# Learning From Mah Jong

*Towards a Multi-Agent System that can Recognize Graphic Units*

Henri ACHTEN and Joran JESSURUN

*Faculty of Architecture, Building and Planning, Eindhoven University of Technology, NL*

**Keywords:** artificial intelligence, games, graphic units, agents

**Abstract:** Sketching is a major means of exploiting the first conceptual developments in architectural design. If we want to support the architect in the ideas-developing phase of design, then we need to understand the conventions of depiction and encoding in drawings. The theory of graphic units provides an extended list of such conventions that are widely used. We propose that a multi-agent system for recognition of graphic units in drawings is fruitful: agents can specialize in graphic units, a multi-agent system can deal with ambiguity through negotiation and conflict resolution, and multi-agent systems function in dynamically changing environments. We first make a multi-agent system that can do something simpler: playing Mah Jong solitary. The Mah Jong solitary system shares the following important features with a multi-agent system that can recognize graphic units: (1) specialized agents for moves; (2) negotiation between agents to establish the best move; (3) dynamically changing environment; and (4) search activity in more advanced strategies. The paper presents the theoretical basis of graphic units and multi-agents systems. The multi-agent framework and its implementation is presented. Various levels of game play are distinguished, and these are correlated to the multi-agent system. The paper shows how the findings form the basis for graphic unit recognition.

## 1 GRAPHIC UNITS

Graphic representations such as sketches, plans, sections, perspectives, and so forth share the property that they are not only very personal documents because of style of drawing, but also that they can be used to share information between professionals. This share-ability is possible because architects use well-established conventions of depiction and encoding. Conventions of depiction are types of images such as plans, sections, and elevations. Conventions of encoding are the line types and hatching patterns that provide further information about the drawing.

In an analytical study of some 220 drawings taken from an extensive period of time (13th to 20th century), it was possible to identify 24 specific kinds of drawing elements that have a well-established use and meaning to architects (Achten 1997). These kinds of drawing elements are termed “graphic units” (Table 1).

**Table 1 Graphic Units**

| Graphic Unit | Description |
|---|---|
| Simple contour | Regular shape showing an outline. |
| Contour | Any irregular shape showing an outline. |
| Measurement device | Measure for establishing (relative) dimensions. |
| Specified form | Contour with specified dimensions. |
| Elaborated structural contour | Outline with structural detail. |
| Complementary contours | Composition of outlines. |
| Function symbols | Textual indication of function. |
| Zone | Area with specific use or function. |
| Schematic subdivision | Schematic depiction of principal subdivision. |
| Modular field | Irregular subdivision of area along coordinating lines. |
| Refinement grid | Grid with smaller module coordinated in other grid. |
| Schematic axial system | Schematic depiction of organisation of axes. |
| Axial system | Organisation of axes applied to building design. |
| Grid | System of modularly repeating coordinating lines. |
| Tartan grid | Double grid based on two alternating modules. |
| Structural tartan grid | Tartan grid with structural elements. |
| Element vocabulary | Set of simple shapes depicting (interior) elements. |
| Structural element vocabulary | Set of simple shapes depicting structural elements. |
| Functional space | Outline combined with function indicator. |
| Partitioning system | Schematic depiction of more detailed subdivision. |
| Proportion system | Diagram showing how proportions are derived. |
| Combinatorial element vocabulary | Precise relationships between elements. |
| Circulation system | Principle layout of circulation. |
| Circulation | Layout of circulation applied to building design. |

The following observations can be made: (1) half of the 24 identified graphic units represent structuring devices (e.g., grid, zone, and axial system) rather than concrete building elements (e.g., contours, functional space, and circulation). This implies that architects have an extensive set of representations for organising the design; (2) graphic units vary from being schematic, indicating global organisation or intention (e.g., contour, schematic subdivision, and schematic axial system), to being very specific and precise about location and dimension of the elements that are depicted (e.g., elaborated structural contour, functional space, and circulation); and (3) graphic units encode things such as composition, layout, modularisation, circulation, and interior in a graphic way. Therefore, even without additional explicit textual information, they convey information that is generally shared by the architectural community. Much of this information is encoded implicitly, but can be derived by examining the drawing. We propose therefore, that graphic units can form the basis for a visual language on which to build more sophisticated design support.

