An Agent Framework for Recognition of Graphic Units in Drawings

Henri Achten, Joran Jessurun

Eindhoven University of Technology, Netherlands
http://www.ds.arch.tue.nl *H.H.Achten@tue.nl*

Architects use graphic conventions in their drawings that have meaningful content to the design task. In previous work, such well-defined sets of graphic entities have been identified and defined. These sets are called graphic units. In this paper, we discuss how graphic unit recognition in drawings can take place using a multi-agent systems approach. This approach seems promising as singular agents may specialize in graphic unit-recognition, and multi-agent systems can address problems of ambiguity through negotiation mechanisms. We present an agent framework for this purpose, how it connects to the theory of graphic units, and how agents for recognizing graphic units are defined. The paper ends with a discussion of current findings and future work.

***Keywords**: Multi-agent systems; graphic representations; graphic units; design support; graphics recognition.*

Introduction

Graphic representations constitute a major means of communication for the architect; not only for exchanging information between the partners in the design process, but also as a medium for reflecting on the progress of design in the early phases. Architects have a rich collection of conventions of depiction and encoding which are shared within the professional community. In previous work we have identified these in the form of graphic units and generic representations.

In the present work we aim to implement the findings on graphic units in a drawing system that will interpret drawings while architects are drawing. Through this implementation we hope to understand the scope and limitations of the theory of graphic units and generic representations.

Our implementation approach is loosely based on findings from cognitive psychology about working of the mind, in particular its modular nature. We do not aim to make an accurate model of perception, however. As will be explained later in this paper, we note that there are interesting resemblances between mechanisms of mind, multi-agents systems, and the concept of graphic units. For this reason, we turn to multi-agent systems for the implementation. As this work is our first reading of the field, we do not claim comprehensiveness or accurate rendering of the state-of-the-art. We do feel however, that the currently identified threads, when combined, form an interesting and productive direction.

The system as we will develop it, does not directly interpret drawings, but rather operates on a metalayer, which is the drawing system. The assumption here is that an image recognition layer, which can process sketches, can be added later, which will yield the primitives that are the basis in the current system.

The agent framework has originally been

designed for the Amanda project (Dijkstra and Timmermans 2000). In this project agents and cellular automata are combined to simulate human behavior in a built environment. The cellular automata framework has already been implemented (Dijkstra et al. 2002). In a similar way, the agents framework will be implemented on the Microsoft Component Object Model (COM). This gives the possibility to use the framework with a lot of different languages on the Windows platform, and also the possibility to connect it to existing applications such as AutoCAD.

Graphic units and generic representations

Architects have well-established conventions of depiction (how things are drawn) and conventions of encoding (understanding what is drawn) in their graphic representations (Akin 1986). Through these properties drawings constitute a stable medium that can be analyzed further. An analytical technique has been developed which identifies well-defined structures of graphic entities that have a specific meaning to the architect (Achten 1997). These structures are termed 'graphic units'. The following graphic units have been identified (Table 1):

Defined concisely, a graphic unit is a set of graphic entities (such as lines, contours, textual elements) that are organized in a specific way and that have a generally accepted meaning to the architect. The identified graphic units demonstrate that architects have a rich graphic vocabulary to their disposal through which to develop their designs. Furthermore, in Achten (1997; 2000) and Achten et al. (1998) we have outlined how graphic units can be used to acquire, retrieve, and present relevant information to the architect.

In the present work we aim to implement the findings on graphic units in a drawing system that will interpret drawings while architects are drawing. Through this implementation we hope to understand the scope and limitations of graphic units. We choose to look at 'drawing in action' for two main reasons: (1) we will be able to use drawing actions as clues for recognizing graphic units (Kavakli et al (1998), McFadzean (1999)); and (2) understanding the drawing in 'real time' is important if we want to add additional support through the system. This support should happen as the designer is working, in a very short time span after the drawing act.

