*Artificial Intelligence for Engineering Design, Analysis and Manufacturing* (2002), *16*, 93–102. Printed in the USA.

DOI: 10.1017.S0890060402020085

# Agent-based collaborative design system and conflict resolution based on a case-based reasoning approach

KYUNG HO LEE[1] AND KYU-YEUL LEE[2]

[1]Korea Research Institute of Ships and Ocean Engineering, KORDI, Design Automation Team, Taejon 305-343, South Korea
[2]Seoul National University, Department of Naval Architecture and Ocean Engineering, Seoul, South Korea

(RECEIVED May 26, 2000; ACCEPTED December 16, 2001)

**Abstract**

Under the concept of a global economy, enterprises are assigning design and production environments around the world in different areas. A serious issue of information exchange emerges as companies use traditional hardware and very distinct software appropriate to their field of expertise. To overcome the problem of low productivity due to the interruption of information, the concept of simultaneous engineering and concurrent design becomes very significant. In this article an agent-based ship design system is developed to support cooperation in distributed ship design environments. Above all, the conflicts that occur while sharing knowledge in the system must be resolved. One approach is to adopt a case-based conflict resolution strategy formulated to resolve current conflict on the basis of similar previously resolved cases in agent-based collaborative design system environments. To do this, conflict cases that occur in the initial ship design stage are extracted. On the basis of the extracted cases, a case base is constructed. In addition, a conflict resolution handler located in the superagent called a facilitator, an agent to control other subagents, is developed to treat conflict problems effectively by case-based reasoning. The case-based conflict resolution strategy is evaluated by applying it to a collaborative design process in the initial ship design stage, especially the machinery outfitting design, the preliminary design, the hullform design, and the structural design. Through the help of the collaboration of the design agents, the facilitator, the conflict resolution handler, and the case-based system, a designer can make decisions based on similar previously resolved cases.

**Keywords:** Agent-Based System; Case-Based Reasoning; Collaborative Work; Concurrent Engineering; Conflict Resolution; Ship Design

## 1. INTRODUCTION

The importance of a design methodology based on concurrent engineering in a distributed environment is increasing day by day. Many attempts have been made to resolve the limitations of communication between distributed and heterogeneous systems and the sharing of the design information with each other. The adoption of agent technology is one of the solutions to solve these problems (Park et al., 1994; Jha et al., 1998). Recently many attempts have been made to construct collaborative systems to support cooperation in distributed ship design environments (Fujita & Akagi, 1998; Kim et al., 1999). Because ship designing is a very complicated process and manipulates quantities of data, it is necessary to construct the Computer-Supported Collaborative Work (CSCW) system based on the agent technology. To construct the agent-based concurrent design system, intercommunication between agents, conflict resolution, and negotiations that occur in the middle of the decision-making process are key issues to be resolved.

This paper describes an engineering knowledge sharing environment for an agent-based ship design system. The focus is on the methodology of conflict resolution, based on the case-based reasoning (CBR) techniques.

## 2. COOPERATIVE SHIP DESIGN

### 2.1. Cooperation in ship design

Recently, enterprises are assigning design and production environments around the world in different areas to follow the concept of a global economy. A serious issue of infor-

Reprint requests to: Kyung Ho Lee, Korea Research Institute of Ships and Ocean Engineering, KORDI, Design Automation Team, 171 Jang-Dong, Yusung-Gu, Taejon 305-343, South Korea. E-mail: khlee@kriso.re.kr

mation exchange emerges as companies use traditional hardware and distinct softwares appropriate to their field of expertise and combine different data from different areas to form common information. To stop the problem of low productivity due to the interruption of information and the independence of different designs and productions, the concept of simultaneous engineering and concurrent design becomes significant in the field of engineering.

Currently, in Korean shipyards, each field of design utilizes different CAD systems, such as the TRIBON system for the preliminary design; SIKOB for hydrostatic calculation; CADRA or AutoCAD in sketch designing; AUTODEF, TRIBON, or Intergraph VDS in hull design; and CADDS in outfitting design. Each of the different fields of design uses a distinct hardware environment and utilizes various tools for the specific design process. In addition, these systems are processed independently under a distributed environment. Presently, these systems are generating digitized design information, not sketch designs, but the occurrences of conflicts in the opinions are frequent among design processes because of the failure in unifying design information from different systems. Improvements in the design productions become challenging because of the independence and the division of design information in different fields. Considering the problems above, the concept of early design is important because it uses concurrent design to unify different information and knowledge among different systems to generate related design information at a preliminary stage. Many researchers have tried to solve the problem. Currently, among the different studies performed, the knowledge sharing approach, which introduces the concept of an agent, is receiving much of the spotlight (Finin & Wiederhold, 1991; Genesereth & Ketchpel, 1994). The intellectual ability of problem solving by the agent with its peculiar ability in unifying different collective groups and the internet-based support of divided environments are important and useful concepts in constructing simultaneous engineering concurrent design.

## 2.2. Agent-based cooperative ship design system

In order to construct the agent-based ship design system, we use several basic concepts. First, JATLite 0.4 is utilized for communication between agents. Second, a multipurpose knowledge interchange format (KIF) interpreter to offer the standard application programming interface (API) is developed. To do this the management of performatives, such as *interested* and *handles*, is necessary. Third, a standard algorithm is designed to process basic knowledge query and manipulation language (KQML) performatives. In this paper the basic KQML performatives such as *interested*, *handles*, *tell*, *ask-one*, and the *reply* are used. JATLite 0.4 has only the function of analyzing KQML performatives by token, but it does not offer the interpretation of the meaning.

