Beetz, J., J. van Leeuwen, and B. de Vries, 2004, Towards a Multi Agent System for the Support of Collaborative Design, In: Van Leeuwen, J.P. and H.J.P. Timmermans (eds.) *Developments in Design & Decision Support Systems in Architecture and Urban Planning*, Eindhoven: Eindhoven University of Technology, ISBN 90-6814-155-4, p. 269-280.

Towards a Multi Agent System for the Support of Collaborative Design

Assembling a toolbox for the creation of a proof of concept

Jakob Beetz, Jos van Leeuwen, and Bauke de Vries
Eindhoven University of Technology

Keywords: Multi Agent Systems, Design & Decision Support Systems, Collaborative Design, Human Computer Interfaces, Machine learning, Data Mining

Abstract: In this paper we are drafting the outline of a framework for a Multi Agent System (MAS) for the support of Collaborative Design in the architectural domain. The system we are proposing makes use of Machine Learning (ML) techniques to infer personalized knowledge from observing a users' action in a generic working environment using standard tools such as CAD packages. We introduce and discuss possible strategies to combine Concept Modelling (CM)-based approaches using existing ontologies with statistical analysis of action sequences within a domain specific application. In a later step, Agent technologies will be used to gather additional related information from external resources such as examples of similar problems on the users hard disk, from corresponding work of team-members within an intranet or from advises of expert from different knowledge domains, themselves represented by agents. As users deny or reward resulting proposals offered by the agent(s) through an interface the system will be enhanced over time using methods like Reinforced Learning.

1. INTRODUCTION

1.1 AEC in global markets

While the complete shift of paradigms from paper-based documents to electronic data processing in the Architecture / Engineering / Construction (AEC) industry has almost been completed, new challenges have emerged.

The increasing number of (globally) distributed collaborative work environments in the recent years has created demands for new software applications to address a number of problems unknown in traditional (locally organized, non-distributed) surroundings:

- Great diversity of software tools and their respective Human Computer Interfaces (HCI), file formats and methodological approaches. Today, no standard formats exist that guarantee the complete, seamless and lossless exchange of building related information.
- Diversity of jurisdictional requirements for building projects. Although within the European context common law standards in some areas of the building industry are being investigated and have partly been implemented, adoption to foreign standards remains one of the key issues in the growing field of international projects.
- Cultural diversity such as building standards, language, working habits. The translation of technical terms, different dimensioning units, divergent drawing semiotics and varying local working cultures often create additional workload and potential error sources.
- Diversity of time zones. Especially in overseas collaborative projects the lack of isochronous office hours forces a participants to adhere to asynchronous communication protocols. Often these are regarded as uncomfortable due to long feedback cycles that are bound to create delays in the overall progress of a project.
- Pressure of time spiral. The more advanced information technology gets, and the faster general production cycles in other industrial fields become, the higher the expectations regarding flexibility, speed and quality become in the building industry. As tools and working methods in the everyday practice often lack behind, these high expectations are often difficult to meet.

1.2 Traditional approaches in Computer Supported Collaborative Work

Over the past years, techniques and applications have been researched and developed that are increasingly getting integrated into the workflow of AEC offices. As many of the collaboration problems are shared by other industries, a wide range of general-purpose techniques and tools have been taken over or adapted from other collaborative environments. However, a lot of effort has also been made to address the domain specific issues in the AEC area. They can be classified roughly into two categories:

