Available online at www.sciencedirect.com

SciVerse ScienceDirect

Physics Procedia 25 (2012) 1569 – 1575

Physics Procedia

2012 International Conference on Solid State Devices and Materials Science

The Study on Collaborative Manufacturing Platform Based on Agent

ZHANG Xiao-yan [a,b],QU Zheng-geng [a,b]

[a]*College of Information Science and Technology Northwest University ,Xi'an, China*

[b]*Deptment of Computer Science ,Shangluo University ,Shangluo, China*

Abstract

To fulfill the trends of knowledge-intensive in collaborative manufacturing development, we have described multi agent architecture supporting knowledge-based platform of collaborative manufacturing development platform. In virtue of wrapper service and communication capacity agents provided, the proposed architecture facilitates organization and collaboration of multi-disciplinary individuals and tools. By effectively supporting the formal representation, capture, retrieval and reuse of manufacturing knowledge, the generalized knowledge repository based on ontology library enable engineers to meaningfully exchange information and pass knowledge across boundaries. Intelligent agent technology increases traditional KBE systems efficiency and interoperability and provides comprehensive design environments for engineers.

Keywords : collaborative manufacturing platform, multi-agent, knowledge

1. Introduction

The rapid advances in computing, Internet technologies and cross-industrial business strategy have given a boost to collaborative manufacturing development, which is promoted as a means for reducing or at least sharing. The collaboration process includes a complex myriad of partners from numerous companies and heterogeneous and large amounts of information resources. There is a strong need for exchanging manufacturing data from different sources in different formats. Moreover manufacturing development is not only collaborative but also knowledge-intensive. Participants which different disciplines and skill levels require to share both explicit and implicit knowledge with others, capture the alliance's insights and experiences, and communicate person-to-person to obtain the know-how [1].

Thus, requirements in two areas must be addressed: 1) How to organize multi-disciplinary participants and heterogeneous tools into a cooperative infrastructure on which they can develop at knowledge level; 2) How to effectively supporting the formal representation, capture, retrieval and reuse of manufacturing

 doi:10.1016/j.phpro.2012.03.277

knowledge including specifications, design rules, constraints and rationale, which may extend the whole manufacturing life and cross corporate boundaries.

We believe multi-agent and ontology technology have distinct advantages in dealing with the above issues [6]. In this paper, a scalable collaboration infrastructure based on agent is presented in section 2. Section3 provides a cooperated process of agents and is followed in Section 4 with communication based on ontology.

2.Architectural overview

Figure1 represents the architecture of our collaborative development platform, in terms of its agents. Multi-agent systems (MAS) offer an efficient and way to integrate heterogeneous applications. The Wrapper Agent provides wrapper service [2][3] enable agents to access technical data in commercial software through an Application Programming Interface (API) or the standard user interface. Based on the translation mechanism, distributed participators enable data sharing via messages of wrapper agent or user agents."

