WPI

Improving Design with Agents,
or,
Improving Agents by Design

David C. Brown

AI in Design Group
Computer Science Department
Worcester Polytechnic Institute
Worcester, MA 01609, USA.

We start by assuming that NASA's future design and synthesis environment will be built as a *real* multi-agent system. In what follows, we will first look at the task that the environment will need to support, and then examine the consequences of using agents for this environment.

ASSUMPTION

- ❐ Assume that the design environment is built using agents.
  i.e., situated, autonomous, flexible
- ❐ Not just a distributed system.

➠ Consider factors that affect agents.

Like all good designers, we examine the requirements for such an environment. It’s immediately clear that most of the design decisions will be critical, and that the activity will be non-routine with creativity involved.

DESIGN PROBLEM REQS

- ❒ Use in Space ➠ hazardous environment
- ❒ High speed ➠ stresses, fast reacting
- ❒ Human users ➠ safety, reliability

i.e., critical design decisions

- ❒ New, very unusual and difficult problems
  ➠ non-routine design,
  creativity

Other requirements on the synthesis environment, due to the designs to be generated, will need to be handled using a distributed, concurrent and integrated approach. Consequently the environment will be very complex.

OTHER ASPECTS

❒ Repairability, etc. ➠ DFX
(life cycle issues)

❒ Complexity ➠ Decomposition

➠

Concurrent Engineering,
teams,
distributed designers,
parallel activity,
integration.

i.e., a very complex system.

Highly reliable designs need to be generated. Design reuse and Simulation are the two software solutions.

RELIABILITY

- ❒ highly reliable design needed

Usual methods:

- ❒ Reuse known reliable designs
  - ◆ less able to do this here
- ❒ Build and test
  - ◆ expensive & slow
- ❒ Simulate
  - ◆ Virtual build and test
  - ◆ Virtual Reality
  - ◆ Simulation Based Design

On the dimension that goes from “common” to “uncommon”, it’s clear that most of the design problems to be tackled using the environment will be quite unusual, with requirements that have not been seen before. This makes both design reuse and design process reuse difficult.

FIRST DESIGN

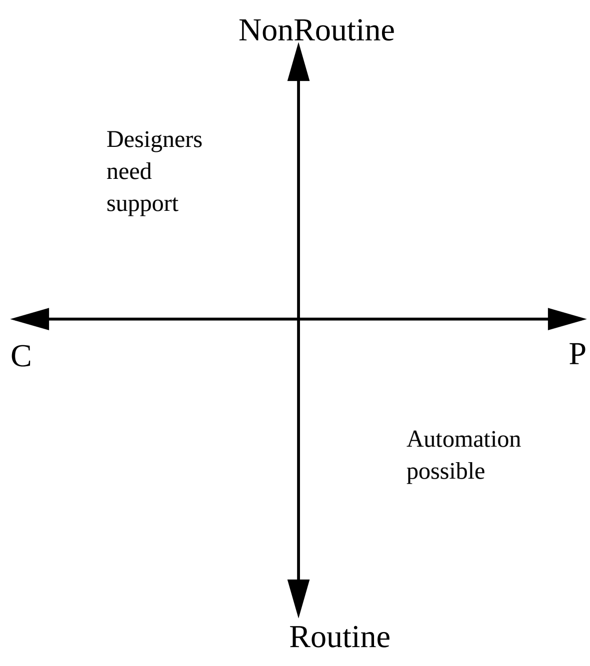
First Design
- new reqs.
- hard to reuse existing designs
- hard to reuse existing design processes

Redesign and Variational design
- based on existing design
- based on existing type
- reuse easier

Design situations vary during the design process depending on the knowledge available and the experience of the designer(s). A Routine situation is recognizable and both the methods and the knowledge can be immediately retrieved for that situation. In a Non-Routine situation this is not the case. The space of design situations is multi-dimensional, but here we just concentrate on the abstractness of what needs to be decided, the Conceptual to Parametric dimension. Moving in the non-routine conceptual direction requires the designers to be provided with support. Routine parametric situations can be automated.

ROUTINENESS

Design situations:



If the need for Creativity is perceived then it can act as a goal to the designer, producing different behavior. As creativity is determined relative to a standard, then designers will attempt to produce non-standard designs or use non-standard design processes. The unusual nature of the design requirements in this synthesis environment will already be forcing the designers towards creativity.