On the basis of the three basic concepts mentioned above, the agent architecture is designed as shown in Figure 1. An

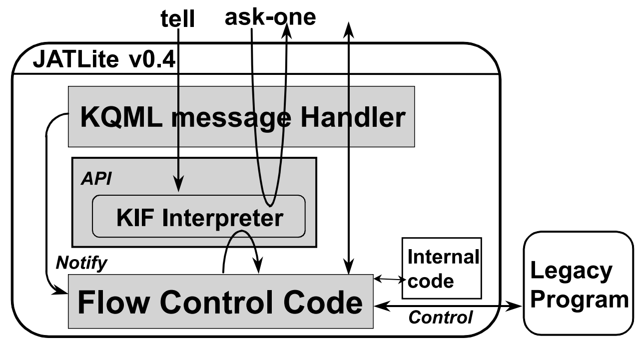


**Fig. 1.** The structure of the primary components for agentification.

agent is operated in a JATLite 0.4 framework, and agent communication language (ACL) messages are transferred to the KQML message handler by way of the communication interface of JATLite 0.4. The contents of the message are transferred to the KIF interpreter once again. The contents of the message with *tell* or *reply* performatives are stored in its own knowledge base, and *handles* and *interested* performatives are given special treatment.

An agent-based ship design system was developed as shown in Figure 2 based on the architecture shown in Figure 1. This section provides more detail on information control in the developed system and content-based message routing in a control module called the facilitator.

To construct the agent-based ship design system, the multiagent concept is adopted. For the communication between the agents, each agent must handle the address of other agents and know their interfaces. This can make the construction of the system complicated and difficult. Hence, this system adopts a federated architecture to reduce the load of communication problems between agents and to manage messages in a centralized structure. Under the federated architecture, the facilitator takes charge of linking, mediation, monitoring, and coordination between the agents. That is, if the KQML message is sent to the facilitator, the facilitator sends the message to an adequate agent according to the contents of the message or the registered interface and replies to the sender again. The core concept of the roles of the facilitator is content-based routing (CoBR). That is, the facilitator interprets the KQML message sent from

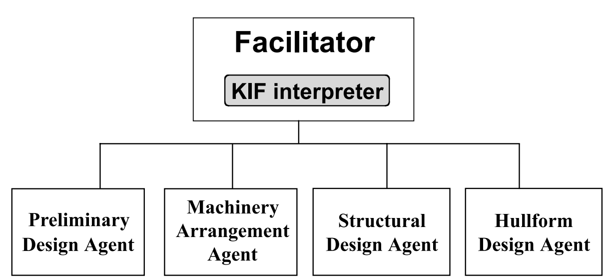


**Fig. 2.** The configuration of the initial ship design agent system.

https://doi.org/10.1017/S0890060402020085 Published online by Cambridge University Press

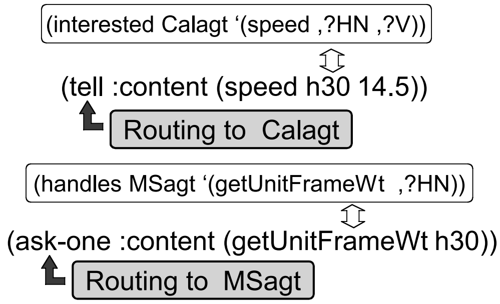


**Fig. 3.** Content-based routing in the facilitator.

an agent and performs the routing of the message based on its contents according to the registered interface.

Figure 3 shows an example of CoBR performed by a facilitator in an agent-based ship design system, where (interested Calagt '(speed ,?HN ,?V)) means that Calagt (ship basic calculation agent) is interested in a (speed ,** ,**) type of sentence, and the question mark denotes that the term is an individual variable. The meaning of the message is that the Calagt agent wants to know the speed of a ship for a ship number to be designed. Figure 4 shows one of the GUI windows implemented in the agent-based ship design system. The given requirement data in this window are routed to the Calagt agent by the facilitator according to the previous registered message. Somewhat (handles MSagt '(getUnitFrameWt ,?HN)) means that the MSagt agent can reply to (getUnitFrameWt ,**) typed questions.

If the following ACL messages are sent to the facilitator, *tell* and *ask-one* performatives are routed to Calagt and MSagt, respectively.

(tell :content (speed h30 15))
(ask-one :content (getUnitFrameWt h30))

## 3. CONFLICT MANAGEMENT

Most design tasks involve the management of conflict. Conflict arises when contradictory requirements are imposed upon characteristics of artifacts, upon the process of their creation, and/or upon their intended use. Even individual design requires trade-offs because of competing design criteria, such as functionality, safety, cost, and so on. The ability of designers to avoid or minimize conflict through judicious trade-offs, careful negotiations, and other methods becomes their most valuable skill.

