

Using Single Function Agents for Design

D. C. Brown

*Computer Science Department, Worcester Polytechnic Institute,
Worcester, MA 01609, USA.*

Ph: (508) 831-5618; FAX: (508) 831-5776; Email: dcb@cs.wpi.edu

Abstract

Our goal is the detailed study of the interactions, conflicts and conflict resolutions that are possible with a multi-agent design system. We are concerned with the “primitives” of knowledge and reasoning. We have stripped down each agent to an extremely simple form, in order to closely study the multi-agent system’s behavior, the functionality required, the interactions that are possible (and necessary), the amount of and types of knowledge needed, and the conflicts that occur. In this way we can build a better understanding of design conflict management.

Keywords

SiFA, Design, Conflict, Selection, Critic, Multi-agent.

1 INTRODUCTION

Our goal is the detailed study of the interactions, conflicts and conflict resolutions that are possible within a multi-agent design system. We are concerned with the “primitives” of knowledge and reasoning for design, redesign, and conflict resolution.

Other work on multi-agent systems that do design has tended to use large agents with powerful capabilities and large quantities of knowledge (e.g., for example [Lander 91]). With so much power and knowledge it is hard to discover what is essential in the system.

In our approach we have stripped down each agent to an extremely simple form, in order to closely study the multi-agent system’s behavior, the functionality required, the interactions that are possible (and necessary), the amount of and types of knowledge needed, and the conflicts that occur. In this way we can build a better understanding of design agent conflict management.

Taking the multi-agent paradigm to such an extreme results in Single Function Agents

(SiFAs), where each agent performs a single function in the design process and therefore contains very specialized knowledge. Each SiFA will have different goals and knowledge. Current SiFA research focuses on parametric design. At present, each SiFA acts on a single parameter.

SIFAs are not the same as the large-grained agents in systems that try to incorporate legacy systems, such as equation solvers, as although those have restricted functionality they are usually very general, and may not be knowledge-based [Kuokka 95].

Using multiple agents brings the possibility of conflicts. Although it may be possible to build a system where all conflicts are resolved during development time, in most design problems this is either impossible or extremely difficult. There are many advantages of run time conflict resolution. Conflicts and their resolution cause the behavior of the system to emerge from the requirements of the particular problem at hand. The system is more flexible and does not suffer from the brittleness of traditional expert systems.

One way of resolving run time conflicts is for the agents to negotiate. This is the most general and flexible way of resolving conflicts. With SiFAs, many more conflicts become visible, due to the small size of each agent, and conflicts can't appear and be resolved within an agent. This is very important if we wish to understand the essence of conflict management.

2 RELATED WORK

To handle conflict resolution, Klein [1991] suggests using a hierarchy of conflicts with the most abstract conflicts at the top and most concrete conflicts at the leaves. There is a corresponding hierarchy of resolution strategies, resulting in domain dependent and domain independent conflicts and their associated resolution strategies. Other recent work on Conflict Management can be found in an AI EDAM special issue [Smith 95].

Negotiation is a common approach to conflict resolution. Sycara defines negotiation to be the process by which resolution of inconsistencies is achieved in order to arrive at a coherent set of design decisions [Sycara 90]. Negotiation proceeds with generation of a proposal, generation of a counter proposal based on feedback from dissenting agents, and then communication of justifications and supporting evidence. Other related work on negotiation includes Lander and Lesser's [1991], and Werkman's Designer Fabricator Interpreter [1992].

The first SiFA system, I3D [Victor 93], was a system that integrated part design and manufacturing plan production for Powder Processing applications. The agents were carefully sequenced, and all possible conflicts were anticipated and removed. I3D+ [Victor 94] had agenda-based scheduling of the agents and allowed conflicts about the values of parameters to occur among agents. SNEAKERS [Douglas 93] was built to train users in Concurrent Engineering. SINE was developed as a platform to build multi-agent design systems using SiFAs [Dunskus 95]. COSINE [Berker 96], is a simple wine glass designer. Other SiFA systems have been built and studied [Grecu and Brown 96].

3.2 The Parameter Block

To allow some potential, realistic actions of agents, all value, estimate, criticism, praise, and evaluation entities are stored separately, as first class objects, with the same status. They are all directly accessible and they can all be the “target” of an agent. All entities associated with the same parameter are organized in a Parameter Block. These blocks are stored centrally.

The root of the parameter block is the name of a parameter. The first level of reference has the parameter’s value and estimated value. The second level has evaluations, criticisms, and praises of the value and the estimate. The third level has evaluations, criticisms, and praises of second level entities, i.e., meta-level information about the design.

Any complete model of the formation, detection and resolution of conflicts between agents with different points of view needs to be able to represent and reason about the information in at least the first three levels of the parameter block.

3.3 Knowledge in a SiFA

Design and redesign knowledge is the first type. The design knowledge allows the agent to carry out its main function (e.g., selection). The redesign knowledge is used when the agent has to produce a response after it has already produced one.