- **Centralized databases** using integrated building information and product models: A universal model for the description of buildings over their complete lifecycle (from early conceptural sketches to construction, maintenance and demolition) has been one of the strongest aims of research and development since the early days of Computer Aided Design in the building industry. However, despite the large amount of approaches to capture all related data, none of the models suggested has been agreed upon. Examples for this category are:
    - **IFC:** In order to develop a common description format that covers all possible aspects of a building over its complete lifecycle, the International Alliance for Interoperability (IAI) was founded in 1994. The original intention to extend the ISO STEP model resulted in a separate standard for the AEC industry, the Industry Foundation Classes (IFC) and its XML-based derivate ifcXML.
    - **IFD for IFC:** Based on the International Standard for Dictionaries, a number of European Projects have begun to develop dictionaries for the (machine-) translation of AEC data between different applications and languages. The Dutch STABU Lexicon project is one of its representatives.
- **Multi-channel communication techniques**. A wide range of asynchronous (e-mail, message boards, group scheduling applications, shared document repositories etc.) and synchronous (instant messaging, video conferencing, 2D/3D whiteboards, application sharing etc.) collaboration techniques have been adapted and integrated into many academic and industrial groupware frameworks. Exemplary works in this category are among others:
    - Fruchter and have developed a system to record and play whiteboard sessions in a multi-disciplinary AEC project for the documentation of the design and working process (Fruchter, and Yen, 2000)
    - Seichter has developed tools to support the distributed architectural design in shared virtual environments (Seichter, 2003)
    - Elger et al have gathered experiences with web-based groupware solutions in many distributed collaboration

projects over a long period of time (Elger and Russel, 2001)

1.3 Remaining problems

Some of the paradigmatic foundations that most of these collaboration techniques are based upon seem to have come to a point, where traditional approaches are bound to face difficulties to get much further:

- Integrated, complete Product Model: A complete Ontology is difficult if not impossible to establish and its technical implementation difficult to maintain. Too many aspects of the various fields involved have to be integrated.
- The integration of a common, unified file format has not been successful yet. The reasons for this are manifold: Technological difficulties to incorporate ever-changing standards (especially for smaller companies), the commercial interest to establish proprietary formats and others.
- Difficulty to maintain large monolithic applications and to adapt them to new situations
- Communication overheads (phone calls to negotiate settings for the video conference etc.) often create more additional workload than the gain from using this technology. This however may vanish as the use of these techniques gets more established.

Hence new technologies should be investigated for their possible use in this area.

2. THE MULTI-AGENT PARADIGM

2.1 Foundations of MAS

Artificial Intelligence (AI) and its sibling Distributed Artificial Intelligence (DAI) has long history in the research community. However, the term ‘agent’ has been used in a lot of different, contradictory and misleading ways, especially since “intelligent agents” became mainstream hype in the booming area of the WWW. To clarify the use of the term agents in the context of this paper, we will give a brief overview of definitions commonly found in literature that we think denote the topic best.

Agents are considered to be software units that exhibit

- Modularity: Agents are small, specialized units designed for the execution of simple tasks.

- Situatedness: Agents are embedded into a continuous environment whose current state is perceived by their sensors and manipulated by their effectors.
- Proactiveness: Agents are not purely reactive software modules that have to be invoked by a user specifically. They incorporate a set of own intentions and objectives that let them act on their own (Even if this happens on behalf of a human user).

A collection of many agents enabled to communicate with each other is generally referred to as a society of agents. Here, additional concepts come into view:

- Mobility: Agents in a MAS are able to (physically) change their location
- Communication: Agents in a MAS are able communications / speech acts
- Synergy: New results emerge from interaction between agents. The outcome is more than the sum of its individual actors

Janca and Gilbert elaborate on the notion of agents as personal assistants in (Janca, P.C. and Gilbert, D., 1998). User interface agents, sometimes also referred to as Personal Digital Assistants (Maes, Liebermann)

Lanier argues, that the (unnoticed) blending of 'intelligent' agents should be avoided as it enslaves the human user by adapting his own behaviour to the agents capabilities. (Lanier, J., 1995)

General introductions to Artificial Intelligence can be found in (Russel, S. and Norvig, P., 1995) and (Nilsson, J.N., 1998) the foundations of Multi Agent Systems and Distributed Artificial Intelligence are covered in (Weiss, G., 1999), (Ferber, J., 1999) and (Jennings, N.R., Wooldridge, M.J, 1998)

2.2 Implementations and usage of MAS in industrial environments

Multi Agent Systems have been successfully applied in a number of industrial-scale applications. The widest range of applications comes from the fields of telecommunication (Weihmayer, R. and Velthuijsen, H., 1995), manufacturing (Shen, W. and Norrie, D.H., 1999), workflow and process management (Jennings et al, 1996), (Singh, M.P. and Huhns, M.N., 1994). An overview of industrial implementations of Multi Agent Systems can be found in (Parunak, H.V., 1998).