Resolution and detection of conflicts are especially difficult when the design task, as well as knowledge concern-

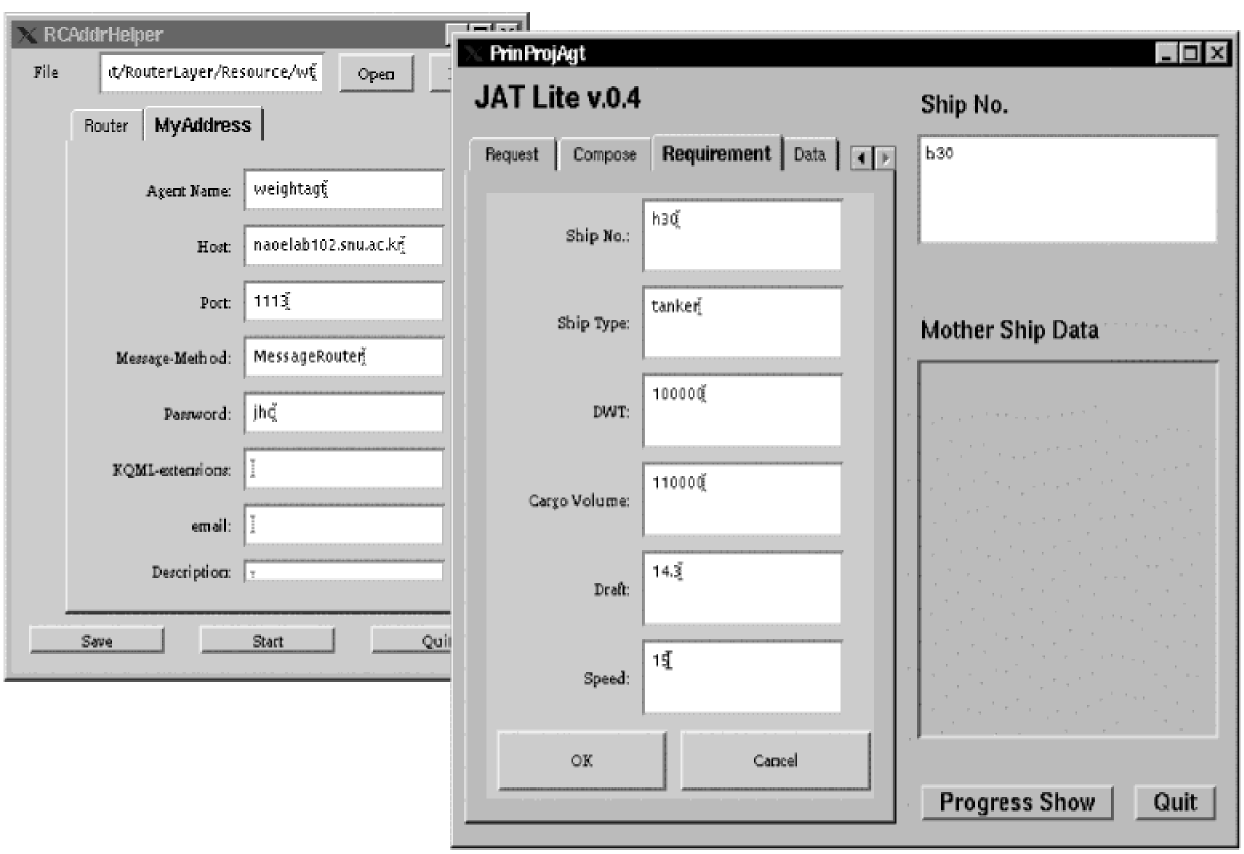


**Fig. 4.** The agent-based initial ship design system.

https://doi.org/10.1017/S0890060402020085 Published online by Cambridge University Press

**Design Viewpoint**

- Preliminary Design : Economic design (Low price, High Performance)
  >> Shorten Ship Length
- Hullform Design : Slender Stern Shape of Ship
- Structural Design : To satisfy structural strength >> Large Web Frame
- Machinery Arrangement Design : Widen Engine Room(E/R) Space

**Conflict** : Machinery Arrangement Design
"Insufficient Space between Machinery and Hull in Engine Room"

**Conflict Resolution**

- Request to Preliminary Design : "Lengthen E/R Compartment"
  Reply of Preliminary Design
  : (Case 1) Make longer E/R as 1 frame ⇨ Applying Design Change Strategy
  (Case 2) Cannot lengthen E/R from economic viewpoint
- Request to Hullform Design : "Widen Stern Shape of Hullform of Ship"
  Reply of Hullform Design
  : (Case 1) Change Stern Shape ⇨ Performance Check !!
  (Case 2) Cannot Change Stern Shape from performance viewpoint
- Request to Structural Design : "Reduce Web Frame Size"
- In Machinery Arrangement Design : Rearrange the Machinery

**Fig. 5.** Examples of design conflicts and their resolution in the initial ship design process.

ing such competing factors, are distributed among different actors with different perspectives. Therefore, conflict resolution is a central issue for problems and technical approaches related to cooperative design (Smith, 1995).

### 3.1. Conflicts in cooperative ship design process

Due to the different design goals of the ship design agents, conflicts can occur in the middle of cooperation between the distributed systems. In this section, the design conflicts and their resolution cases occurring in an agent-based ship design system are extracted. Although many conflicts occur in the ship design process, only some typical conflicts are treated in this paper. Figure 5 depicts conflict problems in the ship design process caused by different design notions of the related agents. That is, the preliminary design agent wants to shorten the length of the ship from an economic point of view. On the contrary, the hullform design agent prefers to lengthen the ship based on a performance viewpoint, such as low wave resistance, high speed, and so on. In the same manner, the machinery arrangement (MA) design agent has to arrange a great deal of machinery in the narrow space of the ship engine room (E/R) located in the stern part of the ship. Hence, the broader the E/R space, the better the viewpoint of layout. However, the goal of the hullform design agent is a slender stern part of the ship from the viewpoint of performance. In addition, the structural design agent is willing to arrange a bigger web frame from the viewpoint of strength. This makes an E/R space narrower. So many conflicts occur in the middle of the ship designing process that are caused by these different notions.