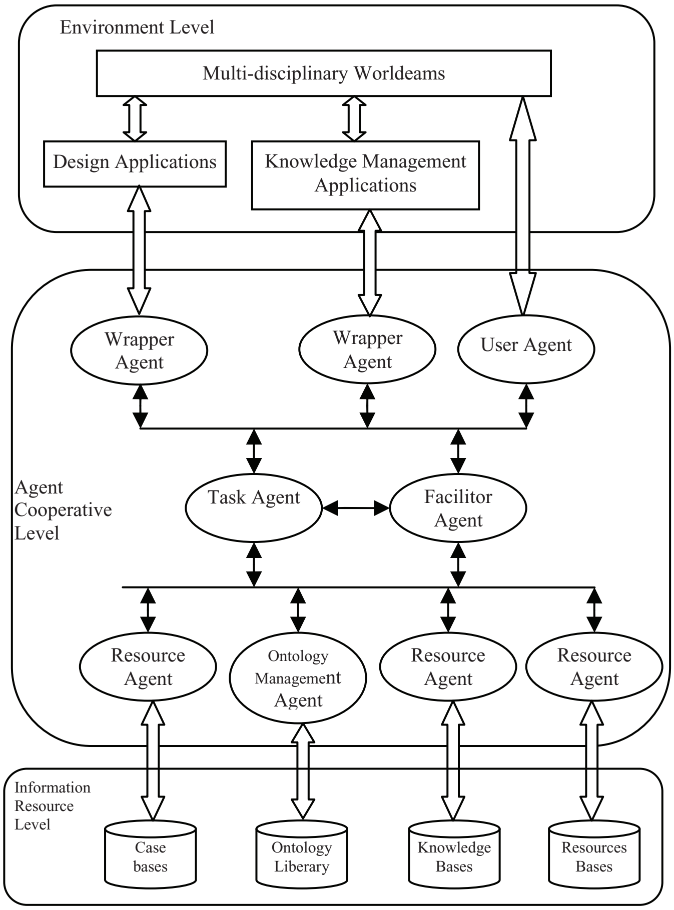

Figure 1. The architecture in terms of agents.

As shown in figure1 the architecture is comprised of three distinct levels. The first level is the "Environment level" which deals with external software that are classified into four categories: geometry model applications (such as CAD systems), knowledge management applications (such as KBE systems), engineering computing applications (such as SYS), and information management applications (such as PDM. systems). Covering the whole manufacturing life cycle areas, these applications would be related to different design phase.

The second level is the "Agent cooperative level". Enabled by a MAS architecture distributed users could complete their missions within the collaborative virtual environment. And Foundation for Intelligent Physical Agents (FIPA) Reference Model [2] is chosen as a technological basis for the system since it provides standards for heterogeneous interacting agents and agent-based systems, and specifies ontologies and negotiation protocols.

- User Agent: interacts with users, knowledge consumer, constituting the user's intelligent gateway into the systems. It uses knowledge of the system's common domain models (ontologies) to assist the user in formulating queries and in displaying agents' computing results.
- Wrapper Agent: can dynamically interface with a software system uniquely described by a software description. It provides a kind of wrapper mechanism used in legacy system which can translates the commands contained in ACL (Agent Communication Language) messages into operations to invoke applications enumerated in the environment level through API, as well as capturing application specific data and translating it to ACL messages.
- Task Agent: coordinates process of user requests processing by identifying and routing requests to the appropriate resource agents or wrapper agents and reassembling the results. It also tracks out the process and performs control for error.
- Facilitor Agent: provides a "yellow page" directory service for all agents. Each agent must advertise its descriptions (such location, name, the language it speaks) and the services it offers to the facilitor agent. Based on this information, the facilitor agent responds to queries from agents as to where to route their specific requests.
- Resource Agent: makes information and knowledge contained in an information source available for retrieval and update. It can answer and translate the queries expressed in ACL into a language consistent with the information sources (such as SQL). It also translates the results of the query into a form understood by other agents.
- Ontology management Agent: performs ontology operations. Ontology acts as a systematic vocabulary to explicitly state and exchange meaning of terminologies of applications, which supports an effective way for knowledge sharing, reuse and integration. The Ontology Management Agent provides a mapping from the common ontology to the database schema and language native to the resource agent represented.

Finally, the third level is the "Information resource level" .With the core of ontology library, which consists of ontologies about various application domains. information resource level include rule bases, design case bases, relational database and so on, called by a joint name "generalized knowledge repository", which store temporary data, expert experiences and all information and knowledge related to engineering design. Each resource is connected to a resource agent and imposes its vocabulary and constraints on it. In the next section we will explain the two main parts which are the basis of our work: multi-agent architecture and generalized knowledge repository.

3.Agent structure

Though the agents are classified into multiple types by their functionalities, each one has uniform logic structure shown in figure2.

Each agent consists of six parts: knowledge base, perception, action, communication, decision-making, and inner-state. Knowledge base stores the knowledge needed for collaborative problem solving, which may be given beforehand relied on the organization roles of agents. Perception module gets facts and state information of outer environments and stores in facts-base of knowledge base. Communication module interacts and cooperates with other agents by FIPA-ACL language. According to the external environmental situation, agent's own internal state (stored in inner-state module) and others' requests, decision-making module will reason by knowledge base module, and then determine the current content of action and communication. Action module makes the appropriate action under the action command passed by decision-making module.

