

# A Multi-Agent Mah Jong Playing System: Towards Real-Time Recognition of Graphic Units in Graphic Representations

H. Achten, J. Jessurun

*In architectural design, sketching is an important means to explore the first conceptual developments in the design process. It is necessary to understand the conventions of depiction and encoding in sketches and drawings if we want to support the architect in the sketching activity. The theory of graphic units provides a comprehensive list of conventions of depiction and encoding that are widely used among architects. These graphic units form useful building blocks to understand design drawings. We investigate whether it is possible to build a system that can recognize graphic units. The technology we are looking at is multi-agent systems. It was chosen for the following reasons: 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. Currently there is no general approach or technology available for multi-agent systems. Therefore, in our research we first set out to make such a multi-agent system. In order to keep the complexity low, we first aim to make a system that can do something simple: 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) a dynamically changing environment; and (4) search activity for more advanced strategies. The paper presents the theoretical basis of graphic units and multi-agents systems, followed by a description of the multi-agent framework and its implementation. A number of systems that can play Mah Jong at various degrees of competence and accordingly degrees of complexity of multi-agent system, are distinguished. Finally, the paper demonstrates how the findings are informative for a system that can recognize graphic units.*

*Keywords: AI, multi-agent systems, games, graphic units, architectural design.*

## 1 Graphic units

Graphic representations such as sketches, plans, sections, perspectives, and so forth are not only very personal documents because of style of drawing, but also 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. Con-

ventions 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 the thirteenth to the twentieth Century, twenty-four specific kinds of drawing elements with a well-established use and meaning to architects based on conventions of depiction and encoding were identified [1]. These kinds of drawing elements are termed "graphic units". They are listed in 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 another grid.
Schematic axial system	Schematic depiction of organization of axes.
Axial system	Organization of axes applied to building design.

Graphic Unit	Description
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 a more detailed subdivision.
Proportion system	Diagram showing how proportions are derived.
Combinatorial element vocabulary	Precise relationships between particular elements.
1Circulation system	Principal 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 organizing the design.
- 2) Graphic units vary from being schematic, indicating global organization or intention (e.g., contour, schematic subdivision, and schematic axial system), to being very specific and precise about the location and dimension of the elements that are depicted (e.g., elaborated structural contour, functional space, and circulation).
- 3) As graphic units are built up of fairly basic graphic entities, it is often only possible to distinguish between graphic units on the basis of their meaning. Thus, interpretation suffers from ambiguity. After the set of graphic entities has been settled as a specific graphic unit, however, the purpose and use are clear.
- 4) Graphic units encode things such as composition, layout, modularization, 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) to provide support through the system as the designer is working, in a very short time span after the drawing actions; and (2) to use drawing actions as clues for recognizing graphic units [2], [3].

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 [4], or more complex shapes in plans [5]. Examples for graphic de-

sign support are the Electronic Cocktail Napkin [6], Hyper-sketch and PHIDIAS [7], Netdraw [8], and EsQUIsE [9]. The sketch functionality and interpretation of EsQUIsE is particularly close to the current work. EsQUIsE interprets strokes in a graphic environment as contours, grids, and functional spaces, and builds a representational model on this. EsQUIsE lacks 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 [10] look at commonalities in sketches, in particular shorthands for drawing the same concepts; McFadzean's Computational Sketch Analyser [3] takes sketch-acting clues as indicators for the current status of the design process; Koutamanis [11] proposes a taxonomy of elements in sketches, breaking them down into organizational units not unlike graphic units. Rodgers et al [12] note a number of mechanisms between sketches without providing a more refined set of criteria to track design development. The works in question corroborate the proposition that there are stable aspects in graphics that are widely shared among architects. The novel aspect in our research lies in the use of acknowledged graphic structures (i.e., graphic units) as a means for computational design support.

## 2 Multi-agent systems

The research area of multi-agent systems came out of the research field of 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 another, 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 [13], [14], [15]. Since the capabilities of such a broadly defined agent range from the very simple to the complex [16], there is as yet no single definition of what an agent is [17], [18], nor a firm theoretical basis for multi-agent systems [19].