CREATIVITY

❒ Creativity is determined by comparison with *a standard*.

❒ The comparison applies to the:
- ◆ Design Process
- ◆ The Design Product

❒ The standard can be set by the past performance of the:
- ◆ Individual (designer or design team)
- ◆ Community

❒ Creativity arises in situations where the designer is forced to produce unusual designs or processes.

❒ Creativity can act as a "goal", changing activity.

Another issue to consider is, how the choice of agents might be made. There are several ways to decompose a system into agents. In a complex system several of these would be competing as candidates. There probably isn't any single correct way.

DECOMPOSITION

- How to decompose into agents?
  - there are many pressures
- by design process
  - tasks & subtasks
  - by reasoning type
    - selection; evaluation; estimation; etc.
- by design product structure
  - systems & subsystems
  - components & subcomponents
- by knowledge available
  - by discipline (e.g., kinematics)
- by convention/historical
  - organizational structure
  - legacy systems (e.g., CAD)

There are many categories of design that appear in the literature. As one moves further away from Parametric, fewer methods and software tools are currently available. Conceptual design, much needed for unusual design tasks, is the hardest to support.

TYPES OF DESIGN

- ❒ Parametric
  - ◆ High Automation,
    many methods.
    - KBS; CSP; CAD; Optimization; etc.
- ❒ Configuration
  - ◆ Medium Automation,
    possible.
    - KBS; Constraint techniques; GAs
- ❒ Conceptual
  - ◆ Low Automation,
    much harder at present.
    - representation difficult
    - procedures less known
    - rarely routine

Agent-based systems can be seen as a configuration of agents, both in the static sense with agents put together to built a system, but also in the dynamic sense, with interacting agents forming configurations in response to the shared task. Large agents, which are quite common and may already exist as legacy systems, have both advantages and disadvantages. Small agents remove many of those disadvantages, but add communication overhead. They would need to be custom built.

ABS AS CONFIGURATION

❐ An agent based system is a configuration
... viewed statically
... viewed dynamically, in response to use

❐ The size of the components to be selected for a configuration makes a difference.

❐ Large agents
- ◆ more functionality
- ◆ more knowledge, goals, constraints
- ◆ more assumptions made
- ◆ more hidden preconditions

➠
- ◆ less predictable
- ◆ less understandable
- ◆ less easy to model

❐ Small agents
e.g., SiFAs: Single Function Agents

From what has already been presented, it appears that the environment will need to be used for unusual, creative, conceptual, non-routine designs. This has many unfortunate consequences for the design of a multi-agent version of the environment.

THE CONSEQUENCES

The more first-time, 1-off, creative, conceptual, non-routine the design is,

the less...

... we can predict the design process.

... we can predict the result.

... we can predict the necessary agents of the MADS.

... we can predict an appropriate organization for the MADS.

... we can predict the necessary ingredients of the agents.

... we can predict the agent-agent interactions.

... existing software systems (including AI in Design) can help.

If we build a Multi-Agent Design System for NASA's design and synthesis environment, we are not likely to get it "right" first time. In order to compensate for this the system must at least act intelligently. A better response is for it to adapt, and consequently to compensate for its inadequacies.

ADAPT

i.e., if we build a Multi-Agent Design System we are not likely to get it "right" first time!

➠ *it must at least act intelligently*

it ought to **adapt**

The use of Learning in Multi-Agent Design systems is quite a new area. Learning might play a part in both the Support and Automation roles of the environment. There are rich opportunities for learning in MADS.

THE CURE?

- ❐ Learning
  i.e., ML in MADS

Rest of the talk:

Support & Automation

Rich Opportunities:
Dimensions of ML in D
Learning needs models
Evaluation of ML in MADS

MADS Research Examples

Conclusions

In a support situation most of the environment's intelligence will need to be added as it gets used, as it won't be possible to anticipate everything *a priori*. There are many possible things to learn, including learning about the user, the design product, etc.