## 1.1 Graphic Unit Recognition

The ultimate goal of the research work is to implement a drawing system that will interpret drawings on the basis of graphic units. We look at 'drawing in action' for two main reasons: (1) use drawing actions as clues for recognizing graphic units (Kavakli et al. 1998; McFadzean 1999); and (2) to provide support through the system as the designer is working, in a very short time span after the drawing actions.

Related work falls in the category of CAD systems and sketch analysis. Computational work on interpreting drawings has focused mainly on bottom-up analysis from primitives to larger constructs, for example in facades (Pelliteri 1997), or more complex shapes in plans (Park and Gero 2000). Examples for graphic design support are the Electronic Cocktail Napkin (Gross 1996), Hypersketch and PHIDIAS (McCall et al. 1997), Netdraw (Qian and Gross 1999), and EsQUIsE (Leclercq 2001). The sketch functionality and interpretation of EsQUIsE is particularly close to the current work. What is missing is a well-founded basis of elements that can be considered for analysis and computational interpretation.

Related research in sketching, aimed at identifying pervasive structures is less common: Do et al. (2000) look at commonalities in sketches, in particular short-hands for drawing the same concepts; McFadzean's (1999) Computational Sketch Analyser takes several sketch-acting clues as indicators what the current status of the design process is; Koutamanis (2001) proposes a taxonomy of elements in sketches, breaking them down into organisational units not unlike graphic units. Rodgers et al. (2000) note a number of mechanisms between sketches without providing a more refined set of criteria to track design development.

# 2 MULTI-AGENT SYSTEMS

The research area on multi-agent systems came out of the general research field on Artificial Intelligence and more specifically Distributed Artificial Intelligence. It acknowledges two basic observations: (1) most of intelligent activity can be considered as distributed in one way or the other, and (2) the isolated symbol-processing approach seems to have reached the limits of what can be achieved.

Based on this, the notion of an "agent" as a situated and autonomous entity capable of interacting with the world and other agents has gradually developed (Russel and Norvig 1995; Nilsson 1998; Weiss 2001). Since the capabilities of such a widely defined agent range from the very simple to complex systems (Müller 1998), there is as yet no single definition what an agent is (Wooldridge and Jennings 1995; Wooldridge 2001), nor a theoretical basis for multi-agent systems (Luck 1999).

Although the mainstream of work on multi-agent systems usually conceives agents in human-like terms, a number of researchers have proposed to apply the multi-agent approach to the cognitive functions of intelligent behaviour (Minsky 1988; Maes 1989; Brooks 1990). Franklin (1995) sums up such directions and identifies agents as a necessary building-block for complex and intelligent behaviour.

Research on multi-agent systems focuses not only on the capabilities of the agents themselves, but moreover on reasoning within groups as cooperating or competitive individuals. It is generally found that conceived this way, multi-agent systems can function robustly in highly dynamic and unpredictable environments, removing some of the brittleness of previous AI systems.

## 2.1 Agents and Graphic Unit Recognition

Important issues to address in graphic unit recognition in drawings concern ambiguities and inaccuracies in the drawing, and resolving conflicting interpretations between candidate graphic units in the drawing. Multi-agent systems seem appropriate for tackling these issues. To summarize:

- An agent can specialize in recognition of one particular graphic unit, building on other agents that recognize more primitive graphic elements (systems approach).
- Agents may engage in conflict identification and resolution; this is necessary 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. Drawing constitutes such an environment.

# 3 DEVELOPMENT STRATEGY

There are two main issues to be resolved for graphic unit recognition: (1) to understand the dynamics of a multi-agent system; and (2) recognition algorithms for graphic units. In this paper we address the question of dynamics and control. The reason for this is to better assess the possible contribution of multi-agent systems to the research on graphic units. This is further discussed in the next section.

## 3.1 Multi-Agent System for Mah Jong

In order to understand the dynamics of multi-agent system, we first make a system that can do something simpler than graphic unit recognition; namely, playing Mah Jong solitary. Mah Jong is a game consisting of 148 bones that are stacked in a specific pattern. The purpose of the game is to clear the board by taking away two bones that form a match in suit and number, and which can be taken away from the left or right (see Figure 1).