As mentioned before, one of the important things in constructing a collaborative design system is to systematically deal with the conflicts arising from the different design notions. That is, conflict resolution is a critical component of the cooperative design process. Ineffective management of conflicts can result in greatly increased product costs and reduced quality. There have been many attempts to resolve conflicts in diverse manners (Klein & Lu, 1989; Sycara, 1990; Klein, 1991). Of these attempts, the approach to resolve a current conflict problem based on previous similar cases is very reasonable. In this section, the case-based conflict resolution approach is presented by using the conflicts described above. An effective strategy for construction of a case base is first established for resolving the conflicts occurring in preliminary design, hullform design, and structural design centering around the machinery arrangement system. Then similar cases are selected through a similarity assessment, and they are presented as "ballpark" solutions. In this paper, a conflict resolution handler is developed to treat a series of conflict processes effectively. The conflict resolution handler is located in the facilitator, and it settles conflict problems that occur in the middle of the information interchange process within agents.

### 3.2. Case-based conflict resolution

#### *3.2.1. Construction of case base*

Generally, a case is composed of three parts such as a situation, solution, and the outcome. Here, to represent the conflicts between predescribed design agents, the case is subdivided into the following four sets.

1. *Title*. As a symbolic meaning, a title classifies the cases according to the kinds of conflicts. The following are titles of implemented cases.

   - conflict between MA design agent and preliminary design agent;
   - conflict between MA design agent and hullform design agent;
   - conflict between MA design agent and midship design agent;
   - conflict in the MA design agent.

2. *Description*. A description is a primary factor in deciding the characteristic of a case in which items for classifying similar cases are described. It plays a role in searching for related cases to solve a given problem. This corresponds to the indexing of cases, which is an important factor for classifying or searching cases. Information such as the following can be described as keywords or natural language.

   - ship type: bulk carrier, tanker, container, and so forth;
   - conflict type: hull clearance, passageway, and so forth;
   - related agent name: preliminary design, MA design, and so forth.

3. *Question/Answer*. After the related cases are selected from the indexing procedure based on the descriptions, question/answer is used to present the closest case through the similarity assessment for characteristic values. A weight can be given for each item.

   - E/R compartment can lengthen? (Yes/No)
   - Is it possible to change hullform of stern part to full? (Yes/No)
   - Can the Web frame be reduced? (Yes/No)

4. *Action*. The action part corresponds to the solution of a given problem. If E/R space has to be longer, a preliminary design agent and a machinery design agent must be reexecuted because of the change of ship length. The implemented actions are as follows:

   - make E/R longer (move FWD BHD);
     - execute redesign process of preliminary design;
     - redesign machinery arrangement process;
   - change the hullform of stern part;
     - execute evaluation process of speed/power;
     - redesign machinery arrangement process (check hull clearance and passageway);
   - reduce web frame size;
     - execute evaluation process of structural strength.

Figure 6 shows an example of implemented cases for conflict resolution in an agent-based ship design system.

#### *3.2.2. Conflict resolution handler*

In this paper, a case-based conflict resolution handler is developed to assist decision making in a cooperative preliminary ship design stage as shown in Figure 7. This handler, located in the facilitator, perceives whether a conflict has occurred or not, based on the information received from agents. When a conflict has occurred, it is resolved promptly and the reply is transferred to related agents (Lee, 1998).

The following is the flow performed by the conflict resolution handler to resolve the conflict problem.

1. *Conflict manifestation*. In this stage, keywords are generated for the occurrence of the conflict problem. That is, a problem description for indexing (e.g., passageway, hull clearance, etc.) and agent names where conflict is occurring (e.g., preliminary design, hullform design, etc.) are described.

2. *Goal generator*. This stage generates an API for indexing, and the information is sent to the conflict library to retrieve related cases.

3. *Indexing*. By keyword matching, all sorts of related cases are retrieved based on the keyword and the agent name.

4. *Similarity assessment*. Similarity scores are evaluated on selected cases by using a nearest neighbor algorithm.

5. *General advice*. The highest scored case is presented as a ballpark solution for a given conflict problem in this stage.

6. *Adaptation*. In general, the presented ballpark case does not exactly agree with the current conflict problem, so some modification is indispensable. For example, if “make longer E/R (move FWD BHD)” is presented as a ballpark solution, it is necessary to know how much one can lengthen the frame.

7. *Execution of conflict resolution*. The generated solution is transferred to related agents in the form of a KQML message and, according to the message, the redesign process is performed in each agent.