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 more cognitive functions of intelligent behavior [20], [21], [22]. Franklin [23] sums up such directions and identifies agents as a necessary building block for complex and intelligent behavior. We adopt this latter view as our basic approach for building a multi-agent system for graphic unit recognition.

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.

It is for these reasons that we turn to multi-agent systems to implement a system that can recognize graphic units.

## 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 a 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 involving 148 bones that are stacked in a specific pattern. Bones come in pairs that are distinguished by suit (e.g., bamboo, circles, or season) and number (ranging from one to nine). The purpose of the game is to clear the board by taking away per move two bones that form a match because they are from the same suit and number, and that have no bones lying to the left or right of them (see Fig. 1).

In comparison to design, Mah Jong is a toy problem. Yet it has some characteristics that make it worthwhile to look at in the perspective of design problems:

- 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) If a particular bone needs to be taken away, and currently does not have a match, then this leads to goal-driven search for other bones to be taken away.

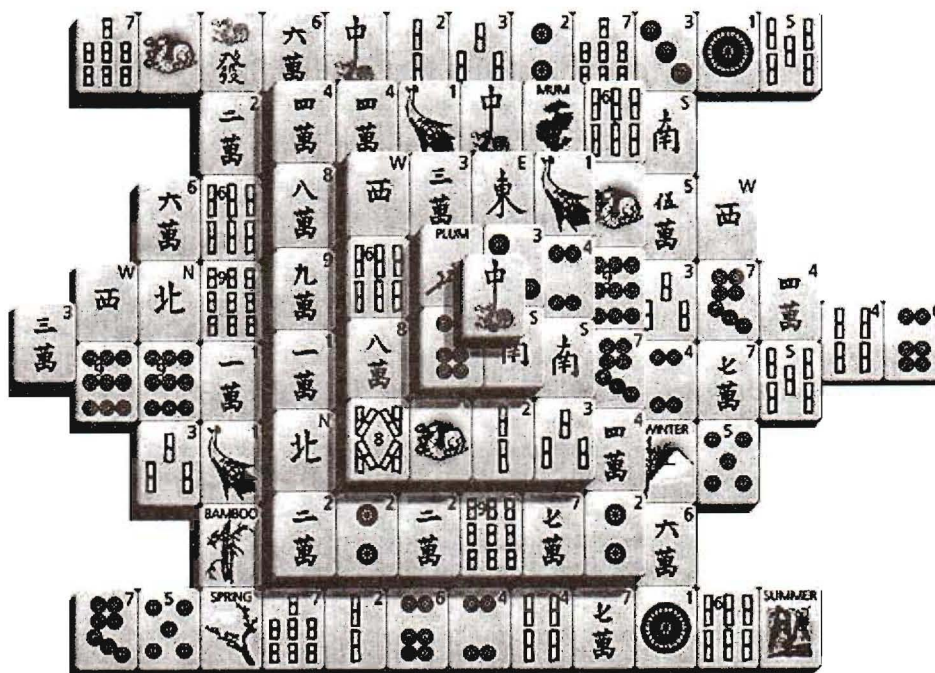


Fig. 1: Mah Jong Initial Layout

- 4) At many points during the game, 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 in which the agent is specialized is being drawn, and recognition for determining whether the graphic unit is actually present. This search is similar to – but more complex than – 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 the moves made by the player and the drawing actions of the architect.

## 4 Agent framework

We 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 broadcasts 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 configuration of simple agents, and is stepwise improved by adding agents that can perform more reasoning. This phased approach is chosen so that it is possible to determine the influence on effectiveness and speed of different kinds of agents. The different systems, called Level 0, I, II, and III, 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* in order to function; more *choosers* increase system speed. Because the *choosers* pick any bone in the layout, many of which cannot be drawn because they are not free, it typically takes many hours before the game 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 all possible matches among the free bones, and a *mover* which removes the first match that the matcher finds. Compared with the Level 0 system, this system speeds up the game-play by only looking at free bones.

The Level I system can terminate prematurely for the same reason as the Level 0 system. The system ends playing when the free agent can find no more free bones. By checking whether this state occurs with the state that there are no more bones in the layout, the system can determine whether the game has been successfully completed or not.