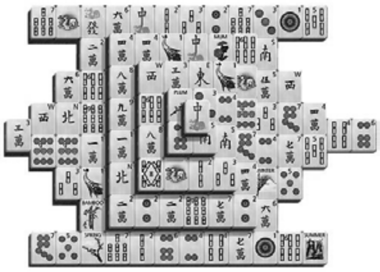

**Figure 1 Mah Jong Initial Layout**

Mah Jong in comparison to design is a toy problem. Yet there are some characteristics that make it interesting to look at: (1) although the game is fully determined and finite, the player does not have full information because of the stacking of bones; (2) complications can occur that may be anticipated by studying the current situation of the board. These can impose constraints on possible draws; (3) taking away a particular bone may lead to goal-driven search for other bones to be taken away before a match can be found; and (4) at many points during the game play, the player can choose between various draws of bones in a move. The choice has to be made with incomplete information, and competing arguments.

A multi-agent system for recognition of graphic units needs to balance search and recognition strategies: search for finding whether the graphic unit for which the agent is specialised is being drawn, and recognition for determining whether the graphic unit is actually present. This is similar to the problem of detecting which move to make. Furthermore, negotiation about best moves resembles negotiation between agents to settle into an interpretation of the drawing. In both cases, the environment changes dynamically through game moves of the player and drawing actions of the architect.

## 4 AGENT FRAMEWORK

We have established an agent framework for developing a multi-agent system. An agent in the framework has input, output, and an internal state and processes that are closed to the outside world. The input part senses the world environment and receives broadcast messages. The output part manipulates the world environment and broadcasting messages. Agents operate independently. It is possible to instantiate any number of agents of a given type. The multi-agent system is multithreaded, having all the agents run continuously at the same time. As in this way it is not possible to predetermine in which order which agents perform their actions, the design of the agents needs to take this into account. We establish implicit control through the use of broadcasts. An agent reads the broadcasts and selects those messages that are relevant. The agent's implementation is basically as follows:

1) Wait for a message.
2) If the message is not interesting, go to 1.
3) Do something with the message.
4) Send messages.
5) Interact with the environment (if the agent can manipulate).
6) Go to 1.

An observer agent is implemented as follows:

1) Observe the environment.
2) Broadcast a message about important changes.
3) Wait for a while.
4) Go to 1.

# 5 THE MAH JONG SYSTEM

For the Mah Jong system, the core system contains three objects: Agent, Message Queue, and Message. The Mah Jong Solitary system is implemented using this core. The system that plays Mah Jong starts from a very simple basis to more improved agents. These are described in the following sections.

## 5.1 Level 0 Mah Jong System

The Level 0 Mah Jong System consists of three types of agents: a *chooser* that picks a random bone in the layout, a *matcher* which checks if the chosen bones form a match, and a *mover* which removes the pair when the bones can be removed. The system needs at least 2 *choosers* to function; more *choosers* increase speed. As this system uses no heuristics, it typically takes many hours before it terminates. The Level 0 system can terminate prematurely by taking away a matching pair that blocks a future pair of bones. The system has no means of detecting whether the game has blocked or ended.

## 5.2 Level I Mah Jong System

The Level I Mah Jong System functions on the principle that it takes as a move the first match it finds. It consists of three types of agents: a *free* agent that broadcasts all the bones that are free for taking, a *matcher* that broadcasts a possible match in the free bones, and a *mover* which removes the first match that matcher finds. This system speeds up the game-play by only looking at free bones. The Level I system can terminate prematurely because of the same reason as the Level 0 system. The system ends playing when the free agent can find no more free bones.

## 5.3 Level II Mah Jong System

The Level II Mah Jong System looks at all possible current matches and tries to choose the best match. Compared to Level I system, it has an additional type of agent: a *heuristic* agent for deciding about the best move. The mover now removes the match of the winning heuristics agent. The choice for best match is based on the following heuristics (each heuristic has one agent):

- Remove two free pairs first: if it is possible to remove two pairs of a particular suit and number, then do so.
- Make long rows short: long rows consist of many bones that may be useful later. If a draw makes a long row shorter, then this is preferred.
- Remove stacks: a stack is two or more bones stacked on each other, where each of the bones could be drawn when the top would be removed.

