AGENT-BASED COLLABORATIVE DESIGN OF SHEET-METAL PARTS

AGENT-BASED COLLABORATIVE DESIGN OF SHEET-METAL PARTS

By

YUQING DING, B. Eng., M. Sc.

A Thesis
Submitted to the Graduate Studies
in Partial Fulfillment of the Requirements
for the Degree
Ph.D. of Engineering

McMaster University

PH.D. OF ENGINEERING (2008)	McMaster University
(Mechanical Engineering)	Hamilton, Ontario

TITLE: Agent-Based Collaborative Design of Sheet-Metal Parts

AUTHOR: Yuqing Ding, B. Eng. and M. Sc.
(Shanghai Jiao Tong University, Shanghai, China)

SUPERVISOR: Dr. T. J. Nye

NUMBER OF PAGES: xiv, 203

Abstract

The key objectives of this research were to develop an integrated design and analysis methodology for sheet-metal product development based on agent-based technology, feature-based design, optimization and finite element analysis techniques, and to study the performance of prototype systems developed based on such a methodology.

To achieve the research objectives, an agent-based framework was proposed for integrating and coordinating activities of participants involved in sheet-metal product development based on the investigation of the industrial requirements and the procedures of the development of sheet-metal products. Prototype systems were developed based on the proposed framework to answer research problems outlined for the design and implementation of agent-based systems, such as agent encapsulation, system architecture, agent communication and agent coordination. The performance of such prototype systems demonstrates that communication and coordination among domain agents can facilitate product development and reduce product cost.

An agent-based optimization approach based on an "A-Teams" approach (Talukdar et al, 1996) was proposed for process optimization in the tooling design stage to combine the utilization of the traditional optimization techniques used to solve sheet-metal forming problems and agent-based approaches. Three test cases were used of varying complexity from a rectangular cup to the NUMISHEET'99 automobile front door panel simulation benchmark for the determination of optimal drawbead restraining forces and blankholder forces when designing draw dies for stamped parts. A network of software agents, each implementing a different numerical optimization technique, was used in combination with metal forming simulation software to optimize process variables. It was found that the performance of each agent (and optimization technique) depended strongly on the complexity of the problem. For a given amount of computational effort, a network of collaborating agents using different optimization techniques always outperformed agents using a single technique in terms of both the best solution found and the variance of the collection of best solutions.

To provide guidance for the design and implementation of real applications, static and dynamic attributes and metrics of such agent-based collaborative systems, which can be evaluated in the preliminary system design stage and the system implementation stage, were proposed to study the impact of system architectures and coordination strategies on system performance. In addition, real-time system performance was statistically studied based on the data collected by the visualiser agent generated with the agent building toolkit. The results of case studies for system performance evaluation demonstrate the applicability of evaluation strategies proposed and can be used as a reference model for performance and scalability analysis on agent-based sheet-metal product development systems. The proposed evaluation strategies are applicable to general applications for product development by taking into consideration other performance indicators.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr. T. J. Nye, for his valuable guidance and consideration throughout the course of this thesis.

I would like to extend my gratitude to Dr. A. D. Spence, Dr. D. Metzger and Dr. W. F. Skip Poehlman for their valuable advice and illuminating comments on my research.

I would like to thank the faculty, staff and students in the Department of Mechanical Engineering for their help during my PhD studies.

And last but not least, I would like to thank my family for their support and understanding during my PhD studies.

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

1.1 Background

Sheet-metal parts are widely used in automotive, electronic, appliance and other industries. Of the manufacturing processes used to produce sheet-metal parts, stamping is perhaps the most common for large batch size production. In this thesis, the term "stamping" is used as a general term to describe all operations for making sheet-metal parts, such as cutting (blanking, piercing, lancing, etc.) and forming (drawing, bending, flanging, embossing, coining, etc.). Sheet-metal parts range from parts with few operations, such as blanking and piercing, to complex parts with a combination of many stamping operations.

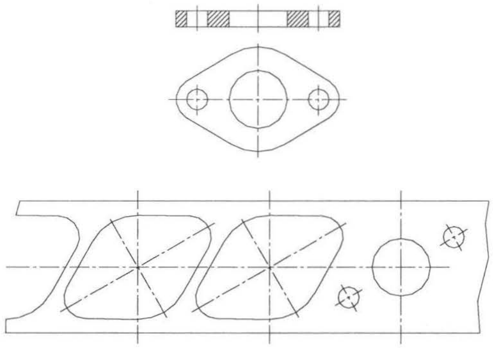


Figure 1-1. A typical sheet-metal part and its strip layout

Figure 1-1 shows a typical stamped part and the strip layout to produce this part with a progressive die. Operations to produce the stamped part comprise three piercing operations to pierce the three round roles, and one blanking operation to cut the perimeter of the part from the strip of metal.

The specific type of die selected depends on the expected production quantities of the part. When the production quantity is large enough, progressive dies are usually needed. These dies contain a series of stations, each of which performs one or more stamping operations on the strip. In Figure 1-1, the strip enters a progressive die from the right. The three round roles are pierced at the first station. Between strokes of the press the strip moves one station to the left where the part is blanked from the strip. Because all operations are built into one large die, the cost of the die is relatively high but is offset by savings in eliminating material handling costs between operations.

Simple dies can be applied in the cases where the production volumes are small and the requirement of quality is low. Each stamping operation is assigned to a separate die that is small and has little complexity, making it relatively inexpensive. Each part, however, must be separately moved from one die to another, leading to significant handling costs. Compound dies are needed if the production quantity is small but the quality requirement is high.

As shown in Figure 1-2, procedures of sheet-metal product development include part design, die design, die manufacturing and stamping production. Traditionally, in the preliminary part design stage, designers determine the shape, dimensions, tolerances and other technical requirements based on customer requirements (i.e. batch size, product functions, assembly requirements, etc.) and their knowledge on manufacturability evaluation and cost analysis. Then, the initial part design will be sent to the die design and manufacturing department. Usually, die designers need to work with part designers and/or customers to determine the nested strip layout because the material usage is one of the most important factors in sheet-metal product cost. Stamping dies are designed according to the technical requirements of parts and information about the stamping equipment. In most cases, iterations of part design are unavoidable due to feedback from the downstream activities, such as process planning, die design and manufacturing and stamping production. For example, Figure 1-3 shows how the product design may affect the strip size. A minor difference in material utilization may have a great impact on the cost of products because even small inefficiencies in material utilization per part can lead to very large amounts of wasted material over a die's life due to the high volumes of metal stampings produced with a given die. The amount of the scrap generated in the blanking operation should be held to a minimum without affecting the manufacturability of the stampings.

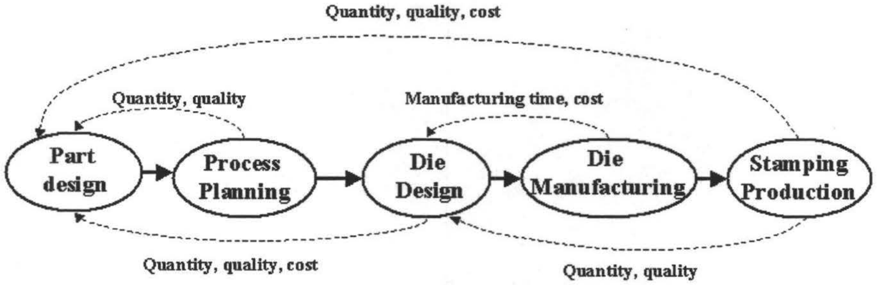


Figure 1-2. Flowchart of sheet-metal product development (Chin and Tang, 2002)

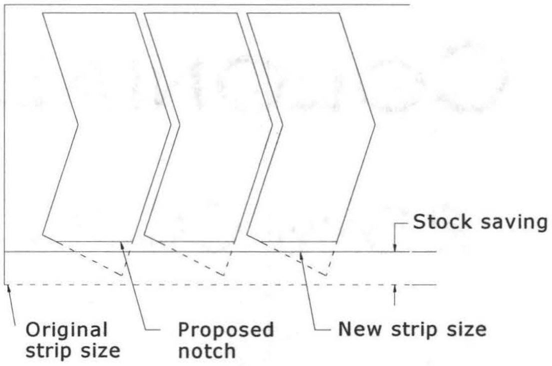


Figure 1-3. Strip layout with unnecessary corners cut off to save material (Wilson et al., 1965)

With the strip layout and technical data of the stamping equipment, die designers can determine the die configuration as either simple, compound or progressive dies according to the needs of the application. Traditionally, the die design process is performed by trial-and-error methods and is experience-dependent and very time-consuming. Numerical simulations have played a very important role in formability analysis for process and tooling design of sheet-metal forming since the late 1970's. Those numerical simulations focus on determination of blank geometry, material selection, press selection, tool configuration and forming conditions. Feedback from the formability analysis usually leads to re-designs of part and/or tooling.

1.2 Research Motivation

Feedback, re-design, and communications among participants in sheet-metal product development may not cause many problems if all participants are working at the same office of a company. However, nowadays, these functions are usually performed in different departments of a company, or across different companies, which may be in one or more countries. Those unavoidable iterations may lead to high cost and long time-to-market (Shen and Wang, 2003). Moreover, information feedback from low-level (e.g. process planning, die design and manufacturing, stamping production) to the high level design (e.g. part design) is usually performed by human interactions. Practical difficulties associated with communication among all the relevant individuals may cause an insufficient product development due to the absence of formability analysis and cost analysis at the preliminary design stage. To solve those problems, life-cycle activities, such as part design, forming process planning, tooling design and manufacturing, have to be considered by the part designer in the preliminary product development stage. The concept of collaborative design has been introduced to solve this problem (Xie et al., 2001; Chin and Tang, 2002).

With the advances of information technology, Internet and World Wide Web (WWW) technologies have been widely used in the field of collaborative design. Web-based design systems for sheet-metal parts have been explored over the past few years. While the Internet facilitates faster communication between different design function areas in the stamped part production process, development time and quality is still limited by the ability of human designers to process information and respond to others involved in the project. This delay is caused by the designers' fundamental speed of performing their tasks, by their workload on other projects, and by time zone and/or language issues when the different participants are widely separated geographically. The requirement that projects should be completed at a given date constrains the number of design iterations, which, in turn, limits the quality of the overall product development and production process.

Usually, Web-based collaborative design systems focus on the sharing of design information, data and knowledge among distributed design modules. Participants of the life cycle design activities contribute and get information via a Web server. However, participants do not communicate and coordinate directly with each other.

The distributed sheet-metal product development environment includes a variety of hardware and software tools, such as different hardware, different operating systems, different network protocols and architectures, different development tools in specific domains, different databases and multidisciplinary knowledge. An integrated design and analysis methodology is needed to encapsulate those existing legacy functions.

As an alternative approach for collaborative design systems, agent-based technology has been used to develop such systems. Besides providing the sharing of design information, data and knowledge among distributed design modules, similar to what Web-based collaborative design systems do, agent-based collaborative design systems can provide better performance of communication and coordination (Shen and Wang, 2003).

Parunak (1998) summarized that "agent technology is appropriate for applications that are modular, decentralized, changeable, ill-structured and complex". Life-cycle activities in sheet-metal product development are usually implemented by different industrial entities with different expertise. It is very suitable to use agents to represent industrial entities. Several companies, located in different countries in some cases, may be involved in the development process of a sheet-metal part. In other words, the sheet-metal product development process is usually organized in a decentralized fashion. A design system of sheet-metal parts has to be changeable and has to have the ability to solve complex problems. Therefore, an agent-based design environment is suitable to sheet-metal product development.

To sum up, considering the sheet-metal industry's requirements, a collaborative product development environment is needed to reduce lead times and costs, and to increase product performance and accuracy of sheet-metal products. The advances of agent-based technology, optimization and finite element analysis of sheet-metal forming make it possible to develop such kind of integrated design environment. However, there are a number of unknowns about the efficiency and reliability of such an integrated design and analysis method based on agent technology and how software agents can support different stages in the life cycle activities of sheet-metal product development. This research was motivated by these problems and the objectives of this research are discussed in the next section.

1.3 Objective and Scope

The objectives of this research are as follows:

Objective 1: To develop an integrated design and analysis methodology for sheet-metal parts through the utilization of recent advances in agent-based technology, feature-based design, optimization and finite element simulations of sheet-metal forming.

Objective 2: To study and gain insights into the performance and behavior of the collaborative design system for sheet-metal parts based on the integrated design and analysis methodology.

Computerized tools based on the integrated design and analysis methodology should provide multiple functions, like early estimation of manufacturability of parts in the preliminary part design stage, optimal process design and cost analysis.

To achieve these research objectives, the following questions have to be answered in this research.

Question 1: What should be specified as agents in the sheet-metal product development system? In other words, what responsibilities fulfilled by participants in the sheet-metal product development process should be assigned to specific agents?

Question 2: What kind of architecture should the agent-based system adopt for sheet-metal product development? How are agents structured internally? How do agents wrap existing functionalities provided by legacy systems or databases?

Question 3: How should agents communicate? What communication protocols should agents use?

Question 4: How should agents coordinate their actions?

Question 5: What design optimization methods are suited for agent-based sheet-metal product development? How do agents collaborate to solve optimization problems?

Question 6: How should the performance of the agent-based sheet-metal product development systems be evaluated?

To answer the above questions, a prototype system for agent-based collaborative design of sheet-metal parts has been developed. The goal of collaborative sheet-metal design is to get near optimal designs with low cost, short lead-time and high quality. To achieve this goal, each domain agent has to contain precise data and knowledge, such as manufacturing costs, die design rules, stamping production costs, shipping costs, etc. How the agent-based integrated design and analysis method can improve the efficiency and accuracy of a specific design optimization task implementation is studied in this research. Several case studies are used to study how software agents can coordinate on specific design optimization tasks to gain insights into the performance and behavior of the collaborative design methodology.

Taking into consideration the time limitation and complexity of sheet-metal product development, limitations of this research are:

- This research is not to develop an agent-based collaborative design system for all kinds of sheet-metal parts. The prototype agent-based design system only handles two kinds of sheet-metal parts, simple deep drawn parts and flat parts with or without holes. The objective of generating the prototype system is to demonstrate how the agent approach can facilitate the sheet-metal product development process and study the performance of the agent-based systems. The flat parts (with or without holes) are used for case studies of construction and demonstration of agent-based collaborative design systems and agent-based distributed sourcing systems for sheet-metal parts. The deep drawn parts are used for studying agent-based process optimization problems.

- The focus of this research is on specific manufacturing process optimization problems rather than the overall design optimization problems. Three optimization algorithms, namely, downhill simplex method, simulated annealing and genetic algorithm, are studied.
- Cost analysis agents and die manufacturing agents only provide rough information based on part features and process planning information in the prototype system. Domain agents are only developed to satisfy the requirements of case studies.

1.4 Thesis Outline

A review of the literature relative to this research is presented in Chapter 2. Previous studies on concurrent design of stampings and dies, formability analysis of sheet-metal forming, optimization of sheet-metal forming process design and agent-based collaborative design are presented.

Construction of an agent-based collaborative design system for sheet-metal parts is introduced in Chapter 3, in which the modularization of sheet-metal product development activities is introduced and a prototype system is developed to demonstrate the benefits of agent communication and coordination.

Chapter 4 presents a study of agent-based process design in sheet-metal forming. Case studies on agent-based optimization of blank-holder force and drawbead restraining forces in sheet-metal forming process are presented.

In Chapter 5, approaches for performance evaluation of agent-based collaborative design systems for sheet-metal parts are discussed.

Overall results of this research are discussed and summarized in Chapter 6, some conclusions are drawn and suggestions for future studies are made.

Chapter 2. Literature Review

As an important manufacturing process, stamping has been a popular research topic for years, leading to the sophisticated tools available today for computer aided design of sheet-metal parts, simulation of sheet-metal forming, strip layout nesting, die design and manufacturing. Research in each of the areas will be examined in the following sections.

2.1 Concurrent Design of Stampings and/or Dies

With the advances of computer graphics and CAD/CAM/CAE technology, some commercial systems and research projects have been developed to solve design and/or manufacturing problems for specific parts. In other words, these systems are restricted to die design/manufacturing or sheet-metal part modeling. Some industrial companies have developed in-house systems for tooling design and manufacturing (Cheok and Nee, 1998). Since the late 1980's, feature- and knowledge-base technologies have been used to develop systems to aid designers for part design and/or die design. Those systems have played a very import role in die design since the 1980's. A review of commercial CAD/CAM systems for die design and manufacturing was given by Cheok and Nee (1998).

2.1.1 Standalone CAD/CAM Systems

Over the past several years, the concept of concurrent engineering has been proposed to overcome the inconsistency of the traditional sequential design method. In the Concurrent Engineering approach, product design, manufacturing, and other life cycle issues are considered throughout all phrases of the development cycle.

Table 2-1 provides a summary of several typical standalone systems for concurrent design of stampings and/or dies. These research projects demonstrate the importance of considering all life-cycle activities at the preliminary design stage. Feature-base design methodology and knowledge-based technology have been widely used in sheet-metal part and die design. General CAD/CAM systems, such as AutoCAD, Pro/E, are selected as platforms for fast development of specific systems for sheet-metal parts (Cheok and Nee, 1998; Mantripragada et al., 1996).

These standalone systems may allow users to input specific design rules for customization. The intents of these systems were to develop a CAD or integrated CAD/CAM system to ease the tedious design work, help users make reasonable decisions, and eliminate some design mistakes by using feature-, knowledge-, case-, and/or constraint-based techniques. These systems were designed to be used by part or die designers separately and were not focused on the knowledge sharing and coordination among participants involved in the sheet-metal product development process.

2.1.2 Web-Based Systems for Stamped Part and Die Design

Recently, there has been increasing use of Internet and World Wide Web technologies in the field of stamping design and die development. Table 2-2 presents the Web-based collaborative design systems for stamping part design and/or die design.

Table 2-1. Standalone concurrent design systems for stampings and/or dies

Project	**Researchers**	**Main characteristics**
?	Poli et al. (1993)	A coding and classification system to analyze the sheet part attributes that impact die construction costs
SMART	Lazaro et al. (1993)	An intelligent design-for-manufacturing system consisting of a feature-based design system and a knowledge-based system to assist designers
?	Lee et al. (1995)	An assessment system consisting of a knowledge-based geometric-analysis module, a finite-element method module and a formability analysis module
?	Mantripragada et al. (1996)	A computer-aided engineering system for the feature-based design concept in developing an interactive design tool that can be used to support the designers to tackle potential stamping problems, defects and failure
IPD	Cheok and Nee (1998)	An intelligent knowledge-based design environment for the design of progressive dies
?	Brinbaum et al. (1999)	An integrated case-based design support system
HPRODIE	Li et al. (2001)	An integrated CAD/CAM system for progressive dies. The system includes 3 features: product feature for the representation of sheet-metal parts, stamping process feature for the strip layout, and the die design feature for building the die assembly model. With feature modeling and feature mapping, most of design processes can be carried out automatically.
?	Tang et al. (2001)	An integrated feature-based design for stamping system
?	Tang and Gao (2007a, 2007b)	A feature- and knowledge- based design system for stampability evaluation and process planning

Table 2-2. Summary of Web-based systems for stamped part design and/or die design

Researchers	**Main characteristics**
Hindman & Ousterhout (1998)	A virtual design system for sheet-metal forming that combined an expert system with manufacturing process simulations and real-time manufacturing
Xie et al. (2001)	A WWW-based integrated product development platform for sheet-metal parts design
Ling et al. (2001)	A Web-based e-Metal Forming system that is targeted at bringing the use of FEM to non-specialist users to study sheet-metal bending.
Chin & Tang (2002)	A Web-based architecture for concurrent stamped part and die development system is presented. The focus of this research is on feature-based design, cost analysis, knowledge-based design of stamped parts and dies.

Usually a client/server model is adopted to build Web-based collaborative design systems. For example, the Web-based architecture proposed by Chin and Tang (2002), as shown in Figure 2-1, is a typical client/server application. In this architecture, clients, such as part designers, die designers and customers, use WWW (World Wide Web) browsers to access design data and rules. The browsers provide user interfaces for clients to create requests, send these requests to the Web server and return the results from the Web server to clients. The Web server that contains design and analysis services plays a very important role in collaborative design. Customers, part designers and die designers distributed among departments, companies or countries can share design information and knowledge and communicate in a concurrent and distributed manner on the Internet.

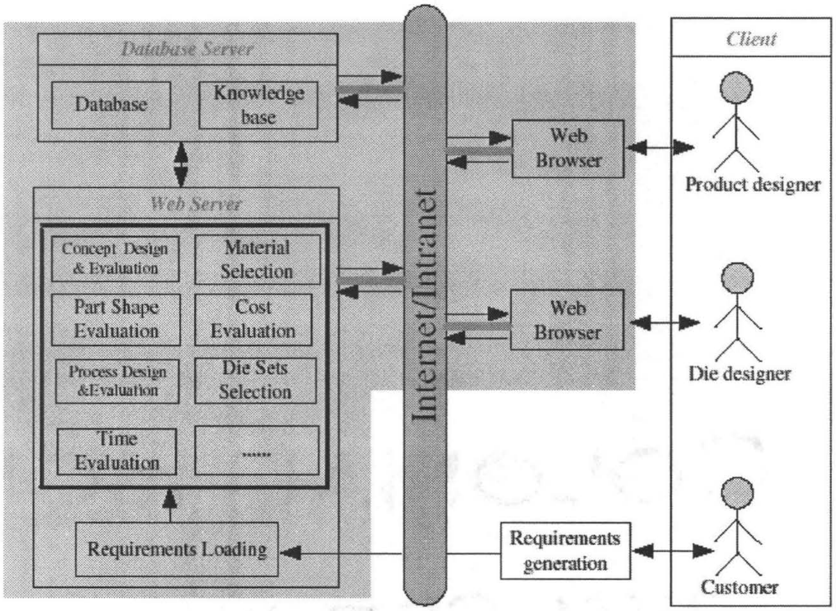


Figure 2-1. Web-based concurrent stamping part and die design
(Chin and Tang, 2002)

These projects focused on developing integrated design systems for information and knowledge sharing based on Internet and Web technologies. Existing technologies presented in these projects do not guarantee enough collaboration among participants, such as product designers, manufacturing engineers within such systems. Agent-based technologies have been proposed for the development of collaborative systems that can support more effective communication and coordination among participants. Agent-based technologies and applications are reviewed in the following sections.

2.2 Agent-Based Technologies and Applications

2.2.1 Definition of Agents and Multi-Agent Systems

There are many definitions for agents. The definition presented by Wooldridge and Jennings (1995) has been widely accepted in academia. According to them, the term "agent" is used is to denote a hardware or software-based computer system that has the following properties:

- **Autonomy**. Agents can work autonomously without assistance from an outside source. They have their own specific problem solving skills and can work on specific tasks separately.
- **Social ability**. Each agent is not alone and can communicate and coordinate with other agents and humans.
- **Reactivity.** Agents can respond in a timely fashion to perceived changes in their environment.
- **Pro-activeness.** Agents do not simply act in response to perceived changes in their environment. They can take initiative in communication and coordination.

Multi-agent systems (MASs) are known as networks of agents that work together to solve problems that are beyond the individual capabilities or knowledge of each agent (Durfee et al., 1989). In the multi-agent environments, agents can be autonomous and distributed and may be self-interested or cooperative (Weiss, 2000). All agents may interact directly with each other, communicate and exchange information to solve problems. Advantages of multi-agent Systems compared with a single agent can be summarized as follows:

- Multi-agent systems can be more effective in solving large and complex problems that are too difficult for a single agent to tackle because of resources limitations. For example, when a system is dealing with enormous quantities of data or when the search space is very large, or in time constrained situations, distributed computation with multiple agents can be more efficient than solving the problem with a single agent (Zhang, 1999).
- Multi-agent systems are preferable when legacy systems need to be incorporated into a wider cooperating agent community. For example, existing CAD/CAM/CAE systems and data and knowledge bases usually need to be incorporated into a collaborative design system for specific products. Since those systems are not open to any user and some knowledge and data are supposed to be kept locally, a single agent is impossible or inefficient to wrap everything.
- Compared with a single agent, multi-agent systems can enhance performance dimensions of computational efficiency, reliability, extensibility, robustness, flexibility and reuse (Zhang, 1999).

2.2.2 Software Agents vs. Direct Manipulation Interfaces

Although the benefits of agent-based approaches are obvious, the debate over the benefits of intelligent software agents versus direct manipulation interfaces has never stopped. Direct manipulation permits novice users access to powerful facilities without the burden of learning to use a complex syntax and lengthy list of commands (Shneiderman, 1981).

A famous debate was held at CHI 1997 (Conference on Human Factors in Computing Systems in 1997). The proponents of direct manipulation believed that direct manipulation interfaces can provide users with a sense of responsibility for and control of the objects represented by the interface and dramatic opportunities exist to develop direct manipulation programming to create end-user programming tools. Agent promoters argued that the high complexity of the modern computing environment and increasing numbers of "naive" users unaccustomed to dealing with this complexity require offloading of responsibility from users to intelligence in the interface. They believed direct manipulation would have to give way to some form of delegation. Direct manipulation proponents were and may be still concerned about two typical issues concerning agent technology, trust and competence. Concerning trust, users of the agent-based system must believe that agents can do what they are required to do. As for competence, agents must have the capability to accomplish the tasks assigned to them. In some cases, these two issues may be critical. But at least, agent approaches can be a very good supplement to direct manipulation interfaces. Good human-computer interfaces can ease the complexity of problem solving and agent approaches can play an important role in conducting necessary communication and coordination within systems in a distributed manner (Shneiderman, 1981).

2.2.3 Agent-Based Collaborative Design and Manufacturing

Agent-based technology has been widely used in concurrent engineering, manufacturing enterprise integration, supply chain management, manufacturing scheduling and control.

Table 2-3 provides a summary of projects or systems on agent-based collaborative design and manufacturing.

These projects (as listed in Table 2-3) focused on applying different internal agent architectures, organizational architecture, communication languages, agent knowledge representation and sharing, coordination protocols and strategies, wrapping technologies for encapsulating legacy systems, agent learning techniques and other necessary development tools to solve specific domain problems. These researches do not provide systematic approaches for domain analysis and performance evaluation.

Performance of those projects has demonstrated that agent-based technologies can integrate activities of participants, facilitate the communication and coordination among participants for design and manufacturing, and therefore, reduce the lead-time and cost of products. Taking into consideration characteristics of sheet-metal product development, a systematic study on the development of agent-based collaborative systems for sheet-metal product development is required.

Important issues for developing multi-agent systems, such as agent communication languages and organizational architectures, are discussed in the following sections and other issues about coordination protocols and strategies and domain ontology representation are discussed in Chapter 3.

Table 2-3. Summary of projects/systems on agent-based collaborative design

Project	**Researchers**	**Main characteristics**
PACT	Cutkosky et al. (1993)	Engineering tools and frameworks are encapsulated by agents. All interactions between agents are mediated by facilitators.
SHARE	Toye et al. (1993)	An infrastructure for design information sharing. Service agents interacting as peers over the Internet. Messages are sent using standard e-mail and TCP/IP transport services.
SiFAs	Brown et al. (1995)	SiFAs provide elementary functionality specific to design tasks. Agents are small, and restricted to a limited number of types.
ICM	Fruchter et al. (1996)	ICM provides a graphic environment as the central interface to reasoning tools, to support design. A communication cycle for collaborative teamwork is implemented.
Madefast	Cutkosky et al. (1996)	Madefast uses the WWW as a corporate memory, sharing design information across the design team, and preserving it for downstream tasks such as maintenance and redesign.
DIDE	Shen et al. (1995)	ToolTalk is used for local communication and an e-mail tool, ELM, is used for remote communication.
MetaMorph I	Maturana and Norrie (1996)	Agents link with other agents through mediators, which assume the role of system coordinators by promoting cooperation among agents and learning from the agents' behavior.
MetaMorph II	Shen et al. (1998)	A hybrid agent-based architecture was proposed to integrate enterprise activities, such as design, planning, scheduling, simulation, execution, etc., with participants within a distributed intelligent open environment.
RAPPID	Parunak et al. (1999)	A project to develop agent-based software tools and methods to coordinate set-based design of discrete manufactured products.

Table Continued On Next Page

Project	Researchers	Main characteristics
?	Hao et al. (2006)	A software prototype collaborative design system was implemented on a FIPA-compliant agent platform for mechanical product design engineering by applying intelligent software agents, Internet/Web, workflow, and database technologies.
A-Design	Campbell et al. (1999)	The A-Design approach is based on artificial intelligence, genetic algorithms, stochastic optimization, multi-objective optimization, qualitative physics, and asynchronous teams.

2.2.4 Agent Communication Languages (ACLs)

Two ACLs, KQML and FIPA ACL, are widely used in agent-based applications in the literature.

KQML (Knowledge Query and Manipulation Language), born from DARPA (Defense Advanced Research Projects Agency)'s Knowledge Sharing Efforts, is both a message format and a message-handling protocol to support run-time knowledge sharing among agents (KQML, 2007). FIPA (Foundation for Intelligent Physical Agents) is a non-profit organization aimed at producing standards for the interoperation of heterogeneous software agents (FIPA, 2005). FIPA has played a crucial role in the development of standards for agent technologies and has promoted a number of initiatives for the development and uptake of agent-based technologies. FIPA ACL is the agent communication language associated with FIPA's open agent architecture.

Annex C of FIPA 1997 (FIPA, 2005) summarizes the primary similarities and differences between FIPA ACL and KQML:

- Both KQML and ACL are based on speech act theory.

- KQML sets out to be simple to parse and generate, but still human readable. FIPA ACL adopts a very similar syntax. Some differences exist in the names of both the message type keywords and the parameter keywords.
- KQML was originally designed to fulfill a very pragmatic purpose and the semantics of the performatives were initially described informally by natural language descriptions. Subsequent research has addressed the need for a more precise semantics.
- KQML aims to serve several needs in inter-agent communication and attempts to define a core set of performatives to meet all of these needs. ACL does not attempt to cover all of these needs within the language.

KQML does not prescribe the language of message content and ontology on which the message content is based. KQML was originally developed for knowledge sharing rather than agent communication. However, FIPA ACL was initially developed for agent communication. Since FIPA standards have been widely accepted by researchers and developers, FIPA ACL has been a natural choice for agent communication in this research.

2.2.5 Agent Architectures for Collaborative Design and Manufacturing

According to Shen et al. (1999), system architectures proposed in the literature for agent-based manufacturing systems can be classified into three categories: autonomous architectures, federated architectures, and hierarchical agent architectures.

The autonomous agent system architecture shown in Figure 2-2 (a) has no global control. All agents in such systems are autonomous and can communicate directly with any other agents in the system. Autonomous agents should have knowledge about other agents. Each agent must know the addresses, capabilities of other agents. Therefore, the autonomous agent architecture is suitable to model systems with a small number of agents.

Federated system architectures are suitable to develop open, scalable multi-agent system architectures (Genesereth and Ketchpel, 1994). Applications of federated architectures can be found in (Cutkosky et al. 1993, Park et al 1994, Maturana 1999, Wiederhold 1992, Gaines et al. 1995, Shen et al. 1998).

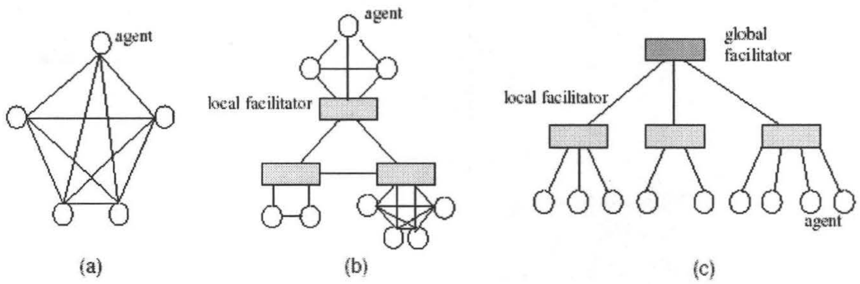


Figure 2-2. System architectures: (a) autonomous or peer-to-peer; (b) federated; (c) hierachical or tree (Lee and Hwang, 2004)

Three approaches have been proposed for federated architectures: the facilitator approach, the broker approach and the mediator approach (Shen and Norrie, 1999).

The facilitator approach was proposed to ease the communication and coordination among agents. With a facilitator, agents do not have to have direct knowledge of other agents for collaboration. Figure 2-2 (b) shows a sample of federated architecture with groups of agents organized by local facilitators. The disadvantage of this kind of federated architecture is that facilitators may become the bottleneck of the system if the communication among facilitators is too busy.

The functional difference between a facilitator and a broker is that a facilitator is responsible for a designated group of agents, but a broker might be asked by any agents in the system for locating the service providers and completing the tasks (Shen and Norrie, 1999).

Mediator agents proposed by Gaines et al. (1995), Maturana and Norrie (1996) and Shen et al. (1998) play the roles of system coordinators by providing the message services and promoting cooperation among intelligent agents and learning from the agents' behavior in addition to the functions that can be provided by a facilitator and a broker.

The hierarchical architecture, as shown in Figure 2-2 (c), is shaped like a tree structure. Local facilitator controls agents and root facilitator controls local facilitators (Lee and Hwang, 2004). The hierarchical architecture is exactly the way that most real enterprises are organized and it is a nature way to model practical agent-based industrial applications. The hierarchical architecture is not suitable for complex and dynamic systems because there may be no, or not enough, information about each new participant agent and relationships between the new agent and existing agents.