Related work falls in the category of CAD systems and sketch analysis. CAD systems for sketching and enhancing the sketch have come into recent attention again. Examples for graphic design support are the "Electronic Cocktail Napkin" (Gross 1996), "Hypersketch" and "PHIDIAS" (McCall et al. 1997), "Netdraw" (Qian and Gross 1999), "SketchBoX" (Stellingwerff 1999), "Piranesi" (Richens 1999), transparent sketch-tool (Trinder 1999), and "EsQUIsE" (Leclercq 2001).

Related research in sketching, aimed at identifying pervasive structures is less common: Do (1997) and Do et al (2000) look at commonalities in sketches, in particular short-hands for drawing the same concepts; McFadzean (1999) has devised a Computational Sketch Analyser, which takes several sketch-acting clues as indicators what the current status of the design process is; Koutamanis (2001) proposes a taxonomy of constituent elements in sketches, breaking them down into organizational units not unlike graphic units. Rodgers et al. (2000) note the importance of supporting vertical transformation (refinement), lateral transformations (divergence), and duplications of earlier sketches, without providing a more refined set of criteria however, to track design development.

Multi-agent systems

The notion of 'agent' has come into use since the early '90ies, especially in the field of (Distributed) Artificial Intelligence. Early developments where the term 'agent' was used concerned autonomous systems that would perform set tasks for users (such as filtering mails to stated areas of interest in the form of

Simple contour (1)	Regular shape showing an outline.
Contour (2)	Any irregular shape showing an outline.
Measurement device (3)	Measure for establishing (relative) dimensions.
Specified form (4)	Contour with specified dimensions.
Elaborated structural contour (5)	Outline with structural detail.
Complementary contours (6)	Composition of outlines.
Function symbols (7)	Textual indication of function.
Zone (8)	Area with specific use or function.
Schematic subdivision (10)	Schematic depiction of principal subdivision.
Modular field (11)	General subdivision of area along coordinating lines.
Refinement grid (12)	Grid with smaller module coordinated in other grid.
Schematic axial system (13)	Schematic depiction how axes are organized.
Axial system (15)	Organization of axes applied to a building design.
Grid (16)	System of regular repeating coordinating lines on a module.
Tartan grid (17)	Double grid based on two different modules.
Structural tartan grid (18)	Tartan grid with one band for structural elements.
Element vocabulary (19)	Set of simple shapes depicting (interior) elements.
Structural element vocabulary (20)	Set of simple shapes depicting structure.
Functional space (21)	Outline combined with function indicator.
Partitioning system (22)	Schematic depiction of more detailed subdivision.
Proportion system (23)	Diagram how proportions are derived.
Combinatorial element vocabulary (24)	Precise relationships between particular elements.
Circulation system (26)	Principle layout of circulation.
Circulation (27)	Layout of circulation applied to building design.

Table 1. List of graphic units (Achten 1997, pp. 90-93)

PDA's – Personal Digital Assistants: see Maes 1990). Such agents were typically conceived as embodied entities; sometimes even human-like. On a different track, Minsky (1988) discusses how intelligent behavior could be conceived as the result of a 'society of agents. ' Agents in his view have very limited capacities, but through the organization of many agents a system can display complex or even intelligent behavior. This view focuses on the internal processes of a system: the agents in such a system are not visible to the outside world. Thirdly, robust systems for example used in air traffic control or electrical power supply were developed where the controlling modules would interact and decide autonomously how to deal with real-time fluctuations or in the case of emergencies. These systems were termed 'multi-agent systems. ' Finally, many early applications were hardware – embodied robots – which could manipulate or move about in the real world.

The wide variety in appearance and scope of agents and multi-agents systems lead to great diversity on the definition what an agent is. Woolridge and Jennings (1995) note that a majority of the researchers ascribe to what they call a weak notion of agency; which is any hardware or software that has the properties of autonomy, social ability, reactivity, and pro-activeness.

After the first wave of developmental work in AI, the discipline of Computer Science aimed to develop a more fundamental basis for agents in the form of 'agent technology' (d'Inverno et al. 1997; Aylett et al. 1998). This work leads to a better understanding of the requirements of such systems, and for methodologies and languages for defining multi-agent systems (e. g. Brazier et al 2001). Müller (1998) proposes the following division of agents with respect to agent architectures (Table 2).