Interesting implementations also stem from the Information retrieval and brokering background.

A number of implementations with promising findings for collaborative design in the AEC field can be found in the mechanical engineering field. Grecu and Brown investigated the use of single function agents in spring design (Grecu and Brown, 1996), Campell, et al developed “A-Design”, a multi agent system for electromechanical design (Campbell, et al, 2000). Cutosky et al. presented an early electro-mechanical concurrent engineering system based on agent technology to communicate, negotiate and share knowledge from different reasoning agents and human users in a heterogeneous environment: the Palo Alto Collaborative Testbed (PACT) (Cutosky et al, 1998)

Unfortunately, implementation of the achievements made in these fields into area of collaborative design in AEC is not possible in a one-to-one manner. Designs from the area of mechanical engineering are composed of assemblies of single parts in most cases. The representation of designs by rigid bodies with properties like mass, moment of inertia, dynamic interaction behavior and so forth is very suitable for the integration into computational environments. The degree of formalization of concepts in the design process found in the building industry does not come close to formalization that has been done in these fields. The ongoing efforts and struggles to find suitable formalizations of the architectural design process is unknown in or solved long ago in many of the aforementioned fields. To make the technology available to the AEC domain, either further work in its formal description has to be done or the use of Multi Agent Systems has to be adapted.

The following section will give a short overview of the contributions that have been done to harness the MAS concepts in the AEC industry.

2.3 Research on MAS in AEC

Contributions made to the application of Multi Agent techniques to address issues in the AEC domain can roughly be categorized into three domains:

MAS for knowledge capturing and recognition in drawings and sketches: Based on previous findings in the fields of classification of architectural plans and sketches, several agent based systems have been developed and proposed (Achten, and Jessurun, 2002), (Leclercq, 2001). Although most often in early prototype stages these contributions are promising for future use in collaborative environments. Important aspects of these works are their formalizations of what still remains the most important means for communicating information in the AEC domain: Graphics.

MAS for simulation and performance of building designs: In the fields of building performance evaluations, especially in those that model and simulate user behaviour, several contributions using multi agent systems have been made. Dijkstra and Timmermans have created AMANDA to simulate pedestrian flows in urban environments, (Dijkstra, 2003).

MAS in collaborative environments: Meissner et al. (Meissner et al., 2004) have created an agent-based system for the support and integration of fire protection engineering into the planning process. Bilek et al (Bilek, et al, 2003) have presented prototypical implementations of MAS for the planning of civil engineering projects and the monitoring of dam structures during a reconstruction project.

2.4 Requirements

Among the requirements of geographically distributed collaborative design and construction environments in the AEC domain, the adaptability to heterogeneous infrastructures is one of the most important ones. Financial and training investments that have been made in software by the domain experts have to be integrated in quickly changing team environments. The quick and effortless establishment of new project-oriented team set-ups is another urgent requirement to be addressed by collaborative software solutions. The coverage of a building over its complete life-cycle with a wide range of different data to be captured also imposes new strategies for a loss-less hand-over of data between the different phases.

3. MULTI-STAGE TRANSITION TOWARDS AN INTEGRATION OF LEGACY APPLICATIONS

3.1 Basic layout

In this section we will describe and propose multiple steps towards an agent based full integration of legacy applications in a collaborative design environment. Every step introduces a new layer with increasing complexity and difficulty to achieve. While the first steps are within achievable bounds and are not relying on any given external infrastructure, proposals of higher levels require the existence of other agent-based services to be useful. Especially these later steps are an outlook to what could be achieved should agent technology be embraced by the industry.