SUPPORT

Conceptual, non-routine, creative, synthesis
➠ Support

❐ Intelligent Support - built in? no
- learned? yes

- ◆ know the user
  - learn
- ◆ know the design product
  - learn
- ◆ know the design process
  - learn
- ◆ know the architecture
  - learn
- ◆ know the agents
  - learn
  - capabilities, limitations, assumptions, ...
  - preferences, knowledge, goals, plans, ...
- ◆ know the interactions
  - learn

In an automation situation, much more of the intelligence can be built in from the start.

AUTOMATION

Parametric, routine, normal, reuse

➠ Automation

❒ Intelligent Automation - built in? yes
- learned? yes

- ◆ same issues
  - • can handle more of them.
- ◆ more concern with efficiency, and more ability to improve it.

In order to learn, agents need to have models. Updating these models constitutes the learning.

AN AGENT'S MODELS

Agents need models to learn:

❐ Model of Agent(s)
e.g., own abilities;
beliefs of others.

❐ Organizational Model
e.g., a hierarchy

❐ Cooperation Model
e.g., delegation

❐ Communication Model
e.g., who to send to

[Based on ideas of S. Labidi, 1997]

There are many variations on learning in design systems. The seven dimensions developed by Grecu & Brown provide a large space of learning activities, and suggest new opportunities.

VARIATIONS IN MLinD

1. What can trigger learning?
   e.g., expectation violations.

2. What elements support learning?
   e.g., Sequences of design decisions;
   Post-design Feedback.

3. What might be learned?
   e.g., Design Preferences.

4. Availability of knowledge for learning
   e.g., via direct communication.

5. Methods of learning
   e.g., Case-based and analogical learning.

6. Local vs. Global Learning
   e.g., Learning between design team agents.

7. Consequences of learning
   e.g., Design improvement;
   Process improvement;
   {Organization improvement}.

[Grecu & Brown 1998c]

Some of the research at the AI in Design Group at WPI is concerned with learning in design. Next we will provide three examples.

ML in MADS EXAMPLES

1 Learning Multidisciplinary Design Methodologies
- to improve integration

2 Adjusting an Agent's Design Preferences
- from agent interactions

3 Learning Key Features
- from expectation violations

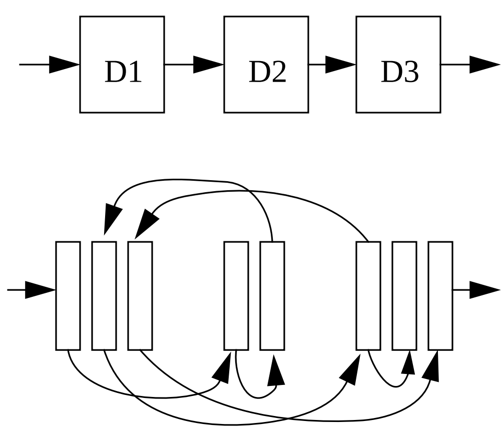
❐ Other ML in MA(D)S work
- Deng & Sycara 1997
- Nagendra Prasad, Lesser & Lander 1997
- plus other ML in MAS work
  http://dis.cs.umass.edu/research/agents-learn.html

This work uses an agent-based system to generate design traces that are turned into design methodologies for multi-disciplinary designs. Agents are built by cutting large blocks of discipline-based knowledge (e.g., D1, below) into smaller pieces. Each piece becomes an agent. The system is exercised with many design problems, generating many traces. Traces are patterns of design methods. These traces are clustered and generalized into methodologies that are appropriate for many design problems. Hence, methodologies are learned from system behavior.

DISCIPLINE PROBLEMS

- ❐ Multidisciplinary design problems.
- ❐ Knowledge: large discipline-based chunks.
- ❐ Leads to less integrated design process.

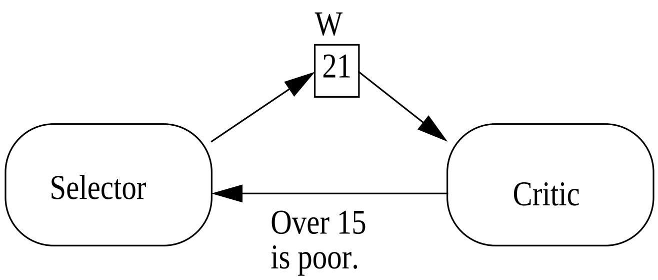
- ❐ Break chunks into smaller methods.
- ❐ Encode as agents.
- ❐ Experiment with resulting ABS.
- ❐ Traces ➠ learned Methodologies



[Shakeri, Brown & Noori 1998]

This work uses a conflict between agents to trigger learning. The Selector sets 21 as the value of W, but the Critic provides a critique indicating that values over 15 are poor. The Selector learns to avoid situations such as this. Learning significantly improved the number of interaction that occurred due to conflict.