If none of the heuristics apply, then the mover takes the first match found from *matcher* (like the Level I system). Performance should improve as the system is better able to avoid bad decisions. By allowing the heuristics agents to establish among themselves a rank-ordering (based on many runs and performance comparisons) the system can learn which priority ranking gives better performance. The system can terminate prematurely by overlooking a blocking pair which is not in the current set of possible moves. From testing it has to be determined whether such cases will occur less frequently than in systems level 0 and I.

## 5.4 Level III Mah Jong System

The Level III Mah Jong System not only looks at current best moves, but also tries to reason about future states of the board. The number of *heuristics* agents is expanded:

- Avoid leaving blocking pairs in the board: if there are two closely lying bones of the same suit and kind in the same row, then efforts should be directed to remove at least one bone of this blocking pair.
- Hindsight: confidence in a draw may be strengthened because of its consequences: (1) A bone has been freed that is necessary to unblock a row (thus removing a blocking couple); (2) the same bone as the removed match appears (that bone would never have been freed in another draw).
- Maximize number of new free bones: a draw can result in zero to six new free bones that can be candidates in the next draw. The current draw should maximize the number of free bones.

- Longest chain: for each draw it is possible to estimate how many following draws can be made by looking which bones will become free. Choose the draw with the longest chain of possible moves.

The Level III Mah Jong System has more sophisticated reasoning than the Level II system. Again, the heuristics agents can establish an internal rank-ordering. Testing has to determine whether it has better performance in terms of leading to a solution, and whether the increased reasoning does not slow down system performance.

## 5.5 Testing the Mah Jong Systems

The Mah Jong system can run in two modes: with a randomly generated layout, which has no guaranteed solution, or with a solvable layout. Each layout is uniquely identified by a seed, which enables us to use the same layout for benchmarking measurements. The systems will be tested on the following aspects by running them through a series of games:

- Average and spread of time for establishing a move. This measure is used to assess performance and stability of the decision-making.
- Average time to resolve a given layout. This measure is used to assess overall performance of the system.
- What kind of problems causes breakdown? This measure is used to find game-playing weaknesses in the system.
- Frequency of problems where the system prematurely ends. This measure is used to determine robustness of the system.
- Stability of the system, in particular in stepwise versus continuous running mode. This measure is used to determine performance of the underlying system architecture.

## 5.6 From Mah Jong to Graphic Unit Recognition

From the Mah Jong systems, we learn how to coordinate decision-making between agents. Also, we learn how the principles of communication by broadcasting in the multi-agent system apply. Thirdly, we can see how combined search- and recognition techniques for heuristics function in a dynamic setting. Finally, we gain some insight in the effects of dynamically rank-ordering heuristics as a simple learning mechanism.

There are two main features that are lacking with respect to a system that can recognize graphic units: (1) there is no sophisticated learning mechanism; and (2) there is no user in the loop. Although the systems can optimise their internal rank-ordering of heuristics, they are not capable to establish new heuristics or modify existing heuristics. This may become an important issue when the system needs to tune in on the architects drawing style. The user in the loop can increase complexity of the environment of the multi-agent system, e.g., by erasing previously drawn objects, changing his mind about what he is drawing, starting a new drawing, and so forth. These issues need to be resolved in further development.

## 6 CONCLUSION

In the current paper we have outlined how multi-agent systems can be used for the recognition of graphic units in drawings. To investigate the dynamics of a multi-agent system, we first implement a multi-agent system that can play Mah Jong. A framework for such a system has been established, and a Level 0 game-play system has been implemented. Heuristics for higher level game-play systems are described and a simple learning mechanism for establishing internal rank ordering. The basics of this work can be used to implement a multi-agent system that can recognize graphic units while the designer is drawing. However, three aspects still need development: algorithms for graphic unit-description, learning mechanisms for the architects' style, and coping with the user in the loop.

## 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: Eindhoven University of Technology.

Brooks, R.A. 1990. Elephants Don't Play Chess. *Robotics and Autonomous Systems,* (6): 3–15. Amsterdam: Elsevier Science Ltd.