The second type is conflict indication, detection, and classification knowledge that is used to carry out the respective tasks of conflict handling. The knowledge in this category also includes the knowledge of the types of conflicts in which the particular agent can be involved.

The third type is negotiation strategy selection, refinement, and execution knowledge. This enables the agent to negotiate with other agents in conflict situations. The system has to go through several steps during a conflict. These are Indication of a possible conflict, Detection of the conflict, Selection of a negotiation strategy to use, and Refinement of that strategy. The actual negotiation is simply the execution of the selected strategy. The result is either a solution to the conflict or the signaling of a failure. Some of these steps are trivial in the SiFA paradigm since the conflicts are very well defined and the number of different kinds of conflicts an agent can be involved in is very limited.

4 CONCLUSIONS

Currently, the negotiation behavior of the agents (their strategies) is very primitive. The simplicity of the negotiation process among the agents is one of the promises of the SiFA paradigm. However, more work is needed to construct and test negotiation strategies. A library of negotiation strategies for selectors, critics, evaluators, estimators, and praisers on all levels of reference would allow rapid (perhaps automated) construction of SiFAs with powerful negotiation techniques. However, in order to construct such strategies, the language used among agents needs to be extended.

History keeping, criticisms and praises, traces of negotiations, and analysis of conflict situations, all facilitate learning. Things that might be learned include dependencies between design parameters, preconditions and postconditions for rules, negotiation strategies, and knowledge about other agents. Learning should improve the quality of the final design, reduce the time required to do the design, and provide better conflict anticipation, avoidance, and resolution.

The SiFA-based model provides a very powerful tool with which to build and study design systems. The model allows a designer to build a system that will consider many different points of view while designing an artifact, without having to predict all possible conflicts between these points of view and resolve them while building the system. The separation of conflict knowledge from the design knowledge in the agents also has very important consequences in terms of the maintainability and the understandability of a system.

A contribution of this work is the analysis of conflicts in the SiFA framework. Since SiFAs provide the building blocks of a multi-agent design system, the conflicts between them represent the primitives of conflict situations in any design. The SiFA approach has allowed the discovery of a rich set of basic conflicts, as well as the knowledge required to handle them. Such an analysis could have only been done with something like the SiFA approach.

Considerable progress has been made in meeting the goals of this work. The work on SiFA's is still new and the model keeps evolving. Work has concentrated on analysis of their properties. A large-scale implementation of a pure SiFA-based system that is capable of solving design problems does not yet exist. Consequently, little attention has been given so far to the performance of large SiFA-based systems.

The SiFA framework provides a new perspective to study design systems, concurrent engineering, multi-agent systems, and conflict management. The major gain in building SiFA based design systems will be to learn more about the essence of conflict management and negotiation. SiFAs have proved to be very helpful in gaining insight into many design related research issues and they have revealed new research paths that should be explored further.

5 REFERENCES

- I. Berker and D. C. Brown. Conflicts and Negotiation in Single Function Agent Based Design Systems. *Concurrent Engineering: Research and Applications*, Special issue, "Multiagent Systems in Concurrent Engineering", D. C. Brown, S. Lander and C. Petrie (Eds.), May 1996.
- R. E. Douglas, D. C. Brown and D. C. Zenger. A Concurrent Engineering Demonstration and Training System for Engineers and Managers. *International Journal of CAD/CAM and Computer Graphics*, Special issue, "AI and Computer Graphics", I. Costea (Ed.), Hermes, Vol. 8, No. 3, 1993, pp. 263-301.
- B. V. Dunskus, D. L. Grecu, D. C. Brown and I. Berker. Using Single Function Agents to Investigate Conflicts. (*AI EDAM*): *Artificial Intelligence in Engineering Design, Analysis and Manufacturing*, Special Issue, "Conflict Management in Design", I. Smith (Ed.), Cambridge, 1995.

