ROLE PLAYING FOR LINKING IDEAS IN THE IDEA ASSOCIATION PROCESS

IH-CHENG LAI  
Graduate Institute of Architecture, National Chiao Tung University, Hsinchu, Taiwan.  
ihcheng@arch.nctu.edu.tw

AND

KUN-TAI KAO, TENG-WEN CHANG  
Graduate School of Computational Design, National Yunlin University of Science & Technology  
g9234701@yuntech.edu.tw, tengwen@arch.nctu.edu.tw

Abstract. Idea association is an interactive behaviour that will invoke dynamic interactions through different participants’ interplay, especially within the conceptual design stages. Such interplay involves distributed interactions. For understanding the distributed interactions of idea association, the research applies a cognitive theory (role play) on to modelling the behaviours of designers with internal, external design situation and linking process. By adopting the mechanisms of Dynamic Agent Role Interplay System (DARIS) framework, some computational characteristics within the distributed interactions are explored. Finally, a computer-supported system for dynamically linking design ideas is investigated through a design example.

1. Introduction

Idea association since ancient times has always been a methodology as well as an important step in the design process, especially within the conceptual design stage. Linking ideas to the action provides an incremental interaction metaphor for generating diverse design ideas. Furthermore, such interactions involve interchange of ideas between different participants’ (Osborn, 1963). Through dynamic interactions among distributed knowledge, the interplay involves distributed interactions. According to different design situations, a designer while participating in the idea association process plays different roles to link his/her internal design knowledge and experience, as well as external participants’ ideas. For example, in a house design, a designer sometimes needs to play the role of the client for linking ideas with actual experiences. On the other hand, the designer also explores the design possibility via playing the role of the design expert. Thus, idea association
involves a dynamic linking process. By studying such interactions via the view of roles, we gain a powerful vehicle and mechanism for understanding the linking process of idea association in the conceptual design stage.

In addition, the distributed interactions involve a process of action and reaction that participants apply according to the ideas they encounter. The reactive nature of these interactions under real design situations is often hard to analyze during the process of idea association. The theory of role-play (Yardlet-Matwiejczuk, 1997) uses role play in modelling psychological situations and provides a useful metaphor for understanding the distributed interactions. This study uses the Dynamic Agent Role Interplay System (DARIS) (Chang and Lai, 2004) framework as a computational role-play environment to explore the distributed interactions. By gaining some insights of distributed interactions via simulating linking process within DARIS, we investigate the computational feasibility of a computer-supported system for dynamically linking design ideas in the idea association process.

1.1. LINKING IDEAS IN THE CONCEPTUAL DESIGN STAGE

The key for idea association is the linking of human long-term memory to the ideas (Rapaport, 1974). Especially in the conceptual design stage, Osborn (1963) argued that the linking process also involves linking different participants’ knowledge in a stimulated discussion among team members. In the design process, ideas need to be related to the task at hand and they need to provide a solution for design problems other than communication (Lugt, 2000). The purpose is to produce diverse design ideas that can serve as leads to the development of possible design alternatives (Petrovic, 1997).

In addition, there is a tendency to use design cases as references and to extract past experience for generating ideas (Oxman, 1994). In the idea association process, designers are used to decomposing a design into several architectural elements, and to using the attributes of these as keys to look for relevant ideas within design cases (Lai, 2004a). The knowledge representation ICF (issue, concept and form) proposed by Oxman (1994) provides an effective linking method to represent design ideas in the conceptual design stage. These ideas, represented by the three elements issue, concept and form, mainly construct a designer’s long-term memory for linking design ideas.

1.2. THREE LINKING BEHAVIOURS OF IDEA ASSOCIATION

Ancient Greeks argued that idea association consists of three linking principles for generating diverse ideas, namely, similarity, contrast and contiguity. These three linking principles, which can be considered as linking behaviours, have their linking mechanisms to link ideas. Based on the textual and linking characteristic of the ICF representation, the three linking behaviours allow designers to link and generate
diverse ideas (Lai, 2004b).

By using textual matching between any two elements within ICF, designers can apply various linking behaviours to link new ideas with another element. For example, the similarity behaviour is used for linking new ideas with similar concept elements. The contrast behaviour is used for linking new ideas with different contrast form elements. Designers can use contiguity behaviours to link the same design solutions with different design problems (issue elements).