1. **Identification of task and task domain**. Situated in some stage of a collaborative design, an agent with a specific reasoning capability will identify the current task at hand. For the integration of the agents into legacy applications, simple mechanisms have to be found to make an effortless extension of

such generic applications possible. As the integration of such agents via APIs, plugin-APIs or other interfaces that a specific application offers lead to a great workload for the developers of such a system, we suggest the exploration of other user monitoring options. One of the approaches we suggest is to identify strings of natural language coming from the GUI (via menus, dialog boxes and so forth) of the specific application. (On MS Windows systems this can be achieved by hooking up to the Win32 message cue and filter out the relevant calls between application and operating system) The user agent then asks agents in its vicinity whether more information for the specific actions / identified objects is available

2. **Retrieval and compilation of related information**. After identifying the current activity of the designer and reasoning about his/her intention in the previous stage, available related information is gathered. Depending on the situation at hand, resources within the design team or external resources will be consulted for relevant material. For internal consultation, those team members have to be identified who are affected by the decision made by a designer or who can offer additional advise for a class of problems. Agents are used to represent a user's expertise, knowledge domain and portfolio. This is done by advertising the specific design task and broadcasting it among team agents. External resources, located i.e. on the WWW could be found be central service marketplaces where expert agents of specific knowledge domains offer their services. Several technological approaches are under development in other areas that will be introduced in section 3.2.
3. **Preparation and presentation of choice of solutions**. The information found in the previous steps will be presented to the user in a meaningful way. An interface will be developed, that seamlessly extends the legacy application with new functionality i.e. by overlay of texts, pictures or suggestions of 2D/3D model fragments. The main focus on this stage is to keep away additional (cognitive) workload and annoyances from the user.
4. **Offer of problem specific advice**. In previous stages, no real problem specific 'intelligence' was introduced. While in former stages general classes of solutions or solution strategies are retrieved and presented to the user (who in turn has to adapt these to the specific problem at hand), solutions in this state should be tailored to the specific needs of a unique problem instance. This implies, that recognition and reasoning not only happens on an abstract conceptual level.

5. **Consideration of personal preferences for provision of solutions**. As the personal problem solution strategies of designers are subject to personal preferences and individual characteristics, an advanced system on this stage should be able to adapt to these. This means that a pre-selection of solutions should be offered to the user and that the user agent acting on behalf of this user must make the personal preferences part of the requirements during the formulation process of information gathering.
6. **Full integration of inhomogeneous services into work environments**. In this ideal last stage all experts in a design project are interconnected and have a large choice of possibilities to share their specific knowledge with other team members or within the context of a larger community.

The stages suggested in the last section are neither mutually exclusive to each other nor completely separable. While some of the stages depend on each other or share some common algorithms and system architectural structures, others can exist and work independently from each other, forming a range of tools in heterogeneous environment with common protocols.

3.2 Available technologies and tools

The situation today, the requirements for future applications and the possible steps to meet them, as described in the previous sections of this paper require a large set of advanced technology from the fields of Multi Agent Systems and the Semantic Web.

Fortunately, a wide range of APIs, communication protocols and formats has been developed in recent years. These are ready to be used in an AEC context without the need for the creation of basic solutions from scratch. To come to the point where an integration of as many different aspects during the design and building process is made possible however, the choice of tools and their future compatibility and extendibility is a crucial issue. In this section we are giving a short overview of existing software tools for the creation of MAS environments that are available today. The use of agent technology frameworks, development environments, APIs and examples were reviewed at large extends by Eiter and Mascardi. (Eiter, T. and Mascardi, V, 2002).

Multi Agent System APIs and communication protocols: Historically stemming from the early DARPA program the Knowledge Query and Manipulation Language (KQML) and the Knowledge Interchange Format (KIF) have been the dominating agent communication protocols in literature and industry for a long time. While KQML is a three-layered speech act protocol between agents, KIF is a standard to capture the contents of these speech acts – the knowledge – itself. In recent years however, a slight shift towards FIPA ACL, the Agent Communication Language standard developed by the Foundation for Intelligent Physical Agents consortium can be noticed in both research and development (Labrou, Finin, Peng, 1999). Today a wide range of different (most often JAVA-based) APIs supporting both KQML and FIPA ACL are available. Among the most popular are JATLite, Zeus, Aglets and AgentBuilder. An extensive survey on available toolkits, frameworks and APIs can be found in (Agentlink.org, 2002)

Semantic Web technologies: While agent technology in general has a much broader field of application, extensions for a machine-readable and machine-processable WWW as mentioned in parts of this paper focuses on different aspects. Although many concepts of the Semantic Web are directly or indirectly inspired by and founded on AI technology new concepts and standards have been added and its common use pushes its development further on a much higher pace than AI techniques. Today, the most established technique from the huge set of proposals, frameworks and concepts developed under the roof of the W3C is RSS. The RDF Site Summary format is currently being used in a broad range of applications to summarize and syndicate news sites, weblogs and WiKis. RDF, the Resource Description Framework in turn has been used as a basis and extended for much more technologies such as DAML-OIL and OWL for capturing knowledge. Among the most important tools to create, process and view knowledge are Jena, Protégé, OntoBuilder, the JRDF API.