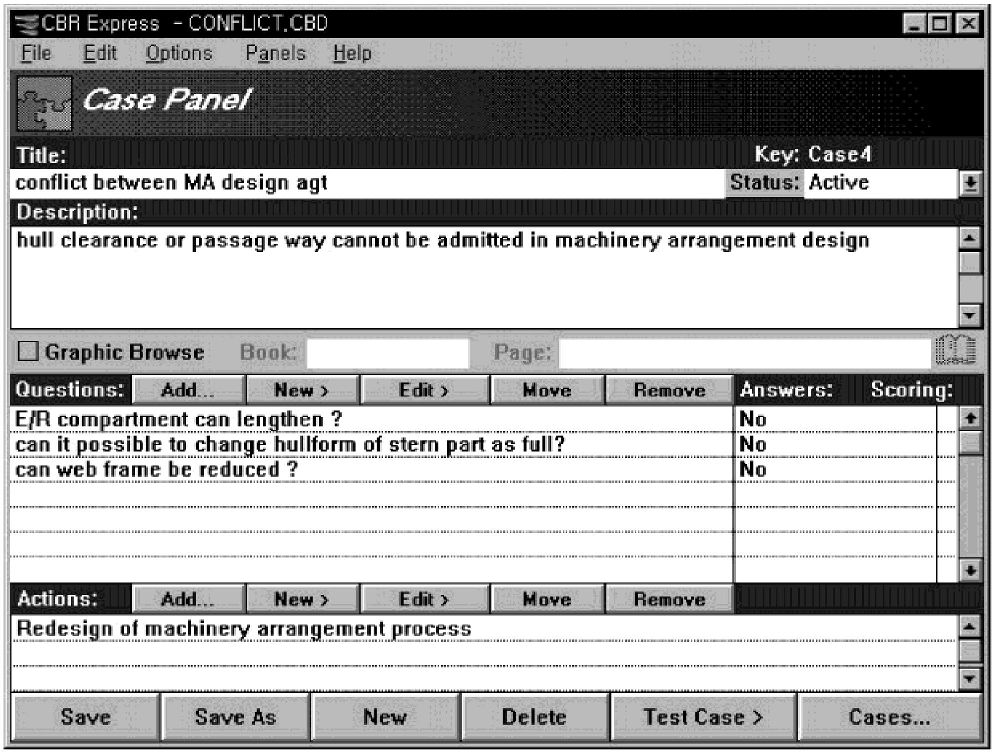


**Fig. 6.** An example of the implemented case base for conflict resolution between design agents.

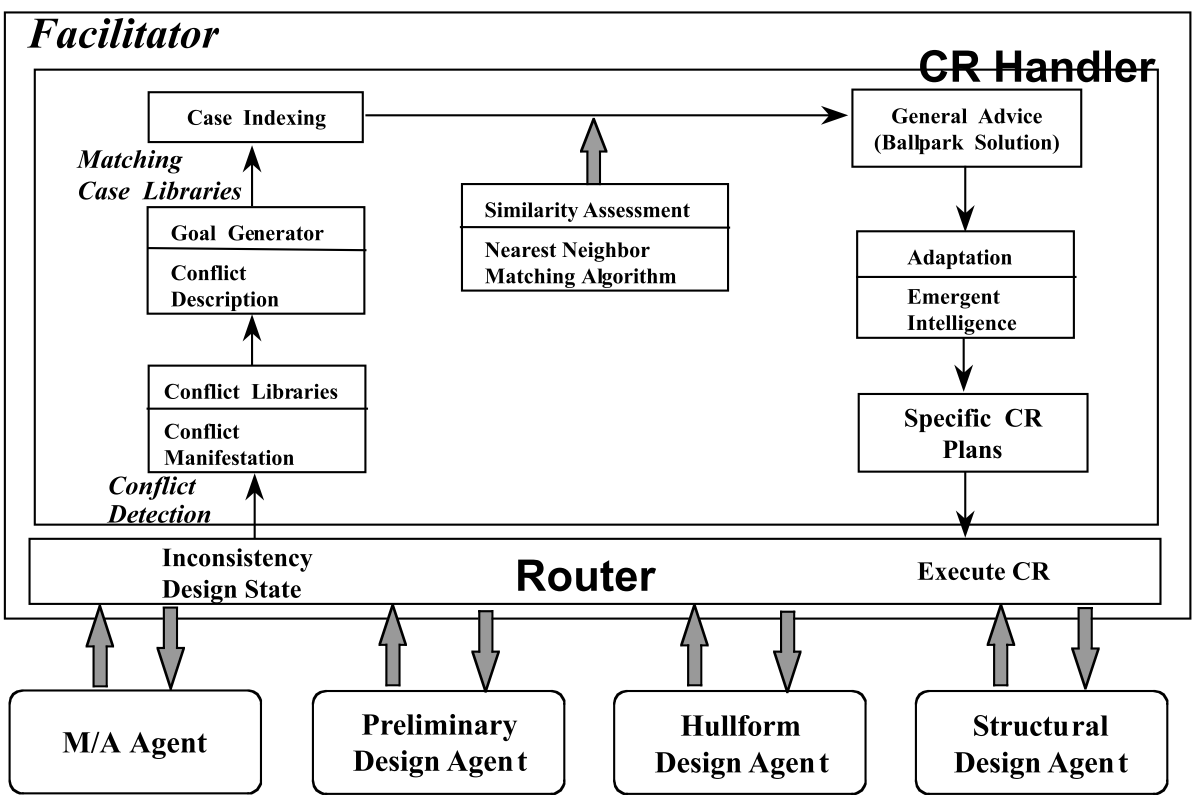


**Fig. 7.** The components of the case-based conflict resolution handler.

https://doi.org/10.1017/S0890060402020085 Published online by Cambridge University Press