These technologies can be used to make (multi) -agent systems. Building on the architectures, Müller proposes the following tentative groups of agent-systems (Table 3).

Research on multi-agent systems is quite young, and for the better part still lacking a sound theoretical basis (Luck 1999). In the next section, we will discuss how multi-agent systems can be used for graphic unit recognition.

Table 2. Agents architectures (Müller 1998, pp. 353-354)

Reactive agents	Agents with little or no internal representation of the environment, acting only on responses to the environment.
Deliberative agents	Agents with comprehensive internal representation of the environment and the capacity to reason and act on this reasoning (mostly through a BDI: Belief-Desire-Intention architecture).
Interacting agents	Agents with the capacity to negotiate and communicate with other agents, possibly using explicit representations of these other agents.
Hybrid systems	Agents with a mix of the above in a layered architecture.
Other agents	Believable agents and softbots/software agents.

Table 3. Possible agent systems (Müller 1998, pp. 370):

1. Autonomous hardware agents	Autonomous Control Systems that operate in embodied form in the real world, e.g. moving vehicles or assembly robots.
2. Autonomous software agents	Programs that exhibit independent behavior in simulated or software environments, e.g. workflow planning
3. Hardware assistant agents	Systems that aid, assist, or entertain humans in embodied form, e.g. in household tasks or mechanical pets.
4. Software assistant agents	Personal Digital Assistants or virtual characters that aid, assist, or entertain humans as software, e.g. mail filters or instructors.
5. Hardware multi-agents	Embodied Autonomous Control Systems that act together in a cooperative manner, e.g. construction robots.
6. Software multi-agents	Systems comprised as programs which act as agents, typically located in dynamic environments, e.g. resource allocation planning or cooperative expert systems.

Agents and graphic unit recognition

The characteristics of the graphic units outline in the first section lead us in the direction of multi-agent systems for implementing graphic unit recognition in drawings. The reasons for this choice are:

- An agent can specialize in recognition of one particular graphic unit, building on other agents that recognize more primitive graphic elements.
- Agents are defined as systems (agents can be defined by lower-level agents), which correlate with the bottom-up definition of graphic units.
- The agents approach puts emphasis on conflict identification and resolution; this seems an appropriate approach to deal with ambiguity in a drawing.
- Functionality is built piecemeal on top of existing agents, so that the system can be developed incrementally.
- Agent-systems can function in dynamically changing environments, where resolution is not always possible. This seems appropriate in the dynamically changing activity of drawing.

As outlined in the section about multi-agent systems, there are many options and approaches available when making multi-agent systems. Since the work concerns graphic unit recognition, which is in some way a visual-cognitive function, we direct our efforts towards a more cognitive approach to agents. Franklin (1995, pp. 17-18) summarizes findings about mind as being composed of separate mechanisms which in a continuous fashion produce as an overriding task the next best action; mind uses memories and sensations to produce information in a constructive process, and it can to some degree be implemented on machines. The modularity of mind is widely accepted in cognitive science and AI (Flanagan 1989; von Eckardt 1993; Franklin 1995). The issue of coordination and control however, is still unclear. Franklin offers a number of control mechanisms, all based on “relatively” simple modules.

The multi-agent system therefore, will consist mostly of simple, reactive agents, much in the way Minsky (1988) and Franklin (1995) propose that complex behavior is achieved. There is most likely no need for these agents to incorporate a Belief-Desire-Intention system. However, at some point, interpretation problems will occur and deliberation must take place to resolve graphic unit recognition. Although deliberation can be assigned to higher-level agents in another layer, we hope to build such activity in the communication structure and behavior between agents.

Agent framework for graphic unit recognition

We have established a framework that has the following characteristics: (a) based on a systems-theoretical view; (b) direct message exchange from one agent to another, and communication through channels inspired by colored Petri-nets; (c) implemented as a discrete system; and (d) implemented using Microsoft’s Component Object Model so it can be used with many languages and applications.