2. Linking Ideas as Role Playing

Idea association involves distributed interactions where participants act and re-act internal and external knowledge through a dynamic linking process. In the process of idea association, participants always play different roles and then employ the different linking behaviours described above to link and generate diverse ideas.

In the cognitive domain, the theory of role-play provides an important insight for our realization of such linking interaction including multi-knowledge interaction, stimulating and the situated analysis.

2.1. ROLE PLAY

Role play can be considered a way of deliberately constructing an approximation of a ‘real life’ episode or experience under the acting/re-acting process (Chang, 2004). Chang and Lai (2004) argue that there are three important mechanisms of role play: situation, engagement and scenario-based interaction. Situation is what people encounter in real life or constructed conditions in experiments; based on the situation, people act/re-act respectively. The interactions between people are what we called scenario-based interaction. The engagement (as-if) represents the induction on the actors’ performance that people make during the play. These three mechanisms provide our understanding of the linking interactions of idea association.

1. Situation: Situation of role play is to investigate the behaviour of designers encountering different design situations. Such design situations influence participants’ decision of what role they play and what linking behaviours they apply.

2. Engagement: The main behaviour of role-play engagement is induction. By inducting different roles, designers can engage in a real design situation for linking ideas.

3. Scenario-based interaction: Based on the specific content of the given circumstances, designers interact with each other and with the material of the play in a design process.

Linking ideas, similar to role playing, is always engaging in the ‘as-if’ condition,
and then acting/re-acting the internal design knowledge and external design situations (such as design task, time duration, participants etc.). Thus three levels of interplay are involved in the process of linking ideas. Within these interplays, each participant can play different roles and apply different linking behaviours to link and generate diverse design ideas. The three interplays will be elaborated in the following section (see in Figure 1).

2.2. THREE INTERPLAYS OF IDEA ASSOCIATION

With the theory outlined by role play, we can identify three important characteristics of role play with respect to the three interplays of idea association (Figure 1). Designers apply different linking behaviours to interact internal and external design situations through a dynamic linking process. These interplays are described below:

1. **External interplay**: External design situations (such as design tasks, participant number and time duration) will influence a designer (role) to use design knowledge to link related ideas.

2. **Internal interplay**: A designer (role) plays different roles and uses different knowledge for linking design ideas to long-term memory.

3. **Linking process**: By interacting with the linking process (such as acting, communicating, playing), designers engage with the idea association process interactively.

Figure 1. Three interplays of idea association.
3. Computing the Distributed Interactions Using DARIS

In the computational domain, agents can be thought of as a distributed computational system. With autonomous, reactive, and communicative behaviours, agents can autonomously participate in dynamic linking play to interact with internal and external design situations. Therefore, integrating the mechanisms of role play with agents provides an effective method to implement the distributed interactions of idea association. This research uses the DARIS framework as a computational role-play environment to explore the idea association process.

3.1. DARIS

DARIS is an agent-based system for implementing distributed interactions in a multi-designer collaboration environment. The original agent model is inspired by the role-play theory identified by Yardlet–Matwiejczuk (1997). DARIS is mainly composed of three different layers: the internal design situation layer, the interprocess layer and the external design situation layer (as shown in Figure 2). Within the three layers, there are dynamic interactions through five different agent entities including the user agent, the role agent, the director agent, the stage agent and the scene agent. Each mechanism of agent entity is described below.

3.1.1. User Agents (UA)
The user agent provides a vehicle for directly interacting with the user as well as facilitating interaction between agents. After role agents finish the user specified task, the user agent is notified and conveys the result to the user.

3.1.2. Role Agents (RA)
Each role agent has its individual knowledge base including design knowledge and skills to solve some specific design problems.

3.1.3. Director Agents (DA)
The director agent plays the same role as the director of the play. The major task of director agent is to be responsible for agents’ participation.

3.1.4. Scene Agent (ScA)
The scene agent controls three main tasks, including defining the role list, enforcing the time schedule and evaluating interplay outcomes from role-agents against design goals. There is only one current ScA within any StA (State Agent) action.