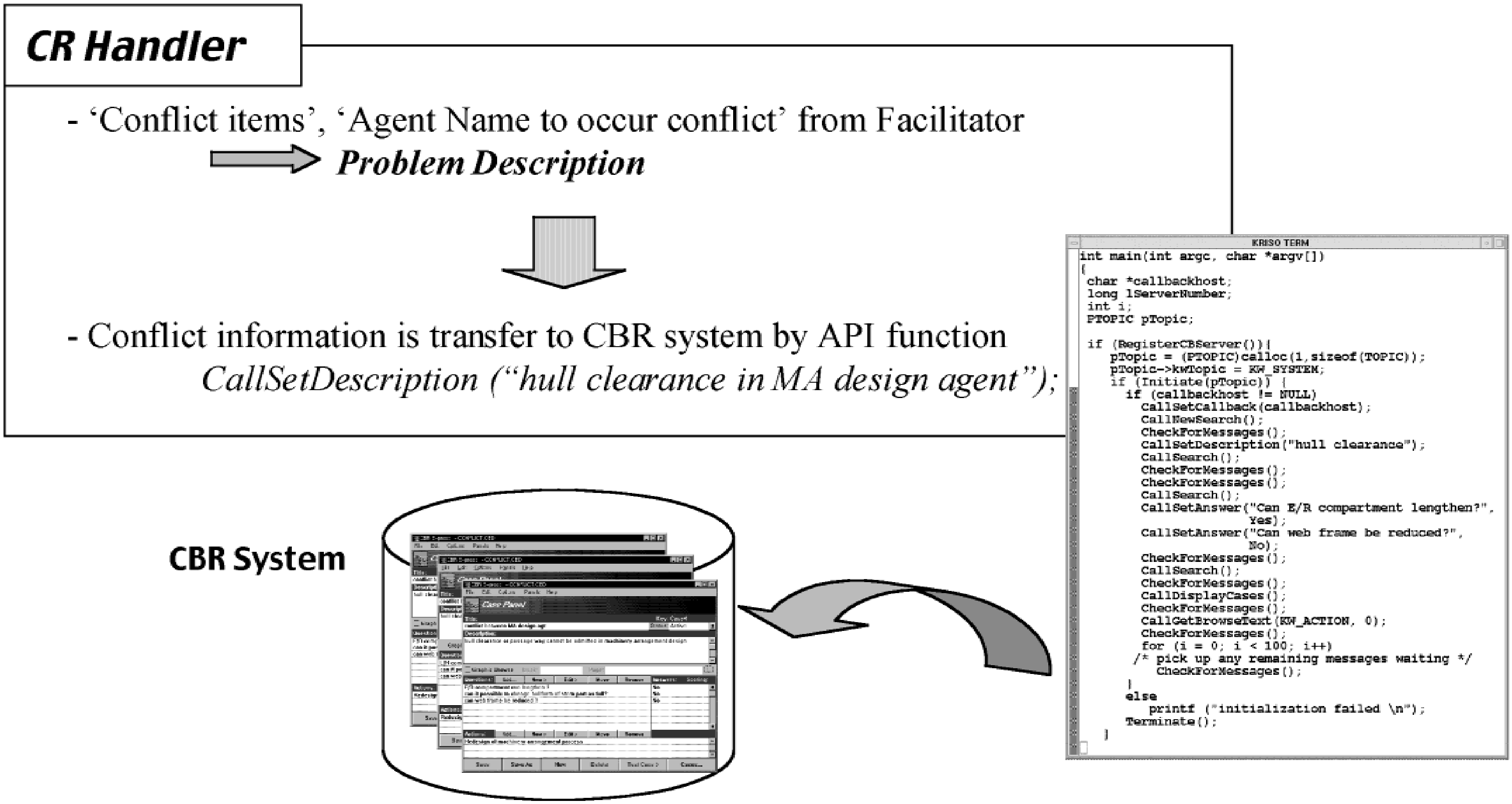


**Fig. 8.** The conflict resolution (CR) process in the CR handler and CBR system.

### 3.2.3. *Process of conflict resolution*

■ Machinery design agent to facilitator

Conflict in machinery design agent
: "Insufficience of clearance between machinery and hull of ship"

```
(tell :sender MA_design_agent
      :receiver facilitator
      :language KIF
      :reply-with MA0
      :content (and (conflict CF1)
            (conflict.type CF1 hullClearance)
            (conflict.shipId CF1 h30)
            (conflict.position CF1 (hullform.frame h30
              13))
            (conflict.source CF1 MA_design_agent)) )
```

In this KQML message conflict ID, type, ship ID, position, and agent name are described by using the conflict definition.

■ Facilitator to conflict resolution handler

The above message is only routed from the facilitator to the handler for resolving the conflict.

■ Conflict resolution handler to case-based system

The problem is described for searching similar cases based on contents and agent name transferred from the facilitator. This information is transferred to the case-base system by using API functions as follows. Figure 8 shows the concept of the conflict resolution process in the conflict resolution handler and CBR system.

CallSetDescriptions("hull clearance in MA design agent");

■ Case-based system

In order to evaluate the similarity score of the cases, the case-based system communicates with related agents by question/answer. The following is a KQML message transferred between the facilitator and preliminary design agent:

```
(ask-if :sender facilitator
        :receiver preliminary_design_agent
        :language KIF
        :reply-with facil1
        :content (ERComp.canLengthen (ship.ERComp
            h30) true) )
(tell   :sender preliminary_design_agent
        :receiver facilitator
        :language KIF
        :reply-with pre0
        :in-reply-to facil1
        :content true )
```