The MetaMorph II project (Shen et al. 1998) based on the mediator approach is a good sample for developing large, complex and dynamic distributed manufacturing systems. Strictly speaking, the system architecture of MetaMorph II is not a pure federated architecture, but a hybrid agent-based architecture with a combination of the autonomous agent approach, the mediator-based federated approach, and a static or dynamic hierarchy imposed for specific tasks. In the hybrid architecture, the system is primarily organized at the highest level through "subsystem" mediators. Each subsystem is connected to the system through a special mediator. Each manufacturing enterprise has at least one enterprise mediator that encapsulates functionality to allow local coordination and interaction with other dissimilar mediator agents. The hybrid agent system architecture combined the advantages of both the federated architecture and the autonomous architecture.

Taking into consideration the complexity and dynamics of sheet-metal product development, a hybrid system architecture, which combines federated, hierarchical and autonomous system architectures, is suitable for developing prototype or real systems for sheet-metal product development in this research. Details will be discussed in Chapter 5.

2.2.6 Agent Building Toolkits

As mentioned in the previous sections, agent-based technologies are appropriate for many application areas. The development of agent-based systems has been a popular research topic. Software developers presented many agent models and system architectures in the literature. However, implementing an agent-based system from scratch requires a lot of time and expertise. To facilitate the development of agent-based systems, some agent building toolkits have appeared on the market.

According to Serenko and Detlor (2002), an agent building toolkit is defined as any software package, application or development environment equipped with a sufficient level of abstraction to implement software agents with desired attributes, features and rules. Agent building toolkits provide a certain level of abstraction in which programmers can develop their objects. Some toolkits may offer only a platform for agent development, whereas others may provide features for visual programming. Agent toolkits may also provide an environment for running, monitoring, analyzing and testing agents. Serenko and Detlor (2002) presented a summary of assessment of some popular agent building toolkits in terms of four categories, mobile agent toolkits, multi-agent toolkits, general-purpose toolkits and Internet agent toolkits.

JATLite (Java Agent Template Lite) (JATLite, 2005), ZEUS (Collis and Ndumu, 1999) and JADE (Java Agent Development Framework) (JADE, 2007) are three popular multi-agent toolkits widely used in concurrent design, intelligent manufacturing and supply chain management.

JATLite allows users to build agents that communicate robustly over the Internet. JATLite provides an infrastructure, in which agents register with an Agent Message Router Facilitator using a name and password, connect/disconnect from the Internet, send and receive messages, transfer files with FTP, and generally exchange information with other agents on the various computers where they are running. JATLite facilitates especially construction of agents that send and receive messages using the emerging standard communication language KQML. The communications are built on open

Internet standards, TCP/IP, SMTP and FTP. However, developers may easily build agent systems using other agent languages, such as the FIPA ACL using the JATLite layers.

JADE is a software framework fully implemented in the Java language. It simplifies the implementation of multi-agent systems through a middleware that complies with the FIPA specifications and through a set of graphical tools that support the debugging and deployment phases. The agent platform can be distributed across machines (which do not even need to share the same operating system) and the configuration can be controlled via a remote GUI. The configuration can even be changed at run-time by moving agents from one machine to another, as and when required.

ZEUS (Collis and Ndumu, 1999) toolkit offers a library of software components and tools that facilitate fast and friendly design, development and deployment of multi-agents, and, as one of the most important features, ZEUS is FIPA compliant. All the ZEUS components are written in Java. ZEUS agents communicate using messages that obey the FIPA ACL specification. Communication between ZEUS agents is via point-to-point TCP/IP sockets, with each message communicated as a sequence of ASCII characters (Collis and Ndumu, 1999). Together, the components of the Agent Component Library enable the construction of an application-independent generic ZEUS agent that can be customized for specific applications by imbuing it with problem-specific resources, competencies, information, organizational relationships and co-ordination protocols. Further details about ZEUS agent components, agent architecture, communication mechanism, ontology definition and knowledge representation can be found in Appendix A.

ZEUS (version 1.2.1) was selected as the agent development toolkit in this research because:

- ZEUS toolkit is suitable for the development of collaborating agent systems (Collis and Ndumu, 1999).
- ZEUS is FIPA compliant. FIPA standards have been widely accepted by agent researchers and developers.
- Communication between ZEUS agents is via point-to-point TCP/IP sockets. TCP/IP is still arguably the single most important network protocol in use today. Nowadays, participants in the supply chain of sheet-metal product development are medium- and small-size enterprises. TCP/IP is suitable for this application.
- ZEUS provides user-friendly graphical interfaces that can facilitate the agent creation and deployment. ZEUS can configure a number of different agents of varying functionality and behavior, organize the agents in varying organizational relationships, and equip agents with different coordination mechanisms.
- ZEUS can automatically generate the executables for the agents based on the user input via the graphical user interfaces. Although, further programming is needed for most applications, ZEUS can save a lot of time for Java beginners.
- ZEUS agents can easily wrap legacy systems. Many CAD/CAM/CAE systems, databases and knowledge bases are used by participants in the supply chain. With ZEUS, it is easy to wrap those legacy systems.

- With the ZEUS toolkit, three kinds of utility agents, namely, name server agents, facilitator agents and visualiser agents, can be easily constructed to coordinate and analyze agent activities. Visualiser agents can be used to view, analyze or debug societies of ZEUS agents.

2.3 Formability Analysis of Sheet-Metal Forming

The design of a stamping die has a strong influence on whether the part will be formed successfully, or will suffer from defects during forming. Since the manufacturing cost of dies is high, it is necessary to evaluate the formability of process design carefully in the part design and process planning stage. An incorrectly designed die can be very expensive to rework. Nowadays, finite element simulations are widely used in metal forming industry and the forming limit diagram (FLD) plays a very important role in formability analysis. The utilization of numerical simulations can be a great help to both part designers and die designers. In the part design stage, designers can avoid part design schemes that may cause bad formability and material wastage.

Two papers (Kobayashi et al. 1978 and Wang et al. 1978) signaled the beginning of the research into finite element simulations of sheet-metal forming processes. Since then, finite element simulation of sheet-metal forming processes has been a rapidly developing research subject because of the importance of metal forming to the economy of industrialized countries. A good literature review of simulation of sheet-metal forming was given by Tekkaya (2000).

The simulation solutions for sheet-metal forming range from implicit, explicit and inverse approaches.

Static implicit methods are the traditional methods used in finite element simulations of metal forming processes. The advantage of static implicit methods is that they can perform the simulation of a sheet-metal forming process without ignoring the fact of its quasi-static nature.

Ignoring the quasi-static nature of most sheet forming processes, dynamic explicit approaches have a lower memory requirement compared to static implicit methods.

Research on inverse methods originated in the late 1980's and developed rapidly in the 1990's. An inverse approach used for estimating the large strains of thin sheet obtained by deep drawing was briefly introduced by Batoz et al. (1995). Mouatassim et al. (1995) introduced an industrial one-step finite element code for sheet-metal forming. Most inverse finite element methods are focused on the forming problems of thin sheets. For sheet-metal forming, the aim of the research into inverse methods was to develop accurate FEM solvers to check quickly the feasibility of stampings and the effectiveness of a tool design solution. Inverse approaches enable users to find the position of the nodes of the blank in its original, horizontal plane, ensuring the equilibrium of the stamped blank under tool forces and restraining actions so that the inverse FEM can predict an initial blank shape from a final deformed shape in a one-step calculation.

The inverse approach can provide information about formability and estimate the blank at the preliminary stage of the design process, even without detailed tooling information. On the other hand, the accuracy of the results in certain cases may be not satisfactory. However, the computational efficiency still makes inverse approaches attractive, especially in this research, an inverse approach can be applied for early estimation of material usage and formability analysis so that the cost and lead-time of product development can be lowered. The computational efficiency of inverse approaches can make it efficient and attractive to use a combination approach with inverse FEM tools and optimization algorithms for manufacturing process optimization in sheet-metal forming.

2.4 Optimal Process Design of Sheet-Metal Forming

Over the past few years, some researchers have combined optimization algorithms with finite element simulations to optimize the process design of sheet-metal forming. The process optimization of sheet-metal forming is a nonlinear constrained optimization problem and usually its objective function and the constraints cannot be described explicitly. Optimizing the design plan in sheet-metal forming is very challenging.

Table 2.4 provides a summary of research on simulation-based optimization in metal forming process design. Mackerle (2004) presents a summary of the research on the application of numerical simulations in sheet-metal forming. The research work in the literature (shown in Table 2.4) has shown that the introduction of optimization in the product development process would result in significant development time reduction and cost savings. Guo et al. (2000) and Naceur et al. (2004) combined an inverse approach with an optimization algorithm to optimize drawbead restraining forces. Explicit or implicit FEM tools were used in other research. Different objective functions, such as geometry constraints, thickness distribution, material usage, etc., were applied in these projects. The research work also mentioned the limitations of optimization algorithms used. For example, Ohata et al. (1996; 1998)'s approaches are only applicable to cases with limited design variables.

Table 2-4. Summary of research on optimization of stamping process design

Researchers	**Optimization Method**	**Main characteristics**
Ohata et al. (1996)	Sweeping Simplex Method	"Sweeping Simplex Method" was introduced into the optimum forming design system in cooperation with FEA code for deep drawing process design with two design variables. Objective function: minimization of the deviation of thickness from uniform average thickness
Ohata et al. (1998)	Improved Sweeping Simplex Method	Improved Sweeping Simplex Method with an operation of expansion was applied to optimum deep drawing design with three design variables.
Liu et al. (2002)	Improved Simplex Method	An improved hybrid optimization algorithm and an improved drawbead restraining force model are presented for the optimization of drawbead design of autobody cover panel. Objective function: minimization of the forming redundancy of the joined panel.
Guo et al. (2000)	SQP (Sequential Quadratic Programming) Method	An inverse approach was combined with an SQP method to optimize the blank shape and drawbead restraining forces in deep drawing. Objective function: minimization of the maximum of thickness variations.
Naceur et al. (2004)	BFGS (Broyden-Fletcher-Goldfarb-Shanno) Algorithm	An inverse approach, a BFGS algorithm and analytical sensitivity analysis were combined to optimize material parameters and restraining forces for deep drawing. Objective function was defined based on the forming limit curve. Design variables include drawing restraining forces and material parameters.

Table Continued On Next Page

Researchers	Optimization Method	Main characteristics
Roy et al. (1997)	Micro Genetic Algorithm	Application: multi-pass cold wire drawing, multi-pass cold drawing of a tubular profile and cold forging of an automotive outer race preform. Objective function for design optimization of multi-pass cold wire drawing: (i) the difference between maximum and minimum effective plastic strains; (ii) the total deformation energy.
Chung and Hwang (1999)	Micro Genetic Algorithm	Application: Optimal design of the die shape in extrusion. Objective function: minimization of: (i) punch load; (ii) effective strain variations; (iii) peak die pressure. Design variables are die shape parameters.
Kok and Stander (1999)	Response Surface Methodology	Application: Optimally design of the preforming process of a wheel centre pressing. The die shape is based on a cubic spline interpolation. The objective is to minimize the blank weight, subject to minimum thickness constraints and assembly constraints.
Huh and Kim (2001)	Response Surface Methodology	Application: design of the draw-bead force and the die shapes in deep drawing processes. A rigid-plastic finite element method was employed for the calculation of the final shape and the strain distribution.
Thiyagarajan and Grandhi (2005)	Response Surface Methodology	Application: a 3-D preform shape optimization problem for the forging process. A multi-level design process was developed to find suitable basis shapes or trial shapes at each level that could be used in the reduced basis technique. Each level was treated as a separate optimization problem until the objective (minimum strain variance and complete die fill) was achieved.

To develop an integrated design and analysis environment for sheet-metal product development, limitations of each optimization algorithm that may be used should be studied and only suitable optimization algorithm(s) should be used for achieving good optimization results. However, it is difficult to find which optimization approach is the best one due to the complexity of most real applications. Most research in this area employed trial-and-error methods.

Process optimization tasks are important to the sheet-metal product development and other manufacturing problems as well. Agent approaches have been proposed to get agents equipped with different optimization algorithms involved in solving specific optimization problems (Davidsson et al., 2003). Details are discussed in Section 2.5.

2.5 Agent-Based Optimization

Davidsson et al. (2003) summarized that agent-based approaches may be preferable to solve problems when the problem domain is large, the time-scale of the domain is short or there is sensitivity information that should be kept locally. They proposed combining agent-based technology and mathematical optimization techniques because the properties of agent-based approaches and traditional optimization techniques complement each other.

Applications of the combination of agent-based approaches and optimization techniques have been demonstrated by several projects (as shown in Table 2.5).

Table 2-5. Summary of research on agent-based optimization

Project	**Researchers**	**Main characteristics**
A-Teams	Talukdar et al. (1996)	The A-Teams (Asynchronous Teams) approach combines design utilities such as optimization techniques with autonomous agents, which perform computations independently from other agents and contribute their results in a parallel and distributed fashion.
A-Design	Campbell et al. (1999)	A-Design is presented as an agent-based approach for conceptual engineering design. A-Design combines aspects of multi-objective optimization and multi-agent systems.
?	Rai & Allada (2003)	This paper proposed a new agent-based optimization framework to handle the multi-objective nature of the product family problem by determining the Pareto-design solutions for a given module set. The proposed multi-agent framework has built in flexibility to handle various constraints such as module compatibility during the optimization process.
?	Siirola et al. (2003)	An agent-based framework is proposed for non-convex optimization problems. Six types of agents (hill climber, simulated annealing routine, genetic algorithm, 'void' filler, data trimmer, and monitoring agents) within the proposed framework coordinate on optimization tasks.
?	Deshpande & Cagan (2004)	An agent-based optimization approach that combines stochastic optimization techniques with knowledge-based search has been proposed and successfully used for process sequence and process parameter optimization for the design of bulk manufacturing process.

In these projects, agents equipped with different optimization algorithms coordinate on solving difficult problems in engineering design (Campbell et al., 1999; Rai and Allada, 2003), process planning (Deshpande and Cagan, 2004) and complex optimization problems (Talukdar et al., 1996; Siirola et al., 2003) in multi-agent frameworks. Performance of these projects has demonstrated the effectiveness and flexibility of such systems based on the combination of agent-based approaches and optimization techniques.

The A-Teams (2007) approach is effective in solving complex problems for which many algorithms are available, but none of them is entirely satisfactory. With the A-Teams approach, those algorithms can cooperate to produce much better results much faster than they could if working alone. Siirola et al. (2003) used the A-Teams approach to solve complex non-convex optimization problems. The results have shown that agent collaboration has a significant impact on system performance.

Similar to the problems of these projects, process design for sheet-metal product development is very complex. It is almost impossible to find one or several optimization algorithms to solve all the optimization problems for process design. The A-Teams approach makes it possible to develop a flexible framework to combine different optimization algorithms to solve complex process optimization problems.

2.6 Application of Agent Technology in Metal Forming Area

Chin and Tang (2002) proposed a Web-based architecture for stamped part and die development system as shown in Figure 2-1 and recommended that an agent-based methodology may facilitate the communication and coordination between part designers and die designers.

Tang (2004) summarized the characteristics of the stamping product development process and roles of stamping part designers and die-makers. A simple prototype of federated multi-agent architecture, as shown in Figure 2-3, is introduced to integrate die-maker's activities into product development. Three agent communities (part design agent, die-maker involvement agent, and coordination agent) are developed based on JATLite in his project. JATLite facilitates the development of agents that send and receive KQML messages including eXtensible Markup Language (XML) contents.

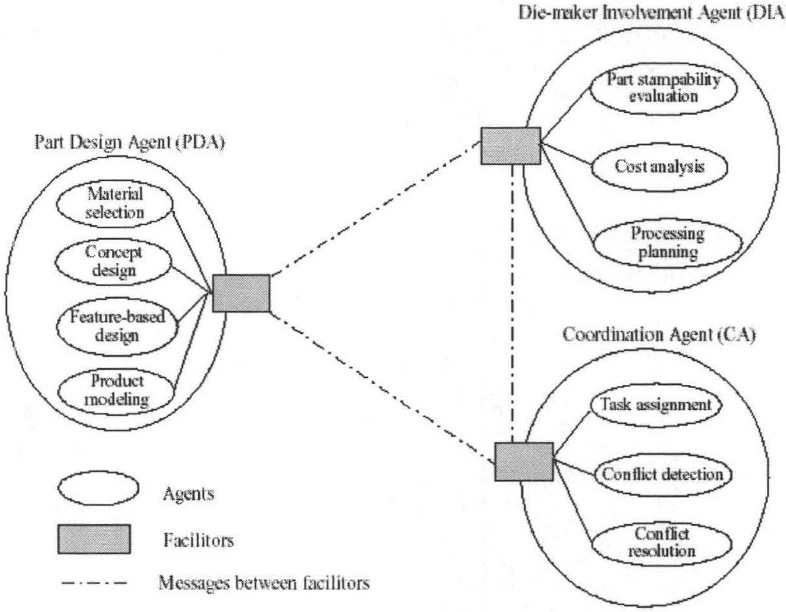


Figure 2-3. Federated multi-agent architecture for collaborative metal stamping development (Tang, 2004)

In Tang (2004)'s research, manufacturability evaluation and cost analysis are based on part design features. Basic feature types include plates, bends, walls, L-brackets, U-channels and boxes. Evaluation rules for basic features, subsidiary features and side-action features are summarized. A cost analysis model for die material cost, die base cost, die manufacturing cost, die setup cost and die maintenance cost is discussed (Chin and Tang, 2002; Tang, 2004). Part design evaluation rules and cost estimation rules are saved in the Die-maker Involvement Agent.

Limitations of Tang (2004)'s research are:

- The federated multi-agent structure for collaborative stamping development is not applicable in some cases. The advantages and disadvantages of different multi-agent architectures have been discussed in section 2.2.3. Hybrid multi-agent structures may be more suitable for some applications.
- Further studies on performance and dynamics of multi-agent architecture are required to improve agent communication and coordination strategies. It is necessary to study the evaluation criteria of multi-agent architectures.
- As a very important area in stamping technology, sheet-metal forming has not been mentioned. Characteristics of sheet-metal forming need to be studied. There is need to undertake further study on how agent-based technology can facilitate the part design, process design and die design for sheet-metal forming.

2.7 Performance Evaluation of Multi-Agent Systems

Over the past few years, performance evaluation of multi-agent systems has unavoidably become an attractive topic due to the rapid growth of applications of multi-agent systems in many areas, such as concurrent design, distributed manufacturing, E-business, etc.

Lee et al. (1998) clarified what performance, scalability and stability mean in the context of multi-agent systems and demonstrated how to analyze the performance and scalability of such systems. A comprehensive definition for performance was given by Lee et al. (1998): "the performance of a multi-agent system is a measure using a set of statistical indicators of the system's major outputs and its consumption of resources, where typical indicators include throughput, response time, number of concurrent agents/tasks, computational time and communication overhead." Scalability is an important issue for the development of practical large-scale multi-agent systems. In the context of multi-agent communities, scalability is defined as the ability of a multi-agent system to continue to function well as it is changed in size. In Lee et al. (1998)'s research, performance and scalability of multi-agent systems with a mesh structure and a hierarchically structure, equipped with a contract-net protocol and an auction protocol, were studied based on the relationships between the computational time and the number of tasks and agents.

Cao et al. (2001) developed a performance modeling and simulation environment to measure the performance of the service discovery system quantitatively by applying metrics. Simulation results of case studies demonstrated the impact of the agent mobility and the choice of optimization strategies on the overall system performance.

Lee and Hwang (2004) identified the possible architectural models for multi-agent systems from the combination of organization structures and coordination strategies, and presented an evaluation approach to evaluate the identified architectural models. In their research, quality characteristics of multi-agent systems were categorized into static and dynamic features. Metrics to calculate quality attributes of static and dynamic features were defined. Static features including complexity, extendability, and availability of systems were evaluated based on the number of links among agents. Dynamic features were evaluated based on the number of interactions rather than the real response time.

Camacho and Aler (2005) tested several frameworks for multi-agent systems by measuring the development time and reusability of the framework and studied the performance and scalability of those systems with respect to the complexity of the task given to the agents in a new Web retrieval domain. Response time and the number of exchanged messages between agents were used for performance evaluation. The behavior of the entire system could be affected if the number of exchanged messages grows quickly. Response time with respect to the number of agents involved in the implemented multi-agent system was measured to obtain a quantitative measure about the scalability of the system.

The above literature demonstrated how the performance of multi-agent systems could be systematically evaluated. Simple case studies were modeled for performance analysis. Performance evaluation approaches proposed by Lee et al. (1998) and Lee and

Hwang (2004) can be extended to develop an evaluation methodology for general applications in concurrent design and manufacturing.

Taking into consideration the characteristics of a specific application domain, performance indicators should be carefully selected from the indicators summarized by Lee et al. (1998). With Lee and Hwang (2004)'s approach, system developers can predict the performance of multi-agent systems by evaluating static system attributes and the dynamic feature, i.e., the number of interactions among agents. Organizational structures and coordination strategies can be selected based on the performance evaluation results. In the system implementation stage, real-time performance factors, such as response time, ratio of coordination time versus computation time, etc., should be evaluated for real applications.

2.8 Summary

The literature review on agent-based collaborative design shows that agent-based technology can facilitate the communication and coordination among participators during the product development process and agent-based systems are flexible and extensible. Agent-based technology can play a very important role in developing collaborative design systems for sheet-metal parts.

The design of human-computer interfaces is one of the most important issues of CAD/CAM/CAE system development. Agent approaches can be a very good supplement to direct manipulation interfaces.

Implicit, explicit and inverse approaches have been combined with optimization methods in simulation-based optimization of sheet-metal forming process. Review on finite element simulation of sheet-metal forming shows that the inverse method is suited for formability analysis and blank estimation at the preliminary stage of product design.

Downhill simplex methods, genetic algorithms and response surface methods are efficient in manufacturing process parameter optimization, part shape or die shape optimization for specific parts. Research projects in agent-based optimization show that combining optimization approaches in a cooperative manner can produce better results than when approaches are executed individually.

Few researchers have worked on agent-based collaborative design systems for sheet-metal parts. Further studies on system design methodology, agent coordination/communication and system evaluation methodology are needed to enhance the application of agent-based technology in metal forming area.

Performance of multi-agent systems can be evaluated by measuring suitable performance indicators selected for specific applications. Organizational structures and coordination strategies should be selected based on the performance evaluation results.

Chapter 3. Construction of Agent-Based Collaborative Design System for Sheet-Metal Parts

3.1 Introduction

To study the performance of agent-based collaborative design systems for sheet-metal parts, a prototype system has to be developed in this research. Usually, the first stage of the construction of an agent-based collaborative design system is initial application analysis. It is necessary to understand the application problem and determine what should be specified as agents at the beginning. Based on the initial application analysis, developers determine what responsibilities agents should fulfill. Then, developers can move to the agent design stage. Agent communication languages and interaction strategies are determined in the agent design stage.

Application analysis of sheet product development, construction of agent-based collaborative system for sheet-metal parts and case studies are going to be discussed in the follows sections.

3.2 Application Analysis

The purpose of the initial application analysis is to understand and model the application problem. In this research, the initial application analysis is to analyze the responsibilities or roles of specific agents when an agent-based system approach is applied to concurrent sheet-metal product design.

3.2.1 Responsibilities of Participants

From the point of view of a supply chain network, participants in sheet-metal product development can be categorized into customers and suppliers. In this case, customers represent the enterprises that design the stamped part and order the stamped parts from suppliers. As shown in Table 3-1, an initial design plan of a stamped part is generated by the customer and the initial part design information is then passed to suppliers. Customers will get feedback about process arrangements and costs from suppliers. If redesign is necessary, customers will communicate with suppliers for any information needed.

Table 3-1. Responsibilities of participants in the supply chain

Participants	**Responsibilities**
Customer	▪ Design stamped parts based on the requirements ▪ Process design
Supplier	▪ Process design ▪ Die design ▪ Die manufacturing ▪ Stamping production ▪ Cost analysis

Suppliers can manufacture stamped parts and dies for the stamping production. Suppliers may include die design departments, die manufacturing departments and stamping production departments. These departments may not belong to one supplier. For example, one supplier may only provide the service of die design, one supplier may only work on die manufacturing and another supplier may provide the service of stamping production. Customers usually give the contracts to the suppliers that can provide the finished stamped parts. If die design, die manufacturing and stamping production are not performed by the same supplier, the supplier that gets the contract to manufacture the stamped parts will decide which supplier will get the contract to manufacture the dies and/or design the dies.

Customers may have information about certain suppliers. Those suppliers can be called acquaintances of customers. When customers need to order stamped parts, they usually contact those acquaintances first. Moreover, a hierarchical structure may exist in a large company. Relationships between participants can be categorized into subordinates, superiors and coworkers. Complex organization structures in sheet-metal product development demands flexibility in the agent-based design methodology in this research.

Responsibilities of all the participants of sheet-metal product development, such as part designers, die designers, die manufacturing engineers and stamping production engineers, are summarized in Table 3-2. The initial part design is based on customer requirements, such as functions that the stamped part will have to provide, production volume, assembly requirements, weight requirements, appearance, time to market and the initial target price. Part designers are supposed to understand the requirements and determine an initial part design based on those customer requirements.

3.2.2 Assigning Roles to Agents

By the end of application analysis, developers of agent-based systems need to allocate those responsibilities or roles to agents.

Two metrics were introduced by Collis and Ndumu (1999) to determine whether candidate entities would make appropriate agents: the sphere of responsibility test and the point of interaction test.

Table 3-2. Task assignment

Participants	Responsibilities
Part designer	▪ Material selection ▪ Part shape design (dimensions and tolerances) ▪ Technical requirements: burr side, grain direction, etc.
Die designer	▪ Unfolding ▪ Nesting strip layout ▪ Process design ▪ Die configuration
Die manufacturing engineer	▪ Die manufacturing process planning ▪ Die manufacturing cost estimation
Stamping production engineer	▪ Stamping equipment arrangement ▪ Quality control ▪ Production cost estimation

The sphere of responsibility test is derived from the fact that agents should be autonomous, i.e. be responsible for the control of resources and provision of services. A developer will need to consider how the application domain will be partitioned.

The second metric, the point of interaction test, considers the social dimension of agents. Agents are often distinguished from other software systems by their ability to interact intelligently and constructively with other agents and people. Hence in an agent application resources and services may not be directly accessible, but invoked by requesting the agent responsible for their control. The purpose of this test is to help separate application resources from the entities that will use them to provide services.

This is particularly relevant to applications where agent systems serve as the interfaces to legacy systems such as databases.

To develop an agent-based collaborative design system for sheet-metal parts, the following agents can be created to play the roles of specific participants.

Table 3-3. Agent Roles

Agent Name	**Participants**
Design Coordinator Agent	Customer Enterprise
Supplier Coordinator Agent	Supplier Enterprise
Part Design Agent	Part Design Engineer
Die Design Agent	Die Design Engineer
Die Manufacturing Agent	Die Manufacturing Engineer
Stamping Production Agent	Stamping Production Engineer
Cost Estimation Agent	Marketing Person from the Supplier Enterprise

Detailed responsibilities of the above agents will be introduced in section 3.4.3. Based on the role analysis of participants in sheet-metal product development, development tools, such as the agent building toolkit, part and die modeling platform and data and knowledge management system, have to be selected at the beginning of the implementation of the agent-based collaborative design system.

3.3 Development Tools

3.3.1 ZEUS Agent Building Toolkit

Implementing an agent-based system from scratch requires a vast amount of work. Agent building toolkits on the market were studied and ZEUS was selected as the agent building toolkit for fast development of the agent-based sheet-metal product development because of its advantages mentioned in Section 2.2.6.

3.3.2 Mechanical Desktop and ObjectARX

Part design and die design processes are very complicated and can be regarded as the most important stages in sheet-metal product development. In this research, we select Autodesk Mechanical Desktop as an engineering software tool to build part design agents and die design agents because Autodesk products, such as AutoCAD and Mechanical Desktop, are widely used in the mechanical design area and powerful customization tools are provided for developers to create new commands and user interfaces for domain applications.

ObjectARX (Autodesk Inc., 2004), which provides an object-oriented C++ application programming interface for developers to use, customize and extend AutoCAD and AutoCAD-based products, is used for the development of part design and die design modules on the Autodesk Mechanical Desktop platform. Part design and die design modules are wrapped by ZEUS agents. The communication between the ObjectARX program and ZEUS agent interface is based on COM (Microsoft Corp., 2004) technology in this research.

3.3.3 MySQL-Data Management System

The database management system used in our research is MySQL (MySQL AB, 2003). MySQL is one of the most popular open source relational database management systems in the market.

MySQL APIs in C, C++ and Java are distributed with the MySQL source. Agents communicate with MySQL server using those APIs.

3.3.4 Programming Languages

ZEUS agent building toolkit provides a code generator to automatically generate some source code in Java for task agents based on the user input. To create a complete application for collaborative design of sheet-metal parts, developers still have a lot of programming work to do for external programs, interaction operations and negotiation strategies.

Some external programs, such as the nesting module of the die design agent and customization of Mechanical Desktop with ObjectARX, were written in Visual C++.

3.4 Application Realization

In this section, a concrete implementation of the agent application will be described. The architecture of a prototype system for agent-based collaborative design of sheet-metal parts and detailed agent design will be outlined.

3.4.1 Architecture of the Prototype System

To achieve the research objectives, a prototype system has to be developed. Based on the application analysis, responsibilities of participants in the supply chain can be assigned to task agents:

- design coordinator agents;

- part design agents;
- supplier coordinator agents;
- die design agents;
- die manufacturing agents;
- stamping production agents;
- cost estimation agents.

With the ZEUS toolkit, a facilitator agent, a global name server agent and local name server agents are added as utility agents to the prototype system as shown in Figure 3-1. Design coordinator agents and supplier coordinator agents also play the roles of local facilitators. Facilitator agents store the abilities of agents, receive and respond to queries from agents about the abilities of other agents. Nameserver agents receive and respond to agents' requests for the addresses of other agents and maintain a society-wide clock.

The proposed architecture shown in Figure 3-1 can deal with situations when there are a large number of agents involved in the product development process. There can be several part design agents and service agents in a customer agent community. It also applies to supplier agent communities. Service agents in the customer agent community can be nesting agents, computing service providers, or evaluation tools for initial product design. Service agents in the supplier agent community can be formability analysis tools, optimization tools, etc.

Local name server agents are not absolutely necessary if the agent community is small and stable and the design or supplier coordinator agent has detailed information of other agents in the same community.

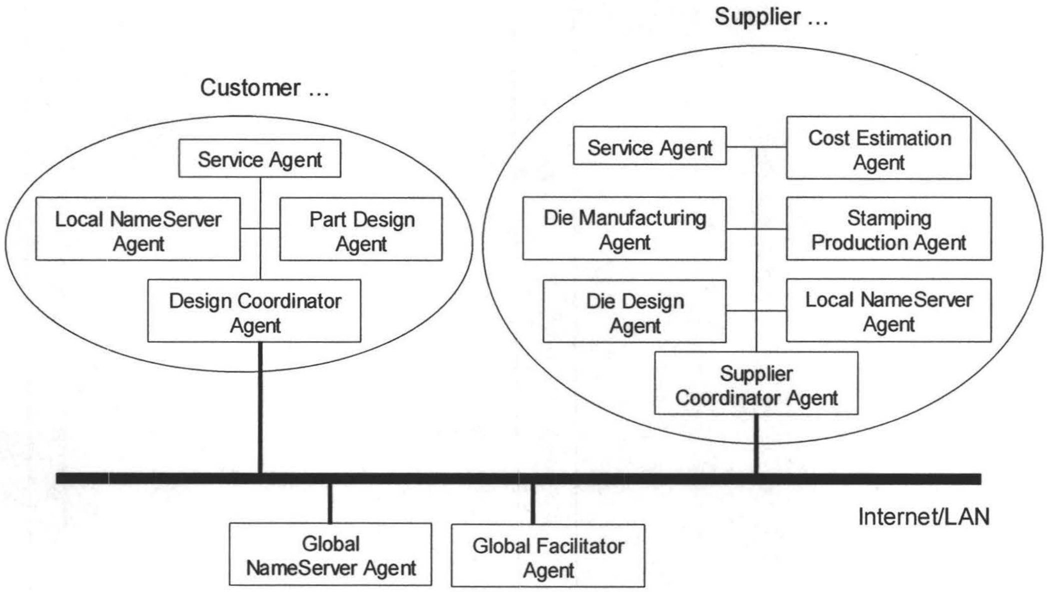
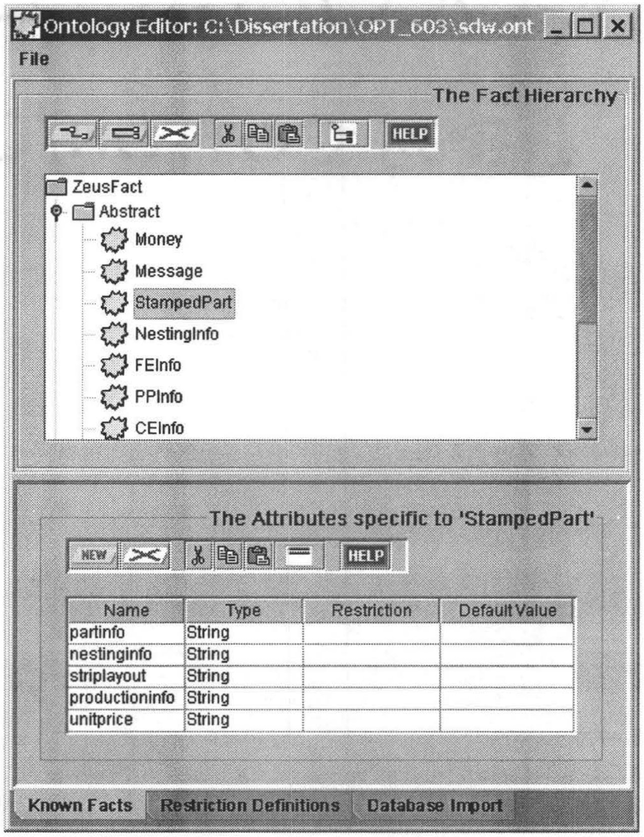


Figure 3-1. Architecture of the prototype system

3.4.2 Ontology Creation

In this research, facts that represent different domain concepts are generated as child facts of Abstract fact. Figure 3-2 shows how new facts are created using ZEUS ontology editor. For example, five attributes were assigned to the fact of "StampedPart".