Agent definition

An agent has an internal state and processes that are closed to the outside world. The agent has contact through input and output. The input part deals with sensing the world environment and receiving messages from other agents either as direct communications from specific agents or as broadcasts. The output part deals with manipulating the world environment, sending messages to specific agents, and broadcasting messages to all agents in the system (see Figure 1).

Systems theoretical view: making agents by agents

From the systems-theoretical view, where elements and relations build up systems, and where elements also can be systems, we propose to build agents by simpler agents. In the spirit of work by Minsky (1988) however, agents are not exclusively nested in one single higher-level agent, but can be used by other

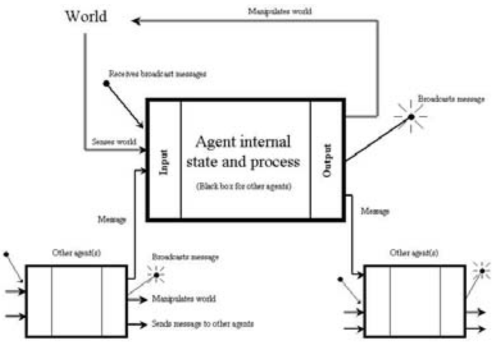

Figure 1. Principle of multi-agent system.

agents as well. The implemented system then, through a series of more complex operations, can recognize graphic units (Figure 2). Mechanisms such as developed by Pelliteri (1997), Cha and Gero (1998), McFadzean (1999), and Park and Gero (2000) may prove helpful in the 'shape' to 'form' and 'form' to 'assembly' transitions. Most deliberation will occur in the transition from 'assembly' to 'interpretation.' Here agents representing graphic units will be productive limiting the combinatorial explosion of complex shapes and shape complexes through the search for grids, zones, contours, axial systems, and so forth.

Communication principles

Agents communicate using messages. In order to determine to which agents a message should be sent, we use message filters. These filters can use the state of an agent as a determinant whether this agent should receive the message (*e. g.* age, time passed since last activation, competence, etc.). The message filters are built in a simple language that describes the required state of the agent for receiving a particular message. Each agent has its own message queue in which it receives its messages.

To enable broadcasting as described above, we use channels. A channel is a global message queue that multiple agents can use to get messages from. The working of a channel is based on the concept of places from colored Petri-nets. An agent inherits its capability to use a channel in a particular way from its parent agent.

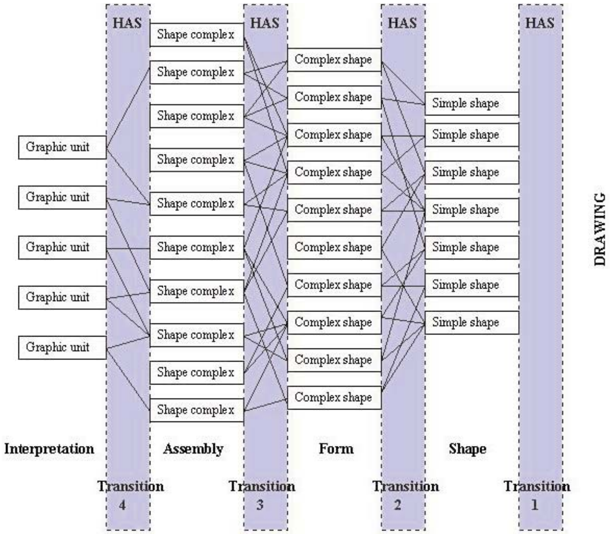

Figure 2. Transition from simple to complex recognition.

Discrete system

The state of the agent is a collection of variables kept by the agent that can contain almost any data from primitives such as integers to objects such as other agents. A step method at top level will fire step events for all agents in the hierarchy. This will be done in a bottom-up order.

Discussion and future work

The present paper gives a theoretical outline of the project. It is work-in-progress before the implementation phase. This means that in particular our starting position relative to the kinds of agents to use and the communication processes and behavior will be tested in the implementation phase. In order to get a better understanding of the dynamics and characteristics of multi-agent systems, we first tackle a simpler problem than graphic unit recognition, through the implementation of a multi-agent system that can play Mah Jong solitary (the system has no need for ambiguity resolution and can be used to test various move-choice behaviors).