### 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 without reasoning about future states of the layout. Compared to the Level I system, it has an additional type of agent: *heuristic* agents for deciding about the best move. A heuristics agent knows one game-playing heuristic. If there are more heuristics, then there are more agents, which have to decide among themselves which heuristic prevails in the current move. The mover now removes the match of the winning heuristics agent. The choice for best match is based on the following heuristics:

- Remove two free pairs: if it is possible to remove two pairs of a particular suit and number, then do so (thus, all four bones of a suit and number are removed).
- Make long rows short: prefer a draw that will shorten a long row above a draw from a shorter row.
- Remove high stacks: a stack occurs when two or more bones are stacked on each other. This heuristic prefers a draw that will make a high stack lower above a draw from lower stacks.

If none of the heuristics apply, or if the result between the heuristics agents is a draw, then the system functions like the Level I system and the mover takes the first match found from *matcher*.

Compared with to system Level I, 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 detect which priority ranking gives better performance. The current listing above reflects the initial rank ordering.

The system can terminate prematurely, for example by overlooking a blocking pair that is not in the current set of possible moves. From testing it has to be determined whether such cases will occur less frequently with the additional heuristics agents 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 reasons about future states of the board. The following *heuristics* agents are added:

- Avoid leaving blocking pairs on 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 the 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 to see which bones will become free. Choose the draw with the longest chain of possible moves.

As in the Level II system, the heuristics agents can establish an internal rank ordering. The current list above reflects the initial rank ordering. Testing has to determine whether system Level III 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 the use of the same layout for benchmarking. The systems will be tested on the following aspects by running them through a series of solvable games:

- Average and spread of time for establishing a move. This measure is used to assess the performance and stability of the decision-making.
- Average time to resolve a given layout. This measure is used to assess the overall performance of the system.
- What kind of problems cause premature termination? This measure is used to find game-playing weaknesses in the system.
- Frequency of problems that cause premature termination. This measure is used to determine the robustness of the system.
- Stability of the system, in particular in stepwise versus continuous running mode. This measure is used to determine the performance of the underlying multi-agent 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 into the effects of dynamically rank ordering heuristics as a simple optimization 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 of establishing new heuristics or modifying existing heuristics. This may become an important issue when the system needs to tune in on the architect's drawing style. The user in the loop can increase the 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 Conclusions

In the paper we have outlined the principal reasons why 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 optimisation mechanism for establishing internal rank ordering. The basics of this work will 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 architect's style, and coping with the user in the loop.

### 7 Acknowledgements

The presented work was carried out by the principal author during a half-year sabbatical leave undertaken at the Faculty of Architecture of the Czech Technical University of Prague. The sabbatical leave was kindly hosted by Doc. Ing. Václav Kučera CSc., head of the Computer Aided Design group. The authors wish to thank both the Czech Technical University in Prague and Technische Universiteit Eindhoven in Eindhoven for supporting the research work.

### References

- [1] Achten, H.: *Generic Representations – An Approach for Modelling Procedural and Declarative Knowledge of Building Types in Architectural Design*. Eindhoven, Eindhoven University of Technology, 1997.
- [2] Kavakli, M., Scrivener, S., Ball, L.: *Structure in Idea Sketching Behavior*. Design Studies, Vol. 19, 1998, No. 4, p. 485–517.