Once a ZEUS agent is initialized, its internal ontology database will store the logical definition of each fact type, attributes of each fact, the range of legal values for each attribute, any constraints between attribute values, and any relationships between the attributes of the fact and other facts.



Name	Type	Restriction	Default Value
partinfo	String		
nestinginfo	String		
striplayout	String		
productioninfo	String		
unitprice	String		

Figure 3-2. Ontology creation

3.4.3 Agent Creation

The agent creation process comprises of several sub-processes, such as the agent definition process, task description process, agent organization process and the agent coordination process.

Detailed responsibilities of each task agent have been summarized in Tables 3-4 and 3-5. Simple unfolding and nesting functions can be assigned to service agents in the customer agent community to improve the initial design. In some cases, it is a natural way to do that because usually it is the designers on the customer side that care about the raw material usage. However, those simple service agents may not have enough capability to deal with complex parts. In that case, die design agents should become involved.

Table 3-4. Description of task agents in the customer agent community

Agent Name	**Responsibilities**	**Development Tools**
Design Coordinator Agent	• To initiate a request for part formability analysis, cost analysis • To negotiate the terms of process design, tooling configuration, stamping equipment arrangement with suppliers and/or supplier coordinator agents • To represent part design agents	Java
Part Design Agent	• It wraps Mechanical Desktop as a design agent • Part modeling • Preliminary feature identification based on knowledge; • Part evaluation; • To submit the design information to the design coordinator agent and receive feedback from the design coordinator agent	Java; Visual C++; ObjectARX; MySQL C++ API; COM
Service Agent	• Unfolding • Nesting • Providing computing services	Java; Visual C++

Table 3-5. Description of task agents in the supplier agent community

Agent Name	Responsibilities	Development Tools
Supplier Coordinator Agent	• To respond to a request for part formability analysis, cost analysis • To represent die design agents, die manufacturing agents, stamping production agents and cost estimation agents	Java
Die Design Agent	• It wraps Mechanical Desktop as a design agent • Unfolding • Nesting • Process design; • Die configuration modeling	Java; Visual C++; ObjectARX; MySQL C++ API; COM
Die Manufacturing Agent	To provide detailed information about process planning of die manufacturing	Java; MySQL Java API
Stamping Production Agent	To provide detailed information about press machines for stamping production	Java; MySQL Java API
Cost Estimation Agent	To estimate the product cost in the very early stage of product development based on part geometry, process configuration and manufacturing processes	Java; MySQL Java API
Service Agent	• Formability analysis • Optimization	Java; Visual C++

Part Design and Die Design Agents

A feature-based design approach is used for the modeling of stamped parts and the representing manufacturing features for process planning for die design.

In this research, features of stamped parts are categorized into basic features (flat and drawn blanks), subsidiary features (holes, embossing, etc.) and connection features (bends and blends). The data structure of these features is implemented in C++.

Features of stamped parts can be mapped into manufacturing features, such as blanking, piercing, bending, etc. A member pointer in the class of a manufacturing feature was defined to point to the basic feature of the stamped part. Specific rules were designed for mapping part features to manufacturing features. Users can also manually create new manufacturing features in the Mechanical Desktop platform. For example, the profile design of cutting (piercing or blanking) processes prior to bending or the final blanking process is very complex. It is difficult to summarize perfect rules to generate those profiles automatically. It is still very helpful to design user interfaces for designers, especially experienced designers, to make some changes to the existing design or create profiles from scratch.

The agent structure of part design agents and die design agents is shown in Figure 3-3. The wrapper is a ZEUS agent created with Java. It runs a ZEUS external program to communicate with the feature-based design module. As mentioned in Section 3.3.2, the communication between the wrapper and the C++ module created with ObjectARX is based on the COM technology. MySQL C++ API is used for the communication between the C++ module and the database, which stores material data and other design information. ZEUS agent wrapping technology, ObjectARX and SQL (Structured Query Language) constitute the technical basis for the implementation of part design agents and die design agents.

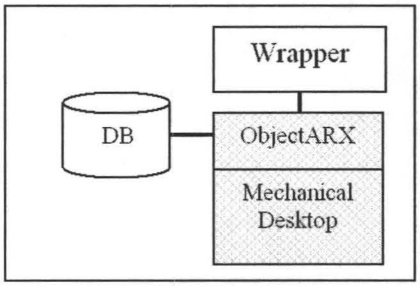


Figure 3-3. Agent structure of part design and die design agents

Stamping Production Agent

The stamping production agent connects with a press database. Press tonnages, stroking rates, bed sizes and other press information are saved in the press database. The stamping production agent is also equipped with press selection rules. More sophisticated agents are envisaged that will interface with a firm's production planning database to help schedule production and control capacity utilization.

Cost Estimation Agent

The cost estimation agent provides cost models to estimate the manufacturing cost of metal stampings. Current cost models have not taken into consideration the delivery costs and tax rates. Die cost is estimated according to the working area of stamping operations along the strip layout. The working area means a rectangle along the strip layout to contain all or part of stamping operations. Details of the cost models will be introduced in Section 3.5.

3.4.4 Agent Coordination

"Coordination is the process by which an agent reasons about its local actions and the (anticipated) actions of others to try and ensure the community acts in a coherent manner" (Jennings, 1996). Agent coordination is necessary because of the decentralization of sheet-metal product development systems and limitations of individual agents.

Task allocation is based on the contract-net protocol in this research. Agent interaction based on the contract-net protocol comprises three phases: task announcement, bidding and awarding. A manager announces a task that needs to be performed and then receives and evaluates bids for the task. Based on its evaluation criteria, the manager awards the contract to the contractor, which provides a satisfactory bid before the bidding deadline.

As with the prototype agent-based collaborative design system for sheet-metal parts, agent coordination is based on the following assumptions:

- Agents have domain-specific expertise. Each agent can make a decision independently based on its own knowledge.
- In this research, 'agents' are cooperative agents. When a conflict arises, agents involved negotiate to find a resolution. All the agents cooperate for better solutions of product development. Agents are honest and always give the best offer.

- Rules for conflict resolution are organized based on previous cases.
- At the current stage, the prototype system is developed for collaborative design. The supplier coordinator agent does not really produce the part. Agents in the prototype system coordinate on getting an optimal or near optimal design. In the long term, an agent-based supply chain management system for sheet-metal parts can be developed based on this work. Equipped with the contract-net protocol, the prototype system shown in Figure 3-1 can work as a supply chain management system for sheet-metal parts. Cost and time-to-market can be set as the evaluation criteria for task allocation in such a system.

To coordinate activities of participants in collaborative design, an extension of the basic contract-net protocol is applied to this research. In this research, a bid offered by a supplier coordinator agent contains not only the cost and time-to-market but also process design, manufacturability analysis and recommendations about conflicts detected, conflict resolutions and/or modifications that can reduce the cost. To solve the task of part evaluation, the design coordinator agent is called the *manager*; the supplier coordinator agents that might be able to solve the task are called potential *contractors*. From a manager's perspective, the process in this protocol is:

- **Step 1:** Receive a request to announce a task of part evaluation from part design agents.
- **Step 2:** Announce the task to potential contractors, i.e. supplier coordinator agents.

- **Step 3:** Receive and evaluate bids from potential contractors.
- **Step 4:** If a suitable bid is found, award a contract to the suitable contractor for the future production; else, iterate the process by issuing a revised CFP (Call for Proposals) to potential contractors that offered a bid in the previous round. Then, go to **Step 3**.

From the perspective of a contractor, i.e. a supplier coordinator agent, the process is

- **Step 1:** Receive task announcements from a design coordinator agent.
- **Step 2:** Decide whether to respond or not.
- **Step 3:** Decline or bid.

A manager agent may not receive bids after the expiration time has been reached. It may then announce the task and request potential contractors that are eligible but busy, ineligible, or uninterested in order to take appropriate actions.

The contract might not be awarded to the most capable agent because it is busy. The manager has no obligation to inform other potential contractors of the contract awarded. Therefore, the most capable agent may not get the contract due to the settings of its negotiation strategies although it still has a room to make concessions.

Since agents are domain-specific, conflicts are unavoidable in the process of agent coordination. For example, a poor part design created by a part design agent may cause poor formability and high manufacturing cost. The conflicts are solved through iterations by issuing revised CFPs and bidding.

3.5 A Case Study

Section 3.4 presents the system implementation approach. In this section, a case study illustrates how the prototype agent-based system works. Three agent communities, including one customer agent community and two supplier agent communities, were created for case studies as shown in Figure 3-4. A nesting agent was added as a service agent in the customer agent community. In Figure 3-4, The "Broker" agent represents the "Design Coordinator Agent" and the "Supplier" agent represents the "Supplier Coordinator Agent" as shown in Figure 3-1. Since there are only a small number of agents involved, local nameserver agents are disabled to reduce the interactions among agents.

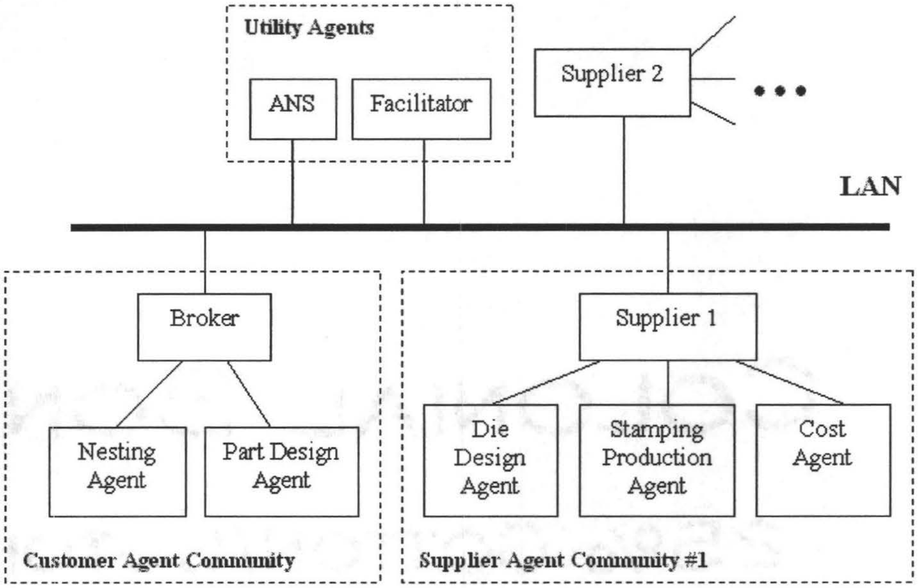


Figure 3-4. Proposed system for case studies

All the systems were deployed on three computers with Windows XP/2000 on a local area network.

Specific details of the suppliers are:

Stamping facility of Supplier 1:

400 KN Power Press (Qty: 1)

Stroking Rate (rpm): 50

Bed Size (mm x mm): 630 x 420

Press Hourly Cost: ($) 120

Stamping facility of Supplier 2:

250 KN Power Press (Qty: 2)

Stroking Rate (rpm): 55

Bed Size (mm x mm): 560 x 360

Press Hourly Cost: ($) 100

A simple sheet-metal part consisting of only one flat wall and holes on the wall, as shown in Figure 3-5, was used in case study. Material properties are shown as follows:

Material: low carbon steel

Thickness: 2 mm

Shear strength (τ): 180 MPa

Quantity: 50,000 pieces.

Material Cost (for all suppliers) 0.02 $/cm^2

A user interface (as shown in Figure 3-6) for part information input was developed using ObjectARX in C++. Part information includes material, thickness, burr side, grain direction, estimated quantity and other requirements of the part. The profile (e.g. outer-profile, holes) of the part is drawn using native Mechanical Desktop commands. A feature tree saves all the part information and is accessed by a Java program by means of COM technology.

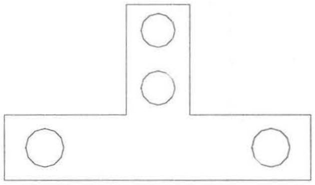
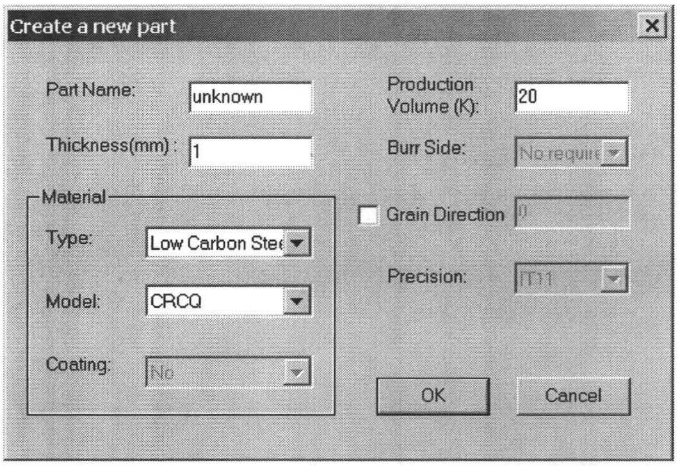


Figure 3-5. Profile of a sheet-metal part



Create a new part

Part Name: unknown

Thickness(mm) : 1

Material

Type: Low Carbon Ste

Model: CRCQ

Coating: No

Production Volume (K): 20

Burr Side: No require

☐ Grain Direction 0

Precision: IT11

Figure 3-6. User interface for part information input

In the process of part design, designers may want to know the possible strip layout and cost of the part in the early part design stage. With the help of the broker (i.e., design coordinator agent), the part design agent gets the address of the nesting agent. Then, the part design agent negotiates with the nesting agent. After a proposal is accepted, the part design agent sends the task to the nesting agent and the nesting agent implements the task and returns the result to part design agent once the task is done.

In this case study, an instance of the Performative class included in ZEUS toolkit is sent to the nesting agent. Below is the content of this performative message.

Content: data (:type NestPoly :id fact_97 :modifiers 0 :attributes ((info "-7.48822173 83.31639883 46.85841676 21.84655333 197.91037614 59.95710507 192.79679443 175.93230295 66.88170526 188.15609988 ")))

Vertex coordinates of the polygon are saved as a list with an attribute name of "info". The nesting agent provides an optimal layout for the part design agent. Pitch, width, angle and utilization rate of the layout are shown in the message content as below.

data (:type NestRes :id fact_47 :modifiers 0 :attributes ((info "Angle -1.526733 Pitch 159.638435 Width 204.170271 Utilization 0.754263")))

The nesting profile is shown in Figure 3-7.

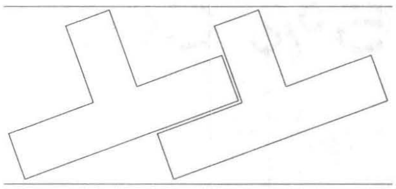
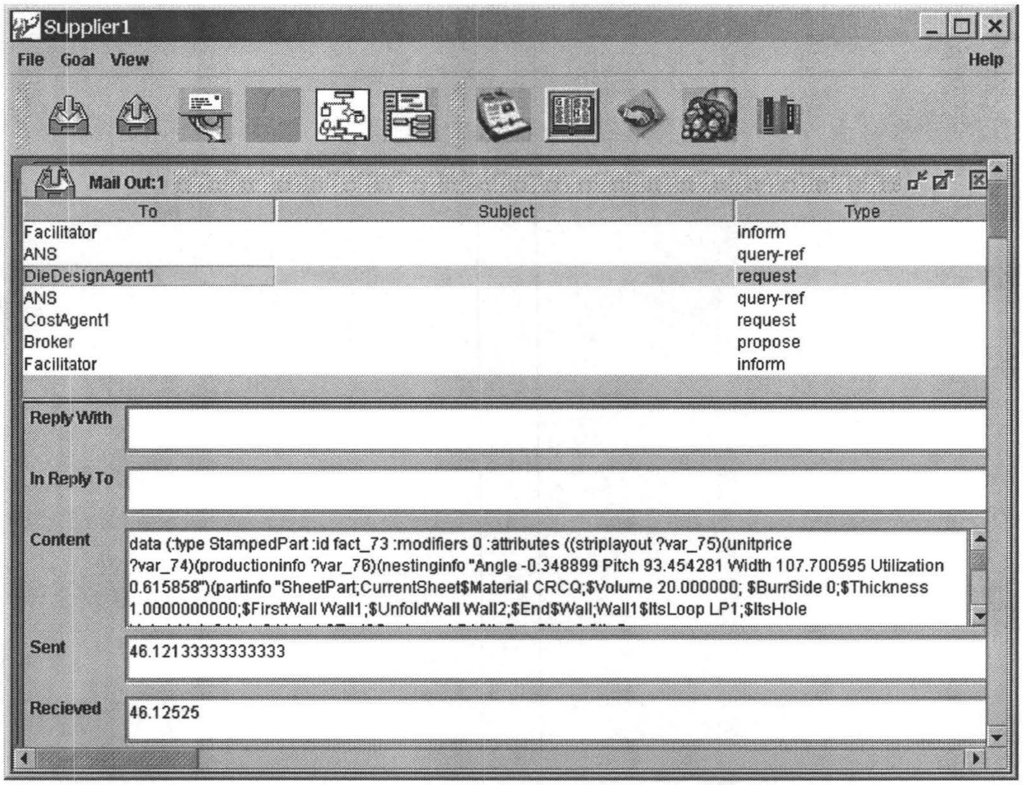


Figure 3-7. Nesting profile

The part design agent then sends a request for part evaluation to the broker and the broker announces the tasks to all potential contractors, i.e., supplier 1 and supplier 2 in this case study. Suppliers then assign the task to die design agents. As shown in Figure 3-8, Supplier 1 sent a request message to the die design agent for part evaluation. The content of the request message includes the details of part information and nesting information.



To	Subject	Type
Facilitator		inform
ANS		query-ref
DieDesignAgent1		request
ANS		query-ref
CostAgent1		request
Broker		propose
Facilitator		inform

Reply With

In Reply To

Content

```
data (:type StampedPart :id fact_73 :modifiers 0 :attributes ((striplayout ?var_75)(unitprice
?var_74)(productioninfo ?var_76)(nestinginfo "Angle -0.348899 Pitch 93.454281 Width 107.700595 Utilization
0.615858")(partinfo "SheetPart;CurrentSheet$Material CRCQ;$Volume 20.000000; $BurrSide 0;$Thickness
1.0000000000;$FirstWall Wall1;$UnfoldWall Wall2;$End$Wall;Wall1$ItsLoop LP1;$ItsHole
```

Sent 46.121333333333333

Recieved 46.12525

Figure 3-8. Mailbox of Supplier 1

In this prototype system we have implemented relatively simple models for estimating costs; obviously any firm will have their own methods and values for calculating costs. Regardless of the formula used for values of specific constants, they can be easily implemented in a production agent.

The particular cost models used in this prototype system are:

Material Cost:

$$C_{Material} = b \cdot p \cdot C_{Strip} \tag{3-1}$$

Where:

$C_{Material}$ = material cost per part, \$

b = strip width, cm

p = pitch of the strip layout, cm

C_{Strip} = strip material cost , \$/cm^2

In this case,

$$C_{Material} = 11 \times 9.3 \times 0.02 = 0.2 \text{ \$/part} \tag{3-2}$$

Cutting Force:

$$F_{Cutting} = L \cdot t \cdot \tau \tag{3-3}$$

Where:

$F_{Cutting}$ = cutting force, N

L = sheared length, m

t = material thickness, m

τ = shear strength material, Pa

In this case, blanking force is:

$$F_{Blanking} = 0.444 \times 0.002 \times 180000000 = 159840 \text{ N} \tag{3-4}$$

While piercing force for the holes becomes:

$$F_{Piercing} = 0.190 \times 0.002 \times 180000000 = 68400 \text{ N} \tag{3-5}$$

Taking into consideration the stripping force and adding a safety factor totaling 1.4, total force required is:

$$F_{Total} = 1.4 \times F_{Cutting} = 1.4 \times (F_{Blanking} + F_{Piercing}) \tag{3-6}$$

In this case,

$$F_{Total} = 1.4 \times (159840 + 68400) = 319536 \text{ N} \tag{3-7}$$

Die Cost:

To estimate the cost of a die, this prototype system simply multiplies the plan area of the die by a constant, as this roughly correlates with die cost. (More sophisticated calculations could be substituted into the future tooling cost agents without affecting other aspects of the system.) Assuming the die is used only for this single production order, cost per part becomes:

$$C_{Die} = A \cdot C_{cpc} / Qty \tag{3-8}$$

Where:

C_{Die} = tooling cost per part, \$

A = projected area of die, cm^2

C_{cpc} = cost per square centimeter of die area, \$/cm^2

Qty = quantity of stamped parts

Production Cost:

Assuming that all production costs, including setups, scrap profits and overhead, are captured in a standard hourly cost for the press, the production cost per part becomes:

$$C_{\Pr oduction} = C_{Hour} / (60 \times R_{Stroking}) \tag{3-9}$$

Where:

$C_{\Pr oduction}$ = production cost per part, \$

C_{Hour} = hourly cost of the press, \$/hr

$R_{Stroking}$ = stroking rate, rpm

Total Cost:

Total cost is simply the sum of these components, or:

$$C_{Total} = C_{Material} + C_{Die} + C_{\Pr oduction} \tag{3-10}$$

The stamping production agent provides the information of a stamping press that is suitable for the real stamping production. The cost analysis agent calculates the cost of the part based on the usage of raw material, die cost and stamping production cost.

Supplier 1:

Comparing the total force require for piercing and blanking (i.e., 320 kN) to the capacity of Supplier 1's press (400 kN) shows that there is ample capacity to make the part in a progressive die with 2 stages. The die design agent predicts the strip layout shown in Fig. 4, with a corresponding projected die area of 235.09 cm^2. If Supplier 1's cost of constructing dies if \$45/cm^2, and other production details are as given above, the Production Agent and Cost Agent negotiate through the supplier's LAN to produce the following:

$$C_{Die} = 235.09 \times 45 / 50000 = 0.21 \tag{3-11}$$

$$C_{\Pr oduction} = 120 / (60 \times 50) = 0.04 \tag{3-12}$$

$$C_{Total} = 0.2 + 0.21 + 0.04 = 0.45 \tag{3-13}$$

This cost (which implicitly includes the firm's desired profits on the contract) is used as the basis of Supplier 1's bid to the Broker agent.

Supplier 2:

Since the total tonnage of both piercing and blanking operations together (i.e., 320 kN) exceed the capacity of either of Supplier 2's presses (i.e., 250 kN), the die design agent decided that two dies must be constructed; one for blanking and one for piercing so that each operation may be carried out separately. The die design agent estimated the projected areas of these two dies to be 108 cm^2 and 102 cm^2, respectively. Assuming Supplier 2's die construction cost is \$55/cm^2,

$$C_{BlankingDie} = 108 \times 55 / 50000 = 0.12 \tag{3-14}$$

$$C_{PiercingDie} = 102 \times 55 / 50000 = 0.11 \tag{3-15}$$

For each press:

$$C_{\Pr oduction} = 100 / (60 \times 50) = 0.033 \tag{3-16}$$

Total cost per part for supplier 2 is then:

$$C_{Total} = C_{Material} + C_{BlankingDie} + C_{PiercingDie} + C_{\Pr oduction}$$

$$= 0.2 + 0.12 + 0.11 + 2*0.033 = 0.496 \tag{3-17}$$

This is the price bid by Supplier 2 for the production contract. The Broker agent accepts and compares the bids and awards the production contract to Supplier 1 due to a better price compared to Supplier 2.

Figure 3-9 shows the strip layout generated by the die design agent of Supplier 1 and Figure 3-10 shows the results sent from the cost estimation agent to Supplier 1.

While this prototype multi-agent system has been confined to a local area network, and the agents so far are limited in the complexity of product data and geometry they can contend with, this illustrative example shows how such a system can automate product and process design tasks. In this example, the product designer received feedback on part nesting, stamping die layout, and estimated product cost in less than a minute.

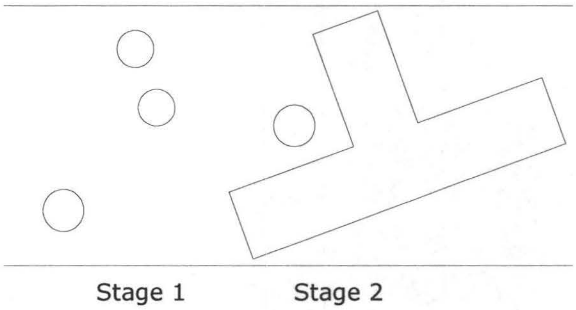
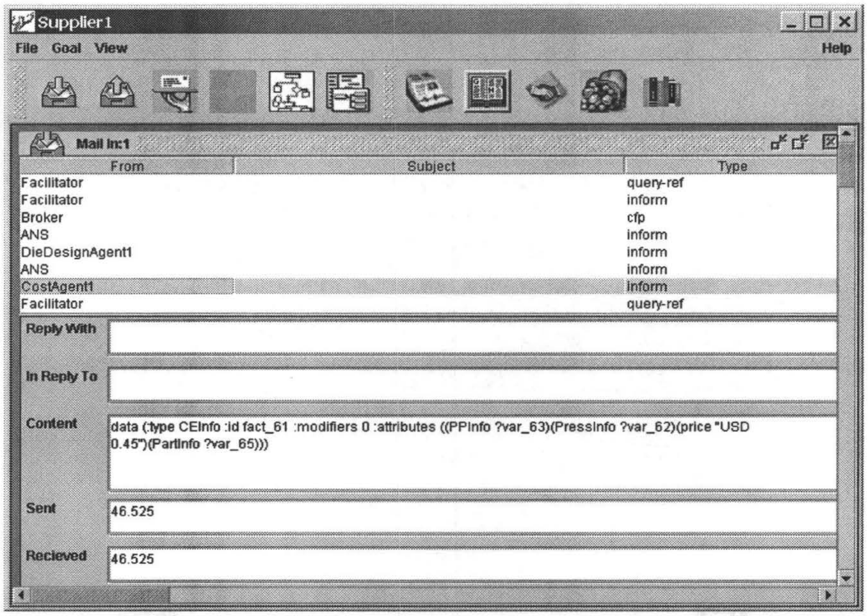


Figure 3-9. The strip layout generated by the die design agent of supplier 1



From	Subject	Type
Facilitator		query-ref
Facilitator		inform
Broker		cfp
ANS		inform
DieDesignAgent1		inform
ANS		inform
CostAgent1		inform
Facilitator		query-ref

Reply With

In Reply To

Content: data (:type CEInfo :id fact_61 :modifiers 0 :attributes ((PPInfo ?var_63)(PressInfo ?var_62)(price "USD 0.45")(PartInfo ?var_65)))

Sent: 46.525

Recieved: 46.525

Figure 3-10. Results obtained from the cost estimation agent of Supplier 1

3.6 Summary

A federated prototype agent-based system for sheet-metal product development based on the ZEUS toolkit was presented. Important issues for the development of agent-based systems, such as domain analysis, legacy system wrapping, ontology representation, agent communication and coordination, etc., have been discussed in this Chapter.

A prototype for collaborative sheet-metal part design based on the integrated design and analysis methodology has been developed as a proof-of-concept system. This prototype system demonstrates the capability of such a multi-agent approach to integrate activities, such as part design, process design, die design and manufacturing and cost estimation. In the following Chapter, this system will be extended to facilitate the communication and coordination among participants in the supply chain. The performance of the prototype system will be evaluated as it is extended to include other service agents to become a comprehensive distributed collaborative design system.

Chapter 4. Agent-Based Process and Tooling Optimization of Sheet-Metal Forming

4.1 Introduction

The prototype agent-based collaborative design system for sheet-metal parts presented in Chapter 3 is focused on the integration of product development processes. The performance of the prototype system demonstrates that communication and coordination among domain agents can facilitate product development and reduce product cost. As shown in Figure 3-1, agents in the prototype system comprise customer agent communities, supplier agent communities and global utility agents. To implement such a system, different kinds of architectures (i.e. autonomous, federated, hierachical or hybrid architectures) can be selected to develop specific agent communities based on the organizational structures of customers or suppliers the agent communities represent. Agents in an agent community representing a customer or a supplier may be distributed either on a local area network, or for complex cases, on the Internet. Tasks undertaken by an agent community can be decomposed to many subtasks. Sometimes a lot of communication and coordination work is needed inside an agent community. To ensure the overall performance and effectiveness of the collaborative design system, it is necessary to study the activities of each agent community.

In an agent-based collaborative design system for sheet-metal parts, a typical agent community representing a supplier may undertake a task of providing a manufacturing plan that can match the product requirements. This task will be decomposed by the supplier mediator agent into several subtasks, such as process and die design, die manufacturing planning, stamping production planning and cost estimation.

These subtasks will be undertaken by task agents in the supplier agent community. For example, die design agents are in charge of process and die design.

As shown in Table 3-2, tasks implemented by die designer include unfolding, nesting, process design and the determination of the proper die configuration. Die design agents need to collaborate with die manufacturing agents, cost estimation agents, stamping production agents and service agents (e.g. optimization agents, nesting agents, formability analysis agents) to achieve optimal or near-optimal process and die design. Meanwhile, substantial computational work may be needed if formability analysis or simulation-based optimization is involved in process and die design.

In this chapter, application of agent-based process and tooling optimization is going to be discussed. A prototype system for agent-based process and tooling optimization has been developed to study the performance of agent coordination in process and tooling optimization.

The following Section 4.2 introduces the characteristics of the process and tooling design in sheet-metal forming. Applications of forming limit diagrams, finite element analysis and simulation-based optimization in sheet-metal forming are discussed.

Section 4.3 introduces the optimization model that is used for case studies in this Chapter.

In Section 4.4, advantages of the combination of agent-based technologies and optimization algorithms are discussed and an agent-based system framework for agent-based process and tooling optimization is proposed. Application analysis and implementation of the prototype system are introduced and detailed information about optimization agents and agent coordination strategies are also presented.

Case studies of optimal process design of a rectangular drawn cup and a front door panel are introduced in Sections 4.5, 4.6 and 4.7.

Discussions and recommendations are made in Section 4.8.

4.2 Process and Tooling Design of Sheet-Metal Forming

4.2.1 Characteristics of Process and Tooling Design

As mentioned in Section 1.1, sheet-metal parts are made with combinations of cutting and forming operations. Typical forming operations include drawing, bending, flanging, embossing, etc. Process and tooling design of sheet-metal forming is experience based and very time-consuming.

Key design issues during the process and tooling design are related to the determination of:

- Die configuration (e.g. geometry of punch and die, punch-die clearance, tooling materials, position and geometry of drawbeads, etc.).
- Operation parameters (e.g. punch load, blankholder force, lubrication).
- Optimal blank geometry.
- Optimal number of stages and stage process arrangement for multi-stage sheet-metal forming.

Figure 4.1 shows an example of sheet-metal forming. The blank is clamped between the *die* and the *blankholder*. Then the *punch* is pushed into the die cavity, simultaneously transferring the specific shape of the punch and the die to the blank.

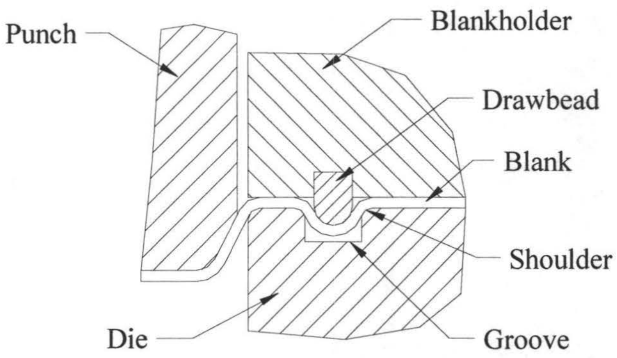


Figure 4-1. Typical die configuration for sheet-metal forming

Prior to the determination of die configuration, proper selection of a press has to be made based on the force, energy, size and speed requirements. Mechanical presses and hydraulic presses are most widely used in sheet-metal forming. Hydraulic presses can meet very high force requirements. Classified by the number of slides, most widely used presses are single-action and double-action presses. Double-action presses are suitable for deep drawing processes.

Once the selection of the forming press has been made, designers can decide what kind of die structure is suitable and then work on the details of tooling design.

Determination of the geometry of punch and die, punch-die clearance, tooling materials and blank geometry is the main design task.

In sheet-metal forming, it is frequently necessary to control the rate of metal flow into the die cavity in order to ensure product quality. Usually the material flow is controlled by using a blankholder or a curved binder to clamp the blank to the die and avoid wrinkling and tearing. The blankholder force may be varied by controlling the clamping force. High blankholder force restricts the flow of material into the die, reducing the tendency to wrinkle but increasing the tendency to fracture (i.e., tearing). Low blankholder force reduces the chance of part fracture, but increases the chance of wrinkling.