The present work is based on graphic units, and therefore it is inherently two-dimensional and oriented towards the sketching and drawing activities of the architect. One can reasonably argue that current developments in CAAD and its use in practice is increasingly three-dimensional from the very start, using quite different and new classes of representation. With respect to these new developments, it still has to be seen whether a stable and rich three-dimensional equivalent of graphic units will develop.

Acknowledgements

The current work is performed within the VR-DIS research programme, which is part of the Design Decision Support Systems programme at the Department of Architecture, Building and Planning at Eindhoven University of Technology. The authors would like to acknowledge the benefit of ongoing presentations and discussions about the use of agent technology within this research programme, in particular Jan Dijkstra for commenting on the paper. Part of the work is developed in a sabbatical leave at the Czech Technical University in Prague (CVUT), at the Faculty of Architecture, in the CAAD group (GAPPA). The authors would like to thank Doc. Ing. V. Ku? era for accommodating and supporting the sabbatical leave.

References

Achten, H. H.: 1997, Generic Representations – An Approach for Modelling Procedural and Declarative Knowledge of Building Types in Architectural Design, Ph. D. Thesis, Eindhoven University of Technology, Eindhoven.

Achten, H. H., Oxman, R. M. and Bax, M. F. Th.: 1998, Typological knowledge acquisition through a schema of generic representations, in Gero J. S. (ed.) Proceedings of Artificial Intelligence in Design ’98, Kluwer Academic Publishers, Dordrecht, pp. 191-207.

Achten, H. H.: 2000, Design case retrieval by generic representations, in Gero, J. S. (ed), Artificial Intelligence in Design’00, Kluwer Academic Publishers, Dordrecht, pp. 373-392.

Akin, Ö.: 1986, Psychology of Architectural Design, Pion Limited, London.

Aylett, R., Brazier, F., Jennings, N., Luck, M., Nwana, H. and Preist, C.: 1998, Agent systems and applications, The Knowledge Engineering Review, 13 (3), pp. 303-308.

Brazier, F. M. T., Jonker, C. M., Treur, J., and Wijngaards, N. J. E.: 2001, Compositional design of a generic design agent, Design Studies 22 (5), pp. 439-471.

Cha, M. Y. and Gero, J. S.: 1998, Shape pattern recognition using a computable pattern representation, in Gero, J. S. and Sudweeks, F. (eds), Artificial Intelligence in Design ’98, Kluwer Academic Publishers, Dordrecht, pp. 169-187.

D’Inverno, M., Fisher, M., Lomuscio, A., Luck, M., Rijke, M. de, Ryan, M. and Wooldridge, M.: 1997, Formalisms for multi-agent systems, The Knowledge Engineering Review, 12 (3), pp. 315-321.

Do, E. Y-L.: 1997, Computability of design diagrams, in Junge, R. (ed), CAAD Futures 1997, Kluwer Academic Publishers, Dordrecht, pp. 171-176.