3.1.5. Stage Agent (StA)
The stage agent contains the sequential orders of the ScA. Once the outcomes of
interplay are approved by ScA and sent back to State Agent (StA), the letter then
either sends the outcomes back to UA for user-approval or stores the acts in the
data repository.

In addition, the *script* is constituted as a list of events that compose a stereotypical
episode. Through the sequential events, the agent entities dynamically interact with
each other to interact different design situations.

![Figure 2. DARIS framework (after Chang and Lai, 2004).](image)

3.2. MAPPING LINKING BEHAVIOURS AND INTERPLAY WITH DARIS

To realize the dynamic role interplay described in this paper, we mapped the three
linking behaviours and interplays to the mechanisms of DARIS using three strategies. They are: (i) embedding each RA’s knowledge base (KB) with a linking behaviour
and an idea library (as its long-term memory); (ii) constructing the internal and
external interplays with DARIS components and knowledge, and (iii) directing a
linking play to interact with the linking process of idea association.

For explaining the distributed interactions, we conducted an experiment using
three human designers and three software agents involved in a brainstorming meeting
in a DARIS environment. The linking mechanisms and the design example are
described in the following section.

4. How Role Plays and Ideas Link

The design task was related to the spatial organization of a single-family house
design in a specific site located in an urban area. Addressing three design issues of circulation, view and sunlight, participants were encouraged to generate diverse ideas.

The three human designers were undergraduate architecture students participating in a design studio course. They were familiar with the DARIS environment. During a brainstorming session, each student inserted a software agent to participate in the design task in different geographic locations. The script and the play are elaborated upon below.

4.1. THE SCRIPT

Script is a structure for describing a sequence of events and their related components that take place during the design process in DARIS, including design tasks, roles’ skills, role list and communication messages. Three layers of scripts for this spatial organization problem are described below:

4.1.1. Internal design situation layer:
In this layer, there are three human agents (UA human_1, UA human_2, UA human_3) and three software agents (RA software_1, RA software_2, RA software_3). Each human agent is an actor who can play two roles: UA and RA. Each of the software agents in this example has pre-set design expert’s knowledge and a linking behaviour seen in Table 1. Each human agent is familiar with these design experts’ knowledge. They also apply different linking behaviours to link design ideas.

| TABLE 1. Design knowledge and linking behaviours for three software agents. |
|-----------------------------------------------|----------------|----------------|----------------|
| Design knowledge                        | RA software_1 | RA software_2 | RA software_3 |
| Rem Koolhaas                              | Le Corbusier  | MVRDV         |
| Linking behaviour                        | Similarity    | Contrast      | Contiguity    |

4.1.2. External design situation layer:
There is a stage agent (StA spatial_organization) and three scene agents (ScA circulation, ScA view and ScA sunlight) in this layer. StA spatial_organization controls the role agent list, the scene agent order and time duration. Each ScA has control knowledge to evaluate the outcomes related to the design issue within ideas in a time duration. For example, ScA circulation has control knowledge related to “circulation” issue.

4.1.3. Inter-process layer:
This layer consists only of the director agent (DA) who is the coordinator of all plays and also acts as a register for availability of role agents. Any communication messages among agents should be passed through the DA.
4.2. THE PLAY

The entire play is comprised of three scenes (circulation, view and sunlight) on a stage. Each scene has a predefined design problem that needs to be solved within the predefined duration of time. The design problem is regarded as an issue element within ICF representation.

In a linking play, the human agent (UA) who starts a new play is a facilitator (DA) guiding the process in the whole play. He or she acts as a facilitator that can terminate each scene and select a design outcome. The other two human agents (UAs) and three software agents (RAs) are invited to generate design ideas. The content within the play is shown in Figure 3.

![Figure 3. The play displayed in the External Interplay interface.](image)

Each scene is comprised of three steps: pre-condition, idea generation process, and termination. The play of three scenes is described below.

4.2.1. Scene 1: Circulation

Pre-condition: UA\textsubscript{human,1} acts as the facilitator and informs DA to start the play. DA receives the message, and sends this message to Sc\textsubscript{circulation} through St\textsubscript{spatial,organization}. Sc\textsubscript{circulation} sends its design problem (circulation issue) to UA\textsubscript{human,2} through DA. A ring-type communication for idea association starts, which involves several loops from UA\textsubscript{human,2}, RA\textsubscript{software,1}, RA\textsubscript{software,2}, RA\textsubscript{software,3} to UA\textsubscript{human,3}.