There are no absolute rules for calculating the blankholder force for a given drawing operation. In most cases, blankholder force values are determined empirically. Blankholder force should be just sufficient to prevent wrinkling, and it depends on work metal thickness and properties, the type of lubrication used, and other forming conditions.

For complex sheet-metal forming operations, drawbeads can be added to provide additional control in addition to the blankholder. As shown in Figure 4-1, a typical drawbead consists of a semi cylindrical ridge in the upper part of the blankholder and a corresponding groove with rounded shoulders in the lower part, or a similar but opposite configuration.

The role of the drawbeads in connection to sheet metal forming is to provide a restraining force to the blank around the die periphery in order to control the material flow into the die cavity. The restraining force should be large enough to prevent the blank from wrinkling, and for certain components this force should be able to produce a certain minimum level of straining in order to reduce springback and improve shape accuracy. On the other hand, the restraining force must not be so big that it causes tearing in the blank material. The position and force of the drawbead are usually determined by experience or by trial and error.

Blank geometry plays an important role in forming operations. Improper blank geometry may lead to poor formability and the waste of raw material. In forming operations, e.g. deep drawing, prediction of the blank geometry is very important. Blank estimation is based on analytical methods or numerical simulations.

With the blank geometry (i.e. unfolded part profile), nesting algorithms can be adopted for generating a nested strip layout with an acceptable material utilization rate. The nested strip layout directly affects the process design, die configuration and the product cost. Coordination between part designers and die designers is usually necessary to generate a proper nested strip layout.

Multi-stage operations are usually needed to produce complex sheet-metal parts. As there can be many combinations of the number of stages and the process arrangement for each stage in a multistage die, it is difficult to determine the optimal die design.

Due to the complexity of process and die design, traditional stamping die design is a trial-and-error based process that may lead to high cost and long time-to-market. Existing technologies, such as forming limit diagrams, finite element simulations and simulation-based process and tooling optimization, have been introduced to facilitate the sheet-metal product development as mentioned in Sections 2.3 and 2.4.

4.2.2 Formability Analysis Based on FLDs

The Forming Limit Diagram (FLD) is a plot of the minor strain vs. major strain at each point measured on the deformed part using the circle grid analysis. The concept of forming limits was first introduced by Keeler (1965) and the standard form of the FLD was later presented by Goodwin (1968). The FLD has been a widely accepted criterion for fracture prediction in sheet metal forming.

FastForm3D (FTI, 2002) software based on an inverse FEM approach was used for formability analysis and blank estimation in this research. Because of its comparatively short computing time, this one-step simulation method has proven very efficient for optimization.

In the simulation result file of FastForm3D, major strain, minor strain, thickness, equivalent strain, and equivalent stress at each Gauss point of elements are listed (Hu, 2001). There are four Gauss points for a quadrilateral element and one for a triangular element (Vamanu, 2003). The strains of each sub-element or node are plotted onto an FLD.

The position of each point on the FLD can be used to determine what forming mode has occurred in an area of the part and whether the part is safe, or has areas with a tendency to split or wrinkle (FTI, 2002).

The Forming Limit Curve (FLC), as shown in Figure 4-2, indicates the forming limits of the material. The FLC is a material property curve dependent on the strain state (Marciniak and Duncan, 1992). To ensure the process is robust and able to tolerate small changes in material or process conditions, a 'safe' curve is defined by offsetting the FLC downward (by 10% for steel in this research). The marginal zone below the FLC means that elements are marginally safe in this zone. Strain levels below the band are considered safe.

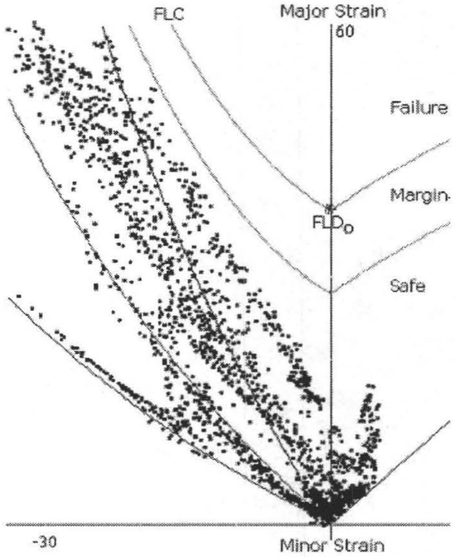


Figure 4-2. A typical forming limit diagram (engineering strains) (FTI, 2004)

Vamanu (2003) used the following formulae developed by Keeler and Brazier based on statistical data collected for deep drawing quality steels to calculate FLC for her research on FEM simulation based optimization. Her research has shown the formulae are acceptable. Keeler and Brazier's formulae are:

For $e_2 \geq 0$ (right side of the FLC):

$$e_1 = FLD_0 + e_2 * (0.784854 - 0.008565 * e_2) \tag{4.1}$$

For $e_2 < 0$ (left side of the FLC):

$$e_1 = FLD_0 + e_2 * (0.27254 * e_2 - 1.1965) \tag{4.2}$$

where e_1 and e_2 are major and minor percent engineering strains on FLC; FLD_0 represents the plane-strain point on a forming limit diagram and can be predicted from the Keeler-Brazier equation as shown below:

$$FLD_0 = (23.3 + 14.14 * t) * n / 0.21 \tag{4.3}$$

where t is thickness in *mm* and n is the strain hardening exponent taken above *10%* strain or at uniform elongation (Keeler, 2002; Vamanu, 2003).

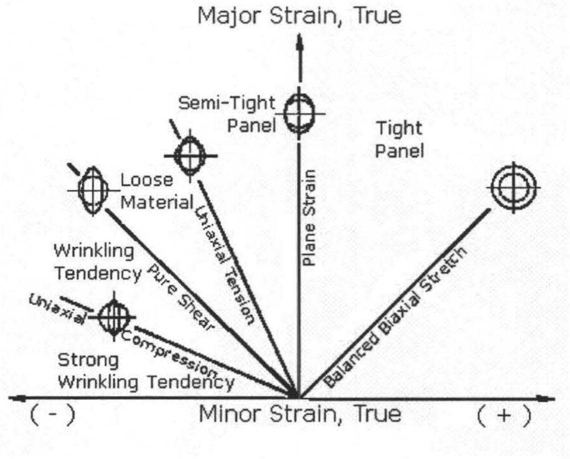


Figure 4-3. Modes of deformation (FTI, 2004)

Modes of deformation shown in Figure 4-3 describe the characteristics in several forming zones. Below the uniaxial compression line in Figure 4-3, high compression forces produce a strong tendency to wrinkle (FTI, 2004). The uniaxial compression line is set as the wrinkling criterion in this research.

The relationship between the true strain (ε) and engineering strain (*e*) is:

$$\varepsilon = \ln(1 + e/100) \tag{4.4}$$

For uniaxial compression:

$$\beta = \varepsilon_2 / \varepsilon_1 = -2 \tag{4.5}$$

where ε_2 is the minor true strain and ε_1 is the major true strain.

Based on

$$\varepsilon_2 = \ln(1 + e_2/100) \tag{4.6}$$

$$\varepsilon_1 = \ln(1 + e_1/100) \tag{4.7}$$

where e_2 is the minor engineering strain and e_1 is the major engineering strain.

From equations (4.5), (4.6) and (4.7), the following equation can be obtained:

$$e_1 = \frac{1000}{\sqrt{100 + e_2}} - 100 \tag{4.8}$$

The equation (4.8) represents the curve of uniaxial compression as shown in Figure 4-3. With a given minor engineering strain and equation (4.8), the critical major engineering strain can be calculated. If the real major engineering strain is lower than the critical major strain, wrinkling occurs. This is the assumption of the wrinkling criterion in this research.

4.2.3 Process and Tooling Optimization

Optimal process design may comprise the optimal number of stages, optimal blank geometry, optimal die configuration, and/or optimal process settings (e.g. blankholder force, position and restraining forces of drawbeads, friction, etc.). For draw dies, in particular, the key tooling and process design function involves determining the drawbeads and blankholder force to be used in the die. Two consequences follow from the optimization of these values. First, if values cannot be determined that will yield a part with acceptable formability in production, the product design is infeasible and this information can be communicated to the product designer for remedial action early in the design process. Second, once feasible values are obtained for these quantities, die design can progress in a highly automated fashion with high confidence the resulting die will work acceptably in production. Blankholder forces and drawbead restraining forces are most popular variables for optimal process design in literature and therefore were selected as optimization variables in this chapter. The simplex method, genetic algorithm and simulated annealing algorithm were adopted as optimization algorithms to be tested.

Typical objective functions that have been considered for process optimization in sheet metal forming are: a) raw material usage; b) deviation of thickness from uniform average; c) specific dimension(s) of the stamped part for assembly or process requirements. Design agents can select suitable objective function(s) based on technical requirements of the part.

4.3 Optimization Model

In this research, a deep drawn rectangular cup and an automobile front door panel were selected for case studies. For the purpose of performance studies of the prototype system, a popular objective function, deviation of thickness from uniform average, was used for case studies. Design variables are blankholder force and drawbead restraining forces. Constraints of this process optimization problem are categorized into thickness constraints and formability constraints. It is possible to add other constraints if desired for a particular application.

4.3.1 Constrained Optimization Model

The constrained process optimization problem is defined as:

Minimize

$$f(V)=\sqrt{\frac{\sum_{i=1}^{m}(t_i-\bar{t})^2}{m}}, \tag{4.9}$$

Subject to:

$$v_{il} \le v_i \le v_{iu},\ i=1,2,...,n, \tag{4.10}$$

$$t_{\max}-T_U \le 0 \tag{4.11}$$

$$T_L-t_{\min} \le 0 \tag{4.12}$$

$$N_W=0 \tag{4.13}$$

$$N_F=0 \tag{4.14}$$

$$N_M=0 \tag{4.15}$$

where:

$V=\langle v_1,v_2,...,v_n\rangle^T$ is the vector of design variables v_i;

m: total number of elements in the FEM mesh;

t_i: thickness of element i;

$\bar{t}$: average thickness of all elements;

v_{il}, v_{iu}: upper limit and lower limit of the i^{th} design variable, i.e. v_i;

n: total number of design variables;

$t_{\max}$: maximum element thickness;

$t_{\min}$: minimum element thickness;

T_U , T_L : upper limit and lower limit of element thickness;

N_W , N_F , N_M : number of elements in the wrinkling, fracture or marginal safety zone, as defined with Forming Limit Diagram analysis.

To apply the optimization model, a proposed forming operation is simulated via FEM. Element thickness values are read from the output file of the FE simulation, and used to compute the value of the objective function and constraints. This objective seeks to minimize the total absolute thickness deviation in the part, that is, to make the part thickness as consistent as possible across its surface. Inequality (4.10) sets practical limits on process variables, i.e. the non-negative blankholder force and drawbead restraining forces. Inequalities (4.11) and (4.12) set absolute limits on part thickness, and equations (4.13) to (4.14) are conditions to make parts with any predicted failures infeasible.

As mentioned before, minor and major strains, thickness, and the equivalent strain and stress of each Gauss point are listed in the output file of FastForm3D. The element thickness and major/minor strains of each element can be calculated approximately as the average thickness and major/minor strains of four Gauss point for a quadrilateral element or one Gauss point for a triangular element. N_W , N_F and N_M are calculated based on the formulae for calculating FLC and wrinkling criterion in Section 4.2.2. The disadvantage of this approach is that N_W , N_F and N_M may deviate from the true situation. The advantage is that optimization results are easily compared with experimental results since they are all based on elements, not nodes or Gauss points.

Since the output results are based on Gauss points in the output file created by FastForm3D, the objective function and constraints can also be calculated based on Gauss points. This means there is no need to calculate the average of thickness and major/minor strains. The advantage of using this approach is that the output results of FastForm3D can be directly used to calculate N_W, N_F and N_M. Major and minor strains of Gauss points are plotted on the FLD to determine N_W, N_F and N_M. Therefore, better accuracy can be achieved compared with the approach based on average values. If a fine mesh density is applied to this model, there should be no big difference between these two approaches. However, for a coarse model, the approach based on average values may sacrifice considerable accuracy.

Both methods have been tested in case studies in Section 4.5.

4.3.2 Penalty Function Method

The constrained optimization problem can be transformed to an unconstrained optimization problem via a static penalty function method (Homaiffar et al., 1994; Kuri-Morales and Gutiérrez-Garcia, 2001).

For example, the general constrained optimization problem is shown as follows:

Minimize $f(v)$

Subject to $h_i(v) = 0 \quad i=1,..,p$ (4.16)

$g_i(v) <= 0 \quad i=p+1,...,j$ (4.17)

The unconstrained optimization problem generated using the static penalty function is as follows:

Minimize $$F(V) = \begin{cases} f(V) & V \in \textit{feasible region} \\ f(V) + \textit{penalty}(V) & V \notin \textit{feasible region} \end{cases} \tag{4.18}$$

where

$$penalty(V) = \sum_{i=1}^{j} R_{ik} \cdot P_i^2(V) \tag{4.19}$$

$$P_i(V) = \begin{cases} |h_i(V)|, & i = 1,\ldots,p \\ Maximum(0, g_i(v)), & i = p+1,\ldots,j \end{cases} \tag{4.20}$$

j is the total number of constraints.

p is the number of equality constraints.

Coefficients, R_{ik}, are set based on 3 penalty levels in this research, i.e., $k = 3$.

Ranges of design variables, i.e. the blankholder force and drawbead restraining forces, can be determined based on empirical methods or a random search method. Checking the constraints of design variables doesn't need FE simulation results. However, thickness and formability constraints, i.e. inequalities (4.11) to (4.12) and equations (4.13) to (4.15), can only be checked when the FE simulation results are available. Considering practicability and optimization efficiency, the original constrained optimization model was transformed to a new constrained optimization model with only constraints of design variables as shown in inequality (4.10). Since the ranges of design variables determined by empirical methods are usually large and safe enough, this approach can avoid wasting time running FE simulations with unsuitable design variables, and meanwhile it is not difficult to deal with the design variable ranges in optimization algorithms used for this research.

4.4 Proposed Framework for Agent-Based Optimization

Due to the complexity of the sheet metal forming process, it is inefficient, if not impossible, to apply one optimization algorithm to all problems. The downhill simplex method (SM), simulated annealing (SA) and genetic algorithm (GA) are selected as the principal optimization algorithms to solve the optimization problem presented in Section 4.3. Detailed algorithms and advantages and disadvantages of these optimization methods are discussed in the following sections.

4.4.1 Optimization Algorithms

Downhill Simplex Method

The downhill simplex search method developed by Nelder and Mead (1965) requires only function evaluations, not derivatives. Computationally, it is relatively uncomplicated, easy to implement and quick to debug. On the other hand, it is slower to converge than the derivative-based methods where such methods are suitable.

The geometric figure formed by a set of $n + 1$ points in an n-dimensional space is called a simplex. In the simplex method, the basic idea is to compare the values of the objective function at the $n + 1$ vertices of a general simplex and move the simplex gradually toward the optimum point during the iterative process. The movement of the simplex is achieved by using three operations: reflection, contraction and expansion.

The traditional downhill simplex method was designed to solve unconstrained optimization problems. To solve the constrained optimization problems with constraints on design variables, modifications have to be made to the traditional downhill simplex method. Figure 4-4 shows the flowchart of an improved downhill simplex algorithm that is used for this research.

The initial simplex can be generated based on a starting point. If there is a starting point P_0, use P_0 as one of those vertices and generate other n points through

$P_i = P_0 + \lambda_i e_i,\ i=1,2,..,n$

where λ_i are constants that characterize the length scale for each vector direction, and e_i are n unit vectors. If P_i falls outside the variable area, re-calculate P_i through reducing by half the value of λ_i.

The optimization process can be terminated when one of the three termination criteria is satisfied:

- The difference among objective values of vertices of the current simplex is less than a predefined small threshold value (*ds_obj*).
- The length between every two vertices of the current simplex is less than a predefined threshold value (*ds_len*).
- The maximum number of iteration cycles, *maxIter*, has been exceeded.

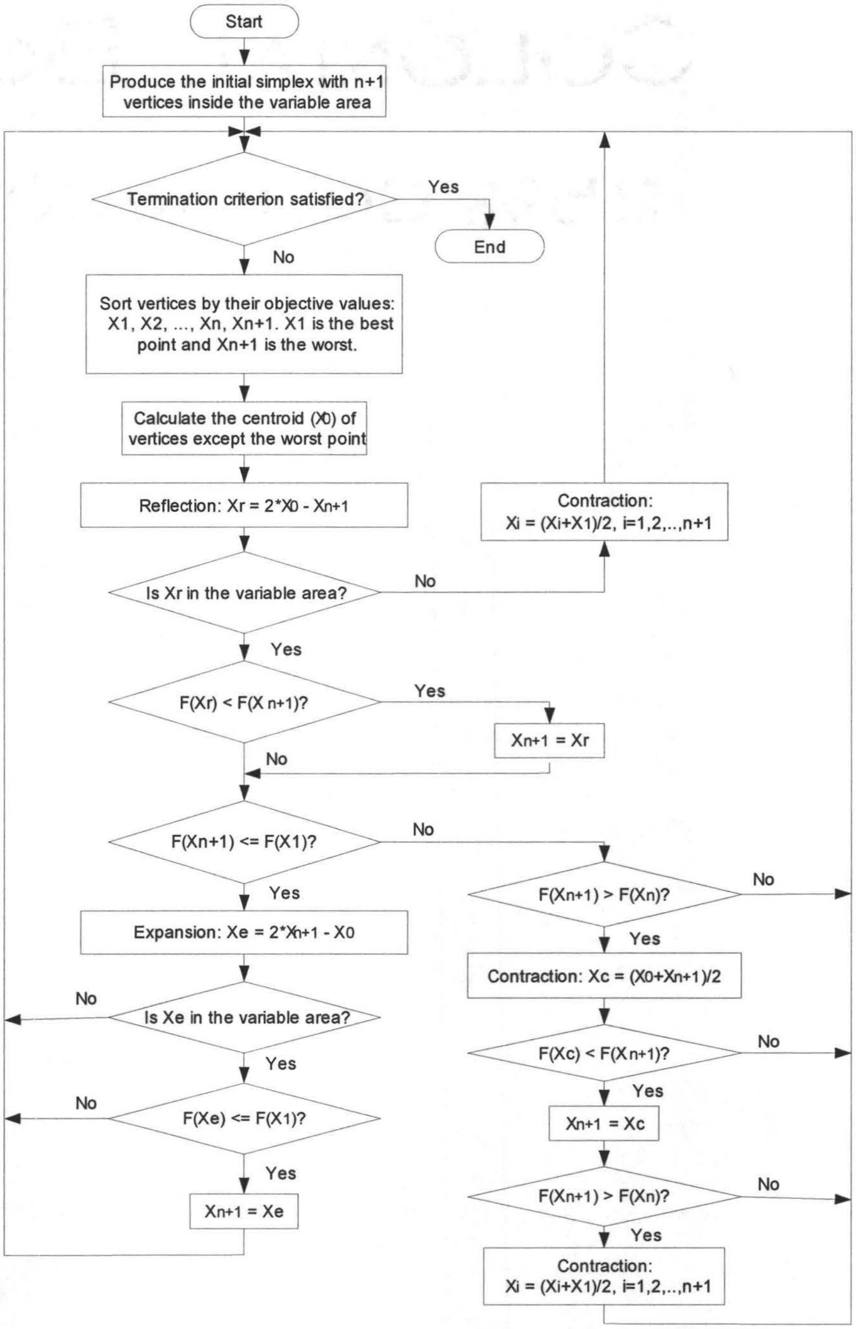


Figure 4-4: Flowchart of the Simplex Method to solve constrained optimization problems

Simulated Annealing

Simulated annealing (Metropolis et al., 1953) is a powerful stochastic computational technique. Its concept is based on the manner in which liquids freeze or metals recrystalize in the process of annealing. Its major advantage over other methods is an ability to avoid becoming trapped at local minima. Simulated annealing has been used in various combinatorial optimization problems (Kirkpatrick et al., 1983; Jain et al., 1992).

The detailed algorithm used in this research is presented in Figure 4-5. Initial control parameter C_0 and cooling parameter k are carefully selected to make sure that there is a high acceptance probability that a new point with a higher value of the objective function can be accepted at the initial stage and the acceptance probability decreases gradually. There is no general way to find the best choices for a given problem. The selection of the number of moves per step, *Nmove*, is based on the number of variables. *Nmove* has to be large enough to make sure that each variable can have a possibility to be updated. *Nstep* is set as the termination criterion for the simulated annealing algorithm since a near optimal solution generated after a certain number of steps is usually good enough for most engineering design problems.

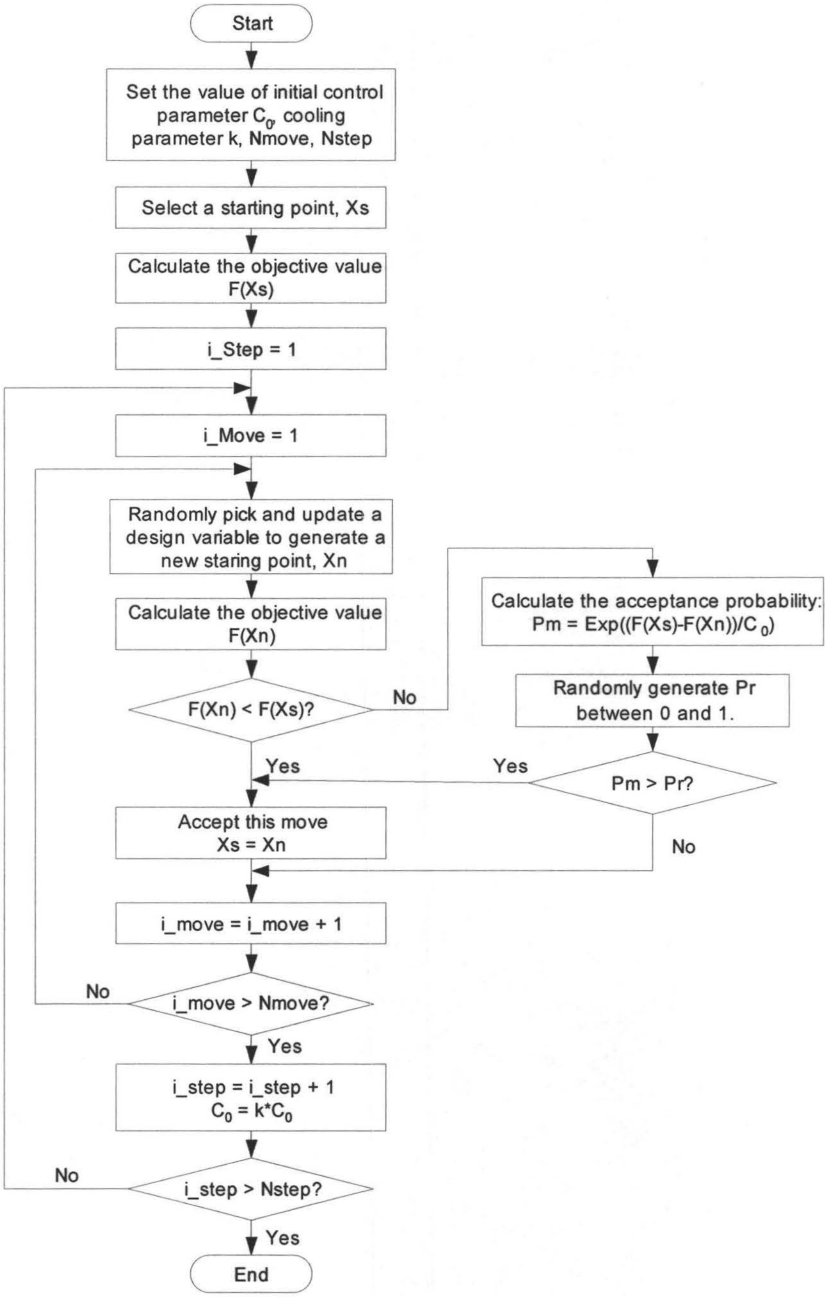


Figure 4-5. Flowchart of the Simulated Annealing algorithm

Genetic Algorithm

GAs, originally developed by Holland (1975), are a search technique that utilizes the concept of natural selection. Aytug et al (2003) summarized that "GAs have eight basic components: genetic representation, initial population, evaluation function, reproduction selection scheme, genetic operators, generational selection scheme, stopping criteria and GA parameter settings".

GAs typically work with a coding of design variables, instead of design variable themselves. To implement a GA, the way in which candidate solutions are encoded is one of the most important factors in the success of a GA (Mitchell, 1996). Binary encodings (i.e., bit strings) are the most common encodings because much of the existing GA theory is based on the assumption of fixed-length, fixed-order binary encodings and there has been a theoretical justification for using binary encodings (Holland, 1975; Mitchell, 1996). In this research, binary encoding, the most widely used way of encoding, is applied.

For example, suppose there are 2 design variables, each encoded to 4-bit resolution. Each possible solution, a combination of binary strings representing both design variables, can then be represented by an 8-bit string, 4 bits for each design variable. Each 4-bit string for a design variable is termed a gene, and then the 8-bit string, made up of the 2 genes, is termed a chromosome.

To determine the string length l for the i^{th} variable in $v_{il} \le v_i \le v_{iu}$, the following inequality has to be satisfied:

$$2^l - 1 \geq \frac{v_{iu} - v_{il}}{\Delta_i} \tag{4-21}$$

where Δ_i is the increment of i^{th} design variable.

The actual value of the i^{th} design variable can be calculated as follows:

$$v_i = \frac{v_{iu} - v_{il}}{2^l - 1}\left[\sum_{j=1}^{l}(B_j)2^{j-1}\right] + v_{il} \tag{4-22}$$

where B_j is the j^{th} bit of the bit string which represents the i^{th} design variable. B_j equals 0 or 1.

The flowchart of the Genetic Algorithm used in this research is shown in Figure 4-6. Fitness is the value assigned to each individual. It is used to sort the array of solutions, compare, and select individual solutions for further evaluation. In this research, the fitness value of an individual is set as the reciprocal of the objective function value of the respective individual. Default weights of both mutation and crossover operators are 1.0. The operator selected will be applied to one or two chromosomes randomly selected from the current population excluding those worst chromosomes that have been eliminated. Tournament selection runs a "tournament" among a number of individuals chosen at random from a designated “population” and selects the best individual. The tournament size in this research is 2.

The most frequently used termination criterion for the Genetic Algorithm is a specified maximum number of generations. The optimization can also be stopped if a desired fitness value is reached or the fitness value of the best individual of the population does not improve after certain number of generations. The strategy used in this study is the maximum number of generations.

GAs work with only function values. There is no need to compute the derivative of the function. GAs can be used to search a wide design space and are less susceptible to becoming trapped at local optima than gradient-based methods. A disadvantage is that they may need many generations to find their solution, if one is found at all. In other words, GAs are computationally expensive. GAs may not converge on the best possible solution if there is not enough randomness in the selection and crossover methods.

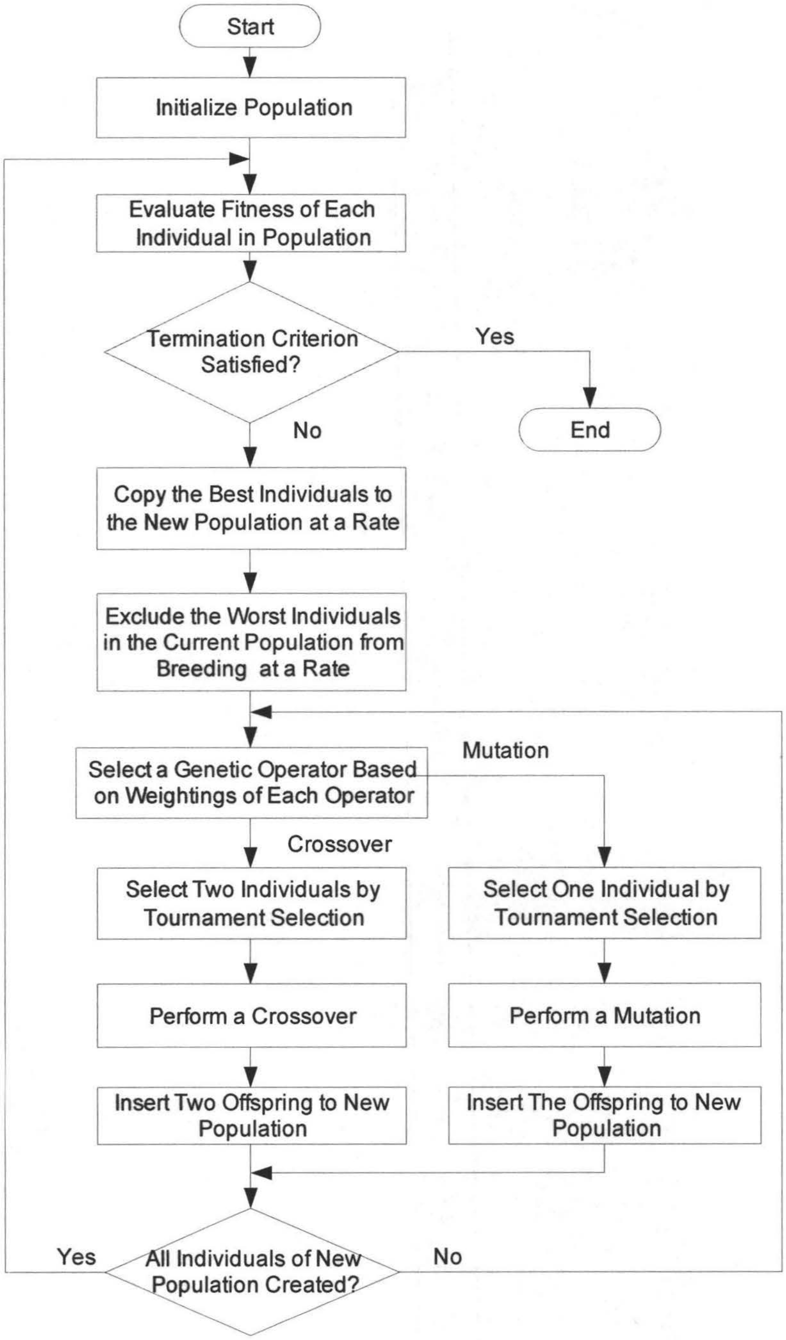


Figure 4-6. Flowchart of the simple Genetic Algorithm

4.4.2 Cyclic Data Flow

Agents equipped with a variety of optimization algorithms can collaborate on a specific optimization task. Considering the advantages and disadvantages of SM, SA and GA, a framework based on an A-Teams approach (Talukdar et al., 1996) has been proposed for agent-based process and tooling optimization by combining agent-based technology and traditional optimization algorithms.

The principle of A-Teams is to solve complex problems with a strongly cyclic computational network of shared memories and agents. Distributed autonomous agents cooperate in parallel by working on one another's results.

4.4.3 System Architecture

Figure 4-7 shows the structure of the proposed agent-based process optimization system. Agents in Figure 4-7 can be seen as service agents in the supplier agent community as shown in Figure 3-1. These agents provide optimization services that are made available to task agents in the supplier agent community. Optimization agents include optimization management (OM) agents, data generating (DG) agents, data removal (DR) agents and different optimization algorithm agents, i.e. downhill simplex method (SM) agents, simulated annealing (SA) agents and genetic algorithm (GA) agents. By separating optimization algorithms and tasks to different agents, multiple agents may be easily distributed across a company's intranet to utilize available computing resources.

Agents encapsulating different optimization algorithms cooperate in parallel by reading data from and posting data to a common shared memory. The database management system used is MySQL (MySQL, 2003). Agents communicate with MySQL server using the APIs for Java or C++.

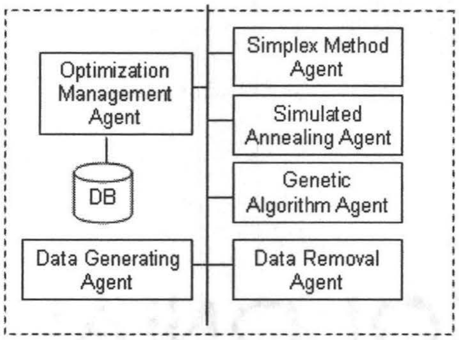


Figure 4-7. Organization of optimization service agents

4.4.4 Agent Coordination

Agents collaborate on optimization tasks by getting initial points from and posting optimization results to the shared database. Optimization results are sent to the optimization management agent and then are saved to the shared database. The results generated by one agent can be used by another agent as initial points. The selection of initial points plays a very important role in the optimization work of the simplex method, simulated annealing and genetic algorithms.

Data Generating Agent

The data generating agent randomly generates a number of initial points in the beginning of the optimization and gets a notification from the optimization management agent when there is any change in the shared database. Based on investigation of the explored space of existing records in the shared database, new initial points are generated to fill the unexplored space. The ranges of design variables are determined based on empirical methods. Grids are made to search the unexplored space of design variables. If a grid is found to be unexplored, the center of the grid will be added to the shared database as a new initial point.

Data Removal Agent

The data removal agent removes the worst initial points in the shared database if a new initial point is added. It has been created but has been disabled in this research because there are usually a small number of points existing in the shared database. Research on the impact of data removal agent is recommended for future work.

Genetic Algorithm Agent

The genetic algorithm agent selects some percentage of best points that have never been used by other GA agents as initial points and then selects the rest of its initial points from the best 90% points in the shared database according to a tournament selection scheme.