- [3] McFadzean, J.: *Computational Sketch Analyser (CSA)*. In: "Proceedings of the 17<sup>th</sup> Conference on Education in Computer Aided Architectural Design in Europe" (Editors A. Brown and M. Knight and P. Berridge), Liverpool: The University of Liverpool, 1999, p. 503–510.
- [4] Pellitteri, G.: *A Tool for a First Analysis of Architectural Facades*. Automation in Construction Vol. 5, 1997, No. 5, p. 379–391.
- [5] Park, S., Gero, J.: *Categorisation of Shapes Using Shape Features*. In: "Artificial Intelligence in Design '00" (Editor J. Gero), Dordrecht: Kluwer Academic Publishers, 2000, p. 203–223.
- [6] Gross, M.: *The Electronic Cocktail Napkin – A Computational Environment for Working With Design Diagrams*. Design Studies, Vol. 17, 1996, No. 1, p. 53–69.
- [7] McCall, R., Johnson, E., Smith, M.: *Hypersketching: Design as Creating a Graphical Hyperdocument*. In "CAAD futures 1997: Proceedings of the 7<sup>th</sup> International Conference on Computer Aided Architectural Design Futures" (Editor R. Junge), Dordrecht: Kluwer Academic Publishers, 1997, p. 849–854.
- [8] Qian, D., Gross, M.: *Collaborative Design with Netdraw*. In "Proceedings of the CAAD futures '99 Conference" (Editors G. Augenbroe and C. Eastman), Dordrecht: Kluwer Academic Publishers, 1999, p. 213–226.
- [9] Leclercq, P.: *Programming and Assisted Sketching – Graphic and Parametric Integration in Architectural Design*. In "Computer Aided Architectural Design Futures 2001" (Editors B. de Vries and J. van Leeuwen and H. Achten), Dordrecht: Kluwer Academic Publishers, 2001, p. 15–31.
- [10] Do, E. et al: *Intentions in and Relations Among Design Drawings*. Design Studies. Vol. 21, 2000, No. 5, p. 483–503.
- [11] Koutamanis, A.: *Prolegomena to the Recognition of Floor Plan Sketches*. In "Design Research in the Netherlands 2000", (Editors H. Achten and B. de Vries and J. Hennessey), Eindhoven: Eindhoven University of Technology, 2001, p. 95–105.
- [12] Rodgers, P., Green, G. McGown, A.: *Using Concept Sketches to Track Design Progress*. Design Studies. Vol. 21, 2000, No. 5, p. 451–464.
- [13] Russel, S., Norvig, P.: *Artificial Intelligence: A Modern Approach*. Upper Saddle River: Prentice-Hall Inc., 1995.
- [14] Nilsson, N.: *Artificial Intelligence: A New Synthesis*. San Francisco: Morgan Kaufmann Publishers Inc., 1998.
- [15] Weiss, G.: *Multiagent Systems – A Modern Approach to Distributed Artificial Intelligence*. Cambridge: The MIT Press, 2001.
- [16] Müller, J. P.: *Architectures and Applications of Intelligent Agents: A Survey*. The Knowledge Engineering Review. Vol. 13, 1998, No. 4, p. 353–380.
- [17] Wooldridge, M., Jennings, N.: *Intelligent Agents: Theory and Practice*. The Knowledge Engineering Review. Vol. 10, 1995, No. 2, p. 115–152.
- [18] Wooldridge, M.: *Intelligent Agents*. In "Multiagent Systems – A Modern Approach to Distributed Artificial Intelligence", (Editor G. Weiss), Cambridge: The MIT Press, 2001, p. 27–77.
- [19] Luck, M.: *From Definition to Deployment: What Next for Agent-Based Systems?* The Knowledge Engineering Review. Vol. 14, 1999, No. 2, p. 119–124.
- [20] Minsky, M.: *The Society of Mind*. New York: Simon & Schuster, 1988.
- [21] Maes, P.: *The Dynamics of Action Selection*. In "Proceedings of the Eleventh International Joint Conference on Artificial Intelligence ICAI-89", (Editor N. S. Sridahran), Detroit: Morgan Kaufmann, 1989, p. 991–997.
- [22] Brooks, R.: *Elephants Don't Play Chess*. Robotics and Autonomous Systems. Vol. 6, 1990, p. 3–15.
- [23] Franklin, S.: *Artificial Minds*. Cambridge: The MIT Press, 1995.

---

Associate professor Dr. Ir. Henri Hubertus Achten  
 phone: +31 40 2472470  
 fax: +31 40 2450328  
 e-mail: h.h.achten@tue.nl

Design Systems Group

Vertigo VRT 9+H05  
 Design Systems Group  
 Faculty of Architecture, Building and Planning  
 Technische Universiteit Eindhoven  
 PO Box 512  
 5600 MB Eindhoven, The Netherlands