Call, R. Mc, E. Johnson and M. Smith. 1997. Hypersketching: Design as Creating a Graphical Hyperdocument. *Proceedings of CAADFutures 1997*, ed. R. Junge, 849-854. Dordrecht: Kluwer Academic Publishers.

Do, E.Y.L., M.D. Gross, B. Neiman and C. Zimring. 2000. Intentions in and Relations among Design Drawings. *Design Studies,* 21(5): 483-503. Amsterdam: Elsevier Science Ltd.

Fadzean, J. Mc. 1999. Computational Sketch Analyser (CSA). *Proceedings of eCAADe 1999*, eds. A. Brown, M. Knight and P. Berridge, 503-510. Liverpool, UK: The University of Liverpool.

Franklin, S. 1995. *Artificial Minds.* Cambridge, Massachusetts: The MIT Press.

Gross, M.D. 1996. The Electronic Cocktail Napkin - A Computational Environment for Working with Design Diagrams. *Design Studies,* 17(1):53-69. Amsterdam: Elsevier Science Ltd.

Kavakli, M., S.A.R. Scrivener and L.J. Ball. 1998. Structure in Idea Sketching Behaviour. *Design Studies,* 19(4): 485-517. Amsterdam: Elsevier Science Ltd.

Koutamanis, A. 2001. Prolegomena to the Recognition of Floor Plan Sketches. *Design Research in the Netherlands 2000*, eds. H. Achten, B. de Vries, and J. Hennessey, 95-105. Eindhoven: Eindhoven University of Technology.

Leclercq, P.P. 2001. Programming and Assisted Sketching – Graphic and Parametric Integration in Architectural Design. *Proceedings of CAADFutures 2001*, eds. B. de Vries, J.P. van Leeuwen and H.H. Achten, 15-31. Dordrecht: Kluwer Academic Publishers.

Luck, M. 1999. From Definition to Deployment: What Next for Agent-based Systems? *The Knowledge Engineering Review,* 14(2): 119-124. Cambridge: Cambridge University Press.

Maes, P. 1989. The Dynamics of Action Selection. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence ICAI-89*, 991-997. Detroit.

Minsky, M. 1988. *The Society of Mind*. New York: Simon & Schuster.

Müller, J.P. 1998. Architectures and Applications of Intelligent Agents: A Survey. *The Knowledge Engineering Review,* 13(4): 353-380. Cambridge: Cambridge University Press.

Nilsson, N.J. 1998. *Artificial Intelligence: A New Synthesis*. San Francisco: Morgan Kaufmann Publishers, Inc.

Park, S-H. and J.S. Gero. 2000. Categorisation of Shapes Using Shape Features. *AID'00*, ed. J.S. Gero, 203-223. Dordrecht: Kluwer Academic Publishers.

Pellitteri, G. 1997. A Tool for a First Analysis of Architectural Facades. *Automation in Construction,* 5(5): 379-391. Amsterdam: Elsevier Science Ltd.

Qian, D. and M.D. Gross. 1999. Collaborative Design with Netdraw. *Proceedings of CAADFutures 1999*, eds. G. Augenbroe and C. Eastman, 213-226. Dordrecht: Kluwer Academic Publishers.

Rodgers, P.A., G. Green and A. McGown, A. 2000. Using Concept Sketches to Track Design Progress. *Design Studies,* 21(5): 451-464. Amsterdam: Elsevier Science Ltd.

Russel, S. and P. Norvig. 1995. *Artificial Intelligence: A modern approach*. Upper Saddle River, New Jersey: Prentice-Hall, Inc.

Weiss, G. 2001. *Multiagent Systems*. Cambridge, Massachusetts: The MIT Press.

Wooldridge, M. and N.R. Jennings. 1995. Intelligent Agents: Theory and Practice. *The Knowledge Engineering Review,* 10(2): 115-152. Cambridge: Cambridge University Press.

Wooldridge, M. 2001. Intelligent Agents. In *Multiagent systems*, eds. G. Weiss, 27-77. Cambridge, Massachusetts: The MIT Press.