All individuals in the final population are sent back to the shared database and then can be used by other optimization agents as initial points.

Simulated Annealing Agent

The SA agent selects an initial point that has never been used by other SA agents from the best 90% points in the shared database according to the tournament selection rule.

In the following Case Study 1, only the final result of simulated annealing is sent back to the shared database. In Case Studies 2 and 3, the intermediate acceptable moves are also sent to the shared database to ensure there are more good initial points available for complex optimization problems.

Simplex Method Agent

The SM agent always uses the best initial point that has never been used by other SM agents.

4.5 Case Study 1: Rectangular Cup, Two Drawbeads

4.5.1 Case Description

A deep drawn rectangular cup shown in Figure 4-8 was selected for case studies. The punch size is 200 *mm* x 100 *mm* and corner radii of the punch are 10 *mm*. Only a quarter of the rectangular cup was modeled because of the symmetry of the part and the anisotropy of the material.

The material was modeled using a power law relation ($\sigma = K\varepsilon^n$). The material strength coefficient (*K*), the strain hardening exponent (*n*) and other material properties are given in Table 4-1. The coefficient of anisotropy, *r-value*, of the material describes the ability to resist thickness change.

In FastForm3D, the blankholder force boundary condition is applied to elements along the edge of the model. The specified blankholder force is translated into a tension, applied normal to the edge of the model. In the following sections, *BHF* represents the edge tension of the blankholder force and the unit of *BHF* is *N/mm*.

This case has 3 design variables, namely, the edge tension of the blankholder force and two drawbead restraining forces.

Two drawbeads were set as shown in Figure 4-8. In FastForm3D, drawbead restraining forces along the drawbead line independent of the element mesh are used for setting forming conditions without using the real geometry of drawbeads.

In this case study, the objective function and constraints are calculated based on average values of thickness and minor/major strains of Gauss point(s).

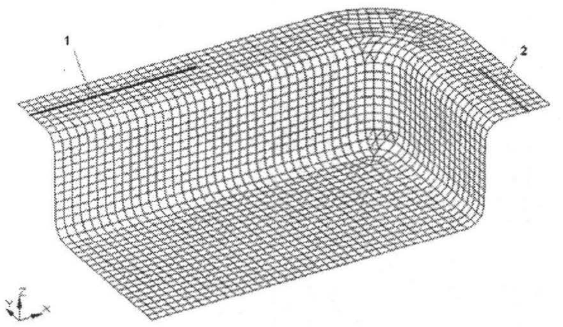


Figure 4-8. Case 1 with 2 drawbeads

Table 4-1. Material properties

Thickness	0.7 *mm*
n	0.2
r-Value	1.6
K	534.48 *MPa*
Yield Stress	186.16 *MPa*
Density	7.84×10^{-6} *kg/mm*^{3}
E	203 *GPa*
Poisson Ratio	0.3

A mesh convergence study for the model was conducted to select optimum mesh density. Only the objective function value in Equation (4-9) is calculated as the evaluation target. Only blankholder force was applied to the case studies for the mesh convergence study. The edge tension of the blankholder force boundary condition was set to *35 N/mm*.

Table 4-2 shows the relationship between element numbers and the deviation of thickness from uniform average as described in Equation (4.9). The deviation of thickness was calculated based on both the average element thickness and the thickness at each gauss point. It is assumed that the mesh density converges when the difference between deviations of thickness is less than *2%*. In Table 4-2, we can find that the difference between deviations of the thickness is less than *2%* when the total number of elements is equal or greater than *1887*. To exploit the symmetry of the part and save the computational time, a quarter of the rectangular cup was meshed with *1887* elements for the remainder of this case study.

Table 4-2. Results of the mesh convergence study

Number of Elements	Deviation of Thickness based on average element thickness ($\times 10^{-2}\ mm$)	Deviation of Thickness based on thickness of each gauss point ($\times 10^{-2}\ mm$)
839	2.836	2.676
1887	2.684	2.627
3033	2.671	2.623
5200	2.643	2.593

The outline of the blank is automatically calculated in FastForm3D. Four tests with process parameters shown in Table 4-3 were run to study the difference among blank outlines. From Figure 4-9, we can find that the maximum differences along X and

Y directions are *1.52 mm* and *3.04 mm*. Compared with the lengths along X and Y directions, *282 mm* and *140 mm*, the differences are ignorable. To simplify the case studies, the outline of the blank was not fixed. In other words, blank outlines may vary from each run.

Table 4-3. Parameters of tests for estimating the blank outlines

Test No.	Edge Tension of the Blankholder Force Boundary Condtion (*N/mm*)	Restraining Force of Drawbead #1 (*N/mm*)	Restraining Force of Drawbead #2 (*N/mm*)
1	0	0	0
2	35	0	0
3	35	13	0
4	25	25	10

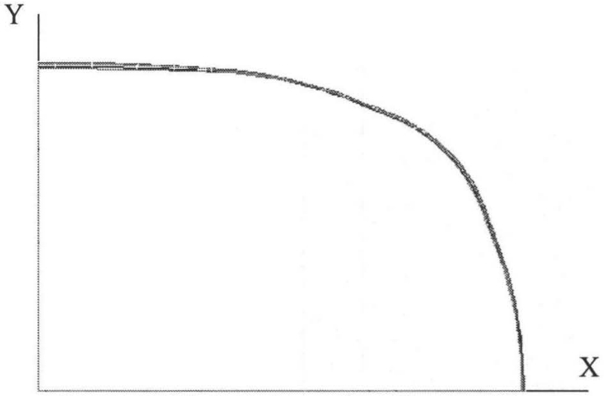


Figure 4-9. Difference among blank outlines

Upper and lower limits of element thickness in Case Study 1 are set to 120% and 77% of the material thickness, respectively. Penalty coefficients used for other constraints are shown in Table 4-4.

Table 4-4. Penalty coefficients for case study 1

Penalty Coefficient	Value	Condition	Constraint No
R1	$0.0394\ mm^{-1}$ $(1\ inch^{-1})$	$0.00254\ mm < (t_{\max} - T_U) < 0.0508\ mm$	(4-11)
	$0.0787\ mm^{-1}$ $(2\ inch^{-1})$	$0.0508\ mm \le (t_{\max} - T_U) < 0.1016\ mm$	
	$0.1575\ mm^{-1}$ $(4\ inch^{-1})$	$0.1016\ mm \le (t_{\max} - T_U)$	
R2	$0.0394\ mm^{-1}$ $(1\ inch^{-1})$	$0.00254\ mm < (T_L - t_{\min}) \le 0.0508\ mm$	(4-12)
	$0.0787\ mm^{-1}$ $(2\ inch^{-1})$	$0.0508\ mm < (T_L - t_{\min}) \le 0.1016\ mm$	
	$0.1575\ mm^{-1}$ $(4\ inch^{-1})$	$0.1016\ mm < (T_L - t_{\min})$	
R3	$0.0127\ mm$ $(0.0005\ inch)$	$0 < N_W < 10$	(4-13)
	$0.0254\ mm$ $(0.001\ inch)$	$10 \le N_W < 30$	
	$0.0508\ mm$ $(0.002\ inch)$	$30 \le N_W$	
R4	$12.7\ mm$ $(0.5\ inch)$	$0 < N_F < 5$	(4-14)
	$25.4\ mm$ $(1\ inch)$	$5 \le N_F < 20$	
	$50.8\ mm$ $(2\ inch)$	$20 \le N_F$	
R5	$0.254\ mm$ $(0.01\ inch)$	$0 < N_M < 10$	(4-15)
	$0.508\ mm$ $(0.02\ inch)$	$10 \le N_M < 40$	
	$1.27\ mm$ $(0.05\ inch)$	$40 \le N_M$	

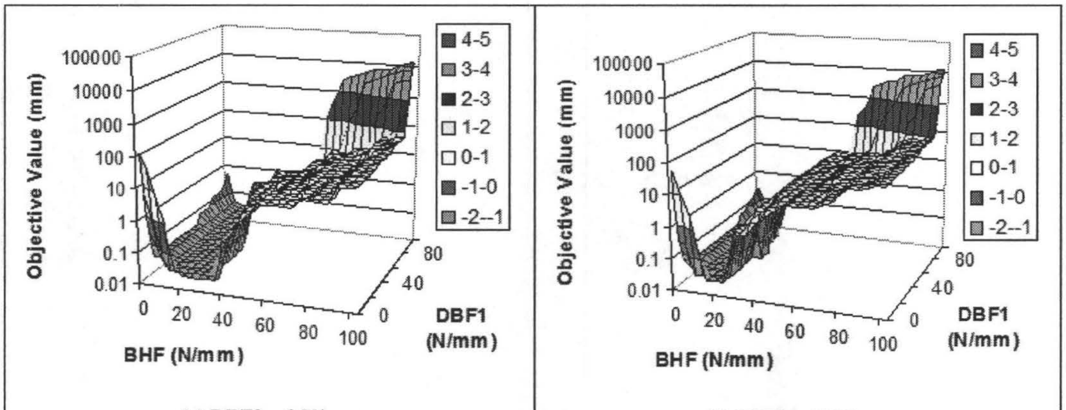
4.5.2 Optimization Trend

To create a baseline measurement for later comparison of the performance of the optimization agents, the shape of the objective function for Case 1 was estimated with a grid search method. The grid search method is computationally expensive if it is applied to a multivariable optimization problem because the model must be evaluated at many points within the grid for each variable. In this case, three design variables are involved. Considering the practicability, the edge tension of blankholder and restraining forces of drawbeads were all set to range between *0* and *100 N/mm*. With a grid size of *4* (*N/mm*), $26 \times 26 \times 26$ simulations were run for this case.

Figure 4-10 presents the trend of the objective function of Case 1. *BHF* represents the edge force of blankholder. *DBF1* and *DBF2* represent drawbead restraining forces of drawbead 1 and drawbead 2, as shown in Figure 4-8.

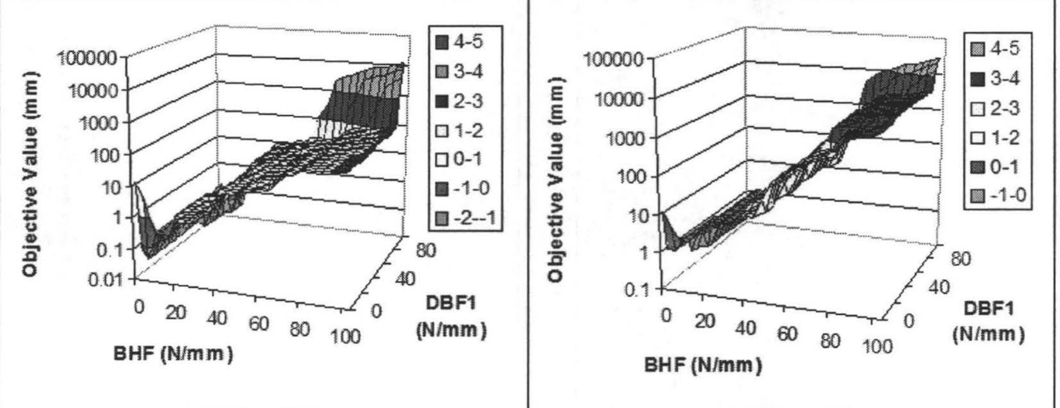
Figure 4-10(a) shows that optimal *BHF* is between *10* and *40N/mm* when *DBF2* equals zero. The optimum area gets narrow when *DBF2* increases as shown in Figures 4-10(b) to (d). Compared with Figures 4-10(a) and (b), objective function values in Figures 4-10(c) and (d) are much larger. Optimal *DBF2* is between *0* and *64 N/mm*. Figures 4-10(e), (f) and Figures 4-10(g), (h) show that optimal *BHF* should be between *20 N/mm* and *40 N/mm* and optimal *DBF1* is less than *40 N/mm*.

Observing Figure 4-10, we can find that optimum areas of the objective function are reasonably “smooth” although there are large “jumps” in other areas due to the usage of the static penalty function.



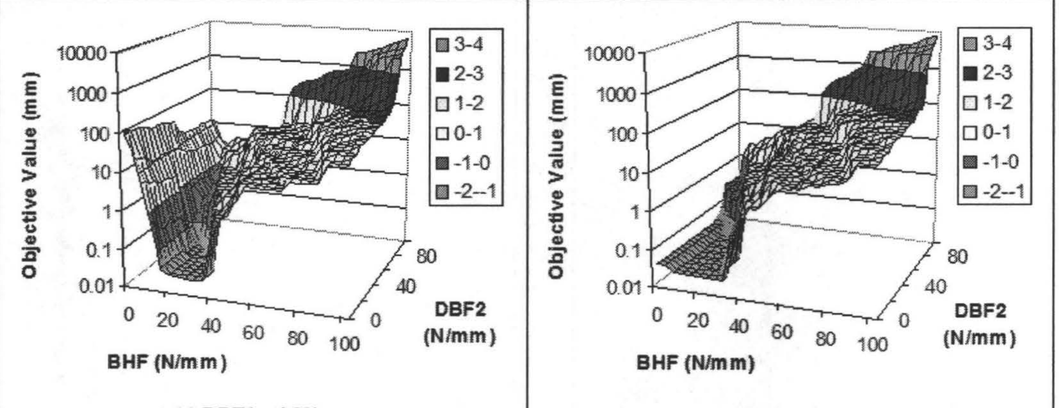
(a) DBF2 = 0 N/mm

(b) DBF2 = 32 N/mm



(c) DBF2 = 64 N/mm

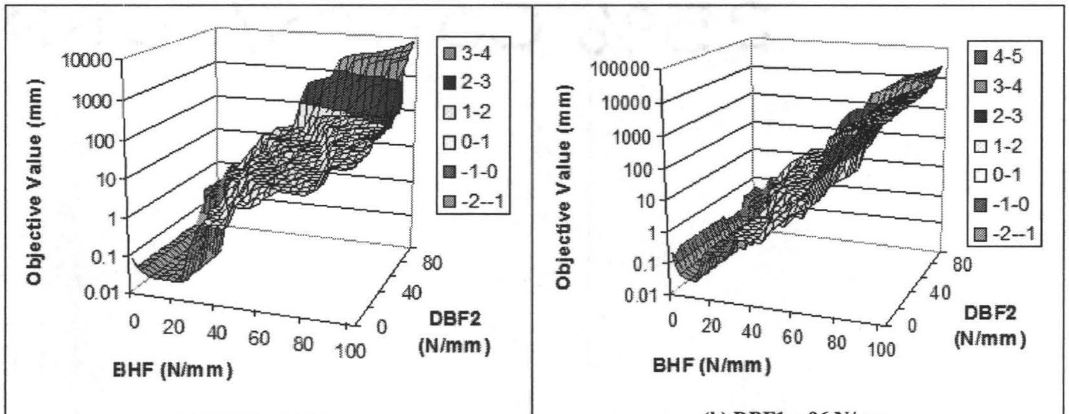
(d) DBF2 = 96 N/mm



(e) DBF1 = 0 N/mm

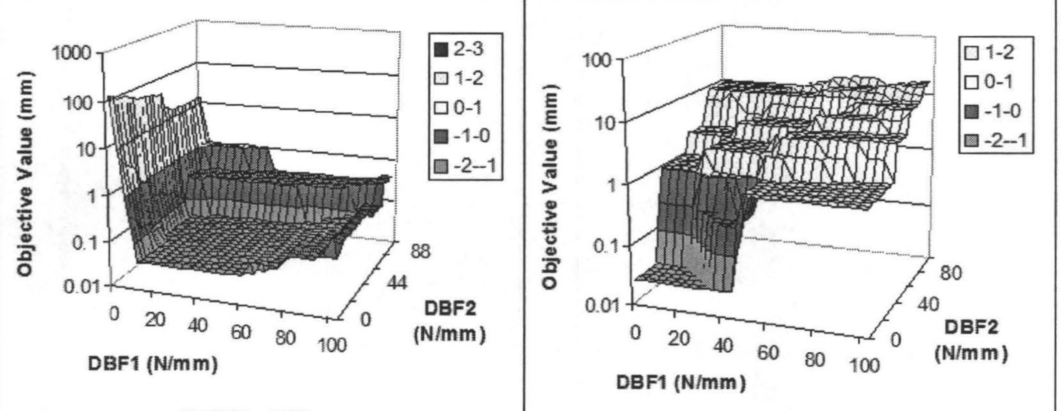
(f) DBF1 = 32 N/mm

Figure 4-10. The trend of the objective function for Case 1



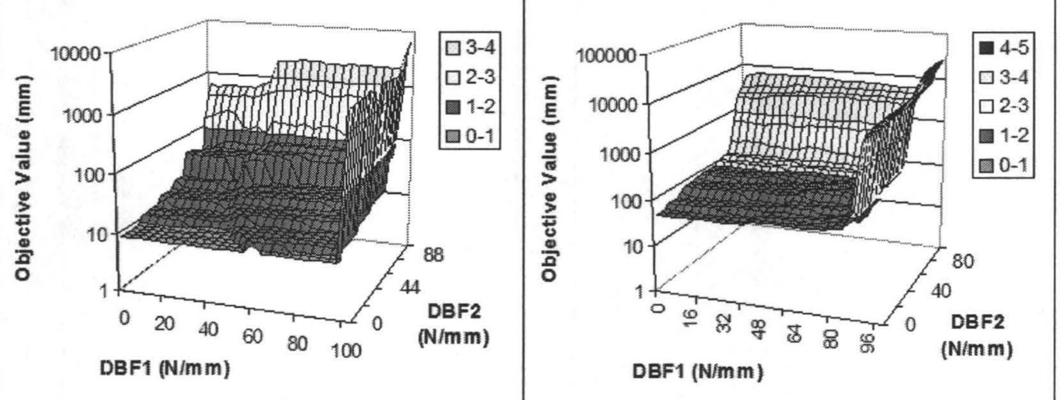
(g) DBF1 = 64 N/mm

(h) DBF1 = 96 N/mm



(i) BHF = 0 N/mm

(j) BHF = 32 N/mm



(k) BHF = 64 N/mm

(l) BHF = 96 N/mm

Figure 4-10. The trend of the objective function for Case 1 (Cont.)

4.5.3 Parameter Selection of Optimization Algorithms

Several tests have been conducted for parameter selection of three optimization algorithms used in this case study.

Simplex Method

To solve the optimization problem of Case 1 with 3 design variables, a simplex consists of 4 vertices. A test with a starting point of (40, 20, 5) was conducted. Figure 4-11 shows the distribution of the objective function value at each of the 4 vertices of the simplex in each iteration. Parameters for the termination criteria are set as follows:

- $ds\_obj = 2.54 \times 10^{-6}\ mm\ (1 \times 10^{-7}\ inch)$
- $ds\_len = 0.1\ N/mm$
- $maxIter = 300$

From Figure 4-11, we can find that the range of objective values of 4 vertices of the simplex decreases significantly as the number of iterations increases, and the optimization process converges after 18 iterations. Several tests with different starting points have been conducted and the results consistently showed that the parameters for termination criteria are suitable.

Simulated Annealing

Determination of the initial control parameter C_0 and cooling parameter k is done by a trial-and-error process. The rule to determine C_0 and k is to make sure that worse points (with higher objective value than that of the starting point) can be accepted with a high possibility in the first steps. The acceptance possibility decreases as the number of steps increases.

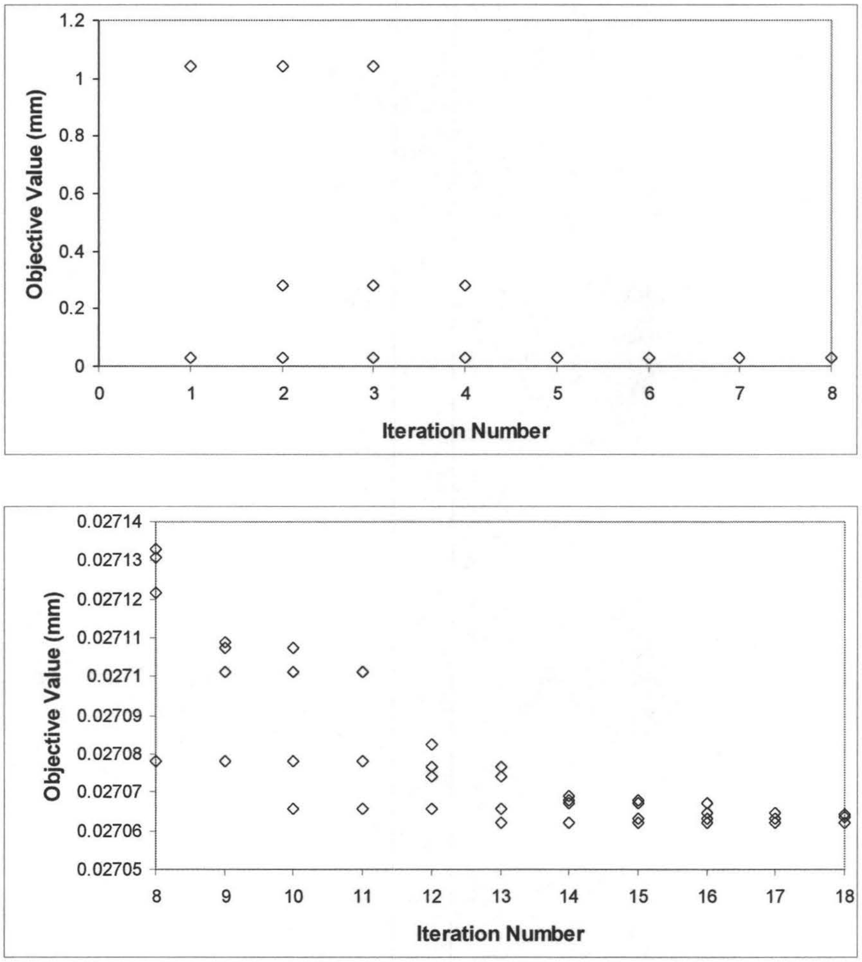


Figure 4-11. Convergence trend of an application of the Simplex Method algorithm

Since Case 1 is a 3-dimensional optimization problem, the number of moves per step was set to 20 to guarantee there are enough chances for each variable to be selected and updated to generate a new starting point. Figure 4-12 shows the convergence trend of the simulated annealing algorithm as shown in Figure 4-5 with the following settings:

$C_0 = 5.0 \times 10^{-6}$; $k = 0.40$; $Nmove = 20$

The starting point for this test is (25, 15, 5).

As seen in Figure 4-12, objective values of points from Step 3 to Step 6 fall between 0.02697 and 0.02713 *mm*. Objective values of points from Step 7 to Step 10 fall between 0.02697 and 0.02699 *mm*. If termination requires a small change of the objective function value in successive iterations and pre-specified threshold value is set as 2.54×10^{-6} *mm* (1×10^{-7} *inch*), the algorithm converges after Step 10. In this research, the maximum number of steps (*Nstep*) was set as the termination criterion for simulated annealing. To reduce the computation cost, *Nstep* is set as a small value of 6 in this case.

Genetic Algorithm

In all case studies presented in this Chapter, mutation rate of GAs was set as 10%. The percentage of population carried forward unchanged from each generation was set as 5% and percentage of population deemed unfit for reproduction and eliminated from the parent generation was set as 40%.

In this 2-drawbead case, population size, i.e. number of chromosomes per generation, was set as 20 and the size of genes was 16.

Figure 4-13 shows the distribution of objective values of a sample test. The initial population of chromosomes was constructed at random. From Figure 4-13, we can find that the best fitness value (i.e., the highest value in Figure 4-13(a)) or the lowest objective value in Figure 4-13(b) doesn't change too much after generation 9. The maximum number of generations was set as 20 for Case 1.

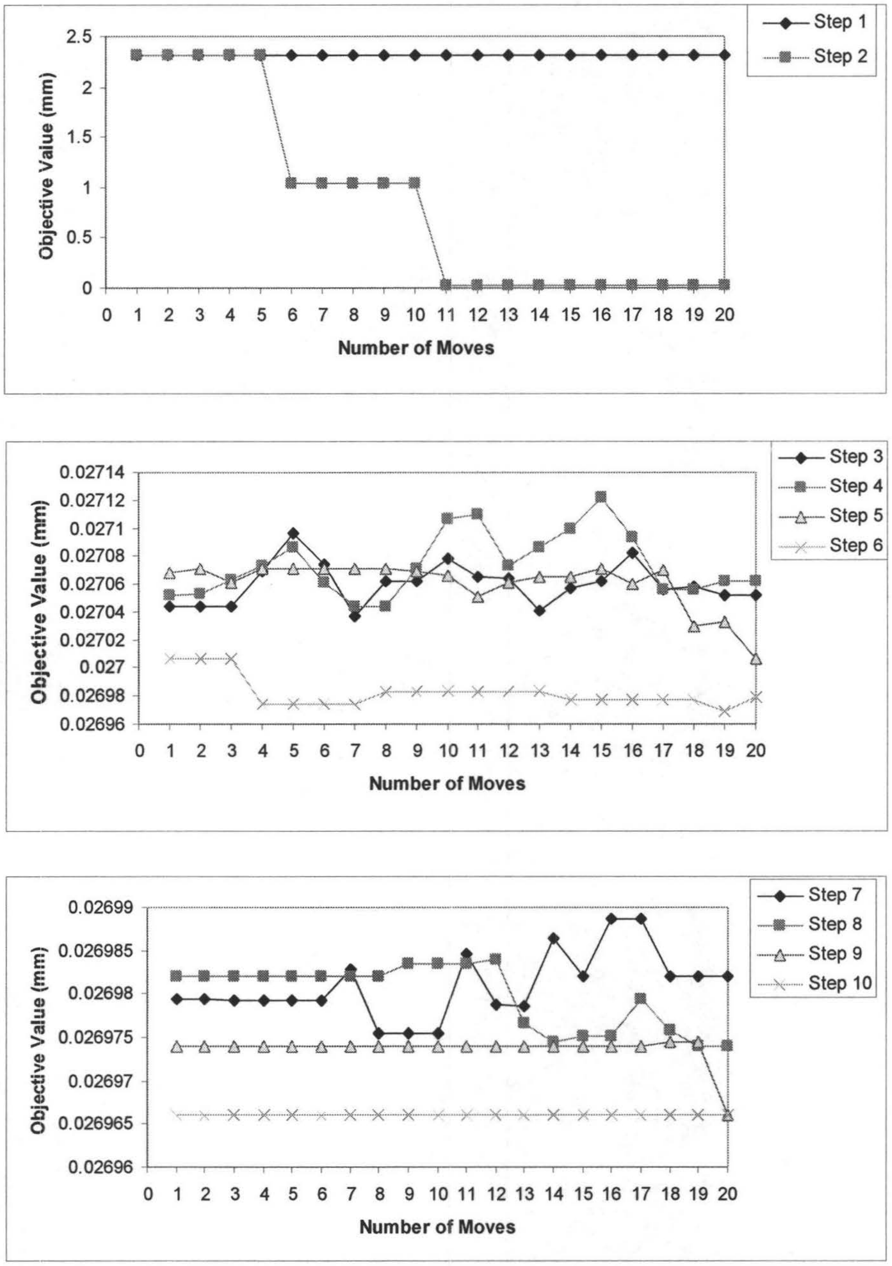
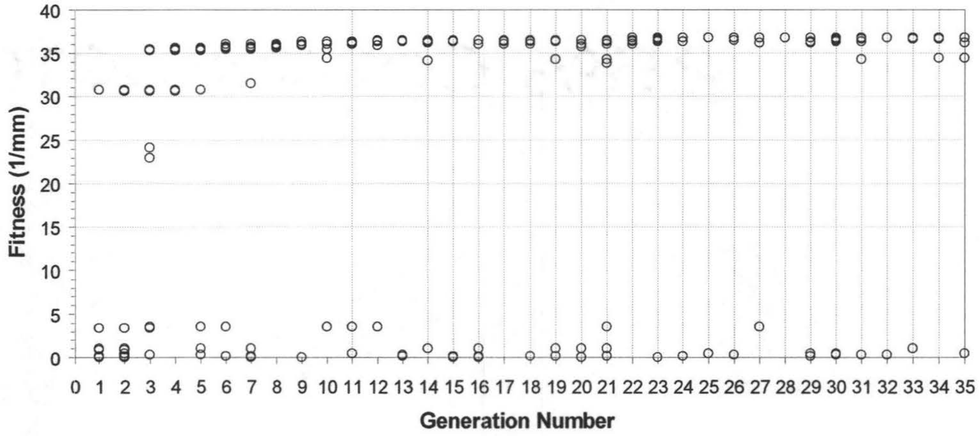
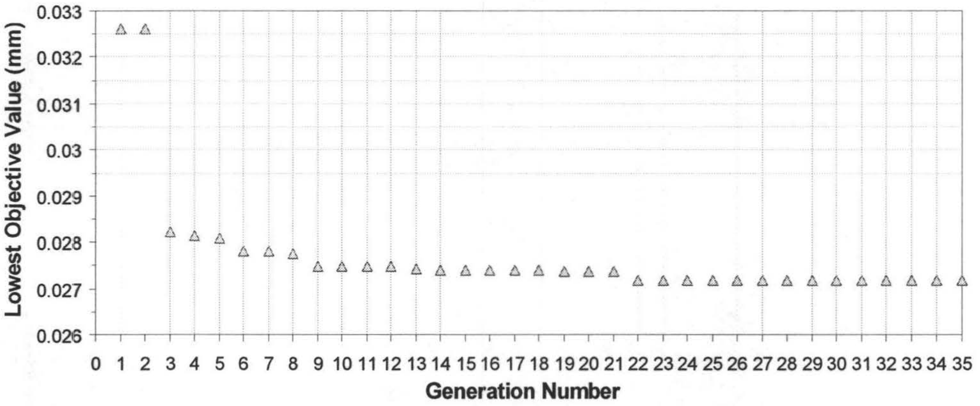


Figure 4-12. Convergence trend of an application of the Simulated Annealing algorithm



(a) Fitness Value vs. Generation Number



(b) Lowest Objective Value vs. Generation Number

Figure 4-13. Convergence trend of an application of the Genetic Algorithm

4.5.4 Result Analysis

The relative number of operator agents was set at 1:1:1:1:1 (simplex method, simulated annealing, genetic algorithm, data generating, data removal) for this case study.

Distributing more optimization agents across a network would likely occur in practice, and study of this is an avenue for further research.

In this case study, the structure of the shared database was defined as shown in Table 4-5. Results generated by SA, SM, GA and DG agents are saved by recording the agent names of senders, design variables, objective values, flags and *N*s. The field of "*flag*" records the history of being selected as a starting point by other agents. "*N*" represents the number of same or very close points generated by the same type of agents. With those data, collaboration among agents and contribution of each type of agents can be studied easily.

Table 4-5. Structure of the shared database

Field	Type	Key	Extra
Id	int(11)	PRI	auto_increment
sender	char(2)		
parm1	double		
parm2	double		
parm3	double		
obj	double		
flag	smallint(6)		
N	smallint(6)		

In the agent collaboration case, the optimization process is triggered by an OM agent once a task of process optimization is sent to the optimization agent by a die design agent. Then a DG agent generates 40 starting points upon the request of the OM agent and passes the results to the OM agent, which saves the results into its database. The OM agent then picks starting points based on respective selection rules for SA, GA and SM agents. SA, GA or SM agents never pick a starting point created by the same kind of agents. SA, GA and SM agents send their results back to the OM agent and new starting points are then sent back to them by the OM agent. Only the final solution, one point, is sent to the OM agent by an SA or SM agent and one starting point is then sent back to the respective SA or SM agent by the OM agent. The final population of the GA agent is sent back to the OM agent and a population of starting points is then sent back to the GA agent by the OM agent. In this case study, a population consists of 20 individuals.

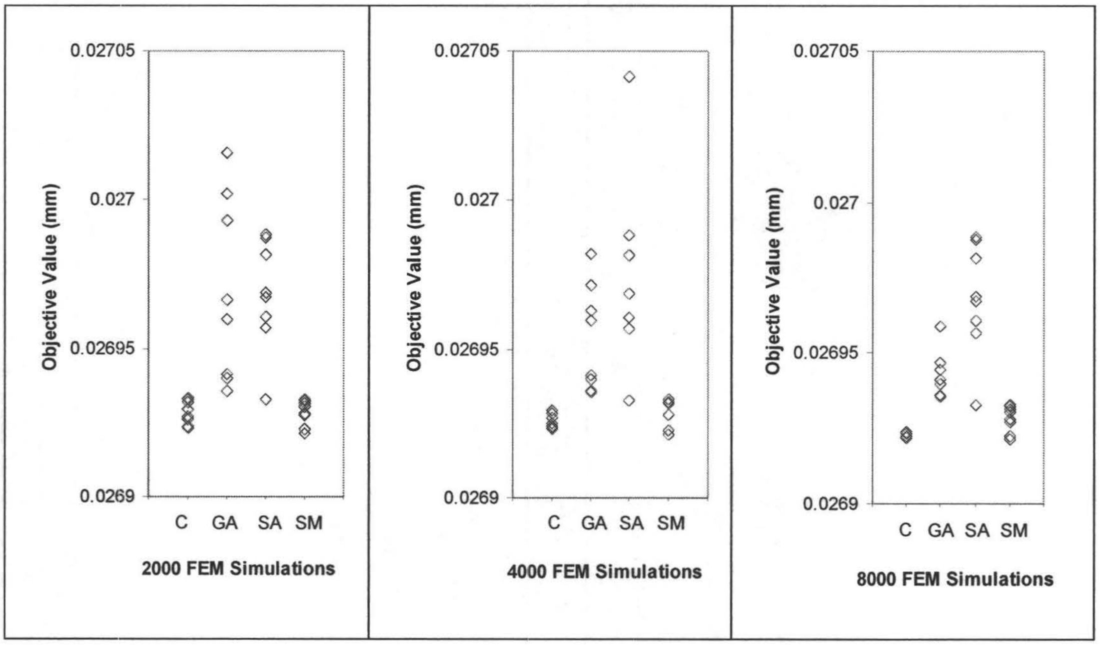
The maximum number of FEM simulations was set as the computational time limitation of the optimization problem. To compare the optimization results with and without agent coordination, each optimization algorithm agent was run separately first and then 3 types of optimization algorithm agents coordinated on the process optimization. Eight trials were made as per 2000, 4000 and 8000 FEM simulations for each case.