A general introduction to the Semantic Web can be found in (Fensel, D., Hendler, J. and Lieberman, H., 2003)

3.3 Steps towards a prototype

In order to verify our suggestions several prototypes will be designed, implemented and tested. Starting from the first layer, aforementioned tools and APIs will be made use of.

For a first test scenario communication processes between two domain specific applications that are used by students during a design task will be monitored and analysed. For example, the file exchange between an architectural application such as AutoCAD™ and an FEM package like Marc Mentat™ and the (presumably repetitive) steps to prepare, ex- and import the data, run the analysis an propagate the findings back to the original design will be logged by a first prototype. Sample data from a series of experiments in a student project will then be used to train agents that will suggest shortcuts or take away workload by predicting next steps and developing solutions on behalf of the user.

The findings from this first step are then used in later steps.

Later stages of the steps proposed require as much applications available on public resources, that exhibit the following properties

1. Standard interface for the integration of agent systems into legacy applications (Common plugin interfaces)
2. Ontology matching and transition capabilities
3. Common Agent communication language(s) and protocol(s)
4. Open, transparent interfaces

4. CURRENT STATE AND NEXT STEPS

In the current early stages of the project, a literature study has been carried out and rough general strategies and concepts to address the problems in the AEC domain identified in an analysis phase have been developed. Next steps include the further narrowing down of concrete ideas and strategies and the design of prototypical solutions using tools mentioned in this paper.

REFERENCES

Agentlink.org, 2002, *Review of Software Products for Multi-Agent Systems,* http://www.agentlink.org/resources/software-report.php

Achten, H. and Jessurun, J., 2002, “An Agent Framework for Recognition of Graphic Units in Drawings”, Proceedings of 20th eCAADe, pp. 246-253

Bilek, J.; Mittrup, I.; Smarsly, K.; Hartmann, D., 2003 "Agent-based concepts for the holistic modeling of concurrent processes in structural engineering", in: *Proceedings of the 10th ISPE International Conference on Concurrent Engineering: Research and Applications,* Madeira Island, Portugal, pp. 47-53

Campbell, M., J. Cagan and K. Kotovsky, 2000, "Agent-based Synthesis of Electro-Mechanical Design Configurations," in: *Jnl of Mechanical Design*, Vol. 122, No. 1, 61-69.

Cukosky, M.R, EngelMore, R.S., Fikes, R.E.,.Genereseth, M.R, Gruber, T.R., Mark, W.S, . Tenenbaum J.M and Weber; J.C., *PACT: An Experiment in Integrating Concurrent Engineering Systems.* IEEE Computer 26(1), pp 28-38

Eiter, T. and Mascardi, V. , 2002, “Comparing environments for developing software agents”, in: *AI Communications* Vol. 14, Issue 4.

Elger, D. and Russel, P., 2001 “Net-based Architectural Design: The Difficult Path from the Presentation of Architectural Design in the World Wide Web to Teamwork in Virtual Planning Offices: A Field Report”, in: Proceedings of the 19th eCAADe Conference , Helsinki, pp. 371-375

Fensel, D., Hendler, J. and Lieberman, H. (ed.), 2003 *Spinning the semantic web : bringing the world wide web to its full potential*, MIT Press, London

Ferber, J., 1999, *Multi-Agent System. An Introduction to Distributed Artificial Intelligence*, Addison Wesley Longman Inc., New York.