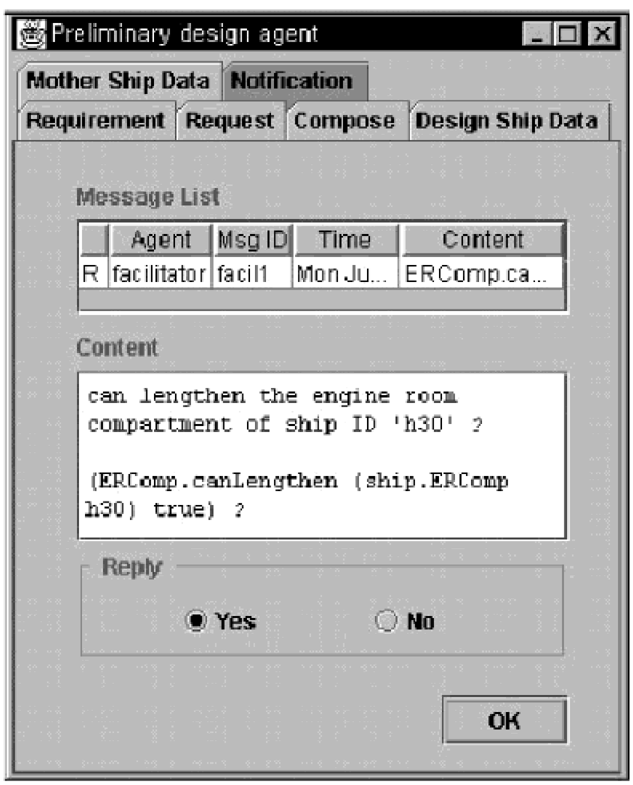


**Fig. 9.** The arrived message and reply in the preliminary design agent.

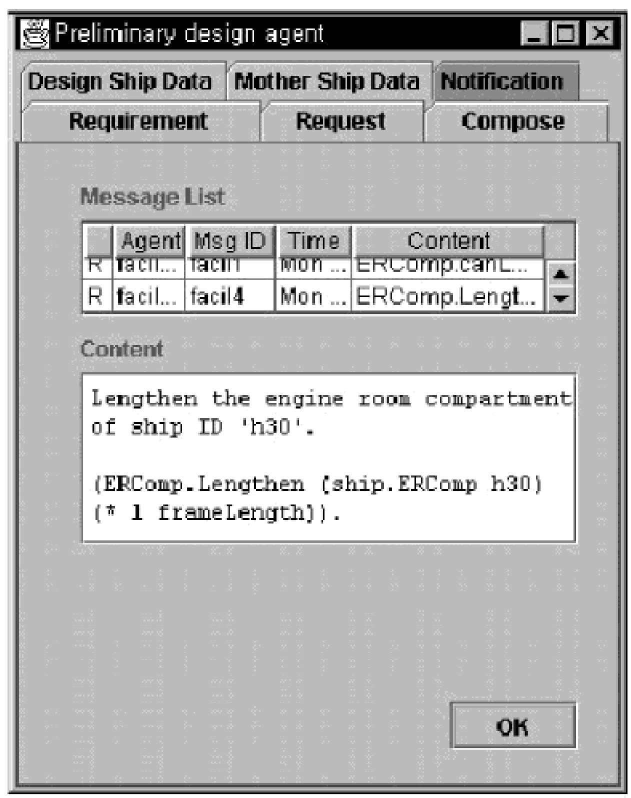


**Fig. 10.** A solution for the conflict and its action in the preliminary design agent.

https://doi.org/10.1017/S0890060402020085 Published online by Cambridge University Press