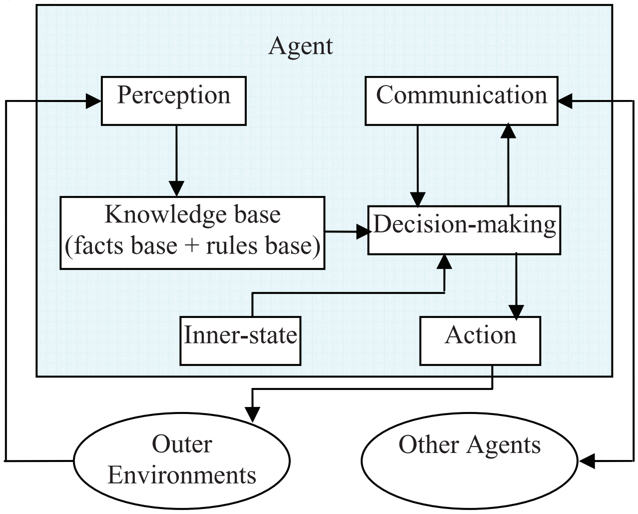

Figure 2. Basic structure of agents.

4.Case diagram of multi-agent system scenario

On the whole when the Wrapper Agent receives a task request, it gets necessary information and knowledge (such as previous design cases) from the Resource Agent. Based on these, it computes some parameters value and embeds them into operation commands on the underlying software systems. The reassembling results will return to users via the User Agent. The procedure of user requests is almost automatically processed because of negotiation and cooperation between agents. That means a deal of repetitive, routine but time-consuming engineering tasks can be automatically done instead. of engineers, whose role is to input the specifications and make the important design decisions.

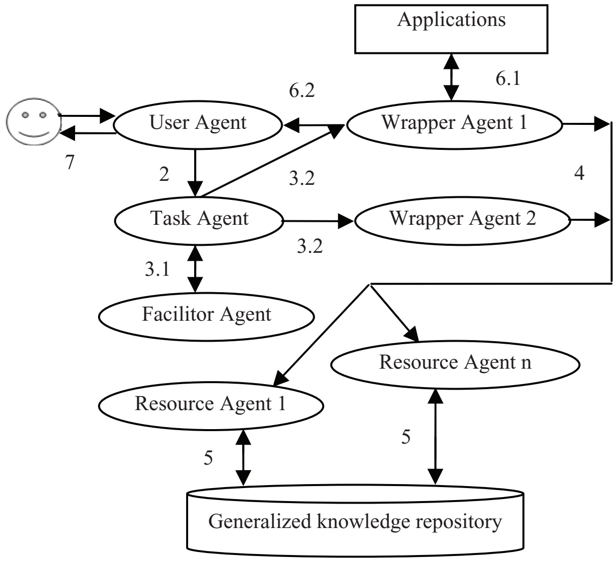

Figure 3. Case-diagram of user request processin.

Figure3 represents case-diagram of user request processing in the multi-agent system. At initialization, every agent need advertise itself to the Facilitor Agent, and get an ontological model from the Ontology Manage Agent in Ontology Library. It is required that every ACL message must contain ontology that is in use by agent, which can help other agents map unknown messages into its familiar form to understand what the messages real mean. The typical scenario is as follows.

- The user inputs a task request into the system, as well as ontology he used, from User Agent with GUI..
- The User Agent translates the request into the terms of ACL messages (contained ontology information) and passes it to the Task Agent for processing.
- The Task Agent decomposes the request into an ordered set of sub-requests and inquires the Facilitor Agent about agent's name who can achieve the sub-requests (3.1). The Facilitor Agent then checks the registered the Wrapper Agents for meeting requirements based on their advertisements at initialization and returns the right ones (including name, location and so on) to the Task Agent who will pass sub-requests to these agents respectively(3.2).
- Similar with step(3) the Wrapper Agent gets relevant the Resource Agent from the Facilitor Agent who can offer needed information and knowledge for completing the sub-request , then sends querying to it (in ACL message format).
- The Resource Agent constructs SQL queries or other type queries varied with schemas in generalized knowledge repository based on the ACL message’s contents, searching the concerning resources for needed information and knowledge. After obtaining the searching results, it returns them to the Wrapper Agent
- The Wrapper Agent composes gained information and knowledge, computing parameter values and invoking underlying software for through an API (Application Programming Interface) (6.1). After capturing application specific data (e.g. analyzed results from Matabele), it encapsulates the result into an internal format, passing to the User Agent (6.2).
- The User Agent translates fit the Wrapper Agent’s result together, convert the result information understandable by user, presenting results to the user for checking.