Fruchter, R., Clayton, M., Krawinkler, H., Kunz, J. and Teicholz, P. 1993, “Interdisciplinary communication medium for collaborative design”, in: *Proceedings of the Third International Conference on AI in Civil Engineering*, Edinburgh, pp. 1—28

Fruchter, R. and Yen, S., 2000, *RECALL in Action*, Proceedings of ASCE ICCCBE-VIII Conference, pp.1012-1021.

Grecu, D.L., Brown, D.C, 1996, Learning By Single Function Agents During Spring Design, in: Gero, J. and Sudweeks, F. (eds.), *Artificial Intelligence in Design '96*, Kluwer Academic Publishers, Boston, pp. 409-428

Shen, W. and Norrie, D.H., 1999, “Agent-Based Systems for Intelligent Manufacturing: A State-of-the-Art Survey”. *Knowledge and Information Systems,* **1**(2), 129-156, 1999.

Janca, P.C. and Gilbert, D., 1998, “Practical Design of Intelligent Agent Systems”, in Jennings, N.R., Wooldridge, M.J, (eds.), *Agent Technology: foundations, applications and markets*, Springer-Verlag, Berlin, pp. 73-89

Jennings, N.R., Faratin, P., Johnson, M.J., Norman, T.J., O’Brien, P. and Wiengand, M.E., 1996, “Agent-based business process management”, in: *International Journal of Cooperative Information Systems* 5(2&3), pp. 105-130.

Jennings, N.R., Wooldridge, M.J, (eds.) 1998, *Agent Technology: foundations, applications and markets*, Springer-Verlag, Berlin.

Labrou, Y., Finin, T. and Peng, Y., 1999, “Agent communication languages: the current landscape”, *IEEE Intelligent Systems,* Vol. 14, Issue 2, pp.45-52

Lanier, J., 1995, “Agents of alienation”, in: *IEEE Interactions* Vol. 2, Issue 3, pp. 66-72

Leclercq, P.P., 2001, “Programming and Assisted Sketching. Graphic and Parametric Integration in Architectural Design”, *Proceedings of the Ninth International Conference on ComputerAided Architectural Design Futures*, pp. 15-31

Maes, P., 1994, "Agents that reduce work and information overload," *Communications of the ACM* 37:7(1994), pp. 30-40

Meissner, U.; Rüppel, U.; Theiß, M., 2004, "Network-Based Fire Engineering Supported by Agents.", in: *Proceedings of the Xth International Conference on Computing in Civil and Building Engineering (ICCCBE)* Weimar, pp. 103

Minsky, M., 1985, *The Society of Mind*, Simon and Schuster, New York.

Nilsson, J.N., 1998, *Artificial intelligence : a new synthesis,* Morgan Kaufman, San Francisco

Parunak, H.V., 1998, “ Industrial and practical applications of DAI”, in *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*, The MIT Press, Cambridge Massachusetts, pp 377-421.

Russell, S. and Norvig, P., 1995, *Artificial Intelligence. A Modern Approach*, Prentice-Hall International, Inc., Englewood Cliffs, New Jersey

Schnabel, M.A., Kvan, T., Kruiff, E. and Donath, D., 2001, “The First Virtual Environment Design Studio, Architectural Information Management”, *Proceedings of the 19th Conf. on Education in Computer Aided Architectural Design in Europe*, Helsinki, pp 394-399

Seichter, H., 2003, “Collaborative Augmented Sketching” ,in *Proceedings of the 2nd International IEEE AR Toolkit Workshop* 2003, Tokyo

Singh, M.P. and Huhns, M.N., 1994, “Managing Heterogeneous Transaction Workflows with Co-operating Agents”, in: Jennings, N.R., Wooldridge, M.J, (eds.) 1998, *Agent Technology: foundations, applications and markets*, Springer-Verlag, Berlin, pp. 219-239

Weihmayer, R. and Velthuijsen, H., 1995, “Distributed AI and Cooperative Systems for Telecommunications”, in: Liebowitz, J. and Prereau, D. (eds.) *Worldwide Intelligent Systems: AI Approaches to Telecommunications and Network Management*, IOS Press, Amsterdam, Chapter 11, pp. 227-260.

Weiss, G., 1999, *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*, The MIT Press, Cambridge Massachusetts.