Comparison of the effects of agent collaboration and the implementation of single optimization algorithm agent is shown in Table 4-6, which shows the eight best points created by 8 trials with 2000, 4000 and 8000 FEM simulations. If no agent collaboration is involved, the SM agents provided the best average of solutions, as well as the lowest standard deviation across different trials, and provided the best overall solution. The reason is that process optimization problem of Case 1 is, relatively, not very complex. The behavior of the objective function is not too challenging, so the simplex method converges faster than the simulated annealing and the genetic algorithm methods. With the in-process collaboration of SM, SA and GA agents, however, the performance is even better than running SM agents alone. With agent collaboration, the optimization results have the lowest average and standard deviation values for the objective function. That means there is higher possibility that better results can be found with agent collaboration.

Table 4-6. Comparison of results for Case 1 with 3 design variables

Number of FEM Simulations	Collaboration Type	Best Solution $(\times 10^{-2} mm)$	Worst Solution $(\times 10^{-2} mm)$	Average of Solutions $(\times 10^{-2} mm)$	Standard Deviation of Solutions $(\times 10^{-6} mm)$
2000	SM No Collaboration	2.692	2.695	2.694	12.02
	SA No Collaboration	2.696	2.711	2.701	55.34
	GA No Collaboration	2.694	2.702	2.697	30.79
	SM-SA-GA Collaboration	2.692	2.693	2.693	3.99
4000	SM No Collaboration	2.692	2.693	2.693	4.85
	SA No Collaboration	2.693	2.705	2.699	41.43
	GA No Collaboration	2.694	2.698	2.695	17.85
	SM-SA-GA Collaboration	2.692	2.693	2.693	2.42
8000	SM No Collaboration	2.692	2.693	2.693	4.24
	SA No Collaboration	2.693	2.699	2.697	18.54
	GA No Collaboration	2.694	2.696	2.694	8.00
	SM-SA-GA Collaboration	2.692	2.692	2.692	0.72

Comparison of the effects of agent collaboration and the implementation of single optimization algorithm agents is also shown in Figure 4-14. From Figure 4-14, we can clearly find that the agent collaboration case provided the best average performance with the lowest range of best points and the SM agents provided the better average performance if compared with GA or SA agents.



(C - Agent Collaboration)

Figure 4-14. Comparison of results for Case 1 with 3 design variables

Figure 4-15 shows the distributions of design variables of best points with a maximum number of FEM simulations of 8000. It is obvious that agent collaboration cases have the lowest ranges of design variables. With reasonably low ranges of design variables of feasible points created, it is much easier for designers to find a feasible solution even if other design criteria have to be considered in the design process.

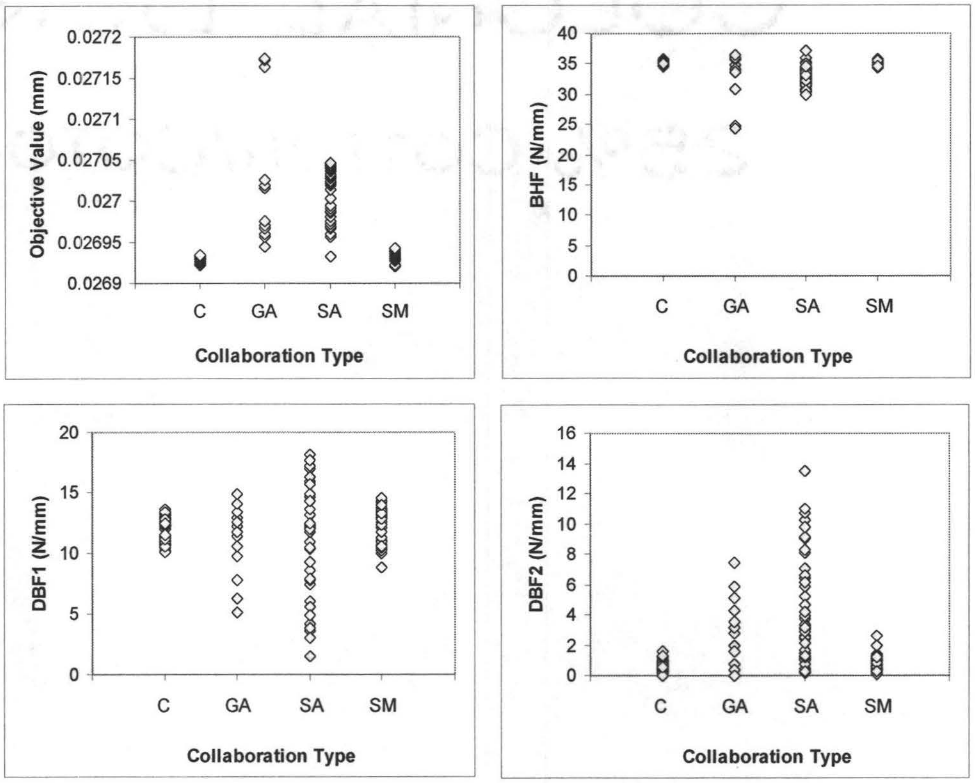


Figure 4-15. Distribution of best 10% points (8000 FEM simulations per run)

The above result analysis has demonstrated the overall performance of agent collaboration. It is also necessary to study the impact of each type of agent on the final optimization result. Fields of "*flag*" and "*N*" of the shared database were assigned especially for impact analysis. As mentioned before, "*flag*" records the history of being selected by other agents and "*N*" records repeating times of the same point in the shared database. Moreover, the name of the original agent that creates the point is recorded in the field of "*sender*". It is also possible to trace the whole history of a point if a field is assigned to record the "ids" of starting points that are used to generate the final result.

In this research, the following items were studied to present the impact of each type of optimization agents:

- How many points have each type of agents contributed to the final result?
- How often have the points generated by each type of agents been selected as starting points by other types of agents?
- How good are the points created by each type of agents?

The first item can be regarded as the quantity of contribution of each type of agents. It is measured by the average number of points contributed as shown in Table 4-6. The second item is measured by the average number of times when the points generated by each type of agents have been reused by other types of agents. The third item presents the quality of the contribution of each type of agents. It is measured by calculating the percentage of points generated by each type of agents in the best 10% points of the final result.

Table 4-7 shows the comparison of impacts of optimization agents. Values of average number of points contributed depend on the parameter selections of optimization algorithms and the distribution of agents. Repeated points were only counted once. From Table 4-7, we can find that points generated by SM agents were reused by other agents at a very high frequency as the number of FEM simulations increased and SM agents contributed a very high percentage of best points in the final best 10% points. In other words, SM agents provided high quality initial points in the process of agent coordination. The significance of GA agents became more apparent as the number of FEM simulations increased. It is not a surprise that better results can be achieved with higher quality of starting points in GA. The trend of SA is not very clear due to the high randomness of SA in this research.

Table 4-7. Comparison of impacts of optimization agents

Number of FEM Simulations	Agent Type	Average Number of Points Contributed	Average Number of Times Points Were Selected by Other Agents as Starting Points	(Number of Points generated)/ (Total Number of Best 10% Points)
2000	SM	9.3	3.9	77. 8%
	SA	3.4	2.9	11.1%
	GA	15.3	5.1	11.1%
	DG	60	37.4	0.0%
4000	SM	19.2	12.1	75%
	SA	7.5	7.6	8.3%
	GA	35.1	14.4	16.7%
	DG	60	53.9	0.0%
8000	SM	37.3	31.2	70.6%
	SA	23.2	17.8	8.8%
	GA	62.4	27.8	20.6%
	DG	60	73.1	0.0%

In this case, the best solution was created by the collaboration of a DG, a GA, and an SM agent. First the starting point in the process of creating the best solution was generated by the DG agent randomly and then used by the GA agent. Then the solution generated by the GA agent was used as a starting point for the SM agent. Finally the best solution was generated by the SM agent. The best solution is:

Edge Tension of Blankholder Force Boundary Condition: *38.14 N/mm*

Drawbead Restraining Forces:

Drawbead 1: *13.24 N/mm*;

Drawbead 2: *0.75 N/mm*.

In practice, drawbead 2 would probably be eliminated since its effect on the forming operation is very small but it would contribute significantly to the manufacturing cost of the die. By including linkages between this optimization and the die design agents in the network, die cost could be added to the optimization model to improve the process design.

The forming limit diagram of the best solution is shown in Figure 4-16. The lack of points falling beyond the forming limit curves demonstrates that the optimized tooling design will lead to good parts in production.

Figure 4-17 shows the distribution of forming behavior over the part surface using the Contour Results mode of FastForm3D in which fully contoured, or smoothed, regions of each forming result are displayed on the part. No elements in the Fail or Strong Wrinkling Tendency zones are found.

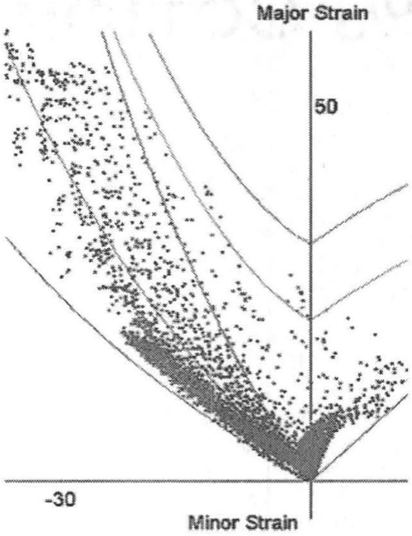


Figure 4-16. FLD of Case 1 with the best solution

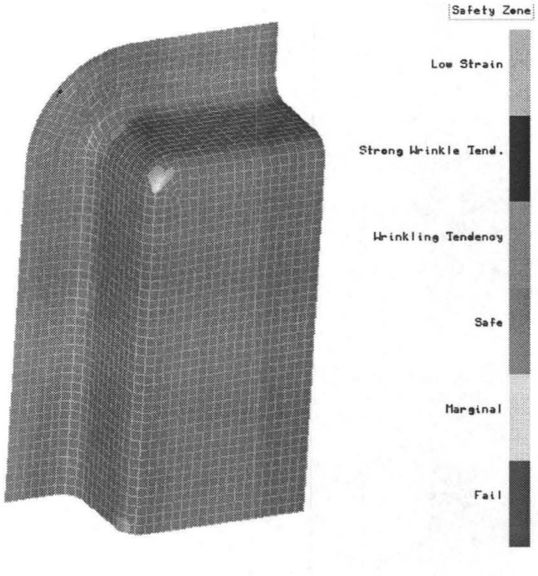


Figure 4-17. Results display based on safety zone for Case 1 with the best solution

Through the result analysis of the 2-drawbead case, we can find that the proposed methodology for process optimization based on agent collaboration can provide better results with a high possibility than running any of the individual optimization algorithms separately. Process optimization problems are complex in formed sheet-metal product and die design. It is usually a hard task to determine which optimization algorithm is suitable for a specific optimization problem. With the agent-based process optimization approach, collaboration of traditional optimization algorithms can improve the overall performance of such systems.

4.6 Case Study 2: Rectangular Cup, Five Drawbeads

4.6.1 Case Description

It has been demonstrated that the proposed agent-based process optimization architecture works well with comparatively simple optimization problems, such as Case Study 1. It is necessary to study how the agent-based architecture works with more complex optimization problems. For this purpose, Case 2 was modeled based on the deep drawn rectangular cup used in Case Study 1. Five drawbeads were set in Case 2, as shown in Figure 4-18. A quarter of the deep drawn cup with 1887 elements was applied, just as in Case 1.

Material, ranges of design variables, upper limit and lower limit of element thickness in Case Study 2 are the same as those in Case Study 1.

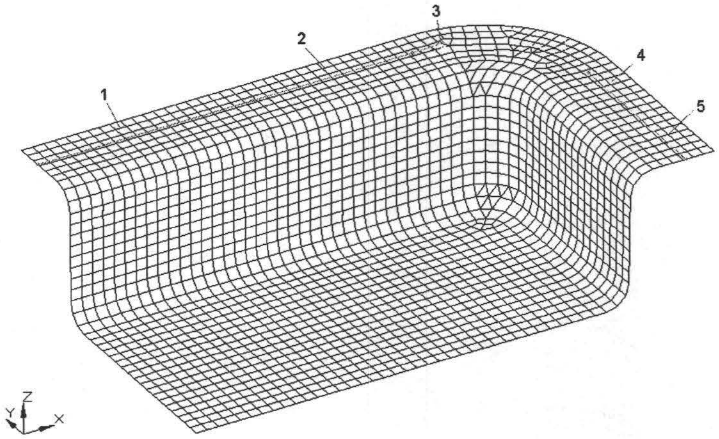


Figure 4-18. Model of the 5-drawbead case (Case 2)

Penalty coefficients are shown in Table 4-8. Penalty coefficients for constraints (4-13) to (4-15) in Case Study 2 are much smaller than those in Case Study 1 because objective function and constraints are calculated based on Gauss points. N_W, N_F and N_M are larger than those in Case Study 1. Small penalty coefficients can ensure the objective function value in Equation (4-9) doesn't change too rapidly when the penalty function method is applied. Meanwhile, the influence of penalty coefficients can be studied by comparing the final results of Case Study 1 and Case Study 2.

Table 4-8. Penalty coefficients for case study 2

Penalty Coefficient	Value	Condition	Constraint No
R1	0.0394 mm^{-1} (1 $inch^{-1}$)	$0.00254\,mm < (t_{\max} - T_U) < 0.0508\,mm$	(4-11)
	0.0787 mm^{-1} (2 $inch^{-1}$)	$0.0508\,mm \le (t_{\max} - T_U) < 0.1016\,mm$	
	0.1575 mm^{-1} (4 $inch^{-1}$)	$0.1016\,mm \le (t_{\max} - T_U)$	
R2	0.0394 mm^{-1} (1 $inch^{-1}$)	$0.00254\,mm < (T_L - t_{\min}) \le 0.0508\,mm$	(4-12)
	0.0787 mm^{-1} (2 $inch^{-1}$)	$0.0508\,mm < (T_L - t_{\min}) \le 0.1016\,mm$	
	0.1575 mm^{-1} (4 $inch^{-1}$)	$0.1016\,mm < (T_L - t_{\min})$	
R3	5.08E-7 *mm* (2.0E-8 *inch*)	$0 < N_W < 20$	(4-13)
	1.27E-6 *mm* (5.0E-8 *inch*)	$20 \le N_W < 40$	
	2.54E-6 *mm* (1.0E-7 *inch*)	$40 \le N_W$	
R4	2.54E-3 *mm* (1.0E-4 *inch*)	$0 < N_F < 10$	(4-14)
	5.08E-3 *mm* (2.0E-4 *inch*)	$10 \le N_F < 20$	
	1.016E-2 *mm* (4.0E-4 *inch*)	$20 \le N_F$	
R5	1.27E-6 *mm* (5.0E-8 *inch*)	$0 < N_M < 20$	(4-15)
	2.54E-6 *mm* (1.0E-7 *inch*)	$20 \le N_M < 40$	
	5.08E-6 *mm* (2.0E-7 *inch*)	$40 \le N_M$	

4.6.2 Optimization Trend

The optimization model for Case 2 has 6 design variables, i.e. the edge tension of the blankholder force boundary condition and 5 drawbead restraining forces. For this case, it is very time consuming to run a grid search like that done in Case Study 1. To simplify the estimation process of the optimization trend of Case 2, a feasible point, as shown in Table 4-9, was generated by a GA optimization and then cases for estimation were generated by updating only one design variable of the feasible point at a time. The grid size was set to 4 and in total $26 \times 6 = 156$ tests were conducted. The distribution of objective values with those settings is shown in Figure 4-19. Since the above estimation process is based on a feasible point, results obtained from these estimation tests do not provide an overall trend of this optimization problem for the total feasible area of design variables and can only be used to study the sensitivity of design variables at a specific point.

Table 4-9. The feasible point for estimation of optimization trend of Case 2

BHF (*N/mm*)	DBF (*N/mm*)					Objective Value ($\times 10^{-2}\ mm$)
	1	2	3	4	5	
42.9	17.1	16.9	8.2	2.1	6.1	2.632

Figure 4-19 shows the relationship between each design variable and the respective objective function when other design variables are set to the values listed in

Table 4-9. Ranges that bracket the position of the optimal value of each design variable are shown in Table 4-10 and were derived from Figure 4-19.

Compared with the distribution of the objective values for Case 1 as shown in Figure 4-10, the distribution of the objective values for Case 2 as shown in Figure 4-19 is much smoother. The reason is that penalty coefficients used in Case Study 2 are much lower than those used in Case Study 1.

Table 4-10. Ranges of design variables for Case 2

Design Variable	Range
BHF	35 ~ 50 *N/mm*
DBF1	10 ~ 30 *N/mm*
DBF2	10 ~ 30 *N/mm*
DBF3	0 ~ 20 *N/mm*
DBF4	0 ~ 10 *N/mm*
DBF5	0 ~ 15 *N/mm*

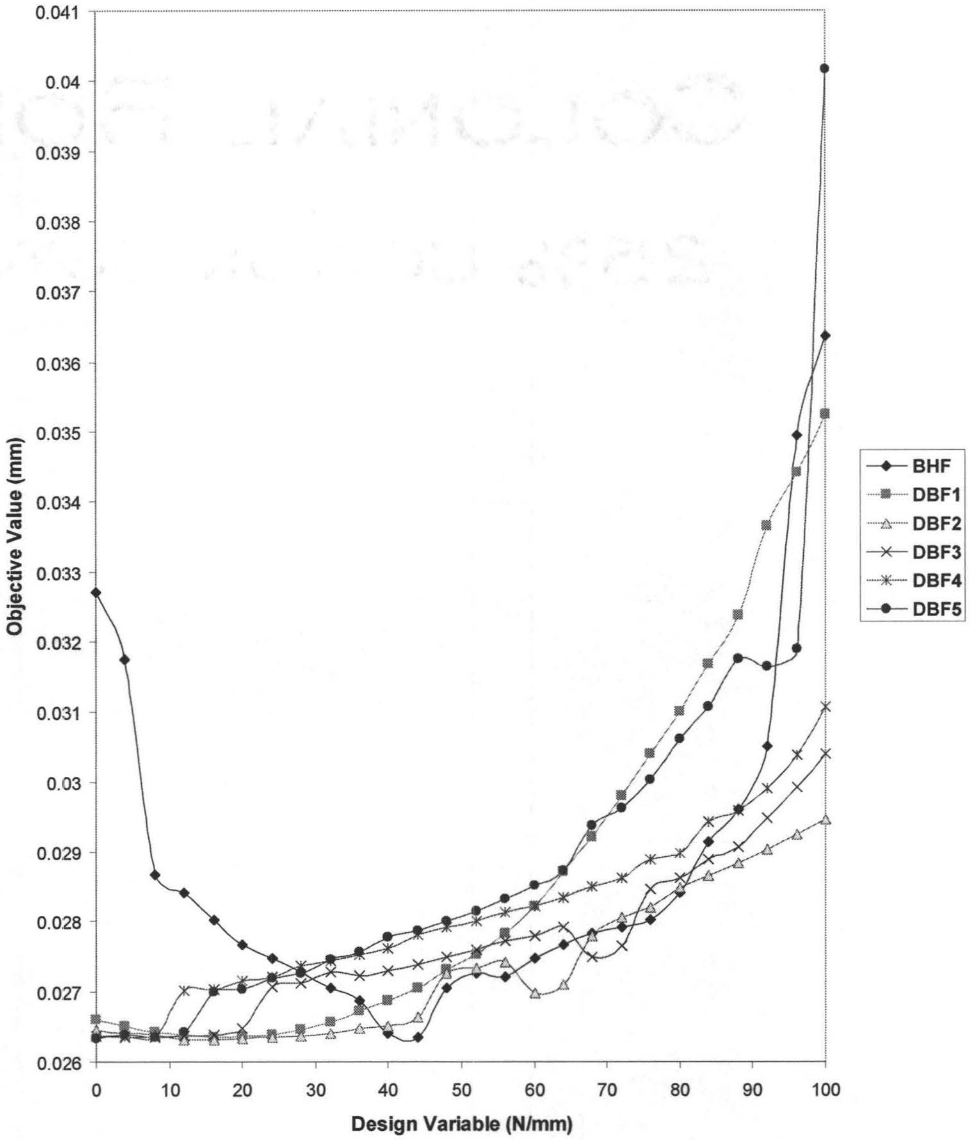


Figure 4-19. Distribution of objective values of tests with varying design variables

4.6.3 Parameter Selection of Optimization Algorithms

Case 2 has 6 design variables. The computational cost of GA, SA and SM for Case 2 is therefore much higher than the computational cost in Case 1. Several tests have been conducted for parameter selection of three optimization algorithms used in this case study.

Simplex Method

In this case study, there are 6 design variables and therefore a simplex consists of 7 vertices. The same termination criteria for the simplex method were used in Case Studies 1 and 2. Two different starting points were used to conduct tests. Results are shown in Figures 4-20 and 4-21, respectively.

The starting point (P1) of the test as shown in Figure 4-20 is (61.5, 8.3, 64.0, 53.8, 37.3, 8.2). It was created by selecting design variables randomly with their ranges and then rounding them off to one decimal place. The objective value of P1 is 0.03185 *mm*. The optimization process as shown in Figure 4-20 converges after 63 iterations and final result is (34.0, 38.2, 47.4, 38.5, 14.9, 37.4). The objective value of the final result is 0.02815 *mm*. The total number of FE simulations involved is 196.

The starting point (P2) of the test as shown in Figure 4-21 is (40.0, 15.0, 10.0, 5.0, 5.0, 5.0). It was picked from the feasible ranges shown in Table 4-10. The objective value of P2 is 0.02644 *mm*. The optimization process as shown in Figure 4-21 converges after 26 iterations and final result is (41.1, 14.3, 12.5, 5.3, 3.5, 7.4). The objective value of the final result is 0.02636 *mm*. The total number of FE simulations involved is 101.

And moreover, objective values of optimal solutions generated in the above two tests, 0.02815 and 0.02636 *mm*, differ greatly. These two sample tests have shown that the selection of initial simplex plays a very important role in simplex methods.

The above two tests with different starting points have also shown that parameters for termination criteria are suitable.

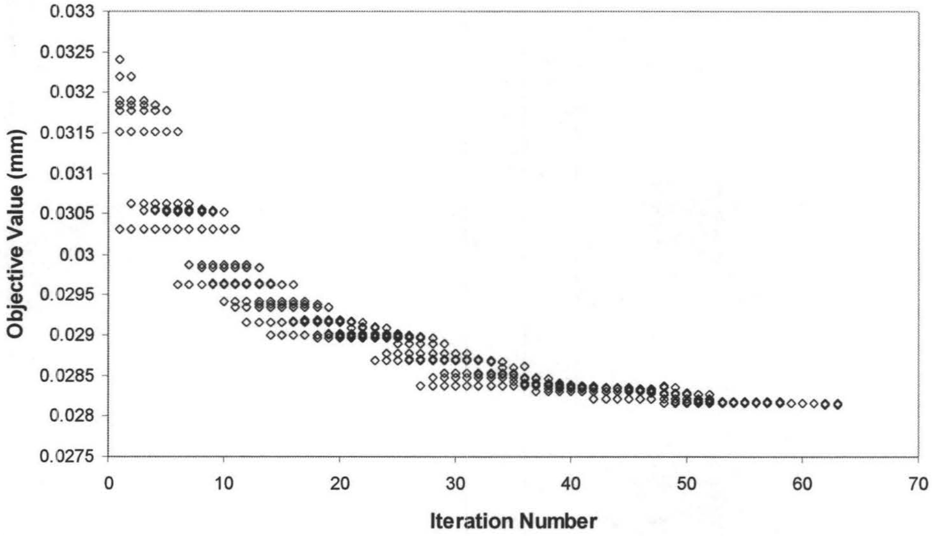


Figure 4-20. Convergence trend of the application of SM with the starting point P1 for Case 2

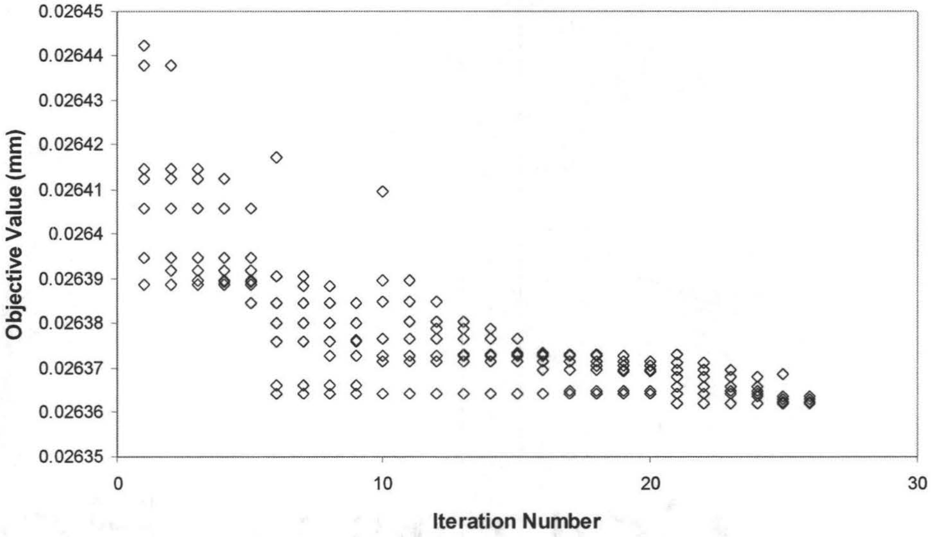


Figure 4-21. Convergence trend of the application of SM with the starting point P2 for Case 2

Simulated Annealing

Since Case 2 is a 6-dimensional optimization problem, the number of moves per step was still set to 20 to guarantee there are enough chances for each variable to be selected and updated to generate a new starting point. As mentioned in Section 4.5.3, it is a trial-and-error process to determine the initial control parameter C_0 and cooling parameter k.

Figure 4-22 shows the convergence trend of the simulated annealing algorithm with the following settings:

$C_0 = 5.0 \times 10^{-6}$; $k = 0.40$; $Nmove = 20$.

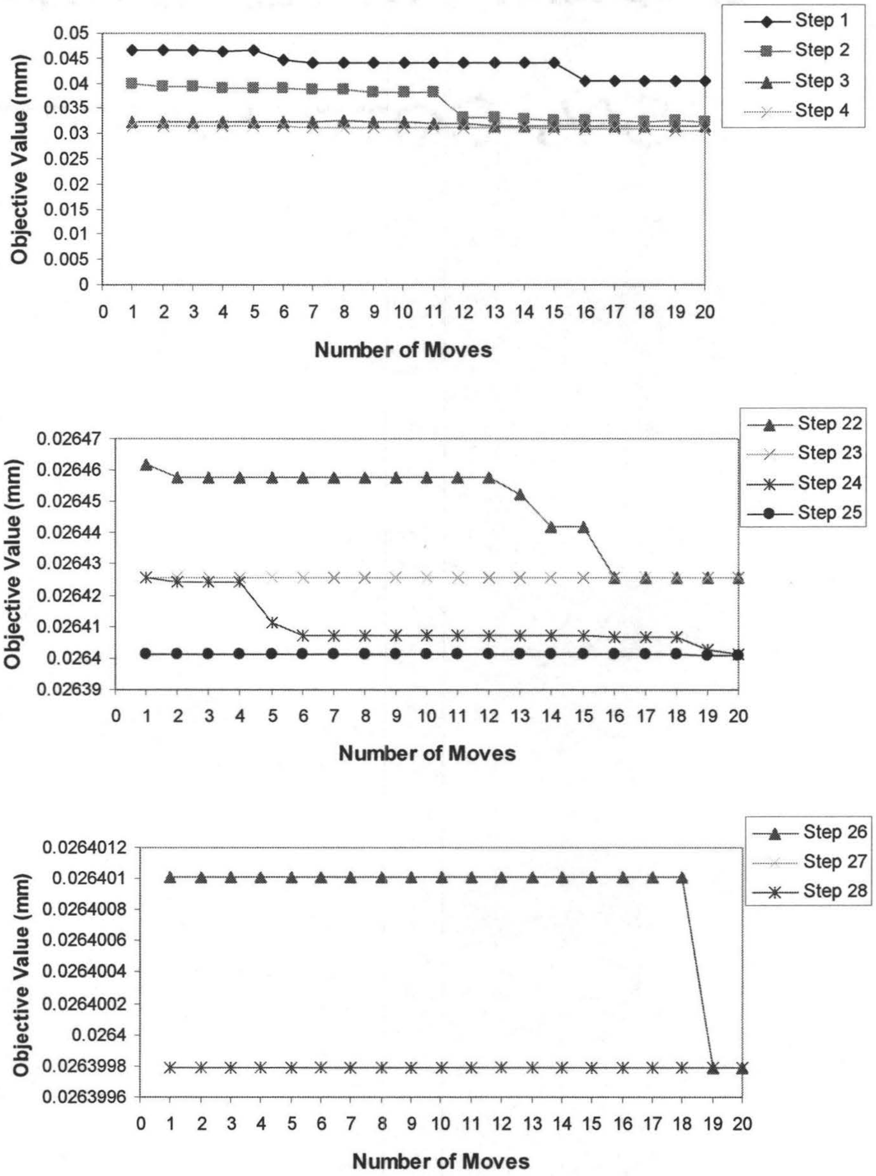


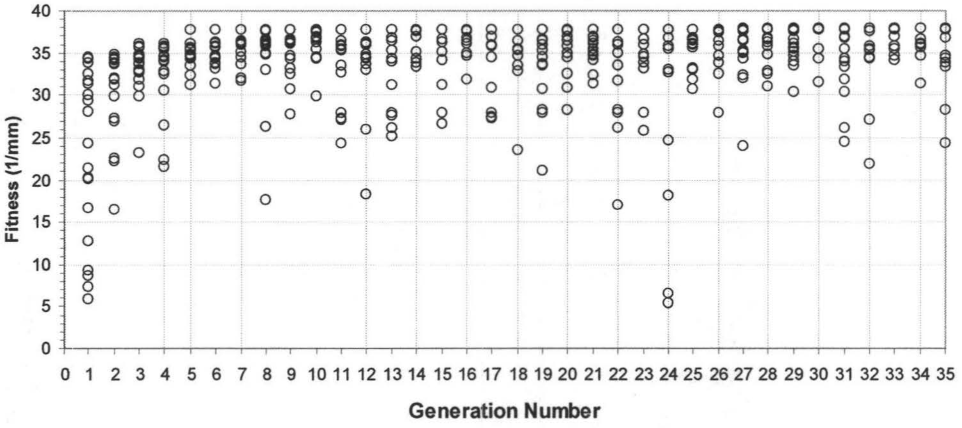
Figure 4-22. Convergence trend of an application of SA for Case 2

The starting point for this test is (75.8, 3.8, 40.8, 11.8, 30.2, 79.4). It was also generated by selecting the design variables randomly within their ranges and rounding them off to one decimal place. The objective value at this point is 0.04662 *mm*. The pre-specified threshold value for this test was set as $2.54 \times 10^{-6}\ mm\,(1 \times 10^{-7}\, inch)$ and the algorithm converges after Step 25. The solution is (40.3, 20.3, 19.5, 14.3, 3.9, 13.5). The objective value of the solution is 0.02640 *mm*.

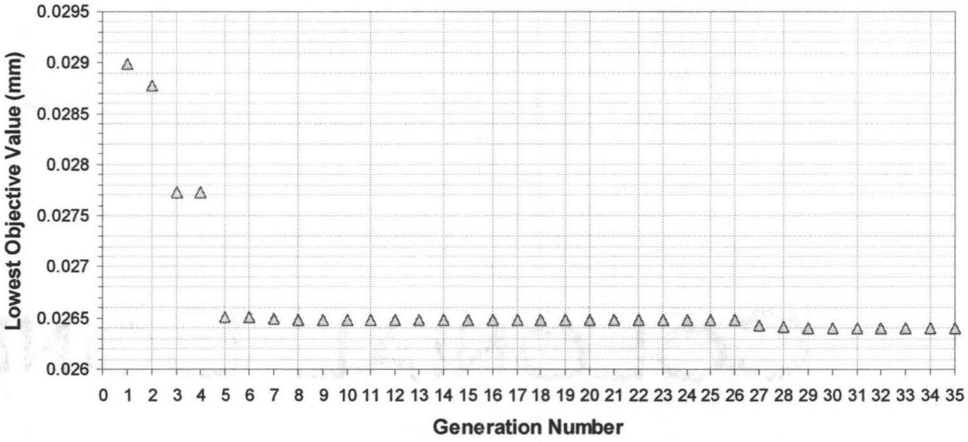
In this case study, the maximum number of steps (*Nstep*) was set as the termination criterion for simulated annealing. A pre-specified threshold value was not used because step numbers of convergence may differ greatly for different starting points and therefore the computational time may be too long for some cases. To reduce the computation cost, *Nstep* is set as a small value of 20 for agent collaboration cases. With a large value of *Nstep*, the acceptance probability becomes very small and the simulated annealing algorithm works almost the same as a random search algorithm.