5.Generalized knowledge repository foundation

Traditional develop process sometimes depended too much upon designers' experience resulting in low efficiency of knowledge utilizing, reusing, sharing and inheriting. As manufacturing development is being more often done within a heterogeneous and collaborative environment, agents in the system have to exchange information and knowledge across various application domains. Semantic conflicts [4][5] are unavoidable when different terms express the same concepts or identical terms for different concepts. We propose the generalized knowledge repository. As figure1 shows, various information resources and knowledge resources, related engineering design are integrated by the ontology library which facilitates data exchange not only at a common syntactic, but also at a shared semantic level.

Exchange of information and knowledge between agents are achieved through ACL messages contained one ontology model. The ACL provides only syntax to encode messages without specifying terms' semantics contained in messages contents. The ontology provides a common vocabulary or lexicon and the relations between terms, which enable agents share the commonly agreed understanding. Shown in figure3 Agent2 in manufacture domain can translate messages sent by Agent1 in design domain into his acquainted format by mappings between different ontologies, which enable agents to capture real meanings of contents in exchanged message to avoid semantics conflicts during negotiation.

6.Conclusions

Collaborative manufacturing development has become important business strategy. In this paper, a framework for autonomous collaborative manufacturing development, in which a set of intelligent agents handle major interact tasks based on ontology, is presented. Finally, communication process between agents is provided, which identifies distributed participators enable to share heterogeneous data via messages of agents.

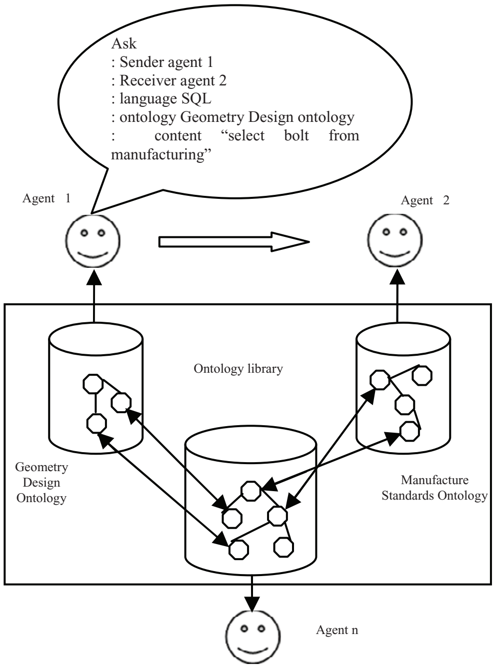

Figure 4. Agent communications based on the ontology library.

References

[1] Margaret Bruce, Fiona Leverick , “Success factors for collaborative manufacturing development: a study of suppliers of information and communication technology”, R&D Management, vol.25,2007,pp.33-44

[2] FIPA Agent Software Integration Specification, http://www.fipa.org/

[3] Chenguang Zhao, Qingshan Li, el al. “An Agent Based Wrapper Mechanism Used in System Integration”,pp.637-640,2008 IEEE International Conference on e-Business Engineering, 2008

[4] Lalit Patil, Debasish Dutta, Ram Sriram, “Ontology-Based Exchange of Manufacturing Data Semantics”, IEEE Transactions on Automation Science and Engineering, vol.2, 2005,pp.213-225

[5] Sudha Ram, Jinsoo Park, “Semantic Conflict Resolution Ontology (SCROL): An Ontology for Detecting and Resolving Data and Schema-Level Semantic Conflicts”, IEEE Transactions on Knowledge and Data Engineering, vol.16,2004,pp.189-202

[6] W. D. Li , Z. M. Qiu. “State-of-the-art technologies and methodologies for collaborative manufacturing development systems”, International Journal of Manufacturingion Research, Vol 44, 2006 , pp. 2525-2559

[7] Amy J.C. Trappey,Tung-Hung Lua,Li-Dien Fu. “Development of an intelligent agent system for collaborative mold manufacturingion with RFID technology” , Robotics and Computer-Integrated Manufacturing,vol 25, 2009, pp. 42-56