- D. L. Grecu and D. C. Brown. Learning by Single Function Agents During Spring Design. *Proc. AI in Design Conference, AID'96*, Stanford, CA, August 1996.
- M. Klein. Supporting Conflict Resolution in Cooperative Design Systems. *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 21, No. 6, Nov./Dec. 1991, pp. 1379-1390.
- D. R. Kuokka and L. T. Harada. Communication Infrastructure for Concurrent Engineering. (*AI EDAM*): *Artificial Intelligence in Engineering Design, Analysis and Manufacturing*, Special Issue, "Conflict Management in Design", I. Smith (Ed.), Cambridge, 1995.
- T. Finin, J. Weber, G. Wiederhold, M. Genesereth, R. Fritzson, J. McGuire, D. McKay, S. Shapiro, R. Pelavin and C. Beck. *Specification of the KQML Agent Communication Language*, DARPA Knowledge Sharing Initiative's External Interfaces Working Group, 1993.
- S. E. Lander and V. R. Lesser. Customizing Distributed Search Among Agents with Heterogeneous Knowledge. *Proceedings of the 5th International Symposium on AI Applications in Manufacturing and Robotics*, Cancun, Mexico, December 1991.
- I. Smith (Ed.) Special Issue, "Conflict Management in Design", *AI EDAM: Artificial Intelligence in Engineering Design, Analysis and Manufacturing*, Cambridge, Vol. 9, No. 4, 1995.
- K. P. Sycara. Cooperative Negotiation in Concurrent Engineering Design. *Cooperative Engineering Design*, Springer Verlag Publications, 1990, pp. 269-297.
- S. K. Victor, D. C. Brown, J. J. Bausch, D. C. Zenger, R. Ludwig and R. D. Sisson. Using Multiple Expert Systems With Distinct Roles in a Concurrent Engineering System for Powder Ceramic Components. *Applications of AI in Engineering VIII. Vol. 1: Design, Methods and Techniques*. G. Rzevski, J. Pastor and R. A. Adey (Eds.), Proc. 8th Int. AI in Engineering Conf., Toulouse, France. Elsevier, June 1993, pp. 83-96.
- S. K. Victor and D. C. Brown. Designing with Negotiation Using Single Function Agents. *Applications of Artificial Intelligence in Engineering IX*. G. Rzevski, R. A. Adey and D. W. Russel (Eds.), Proc. 9th Int. AI in Engineering Conference, Pennsylvania, USA. Computational Mechanics Publications, July 1994, pp. 173-179.
- K. J. Werkman and M. Barone. Evaluating Alternative Connection Designs Through Multi-agent Negotiation. *Computer Aided Cooperative Product Development*. D. Sriram, R. Logcher, S. Fukuda (Eds.), Lecture Notes, No. 492, Springer Verlag, 1992, pp. 298-333.

6 BIOGRAPHY

David C. Brown is a Professor of Computer Science at WPI, and is a member of the ACM, IEEE Computer Society, the AAAI, and IFIP WG5.2. He is on the Editorial Boards of the Journals "AI in Engineering, Design, Analysis and Manufacturing" and "Concurrent Engineering: Research and Applications". His research interests include computational models of Engineering Design, and the applications of Artificial Intelligence to Engineering. He is the author, with B. Chandrasekaran, of the book "Design Problem Solving: Knowledge Structures and Control Strategies", Pitman Publishing, Ltd., and a co-editor of "Intelligent Computer Aided Design", Elsevier Science Publishers B.V. (North-Holland).

3 THE SINGLE FUNCTION AGENT PARADIGM

A SiFA is an agent that performs a single **function** on a single **target** from a single **point of view**. The function performed by a SiFA determines its type. At present only a limited number of functions have been used. We conjecture that a set of agents with these functions is sufficient for most design problem solving activities, although we admit that this is far from clear for more conceptual design activities. These agents types are:

- Selector:** Selects a value for a parameter by picking a value from a list. These are usually suitable for discrete valued parameters.
- Advisor:** Produces a value for a parameter by some means other than picking a value from a list. Advisors and selectors are where most of the design knowledge is stored.
- Estimator:** Produces estimates of values for a parameter. Unlike selectors, estimators can work with insufficient information.
- Evaluator:** Evaluates the value of a parameter, producing a measure of goodness.
- Critic:** Criticizes values of parameters by pointing out constraints or quality requirements that are not met by the current values.
- Praiser:** Praises values of parameters by pointing out why the values are desirable.
- Suggestor:** Suggests what to do to remove an existing conflict or to avoid a conflict altogether.

The target of a SiFA is a single parameter of the design. The point of view of an agent is some aspect of the design that the agent considers while doing its work. Usually, the point of view of the agent is a goal that the agent is trying to satisfy or optimize.

SiFAs implement the basis of Concurrent Engineering, i.e., early consideration of aspects of the product's life-cycle, by allowing agents, such as selectors and critics, to represent the points of view of these aspects at any decision making point in the design. Hence a decision about a length might be made from the points of view of packaging and of maintenance, and criticized from the point of view of cost.

3.1 Communication and Control

Analogous to team members talking to each other, SiFAs communicate by sending each other messages. Each agent can communicate directly with any other agent without the need for an arbitrator. The communication language used is based on KQML (Knowledge Query and Manipulation Language) [Finin 93]. Even though agents exchange messages directly, the current state of the design is stored centrally and is accessible to all agents in the system.

For current SiFA implementations an agenda mechanism is used to schedule agents. Agents are polled by the agenda and each agent indicates what it wants to do. Based on the replies, the agenda orders the agents according to the importance of the tasks they will carry out; e.g., conflict detection and resolution is more important than doing design. Then the agenda gives control to that agent. When the agent's work is finished, the cycle is repeated.