<u>Genetic Algorithm</u>

In this 6-drawbead case, population size was set as 20. Other parameters, such as mutation rate, percentage of population carried forward unchanged from each generation, percentage of population deemed unfit for reproduction, and weights for each operator were set to the same values as those in Case Study 1.



(a) Fitness Value vs. Generation Number



(b) Lowest Objective Value vs. Generation Number

Figure 4-23. Convergence Trend of an Application of GA for Case 2

Figure 4-23 shows the distribution of fitness values of a sample test. The initial population of chromosomes was constructed at random. From Figure 4-23, we can find that the best fitness value (i.e., the highest value in Figure 4-23 (a)) or the lowest objective value in Figure 4-23(b) doesn't change too much after generation 5, although, if more generations are conducted, better optimization results may be achieved. The maximum number of generations was set as 20 for this case taking into account the computational cost.

4.6.4 Result Analysis

To study the performance of agent collaboration, the relative number of operator agents was also set at 1:1:1:1:1 (simplex method, simulated annealing, genetic algorithm, data generating, data removal) for this case study and the same kind of structure of the shared database was used as Case 1.

In general, collaboration strategies used in Case Studies 1 and 2 are the same. The only difference is that in Case Study 1 only the final solution, one point, is sent to the OM agent by an SA agent, however, in Case Study 2 all the acceptable moves are sent to the OM agent by each SA agent. This change can increase the randomness of the initial starting points in the database. It can also be expected that SA agents will have more important impact on the final results due to this change.

Comparison of the effects of agent collaboration and the implementation of single optimization algorithm agent is shown in Table 4-11. Eight best points created by 8 trials with 2000 and 4000 FEM simulations are compared.

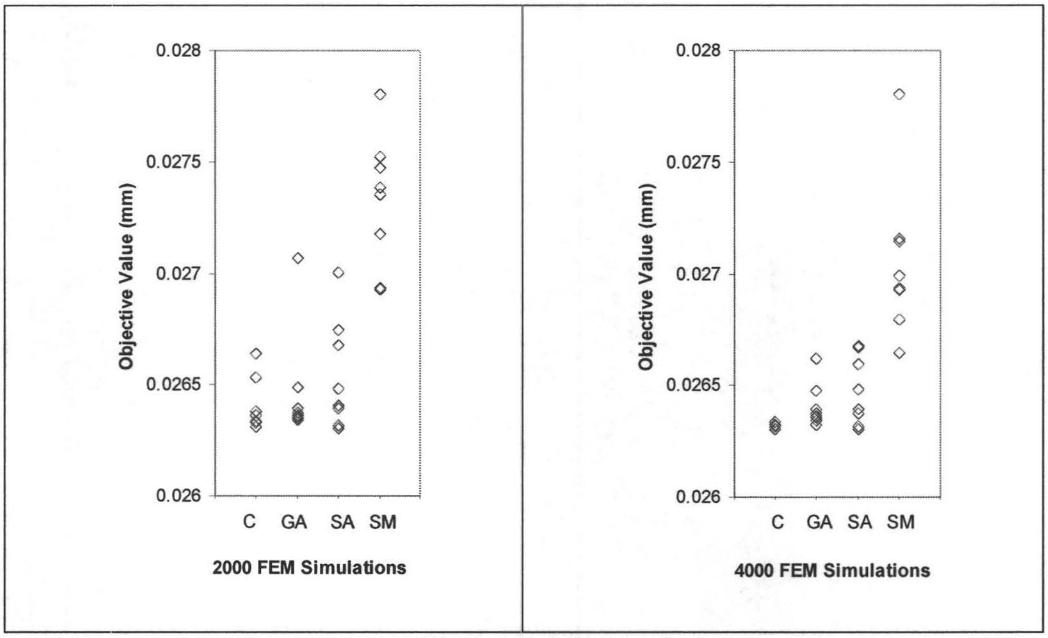
As shown in Table 4-11, if no agent collaboration is involved, GA agents provided the best average performance, as well as the lowest standard deviation across different trials, and provided the best overall solution. As mentioned before in Case Study 1, if no agent collaboration is involved, SM agents provided the best average performance, as well as the lowest standard deviation across different trials, and provided the best overall solution. The reason is that process optimization problem of Case 2 is more complex. The behavior of the objective function is challenging. The simplex method and simulated annealing methods converge very slowly due to this complexity. With limited computational time allowed, GA may work well.

Table 4-11. Comparison of results for Case 2 with 6 design variables

Number of FEM Simulations	Collaboration Type	Best Solution $(\times 10^{-2} mm)$	Worst Solution $(\times 10^{-2} mm)$	Average of Solutions $(\times 10^{-2} mm)$	Standard Deviation of Solutions $(\times 10^{-6} mm)$
2000	SM No Collaboration	2.693	2.781	2.732	299.60
	SA No Collaboration	2.631	2.701	2.654	247.76
	GA No Collaboration	2.634	2.707	2.647	248.34
	SM-SA-GA Collaboration	2.631	2.664	2.640	119.87
4000	SM No Collaboration	2.665	2.781	2.705	348.44
	SA No Collaboration	2.631	2.668	2.648	152.38
	GA No Collaboration	2.632	2.662	2.641	98.55
	SM-SA-GA Collaboration	2.630	2.634	2.632	10.63

With the in-process collaboration of SM, SA and GA agents, however, the performance is even better than running GA agents alone. With agent collaboration, the optimization results have the lowest average and standard deviation values for the objective function. That means there is higher possibility that better results can be found with agent collaboration.

Comparison of the effects of agent collaboration and the implementation of single optimization algorithm agents is also shown in Figure 4-24. From Figure 4-24, we can clearly find that the agent collaboration case provided the best average performance with the lowest range of best points and without collaboration the GA agent provided the best average performance if compared with the SM and SA agents.



(C: Agent Collaboration)

Figure 4-24. Comparison of Results for Case 2 with 6 design variables

SA agents send all the acceptable moves to the OM agent. This greatly increases the total number of points saved in the database of the OM agent. Therefore, only the best 10 points generated with or without agent collaboration are compared in this Case Study. Figure 4-25 shows the distributions of objective values and Figure 4-26 shows the distributions of design variables of the best 10 points with a maximum number of FEM simulations of 4000. In general, agent collaboration cases have the lowest ranges of objective values and the lowest ranges of design variables.

Table 4-12 shows the comparison of impacts of optimization agents. Values of average number of points contributed depend on the parameter selections of optimization algorithms and the distribution of agents. Repeated points were only counted once. From Table 4-12, we can find that points generated by SM agents were reused by other agents at a very high possibility as the number of FEM simulations increased. GA agents contribute most points among the best 5% points generated by agent collaboration with a maximum number of FEM simulations of 2000. SA agents contributed a very high percentage of best points in the final best 5% points with a maximum number of FEM simulations of 4000 because results of all the acceptable moves were sent to the database of the OM agent to. It is important to mention that SM agents still contribute the highest percentage of the most best points. For example, 15 of 20 best points with the maximum number of FEM simulations of 4000 were generated by SM agents, 4 of 20 best points were generated by GA agents and only one of them was generated by the SA agent.

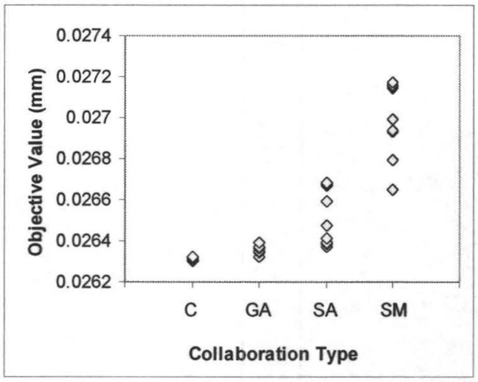


Figure 4-25. Distribution of the best 10 points (4000 FEM simulations per run)

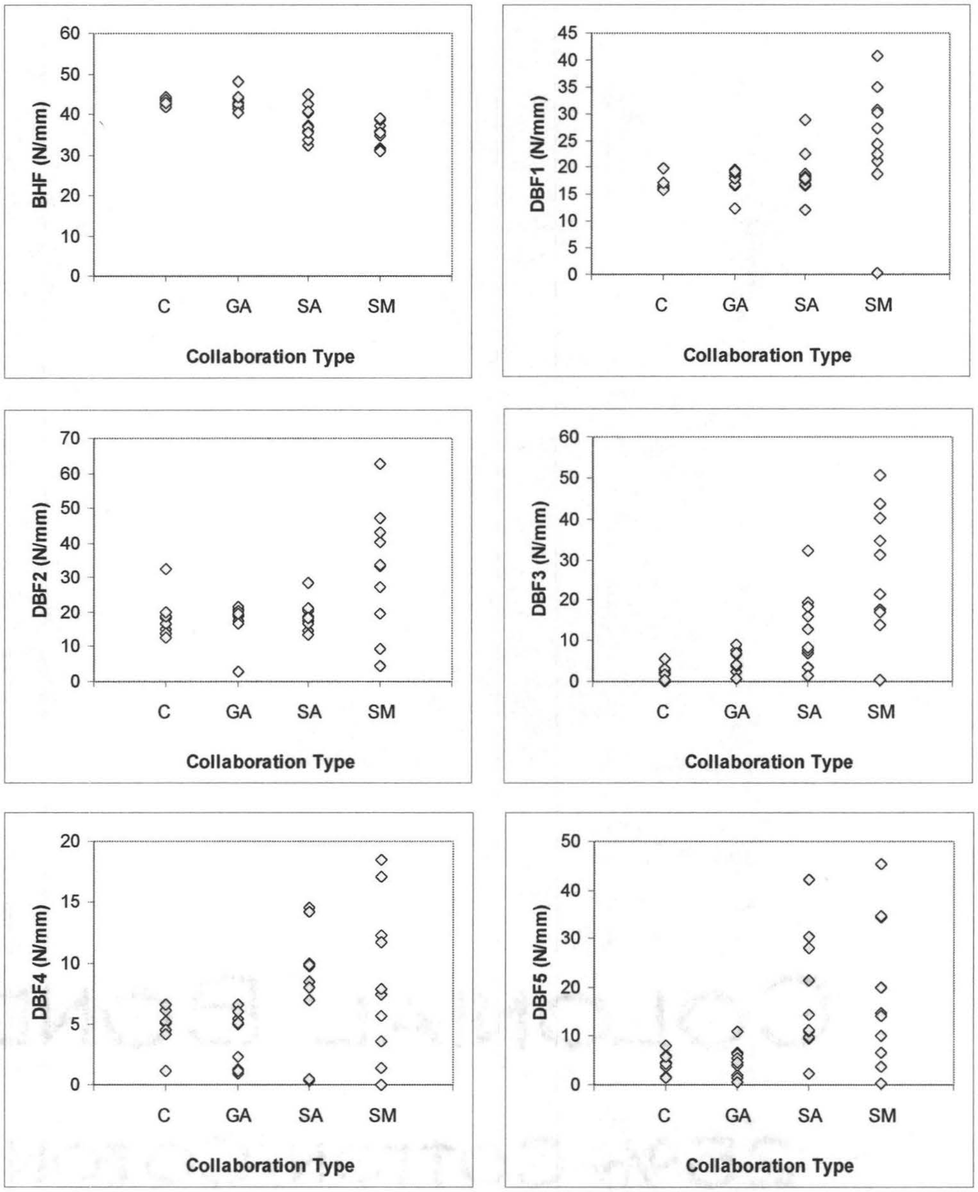


Figure 4-26. Ranges of design variables of the best 10 points (4000 FEM simulations per run)

Table 4-12. Comparison of impacts of optimization agents

Number of FEM Simulations	Agent Type	Average Number of Points Contributed	Average Number of Times Points Were Selected by Other Agents as Starting Points	(Number of Points generated)/ (Total Number of Best 5% Points)
2000	SM	4.8	1.4	22.6%
	SA	191.4	15.8	37.7%
	GA	25.9	2.1	39.6%
	DG	60	25.3	0.0%
4000	SM	10.3	5.1	30.6%
	SA	399.1	35.8	53.1%
	GA	38.3	6.8	16.3%
	DG	60	26.9	0.0%

4.7 Case Study 3: Front Door Panel

4.7.1 Case Description

Cases 1 and 2 were designed based on a comparatively simple deep drawn part with a coarse mesh to save computational time. A complex automotive part, an Audi front door panel, was selected as Case 3 to study how the proposed agent-based process optimization approach works with optimization problems for complex automotive parts. The Audi front door panel served as a benchmark for the Numisheet conference, held in 1999. The geometry of the part and tooling and material properties can be found in Numisheet'99 benchmarks (1999). Position of drawbeads and sections are included in

Numisheet'99 (1999). The front view of the drawing tools and the initial blank is given in Figure 4-27. Steel material is applied in this case study.

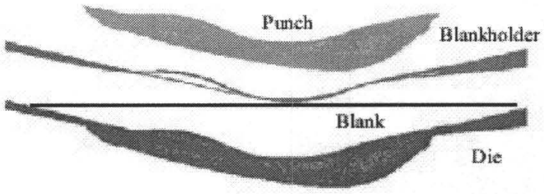


Figure 4-27. Drawing tooling and initial blank for the front door panel

To save computation time, the front door panel was meshed with 7485 elements. Element size is variable. The minimum element size was set to 2.178 *mm* and the maximum element size was 12 *mm*. Chordal deviation was set to 0.05. As shown in Figure 4-28, total number of drawbeads is 8.

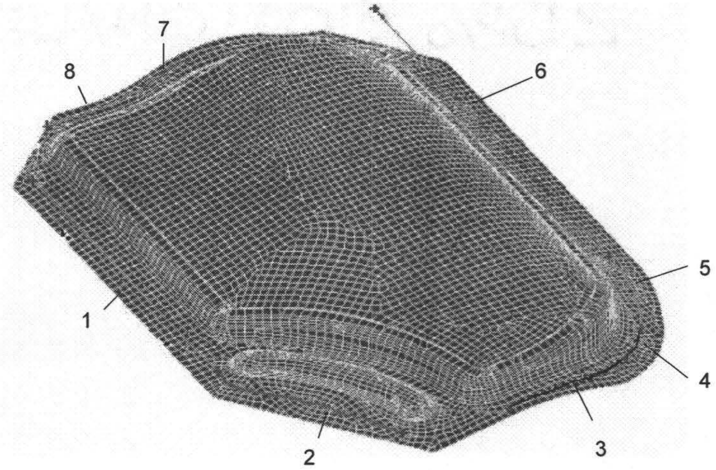


Figure 4-28. The FEM model of Case 3

Upper and lower limits of element thickness and penalty coefficients used for other constraints in this case study are the same as those in Case Study 2. The range of the edge force of the blankholder force boundary condition was set to 0 and 100 *N/mm*. The range of each drawbead restraining force was set to 0 and 500 *N/mm*.

4.7.1 Optimization Trend

The optimization model for Case 3 has 9 design variables, i.e. the edge tension of the blankholder force boundary condition and 8 drawbead forces. As was done in Case Study 2, test cases were generated by updating only one design variable of a feasible point at a time. The feasible point was also generated by a GA optimization and its value is listed in Table 4-13. The grid size was set to 4 for the edge tension of the blankholder force boundary condition and 20 for the drawbead forces and totally $26 \times 9 = 234$ tests were conducted. The distribution of objective values with those settings is shown in Figure 4-29.

Table 4-13. The feasible point for estimation of optimization trend of Case

BHF (*N/mm*)	DBF (*N/mm*)								Objective Value $\left(\times 10^{-2} mm\right)$
	1	2	3	4	5	6	7	8	
72.6	138.1	24.3	44.0	7.9	23.6	4.3	164.2	35.9	1.760

Ranges that bracket the position of the optimal value of each design variable are shown in Table 4-14 and were derived from Figure 4-29.

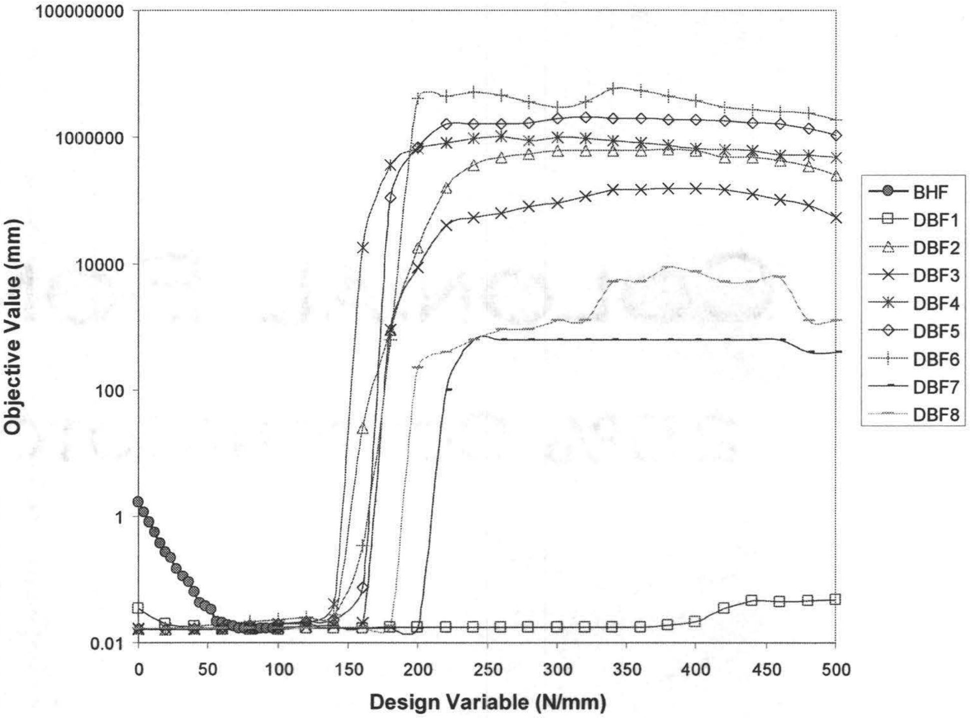


Figure 4-29. Distribution of objective values of tests with varying design variables

Table 4-14. Ranges of design variables for Case 3

Design Variable	Range
BHF	50~100 *N/mm*
DBF1	20~400 *N/mm*
DBF2	0~140 *N/mm*
DBF3	0~140 *N/mm*
DBF4	0~140 *N/mm*
DBF5	0~140 *N/mm*
DBF6	0~140 *N/mm*
DBF7	0~200 *N/mm*
DBF8	0~180 *N/mm*

4.7.2 Parameter Selection of Optimization Algorithms

Case 3 has 9 design variables and the FEM model is more complex than that of Case 2. It takes a much longer time for the three optimization algorithms to converge. Therefore, the termination criteria should be modified. For the Genetic Algorithm, the maximum value of generations should be increased. A large value of *Nstep* or *maxIter* may be suitable for the Simulated Annealing and Simplex Method respectively. However, the optimization process for this case study is very time consuming and the acceptable total number of FEM simulations is limited. It means that there may be not enough collaboration if each optimization algorithm consumes a large number of FEM simulations. Since *maxIter* for the Simplex Method is still large enough to ensure a good result can be obtained and the Genetic Algorithm and Simulated Annealing will not improve very much for this kind of complex optimization problems even the maximum number of generations and the value of *Nstep* are increased a little bit, termination criteria are the same as used in Case 2 to ensure the case study can be focused on performance study of agent collaboration.

4.7.3 Result Analysis

Distribution of agents and collaboration strategies used in Case Study 3 and Case Study 2 are the same. Comparison of the effects of agent collaboration and the implementation of a single optimization agent is shown in Table 4-15. The four best points created by 4 trials with 1000 and 2000 FEM simulations are compared.

As shown in Table 4-15, if no agent collaboration is involved, GA agents provided the best average performance, as well as the lowest standard deviation across different trials, and provided the best overall solution. Running SA or SM agents alone led to very poor results. The reason is that process optimization problem of Case 3 is complex, and therefore, SM and SA methods converge very slowly. With limited computational time allowed, GA methods may work well.

Table 4-15. Comparison of results for Case 3 with 9 design variables

Number of FEM Simulations	Collaboration Type	Best Solution $(\times 10^{-2} mm)$	Worst Solution $(\times 10^{-2} mm)$	Average of Solutions $(\times 10^{-2} mm)$	Standard Deviation of Solutions $(\times 10^{-6} mm)$
1000	SM No Collaboration	1.245E+05	1.887E+09	6.332E+08	8.588E+10
	SA No Collaboration	4.917E+06	1.286E+09	3.266E+08	6.406E+10
	GA No Collaboration	1.701	2.562	2.110	35.40
	SM-SA-GA Collaboration	1.785	1.999	1.881	10.01
2000	SM No Collaboration	1.245E+05	8.428E+08	3.340E+08	4.039E+10
	SA No Collaboration	1.245E+05	2.642E+07	9.210E+06	1.172E+09
	GA No Collaboration	1.701	2.051	1.910	16.53
	SM-SA-GA Collaboration	1.655	1.751	1.697	4.03

With the in-process collaboration of SM, SA and GA agents, the performance is much better than running GA agents alone. With agent collaboration, the optimization results have the lowest average and standard deviation values for the objective function.

Design variables of the best point (with the minimum objective value of 0.01655 *mm*) are shown as follows:

BHF (N/mm): *84.25 N/mm*

DBF 1-8 (N/mm): *138.142, 24.268, 43.983, 7.878, 23.623, 4.309, 89.222, 35.881*

The blankholder force of the best point is *813 KN*. With the process parameters of the best point, no elements fail or fall into the marginal area as shown in Figure 4-30. Some wrinkles occur in the zone under the blank holder.

Thickness distribution of the formed part with process parameters of the best point is shown in Figure 4-31. The range of thickness is between 0.859 and 1.036 *mm*. Compared with the thickness ranges (between 0.7941 and 1.149 mm before optimization and between 0.7876 and 1.0104 mm after optimization) presented in the literature (Ayed et al. 2005), the best point generated with agent collaboration can provide better thickness distribution.

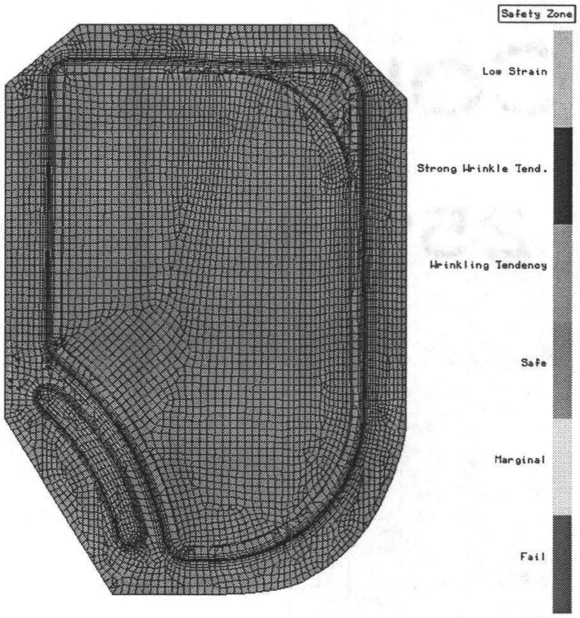


Figure 4-30. Results display based on safety zone for Case 3 with the best solution

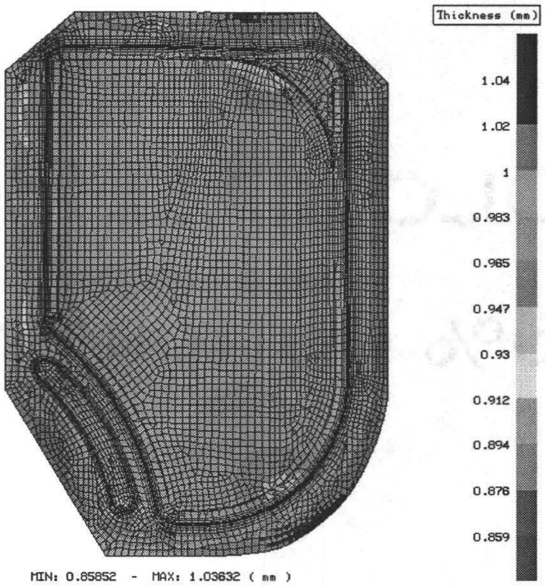


Figure 4-31. Thickness distribution of the formed part

4.8 Summary

Result analysis of the case studies has shown that the agent-based process optimization approach proposed in this chapter works well with both simple and relatively complex optimization problems. The agent-based process optimization approach is especially suitable for the situation when it is difficult to determine which optimization algorithm may be effective in solving complex optimization problems.

The evaluation approach proposed for impact analysis of different optimization algorithm agents involved in the optimization process can provide fundamental insights for deploying and configuring agents.

Research in this chapter has demonstrated that agent collaboration can also play a very important role in process optimization. A seamless design optimization environment can be developed based on this research.

Chapter 5. Performance Evaluation of Multi-Agent Systems for Sheet-Metal Product Development

5.1 Introduction

Important issues on the development of agent-based collaborative sheet-metal product development systems, such as domain analysis, role assignment, agent communication and agent coordination, have been discussed in Chapters 3 and 4. Chapter 3 described the detailed design and implementation process of the prototype agent-based system for sheet-metal product development based on the ZEUS agent building toolkit. Chapter 4 presented an agent-based framework for process optimization and studied the performance of the collaboration among different optimization agents. Research in those two chapters was focused on the design and implementation of a workable prototype system that may facilitate the communication and coordination among all participants in the product development process. The selection of organizational structures and coordination strategies and the distribution of agents were based on the experience gained from legacy projects on agent-based concurrent design and manufacturing in the literature, legacy systems for standalone or collaborative design of sheet-metal parts and dies, and the domain analysis in the preliminary system design stage. For real applications, e.g. sheet-metal product development, performance and scalability have to be studied before the decision can be made on the selection of organizational structures and coordination strategies for the multi-agent systems and the distribution of agents because of the complexity and dynamics of such applications.

The impact of the number of the agents, the organizational structure of agents and the coordination strategy employed on the scalability and performance of agent-based sheet-metal product development systems is studied in this chapter. Quality attributes and metrics will be defined for the evaluation.

This chapter is structured in the following sections. Sections 5.2 and 5.3 present the applicable organizational structures and coordination strategies for agent-based sheet-metal product development systems. The ZEUS agent building toolkit is used to model the case studies and the benefit is introduced in Section 5.4. Performance evaluation strategies are presented in Section 5.5. Case studies and summaries are provided in the last two sections. In this Chapter, the term of "architecture" means the system architecture.

5.2 System Architectures

Applicable organizational structures of multi-agent systems for sheet-metal product development should have the capability to enable such systems to be open, easy to change, secure and stable. Previous research has demonstrated that federated structures are suitable for such applications. For example, in the prototype system as shown in Figure 3-1, design coordinator agents and supplier coordinator agents act as representatives of local agent communities, play a role of local facilitators and manage the communication and coordination activities between customers and suppliers. Local task agents in an agent community communicate with other local task agents in other agent communities through design coordinator agents or supplier coordinator agents. It is easy to add new customer or supplier agent communities without notifying each participant in the system, except informing the coordinator agents of abilities of the new agent communities and notifying local name server agents of addresses of respective agents if applicable. The disadvantage of such systems comes with the advantage mentioned above. Design and supplier coordinator agents may become bottlenecks under heavy load.

To ease the communication load of coordinator agents, agents from different agent communities can be allowed to communicate directly if approved by the local coordinator agents involved. In local agent communities, both hierarchical and peer-to-peer architectures are applicable. This type of hybrid system architecture can benefit from the advantages of peer-to-peer, hierarchical and federated architectures and is a good choice for the development of practical agent-based sheet-metal product development systems.

Influence of various combinations of system architectures will be studied in case studies.

5.3 Coordination Strategies

Various coordination strategies can be found in literature, e.g. Nwana et al. (1996), Green et al. (1997) and Oliveira et al. (1999). According to the above literature, the contract-net protocol and its variants were most widely used for dynamic task allocation in multi-agent systems. An iterative contract-net protocol was used for case studies in Chapter 3 and its details were introduced in Section 3.4.4. The iterative contract-net protocol is suitable for developing open, complex and dynamic federated systems with a large number of agents due to the flexibility requirements of such systems. To simplify the research problem, only the non-iterative version of the contract-net protocol is going to be studied in the case studies in this chapter.

5.4 Architecture Modeling and Evaluation Tools

Research results in Chapters 3 and 4 have shown that ZEUS agents are suitable for the rapid development of flexible, open multi-agent systems for sheet-metal product development. Meanwhile, ZEUS is a flexible toolkit for the development and deployment of multi-agent systems with different organization structures and coordination strategies.

First, ZEUS provides a good user interface to define relationships among agents. There are four different types of relationships that can exist between agents. Agents can be peers, superiors, subordinates or co-workers (Collis and Ndumu, 1999).

Second, ZEUS also provides a good user interface to equip agents with the coordination protocols and expertise required for social interaction with other agents.

Moreover, a visualiser agent equipped with a statistics tool can be created by the ZEUS toolkit to analyze or debug the interactions of agents in the system design stage. Such a tool can provide foundations upon which the organizational structure, coordination strategies and the distribution of tasks can be determined to achieve satisfactory system performance.

The statistics tool supported by the visualiser agent has the capability to collect the following statistics:

- the number and type(s) of messages sent by the agents over a time period;
- the number of messages exchanged among agents in coordinating different goals;

- the average load of agents, i.e. the proportion of time agents spend actually executing tasks; and,
- the ratio of coordination time versus computation time, i.e. time spent coordinating tasks as opposed to executing them.

In this research, the statistics tool is used to record agents' interaction sessions for analysis of the impact of organizational structures and coordination strategies on system performance.

5.5 Evaluation Strategies

The systematical evaluation approach that will be presented in this research is an extension to Lee and Hwang (2004)'s research. Different to Lee and Hwang's approach, dynamic features will be evaluated based on not only the number of interactions, but also the real response time because response time is one of the most important performance indicators for the deployment of agents.

5.5.1 Static Quality Attributes

Static quality attributes include complexity, extendability and availability of agent-based systems in Lee and Hwang (2004)'s approach. Complexity of agent-based systems is measured by the number of links among agents or facilitators. It is assumed that the larger the number of links, the more complex the system is. Extendability of system is represented by the increase of the number of links when an agent or a facilitator is added to an agent-based system. Availability of system is also induced from the complexity of system and can be used to study the availability of resources when an agent or a facilitator is abnormal.

Table 5-1 describes the metrics for the measurement of complexity, extendability and availability of agent-based systems based on three types of typical structures shown in Figure 2-2 according to Lee and Hwang (2004)'s approach. The assumption made for the federated architecture is that agents under each local facilitator are structured in a peer-to-peer manner. In a hierarchical architecture, each agent only communicates with its root and child agents.

Table 5-1. Metrics for static quality attributes (following Lee and Hwang, 2004)

Organization Structure	Complexity	Extendability	Availability
Peer-to-Peer (Autonomous)	$m(m-1)/2$	(Total number of links with *i+1* agents) – (Total number of links with *i* agents)	(Total number of links with *i-1* agents)/(Total number of links with *i* agents)
Federated (Group with facilitators)	$n(n-1)/2+\sum_{i=1}^{n} m_i(m_i+1)/2$	Same as above	Same as above
Hierarchical (Tree)	$n+\sum_{i=1}^{n} m_i$	Same as above	Same as above

In Table 5-1, m is the total number of agents, n is the number of facilitators and m_i is the number of agents under the i^{th} facilitator.

As mentioned in Section 5.3, federated or so-called hybrid architectures are appropriate for practical agent-based sheet-metal product development systems. In this chapter, the research focus of architecture modeling is on federated and hybrid architectures.

In a federated architecture, agents under each facilitator (i.e. a design or supplier coordinate agent in Chapters 3 and 4) can be organized in a peer-to-peer style with the facilitator. That means the facilitator or each agent has the necessary information about addresses and capabilities of other agents in the same agent community and can communicate with each other directly. The facilitator acts as a coordinator of the agents in the community. To ease the communication among agents, the facilitator makes the decision on task allocation based on the information it has about other agents in the community. Agents in the community can also have their subordinates. The relationship between agents needs to be clearly defined in the system design stage to ease the communication. In such a system, agents can be categorized into several hierarchies. Local facilitators (i.e. coordinator agents) and global utility agents constitute the top hierarchy.

The hybrid architecture is based on the federated architecture above. The only difference between them is that agents from different agent communities can communicate directly if approved by local facilitators.

Cases based on different architectures will be modeled for the case studies.

5.5.2 Dynamic Quality Attributes

The number of interactions occurring from the time of service request to the time of service provision was proposed as the dynamic quality attribute of agent-based systems by Lee and Hwang (2004). Since the number of interactions is independent of network protocols and communication capacities, this approach can be used for system performance evaluation at the preliminary system design stage.

Metrics to calculate the number of interactions for typical system architectures shown in Figure 2-2 were presented in Lee and Hwang (2004). A simple case based on the peer-to-peer architecture employed with a non-iterative contract-net protocol is presented as follows.