Process: UA\textsubscript{human,2} starts generating the first idea. RA\textsubscript{software,1} receives the idea, and use similarity behavior to generate the new idea. Since the concept element “folding” has conflicts with RA\textsubscript{software,2}, RA\textsubscript{software,3} automatically generates the next idea as a reaction by using contiguity behavior. Finally, UA\textsubscript{human,3} generates the last idea in the first loop. The following loops involve the same linking sequence.

Termination: When the time is over, Sc\textsubscript{circulation} terminates the scene, and informs St\textsubscript{spatial,organization} to store these idea entities in the data repository. UA\textsubscript{human,1} receives the idea alternatives and selects one generated idea through DA. DA then informs St\textsubscript{spatial,organization} to send the idea back to Sc\textsubscript{view} as the input of Scene 2.
4.2.2. Scene 2: View

Pre-condition: DA receives the message, and sends this message to ScA_view through StA_spatial_organization. ScA_view sends its design problem (view issue) to UA human_3 through DA. A ring-type communication for idea association starts, which involves several loops from UA human_3, RA software_2, RA software_3, RA software_1 to UA human_1.

Process: In the first loop, UA human_3 generates the first idea in this scene. RA software_2 does not have ideas and skips his turn. RA software_3 automatically generates the next idea as a reaction by using contiguity behaviour. RA software_1 then generates the next idea. Finally, UA human_1 informs ScA_view that he does not have ideas and skips his turn. Several loops follow the same linking sequence.

Termination: When the time is over, ScA_view terminates the scene, and informs StA_spatial_organization to store these ideas in the data repository. UA human_1 receives these idea alternatives and selects one generated idea through DA. DA then informs StA_spatial_organization to send this idea to ScA_sunlight as the input of Scene 3.

4.2.3. Scene 3: Sunlight

Pre-condition: DA receives the message, and sends this message to ScA_sunlight through StA_spatial_organization. ScA_sunlight sends its design problem (sunlight issue) to UA human_3 through DA. A ring-type communication for idea association starts, which involves several loops from UA human_1, RA software_3, RA software_1, RA software_2 to UA human_2.

Process: UA human_1 starts generating the first idea for the sunlight issue. RA software_3 receives the idea, and then generates a new idea by using contiguity behaviour. Then RA software_1 uses similarity behaviour to generate another idea. Finally, RA software_2 passes his term, and then UA human_2 generates the last idea in the first loop. The following loops involve the same linking sequence.

Termination: When the time is over, ScA_sunlight terminates the scene, and informs StA_spatial_organization to store these ideas in the data repository. DA then informs StA_spatial_organization to terminate the whole play, and sends all the generated ideas to UA human_1, UA human_2, and UA human_3 through DA. All the selections and decisions are recorded for further learning.

Finally, the linking play among these six agents (human and machines) generates an idea map by dynamically linking related design ideas in each scene. The generated idea map, the linking process and the internal interplay are displayed on the Linking Interplay interface (Figure 4).

5. Discussion and Conclusion

Through the design example simulation, some lessons from the distributed interactions are observed. These characteristics provide distributed linking mechanisms that can support idea association in the distributed collaboration environment. These characteristics are described as follows:
1. Multiple participants involve distributed interactions with human and machine. It will therefore be helpful for idea association in a distributed computational environment.

2. Distributed knowledge connects the blocked linking process. Through these interactions among agents, the blocked linking process can be connected effectively, including combining ideas or passing to the next agent automatically.

3. The design task is decomposed into two main parts: actions and contents, which can be effectively executed by different agent entities through the linking behaviours and interplays.

In addition, users need to have direct control over the linking process in order to improve the efficiency of interactions and shorten the long learning curve that automated methods take. Following the development of our system, an essential prerequisite of preparation for supporting distributed interactions in creative problem solving meetings can be established. Further research such as a brainstorming session for problem solving in a design studio can be pursued under this framework.

Figure 4. The Linking Interplay interface.
References


