 Conference on IT in Construction, Auckland, New Zealand.

Maher, M.L.and Simoff, S.: 2000, Collaboratively Designing within the Design, Proceedings of Co-Designing 2000, pp. 391-399.

Moloney, J, Amor, R., Furness, J., and Moores, B.: 2003, Design Critique Inside a Multi-Player Game Engine, Proceedings of the CIB W78 Conference on IT in Construction, Auckland, New Zealand.

Saunders, R and Gero, J. S.: 2001, A Curious Design Agent, in Gero, J. S., Chase, S. and Rosenman, M. (eds), CAADRIA'01, Key Centre of Design Computing and Cognition, University of Sydney, pp. 345-350.

Shalloway, A. and Trott, J.R.: 2001, Design Patterns Explained: A new Perspective on Object-oriented Design, Addison-Wesley.

Shiratuddin, M and Thabet, W.: 2003, A Framework for a Collaborative Design Review System Utilizing the Unreal Tournament (UT) Game Development Tool, Proceedings of the CIB W78 Conference on IT in Construction, Auckland, New Zealand.

Smith, G. and Gero, J. S.: 2001, Situated Design Interpretation Using a Configuration of Actor Capabilities, in Gero, J. S., Chase, S. and Rosenman, M. (eds), CAADRIA'01, Key Centre of Design Computing and Cognition, University of Sydney, 2001, pp. 15-24.

Wooldridge, M.: 1999, Intelligent agents, in Weiss, G. (ed.), Multiagent Systems, A Modern Approach to Distributed Artificial Intelligence, MIT Press, MA, pp.27-78.

Wooldridge, M. and Jennings, N. R.: 1995, Intelligent Agents: Theory and Practice, in Knowledge Eng. Review 10(2), pp. 115–152.