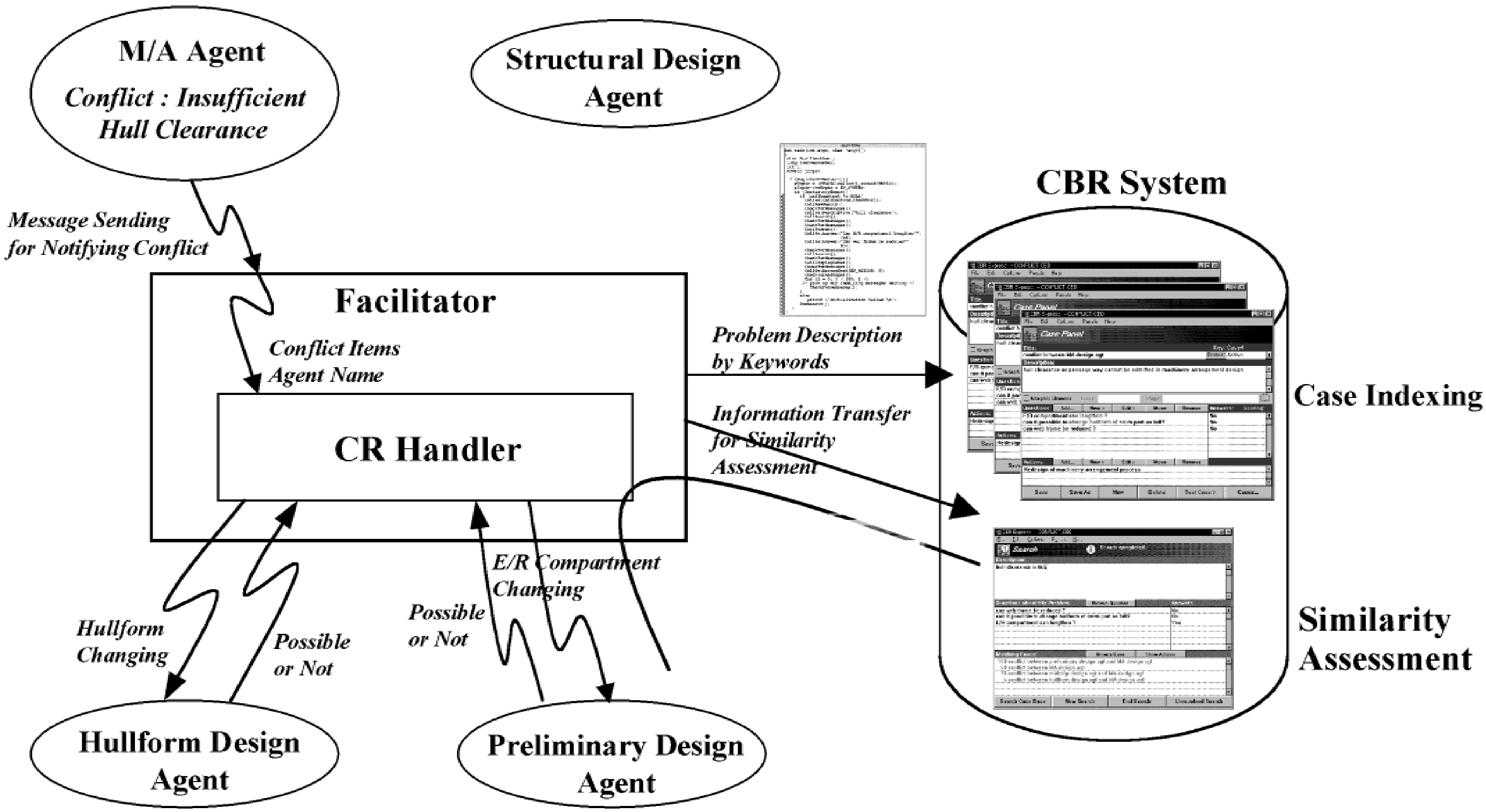


**Fig. 11.** The conflict resolution process in the collaborative ship design agent system.

After awhile an *ask-if* message is shown at the reply window as in Figure 9 in the preliminary design agent; if the designer answers the question, the reply is transferred to the *tell* KQML message in the above and sent to the handler via the facilitator.

Similar processes are performed between the facilitator and other agents such as the hullform design agent, the structural design agent, and the machinery design agent. Through these communication processes, the case-based system decides to lengthen the engine room compartment 1 frame. The message is transferred to the preliminary design agent in which the redesign process is carried out. Figure 10 depicts a solution for the given conflict problem based on CBR.

In this section, a detailed KQML message and the procedure for case-based conflict resolution are described through a scenario of problem solving. Figure 11 illustrates the concept of conflict resolution in the machinery design of a ship and collaboration in an agent-based ship design system.

## 4. EVALUATION AND TEST RESULTS

In this study, the implemented agent-based system reuses conventional design systems, while at the same time effectively supporting a distributed design environment by sharing and exchanging information appropriately. Therefore, the design results are already verified by conventional design systems and information exchange between distributed design groups is made much easier. In addition, conflict resolution is also supported by the system, thus reducing the design spiral. At this point the productivity of the initial ship design is enhanced by about 30% from the viewpoint of the designing period. The system is being applied to small and medium sized shipyards for practical use.

## 5. CONCLUSION

We discussed two topics in a collaborative ship design environment. One is the construction of an agent-based ship design system to realize a concurrent collaborative ship design environment; the second is the resolution of the conflicts that occur in the middle of cooperative design.

First, a framework of an agent-based ship design system to cooperate and share the information with each other under the distributed and heterogeneous system environments is developed. Second, a conflict resolution strategy based on a CBR technique is implemented. The conflict resolution handler plays a special role in detecting conflicts in the middle of cooperative work and treats conflict problems in cooperation with a CBR system. A case-based approach to resolve a current conflict problem based on previous similar cases is very reasonable. Through the help of the cooperation among the design agents, facilitator, conflict resolution handler, and case-based system, a designer can be supported effectively in his/her decision making based on similar previously resolved cases under the collaborative design environment.

## REFERENCES


Finin, T., & Wiederhold, G. (1991). *An Overview of KQML: A Knowledge Query and Manipulation Language*. Technical Report. Stanford, CA: Logic Group, Computer Science Dept., Stanford University.

Fujita, K., & Akagi, S. (1998). Agent-based distributed design system architecture for basic ship design. *Proc. 5th ISPE International Conference on Concurrent Engineering (CE98)*, Tokyo.

Genesereth, M.R., & Ketchpel, S. (1994). Software agents. *Communications of the ACM 37(7)*.

Jha, K.N., et al. (1998). Agent support for design of aircraft part. *Proc. DETC '98 ASME Design Automation Conference*, Atlanta, GA.

Kim, D.H., et al. (1999). A study on ship initial design agent system based on ACL and CORBA. *Transactions of the Society of CAD/CAM Engineers 4(4)*, 360–370 [in Korean].

Klein, M. (1991). Supporting conflict resolution in cooperative design systems. *IEEE Transactions on Stems, Man, and Cybernetics 21(6)*.

Klein, M., & Lu, S. (1989). Conflict resolution in cooperative design. *Artificial Intelligence in Engineering 4(4)*, 168–180.

Lee, K. H. (1998). *Case-Based Conflict Resolution and Negotiation in Agent-Based Design System Environment*. Doctoral Dissertation. Seoul: Seoul National University.

Park, H., et al. (1994). An agent-based approach to concurrent cable harness design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 8*, 45–61.

Smith, I.F.C. (1995). Special Issue: Conflict Management in Design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 9*, 245–246.

Sycara, K. (1990). Negotiation planning: An AI approach. *European Journal of Operational Research 46*, 216–234.


---


**Kyung-Ho Lee** is a Research Engineer in the Design Automation Team at the Korea Research Institute of Ships and Ocean Engineering (KRISO). He received his PhD in naval architecture and ocean engineering from Seoul National University in August 1998 and is a Professional Engineer in ship automation design. His research interests are artificial intelligence in design, concurrent engineering, machine learning, intelligent decision-making systems, and their application to the shipbuilding industry. His current research focuses on a collaborative system for ship design, intelligent navigation systems, and decision-making systems for damaged ships.

**Kyu-Yeul Lee** is a Professor in the Department of Naval Architecture and Ocean Engineering, Seoul National University. He received his PhD from Technical University of Hannover, Germany, in 1982. His research interests are geometric modeling, optimum design, and computerized ship design systems. His current research focuses on collaborative system for ship design, computer-aided design, and multidisciplinary design optimization.