Do, E. Y-L., Gross, M. D., Neiman, B. and Zimring, C.: 2000, Intentions in and relations among design drawings, Design Studies, 21 (5), pp. 483-503.
Dijkstra, J. and Timmermans, H. J. P.: 2000, Towards a Multi-Agent System for Network Decision Analysis, Design Systems Report 2000/2, Design Systems, Technische Universiteit Eindhoven.
Dijkstra, J., Jessurun, J. and Timmermans, H.: 2002, A multi-agent cellular automata model of pedestrian movement, in Schreckenberg, M. and Sharma, S. D. (eds.), Pedestrian and Evacuation Dynamics, Springer-Verlag, Berlin, pp. 173-179.
Eckardt, B. von: 1993, What is Cognitive Science? The MIT Press, Cambridge, Massachusetts.
Flanagan, O. J.: 1984, The Science of the Mind, The MIT Press, Cambridge, Massachusetts.
Franklin, S.: 1995, Artificial Minds, The MIT Press, Cambridge, Massachusetts.
Gross, M. D.: 1996, The Electronic Cocktail Napkin – a computational environment for working with design diagrams, Design Studies, 17 (1), pp. 53-69.
Kavakli, M., Scrivener, S. A. R., and Ball, L. J.: 1998, Structure in idea sketching behavior, Design Studies 19 (4), pp. 485-517.
Koutamanis, A.: 2001, Prolegomena to the recognition of floor plan sketches – A typology of architectural and graphic primitives in freehand representations, in Achten, H., Vries, B. de and Hennessey, J. (eds.), Design Research in the Netherlands 2000, Eindhoven University of Technology, Eindhoven, pp. 95-105.
Leclercq, P. P.: 2001, Programming and assisted sketching – graphic and parametric integration in architectural design, in Vries, B. de and Leeuwen, J. P. van and Achten, H. H. (eds.), Computer Aided Architectural Design Futures 2001, Kluwer Academic Publishers, Dordrecht, pp. 15-31.
Luck, M.: 1999, From definition to deployment: what next for agent-based systems? The Knowledge Engineering Review 14 (2), pp. 119-124.
Maes, P. (ed.): 1990, Designing Autonomous Agents, The MIT Press, Cambridge, Massachusetts.
McCall, R., Johnson, E. and Smith, M.: 1997, Hypersketching: design as creating a graphical hyperdocument, in Junge, R. (ed), CAAD Futures 1997, Kluwer Academic Publishers, Dordrecht, pp. 849-854.
McFadzean, J.: 1999, Computational Sketch Analyser (CSA): extending the boundaries of knowledge in CAAD, in Brown, A., Knight, M. and Berridge, P. (eds), Architectural Computing from Turing to 2000 – Proceedings of the 17th Conference on Education in Computer Aided Architectural Design in Europe, The University of Liverpool, Liverpool, pp. 503-510.
Minsky, M.: 1988, The Society of Mind, Simon & Schuster, New York.
Müller, J. P.: 1998, Architectures and applications of intelligent agents: a survey, The Knowledge Engineering Review 13 (4), pp. 353-380.
Park, S-H. and Gero, J. S.: 2000, Categorisation of shapes using shape features, in Gero, J. S. (ed), Artificial Intelligence in Design ’00, Kluwer Academic Publishers, Dordrecht, pp. 203-223.
Pellitteri, G.: 1997, A tool for a first analysis of architectural facades, Automation in Construction 5 (5), pp. 379-391.
Qian, D. and Gross, M. D.: 1999, Collaborative design with Netdraw, in Augenbroe, G. and Eastman, C. (eds), Computers in Building – Proceedings of the CAADfutures ’99 Conference, Kluwer Academic Publishers, Dordrecht, pp. 213-226.
Richens, P.: 1999, The Piranesi system for interactive rendering, in Augenbroe, G. and Eastman, C. (eds), Computers in Building – Proceedings of the CAADfutures ’99 Conference, Kluwer Academic Publishers, Dordrecht, pp. 381-398.
Rodgers, P. A., Green, G. and McGown, A.: 2000, Using concept sketches to track design progress, Design Studies 21 (5), pp. 451-464.
Stellingwerff, M.: 1999, SketchBoX, in Brown, A., Knight, M. and Berridge, P. (eds), Architectural Computing from Turing to 2000 – Proceedings of the 17th Conference on Education in Computer Aided Architectural Design in Europe, The University of Liverpool, Liverpool, pp. 491-497.
Trinder, M.: 1999, The computer’s role in sketch design: a transparent sketching medium, in Augenbroe, G. and Eastman, C. (eds), Computers in Building – Proceedings of the CAADfutures ’99 Conference, Kluwer Academic Publishers, Dordrecht, pp. 227-244.
Woolridge, M. and Jennings, N. R.: 1995, Intelligent agents: theory and practice, Knowledge Engineering Review, 10 (2), pp. 115-152.