DESIGN Preferences



❐ Learning is triggered by conflict
e.g., Selector-Critic; Selector-Selector

❐ Use concept formation to learn responses of agents in particular situations.

❐ Depending on experiences, interactions reduced by 9% - 55%

[Grecu & Brown 1996]

In this work, what triggers learning is expectation violations. The agent reasons out what features might have contributed to the violation, and then uses some learning experiments to determine the key features, i.e., those that are most predictive of the violation.

KEY FEATURES

- ❐ Agents have Expectations
  e.g., values,
  response time of an agent,
  quality of agent's response, ...
- ❐ Agent detects expectation violations
- ❐ Knowledge used to produce list of features that might have produced this violation.
- ❐ Determine key feature(s) using inductive learning experiments.
- ❐ Incorporate learned relationship into knowledge.

[Grecu & Brown 1998a]

A complex multi-agent design system requires very careful and comprehensive evaluation, as there are many possible effects that might alter its performance.

ML in MADS EVALUATION

❐ What to consider when evaluating distributed learning in design systems.

For example:

❐ The response to objectives
e.g., low cost

❐ Learning processes shared by multiple objectives.

❐ Interference of learning processes.

❐ Cross-talk resulting from training on several classes of design problems.

[Grecu & Brown 1998b]

The multi-agent implementation of the design and synthesis environment will have faults built into it. It will need to learn in order to compensate for and correct these problems. The area of learning in multi-agent design systems is an important and exciting new challenge that will have significant payoffs.

CONCLUSIONS

- ❐ The nature of the design problem affects the use of Agents for constructing an environment for the design of future aerospace systems.
- ❐ Any MADS built will be inefficient and ineffective relative to the task.
- ❐ It will need to compensate for these weaknesses by learning.
- ❐ ML in Design research is flourishing.
- ❐ ML in MAS research is flourishing.
- ❐ ML in MADS research is newer but is an area of where major opportunities exist for significant advances.

This is a very small selection of the references about Agents, Learning in Design, Learning in Multi-agent systems, and Learning in Multi-agent Design systems.

REFERENCES

D. L. Grecu & D. C. Brown, “Learning by Single Function Agents During Spring Design”, *Artificial Intelligence in Design’96,* (Eds.) J. S. Gero & F. Sudweeks, Kluwer Academic Publishers, 1996, pp. 409-428.

D. L. Grecu & D. C. Brown, “Guiding Agent Learning in Design”, *Proc. KIC3: Third workshop on Knowledge Integrated CAD,* Tokyo, Japan, 1998a.

D. L. Grecu & D. C. Brown, “Evaluating the impact of distributed learning in real-world design problems”, In: *Preprints AID98 Workshop on Machine Learning in Design,* Lisbon, Portugal, 1998b.

D. L. Grecu & D. C. Brown, “Dimensions of Machine Learning in Design”, *AI EDAM,* special issue on Machine Learning in Design, (Eds.) A. H. B. Duffy, D. C. Brown & A. K. Goel, Cambridge U.P., 1998c, pp. 117-121.

S. Labidi, “Managing Multi-Expertise in Design of Effective Cooperative Knowledge-Based Systems”, *Proc. IEEE KDEX’97, Knowledge & Data Engineering Exchange,* Newport Beach, CA, 1997.

M. V. Nagendra Prasad, V. R. Lesser & S. E. Lander, “Learning Organizational Roles for Negotiated Search in a Multi-agent System”, In: special issue on Evolution and Learning in Multiagent Systems, *International Journal of Human-Computer Studies*, 1997.

C. Shakeri, D. C. Brown & M. N. Noori, “Discovering Methodologies for Integrated Product Design”, *Proc. Artificial Intelligence and Manufacturing: Second Bi-annual AI & Mfg. Workshop,* Albuquerque, New Mexico, 1998.

D. Zeng & K. Sycara, “Benefits of Learning in Negotiation”, *Proceedings AAAI-97.*