When a non-iterative contract-net coordination mechanism is applied in a peer-to-peer framework, the number of interactions occurring from service request to service provision can be calculated as $3(m-1)+1$, where m is the total number of agents. The assumptions made for the above calculation are:

- One agent among m agents is the "manager" and other agents are "bidders".
- A task can be decomposed into several subtasks. However, this situation is not to be taken into consideration for performance evaluation in this chapter. The time period from service request to service provision means the communication and execution time of the task.
- During each negotiation cycle, the manager announces a task to all bidders and each bidder participates in the bidding. The manager awards the contract to one bidder and rejects other bids. The selected bidder submits the results of executing the contract.

If an iterative contract-net coordination mechanism is employed, the number of interactions should be calculated as $3(m-1)\times i+1$, where m is the total number of agents and i is the total number of iterative cycles during the coordination.

More details on the description of metrics to calculate dynamic quality attributes based on the combination of system architecture and the coordinate strategy employed will be presented in the case studies of this chapter.

Besides the number of interactions, a real-time performance indicator, the response time, is going to be evaluated in the system implementation stage. Response time is one of the most important real-time performance indicators for dynamic performance evaluation.

5.6 Case Studies

5.6.1 Architecture Modeling

To demonstrate the applicability of the evaluation strategies mentioned above, case studies are modeled based on the schematic architecture shown in Figure 5-1.

Agents in Figure 5-1 include utility agents and task agents. Task agents can be categorized into three layers. First-layer agents include the design coordinator agent, brokers and independent supplier coordinator agents (that communicate directly with the design coordinator agents, not through brokers). Supplier coordinator agents that act as child agents under brokers are second-layer agents. Other service provider agents in customer and supplier communities, i.e. the part design agent, die design agents, die manufacturing agents, stamping production agents and cost estimation agents, are categorized into the third-layer.

Compared with the system architecture proposed in Figure 3-1, Figure 5-1 does not include local nameserver agents and service agents. A visualiser agent is included as a utility agent for debugging and performance evaluation. A ZEUS agent society may contain any number of these utility agents, with at least one nameserver agent that is created to maintain the society-wide clock.

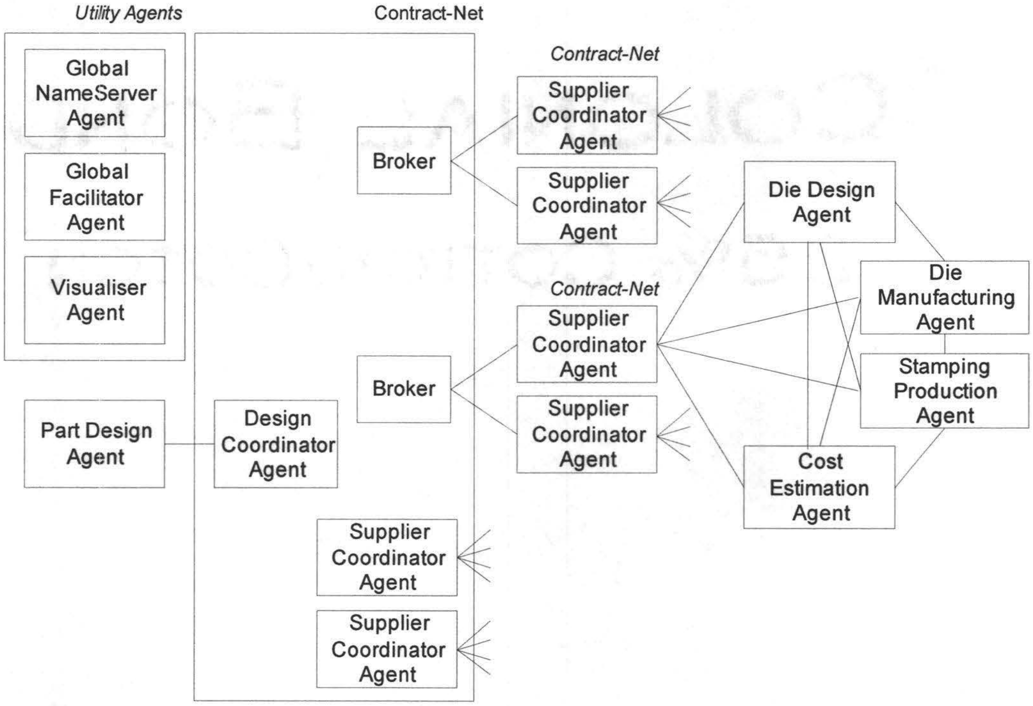


Figure 5-1. Schematic organizational structure

More simplifications have been made as follows:

1) Only one customer agent community is included.
2) The design coordinator agent communicates and coordinates with broker agents and independent supplier coordinator agents via the contract-net protocol.

3) Broker agents act as the root agents and communicate and coordinate with their subordinate supplier coordinator agents via the contract-net protocol.
4) Supplier coordinator agents under broker agents do not communicate and coordinate with the design coordinator agent directly but through the brokers.
5) Supplier coordinator agents are independent. They do not need to know the existence of other supplier coordinator agents or broker agents.
6) Each supplier coordinator agent has four subordinate agents, i.e. a die design agent, a die manufacturing agent, a stamping production agent and a cost estimation agent. The design coordinator agent only has one subordinate agent, i.e. a part design agent.

Table 5-2 describes the cases modeled based on the proposed schematic organizational structure for performance evaluation. There is no broker in cases 1 to 8 and there are two brokers in cases 9 to 24. In cases 1 to 8 and cases 17 to 24, only the first-layer supplier coordinator agents can be added or removed. In cases 9 to 16, only the second-layer supplier coordinator agent can be added or removed. In the case studies of this chapter, the supplier agent community should be regarded as the basic unit of service provider. A supplier coordinator agent will be added or removed together with its subordinates.

Table 5-2. Cases for performance evaluation

Case No.	Number of Design Coordinator Agents	Number of Brokers (n_b)	Number of Supplier Coordinator Agents under Broker *i*, $i=1,..,n_b$	Total Number of Supplier Coordinator Agents	Total Number of Task Agents (Excluding Utility Agents)
1	1	0	0	6	32
2	1	0	0	7	37
3	1	0	0	8	42
4	1	0	0	9	47
5	1	0	0	10	52
6	1	0	0	11	57
7	1	0	0	12	62
8	1	0	0	13	67
9	1	2	2,2	6	34
10	1	2	3,2	7	39
11	1	2	4,2	8	44
12	1	2	5,2	9	49
13	1	2	6,2	10	54
14	1	2	7,2	11	59
15	1	2	8,2	12	64
16	1	2	9,2	13	69
17	1	2	2,1	6	34
18	1	2	2,1	7	39
19	1	2	2,1	8	44
20	1	2	2,1	9	49
21	1	2	2,1	10	54
22	1	2	2,1	11	59
23	1	2	2,1	12	64
24	1	2	2,1	13	69

5.6.2 Evaluation on Static Quality Attributes of Systems

Static quality attributes, i.e. complexity, extendability and availability, are calculated based on metrics shown in Table 5-1.

Complexity is measured by the number of links of the agent-based system described in Figure 5-1. The number of links can be calculated by the sum of the following items:

1) Number of links between first-level agents and utility agents: $n \times n_u$, where n is the total number of the first-level agents; n_u is the number of enabled utility agents and equals 3 in case studies of this chapter.
2) Number of links among first-level agents: $n(n-1)/2$, where n is the total number of the first-level agents.
3) Number of links between broker agents and their subordinate agents: $\sum_{i=1}^{n_b} m_i$, where n_b is the total number of broker agents and m_i is the number of supplier coordinator agents under the ith broker agent.
4) Number of links among supplier coordinate agents and their subordinate agents: $\sum_{i=1}^{n_s+n_{bs}} p_i(p_i+1)/2$, where n_s is the total number of first-layer supplier coordinator agents; n_{bs} is the total number of subordinate supplier coordinator agents under broker agents (, i.e. $\sum_{i=1}^{n_b} m_i$); p_i is the number of agents under the ith supplier coordinator agent and equals 4 in case studies of this chapter.

5) Number of links between the design coordinator agent and its subordinate part design agent: 1.

From the above definitions of n, n_b and n_s, the following equation can be derived:

$$n = n_b + n_s + 1 \tag{5-1}$$

Extendability and availability are induced from the attribute of complexity, i.e. number of links. Extendability is measured by the change of the number of links when a supplier agent community is added. Availability is measured by the ratio of the number of links when a supplier agent community is removed versus the number of links of the original system.

Evaluation results for complexity, extendability and availability of systems are shown in Tables 5-3, 5-4 and 5-5. Figure 5-2 was generated from the data in Tables 5-3, 5-4 and 5-5 to graphically display the evaluation results.

Table 5-3. Results for static attributes of systems without broker agents

Case No.	1	2	3	4	5	6	7	8
Number of Supplier Agent Communities	6	7	8	9	10	11	12	13
Complexity	103	123	144	166	189	213	238	264
Extendability	20	21	22	23	24	25	26	27
Availability	0.816	0.837	0.854	0.867	0.878	0.887	0.895	0.902

Table 5-4. Results for static attributes of systems in which only the number of the second-layer supplier agent communities is changeable

Case No.	9	10	11	12	13	14	15	16
Number of Supplier Agent Communities	6	7	8	9	10	11	12	13
Complexity	90	101	112	123	134	145	156	167
Extendability	11	11	11	11	11	11	11	11
Availability	0.878	0.891	0.902	0.911	0.918	0.924	0.929	0.934

Table 5-5. Results for static attributes of systems in which only the number of the first-layer supplier agent communities is changeable

Case No.	17	18	19	20	21	22	23	24
Number of Supplier Agent Communities	6	7	8	9	10	11	12	13
Complexity	97	116	136	157	179	202	226	251
Extendability	19	20	21	22	23	24	25	26
Availability	0.814	0.836	0.853	0.866	0.877	0.886	0.894	0.900

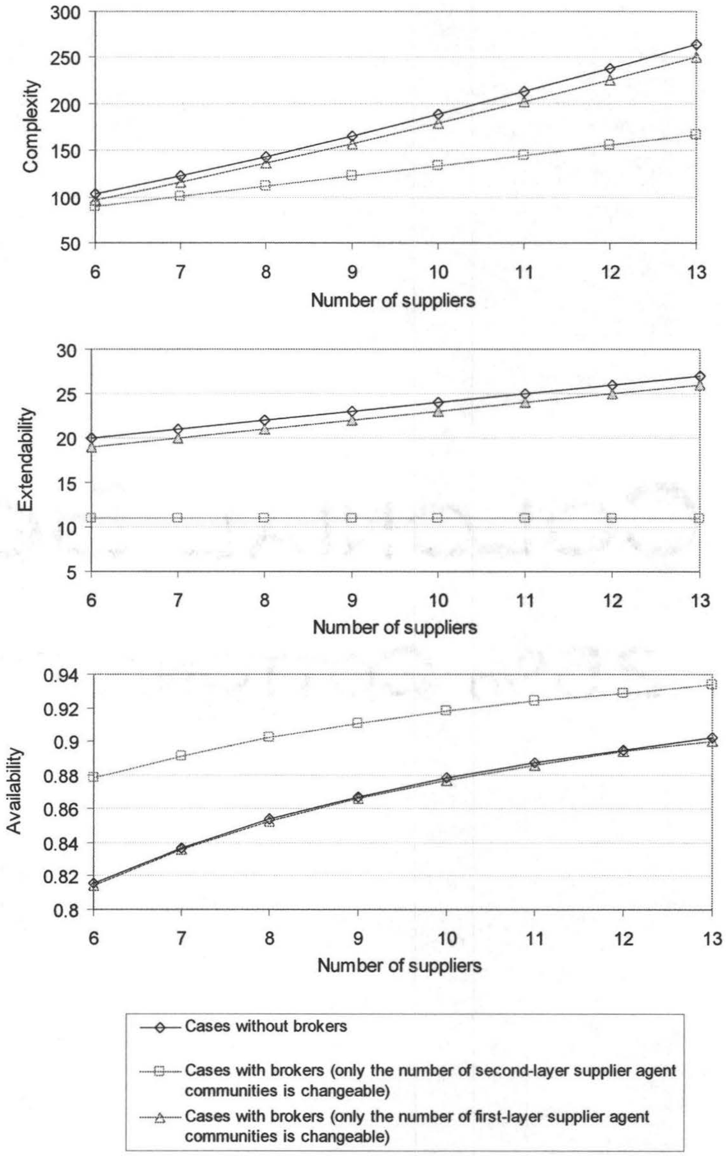


Figure 5-2. Evaluation results for static attributes

As shown in Figure 5-2, the number of links of systems without brokers grows faster as the number of suppliers increases. The reason is that the broker agents and their subordinate supplier coordinator agents are organized in a hierarchical architecture when brokers are employed in the hybrid architecture.

Figure 5-2 also shows that hybrid architectures can be easily extended when broker agents are employed and only the number of second-layer supplier agent communities under broker agents is changeable. The availability for this kind of architecture has a higher value. The hierarchical architecture reduces the communication between root agents and their subordinates. If only the number of first-layer supplier agent communities is changeable and the number of brokers and their subordinates is fixed, static features of such cases with or without broker agents are very close. The only difference is that there are 2 more broker agents in the cases with broker agents than in those without broker agents. For example, case no. 1 has 32 agents and case no. 17 has 34 as shown in Table 5-2.

5.6.3 Evaluation on Dynamic Quality Attributes

Dynamic quality attributes will be measured by the number of interactions and response time based on the combinations of organizational structures and coordination mechanisms. Similar to case studies in Chapter 3, a pseudo contract for the provision of the sheet-metal part is announced by the design coordinator agent and the negotiation strategy is based on the information about the process design and the cost for an initial design created by the part design agent. The pseudo contract is awarded to a broker agent or directly to a supplier coordinator agent in the first layer.

5.6.3.1 Number of Interactions

Interactions between utility agents and task agents are described in Figure 5-3. Task agents inform the name server agent of their addresses when they are added to the system. The facilitator agent periodically queries all the agents in the society about their abilities and task agents might take the initiative to advertise their abilities to the facilitator agent.

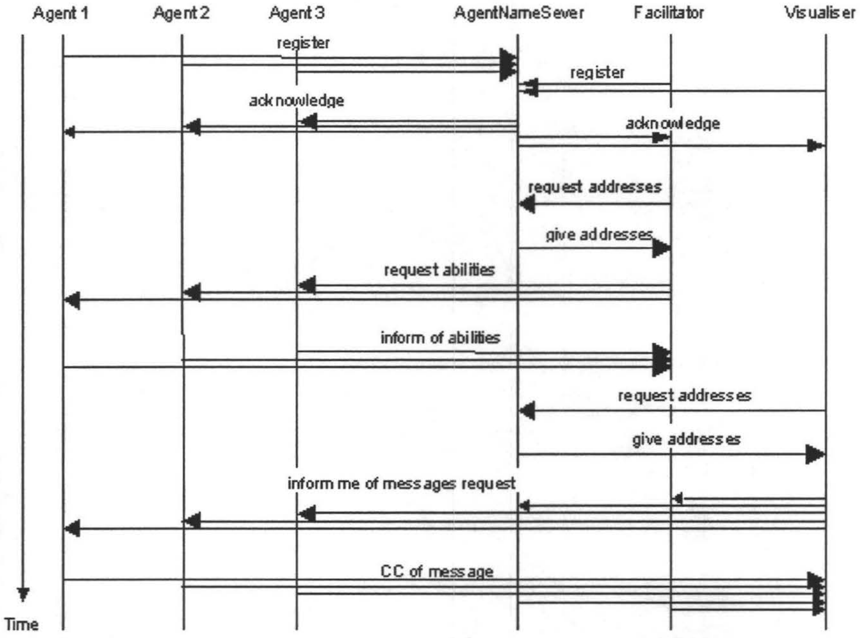


Figure 5-3. A schematic diagram of interactions of a ZEUS agent society (Collis and Ndumu, 1999)

In this case study, the following assumptions have been made to simplify the research problem:

1) The facilitator agent only queries task agents about their abilities at the initialization stage of the system;
2) The task agents do not take the initiative to update the facilitator with their abilities;
3) Visualiser agents are only designed for system debugging and analysis and the interactions between task agents and visualiser agents are not included for the calculation of the number of interactions.

In architecture models shown in Figure 5-1, only first-layer agents directly communicate with global utility agents. The design coordinator agent, first-layer supplier coordinator agents and broker agents record the addresses and abilities of their subordinate agents.

As shown in Figure 5-3, interactions between task agents and utility agents (excluding the visualiser agent) at the system initialization stage include:

1) The first-layer agents and the facilitator agent register with the nameserver agent and then the nameserver agent sends them acknowledgements. The number of these interactions is calculated as $2(n+1)$, where n is the total number of the first-layer agents.
2) The facilitator agent requests addresses for the first-layer agents and the nameserver agent replies with the names and addresses of the first-layer agents.

The number of these interactions is equal to 2 because only one nameserver agent and one facilitator agent are employed for this case study.

3) The facilitator agent queries the first-layer agents about their capabilities and those agents respond to its requests. The number of these interactions is calculated as $2n$, where n is the total number of the first-layer agents.

The design coordinator agent coordinates with other first-layer agents (i.e. broker agents and the first-layer supplier coordinator agents) via a non-iterative version of the contract-net protocol that is also employed for the coordination between the broker agents and their subordinate supplier coordinator agents.

Interactions from the time of service request to the time of service provision are summarized as follows:

1) The part design agent creates the initial design and sends it to the design coordinator agent. The design coordinator agent passes the feedback it receives from potential contractors to the part design agent. The part design agent updates its design based on the feedback and sends the updated design to the design coordinator agent when the iterative contract-net protocol is employed. After the execution of the task, the design coordinator agent sends the result to the part design agent. To sum up, the number of interactions between the part design agent and the design coordinator agent can be calculated as 2 if a non-iterative version of the contract-net protocol is employed.

2) The design coordinator agent queries the facilitator about potential contractors that have the capability to fulfill the task and the facilitator responds to the query. Then, the design coordinator agent makes a request to the nameserver agent on addresses of those potential contractors and the nameserver agent replies. The number of interactions occurring in this period is 4.
3) The design coordinator agent negotiates with potential contractors, i.e. the broker agents and the first-layer supplier coordinator agents. The number of interactions between the design coordinator agent and those potential contractors is calculated as $3(n_b + n_s) + 1$, where n_b is the total number of broker agents and n_s is the total number of first-layer supplier coordinator agents.
4) Broker agents negotiate with their subordinate supplier coordinator agents. If none of the broker agents is awarded the contract, the number of interactions occurring in this period is calculated as $3n_{bs}$, otherwise, the number of interactions is $3n_{bs} + 1$, where n_{bs} is the total number of subordinate supplier coordinator agents under broker agents.
5) Agents in a supplier agent community coordinate on the tasks of process design and cost estimation. Those agents undertake sub-tasks of die design, die manufacturing, stamping production and cost estimation. As mention above, a pseudo contract is set for case studies and the supplier coordinator agent that is awarded the contract does not really produce the part. The number of interactions is calculated by $\sum_{i=1}^{n_s+n_{bs}} C_i$, where C_i is the number of interactions

occurring among agents in the i^{th} supplier agent community. Based on the assumptions made for the structure shown in Figure 5-1, C_i is equal to 13. The interactions are described as follows:

i) The supplier coordinator agent informs all the four subordinates, i.e. the die design agent, the die manufacturing agent, the stamping production agent and the cost estimation agent, that process design and cost estimation for an initial design are going to be conducted. Four interactions occur here.

ii) The stamping production agent creates the plan for stamping production and sends the equipment information and process information to the die design agent, the cost estimation agent and the supplier coordinator agent (for progress monitoring). Three interactions occur.

iii) The die design agent works on die design and sends the results to the die manufacturing agent, the cost estimation agent and the supplier coordinator agent. Three interactions occur here.

iv) The die manufacturing agent generates the manufacturing process based on the die design and sends its results to the cost estimation agent and the supplier coordinator agent. Two interactions occur.

v) The cost estimation sends the result of cost estimation to the supplier coordinator agent. One interaction occurs.

To sum up, if none of the broker agents is awarded the contract, the equation to calculate the number of interactions (N) is:

$$N = 2(n+1) + 2 + 2n + 2 + 4 + (3(n_b + n_s) + 1) + 3n_{bs} + \sum_{i=1}^{n_s + n_{bs}} C_i$$

$$= 4n + 3n_b + 16n_s + 16n_{bs} + 11 \tag{5-2}$$

Considering Equation 5-1, Equation 5-2 can be simplified to:

$$N = 7n_b + 20n_s + 16n_{bs} + 15 \tag{5-3}$$

If a broker agent wins the contract, one more interaction needs to be added. This situation is ignored for simplification and therefore Equation (5-3) is employed for the following calculation of number of interactions for cases shown in Table 5-2.

Tables 5-6 to 5-8 present the evaluation results based on number of interactions. Figure 5-4 graphically shows the evaluation results. When the number of supplier agent communities is over 6, i.e. cases 9 to 16, in which broker agents are employed and only the number of second-layer supplier agent communities (under broker agents) is changeable, have lower values of number of interactions. The reason is that fewer interactions are required with the involvement of broker agents because each broker agent works as a superior agent and knows the capabilities and addresses of its subordinate supplier agent communities. Cases 1 to 8 and cases 17 to 24 have very close values of number of interactions although 2 brokers are included in cases 17 to 24. The involvement of brokers decreases the number of interactions if the total numbers of agents are the same.

Table 5-6. Numbers of interactions for systems without brokers

Case No.	1	2	3	4	5	6	7	8
Number of Supplier Agent Communities	6	7	8	9	10	11	12	13
Number of Interactions based on Non-Iterative CNP	135	155	175	195	215	235	255	275

Table 5-7. Numbers of interactions for systems in which there are brokers and only the number of the second-layer supplier agent communities is changeable

Case No.	9	10	11	12	13	14	15	16
Number of Supplier Agent Communities	6	7	8	9	10	11	12	13
Number of Interactions based on Non-Iterative CNP	133	149	165	181	197	213	229	245

Table 5-8. Numbers of interactions for systems in which there are brokers and only the number of the first-layer supplier agent communities is changeable

Case No.	17	18	19	20	21	22	23	24
Number of Supplier Agent Communities	6	7	8	9	10	11	12	13
Number of Interactions based on Non-Iterative CNP	137	157	177	197	217	237	257	277

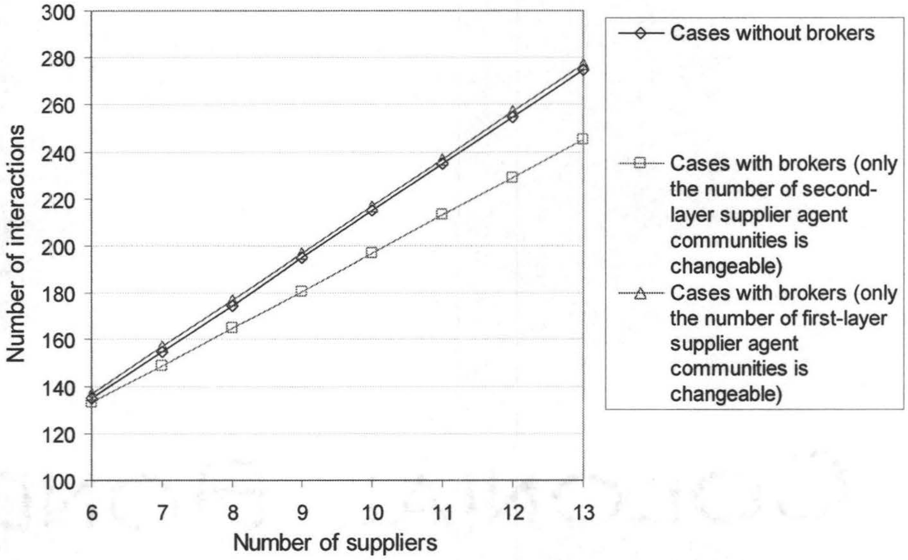


Figure 5-4. Comparison of number of interactions

5.6.3.2 Response Time

Evaluation results based on the above static attributes provide a preliminary assessment of the system's performance. For more details on the real-time performance, dynamic attributes, such as response time, average load of agents, etc., need to be studied. As a typical performance factor, response time is very importance for system evaluation and is going to be studied in this case study.

To implement the systems as shown in Figure 5-1 with the ZEUS agent building toolkit, one computer can be sufficient for hosting all agents in a small-scale system. However, the system's performance degrades significantly as the number of agents grows. Pervious studies demonstrate that the degradation in the system performance is a polynomial function of the number of tasks and agents and the degradation begins to grow dramatically with the load as the number of tasks and agents both exceeds a critical point because memory required by each agent exceeds the limitation (Lee, et al., 1998). Since the system is usually well distributed for sheet-metal product development, there is not a high possibility to host a lot of agents with one machine. Only small-scale systems are going to be studied. Cases modeled for the evaluation of the system's response time are shown in Table 5-9.

All cases shown in Table 5-9 can be categorized into three groups. Two computers running Windows XP are employed for the case study. Group 1, which includes cases 1 to 6, is hosted on one computer only. Group 2 (cases 5-10) and group 3 (cases 11-16) are distributed on two computers considering the balance of the number of agents. Both computers have 256 MB of RAM. The hardware is sufficient when the case study is focused on small-scale systems and simplifications can be made to the prototype agent-based system for this case study.

To simplify the research problem, the prototype system for this case study does not include Mechanical Desktop as the user interface. A pre-saved text format file is used as the initial design by the part design agent and is passed to the design coordinator agent as the part of request for manufacturability evaluation and cost estimation. The response time is calculated from the time when the request is sent to the design coordinator agent to the time when the request has been satisfied and all the subsequent activities (such as sending rejections to bidders that can not get the contract) are cleared. Idle die manufacturing agents are included. Die design agents only undertake the tasks of nesting and the determination of the strip layout. Rough cost models presented in Chapter 3 are used for cost estimation by cost estimation agents.

Table 5-9. Cases for the evaluation of response time

Case No.	Number of Design Coordinator Agents	Number of Brokers (n_b)	Number of Supplier Coordinator Agents under Broker *i*, $i=1,..,n_b$	Total Number of Supplier Coordinator Agents	Total Number of Task Agents and Utility Agents
1	1	0	0	1	10
2	1	0	0	2	15
3	1	0	0	3	20
4	1	0	0	4	25
5	1	0	0	5	30
6	1	0	0	6	35
7	1	0	0	7	40
8	1	0	0	8	45
9	1	0	0	9	50
10	1	0	0	10	55
11	1	2	1,2	5	32
12	1	2	2,2	6	37
13	1	2	3,2	7	42
14	1	2	4,2	8	47
15	1	2	5,2	9	52
16	1	2	6,2	10	57

The relationship of response time versus number of supplier agent communities for Groups 1, 2 and 3 is shown in Figures 5-5 to 5-6.

Figure 5-5 shows that the system's performance degrades significantly as the number of suppliers is over 3 and the total number of agents is over 20. That is reasonable because the implementation of a multi-agent system with 20 agents consumes about 160 MB of memory and the performance of the computer is very poor with so many active processes of "java.exe".

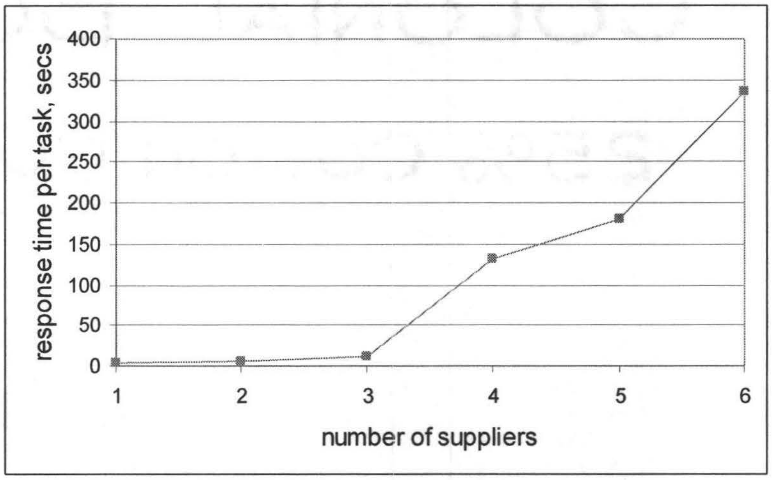


Figure 5-5. Response time versus the number of suppliers for group 1

Figure 5-6 shows that response times needed for groups 2 and 3 are very close. Cases in group 3 need longer response time compared with cases in group 2 because:

- Two more agents (i.e. broker agents) are included in cases in group 3 and therefore more memory is consumed and the system's performance degrades in accordance.
- Broker agents coordinate with their subordinate supplier agent communities via the contract-net protocol. Some delay occurs when broker agents initiate the new

request for proposal, wait for and evaluate the proposals from their subordinates. There is more idle time existing in the coordination.

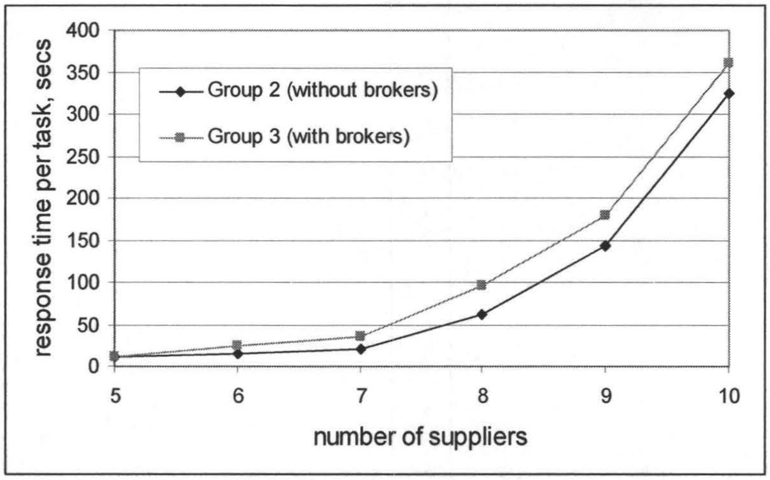


Figure 5-6. Response time versus the number of suppliers for groups 2 and 3

The evaluation results may vary very much if different hardware and agent distributions are applied. The simplifications made to the prototype system play a significant role in the evaluation of response time.

5.7 Conclusions

This chapter presents a systematic approach on performance evaluation of agent-based sheet-metal product development systems. Static and dynamic quality attributes and metrics have been defined and the impact of organizational structures and coordination mechanisms has been studied based on cases modeled with the proposed schematic organizational structure. Real-time system performance is statistically studied based on the data collected by the visualiser agent generated with the ZEUS agent building toolkit. The performance evaluation approach can be used in the preliminary system design stage and the later system implementation stage.

The results presented in this chapter demonstrate the applicability of evaluation strategies proposed and can be used as a reference model for performance and scalability analysis on agent-based sheet-metal product development systems. The performance evaluation approach presented can be used for the selection of organizational structures and coordination mechanisms and real-time performance of overall system and each agent can be statistically studied so that high quality systems can be created. The efficiency of system architecture and coordination strategies can be further investigated based on other important performance indicators, e.g. the average load of agents, i.e. the proportion of time spent in executing tasks.

The proposed evaluation approach is also applicable to general applications for product development by taking into consideration other performance indicators.

Chapter 6. Summary, Conclusions and Future Work

This chapter summarizes the work in this research, draws conclusions and offers some recommendations for future research.

6.1 Summary

Based on the analysis of the characteristics of the sheet-metal product development process, demands on integrated collaborative product development environments for the sheet-metal industry were summarized. With these demands in mind, this research aimed to develop an integrated design and analysis methodology for sheet-metal product development based on agent-based technology, feature-based design, optimization and finite element analysis techniques, and to study the performance of prototype systems developed based on such an integrated methodology. To achieve the research objectives, research questions (Questions1-6) to be answered were proposed in the Introduction of this thesis. Research work done to answer those research questions is summarized as follows.

- In response to Question 1 (What should be specified as agents in the sheet-metal product development system?), roles played by participants among the sheet-metal product development process were analyzed and assigned to specific agents which wrap typical design and manufacturing analysis functions, such as part design, unfolding, nesting, strip layout planning, die design, formability analysis and cost estimation.
- Based on the analysis of the roles of participants and the coordination style among participants in sheet-metal product development, a federated agent-based system

architecture was proposed and the ZEUS agent building toolkit was selected for the fast development and implementation of such prototype systems in response to Question 2 (What kind of architecture should the agent-based system adopt for sheet-metal product development? How are agents structured internally? How do agents wrap existing functionalities provided by legacy systems or databases?). The prototype system included task agents and utility agents. Task agents were categorized into two groups, customer agent community and supplier agent community. Design coordinator agents and supplier coordinator agents played the roles of local facilitators. Other task agents, such as part design agents, die design agents, etc., were equipped with problem solving capabilities and knowledge to implement specific tasks. Utility agents included facilitator agents and nameserver agents. Facilitator agents store the abilities of agents, receive and respond to queries from agents about the abilities of other agents. Nameserver agents receive and respond to agents' requests for the addresses of other agents and maintain a society-wide clock. Both task agents and utility agents can be called ZEUS agents, which are equipped with communication and coordination mechanisms. ZEUS agent wrapping technology was adopted for encapsulation of legacy functionalities, such as part and die modeling tools, data and knowledge management systems.

- For Question 3 (How should agents communicate? What communication protocols should agents use?), agents communicated using messages that obey the

FIPA ACL specification and TCP/IP was selected as the network communication protocol in this research.

- As for Question 4 (How should agents coordinate their actions?), the contract-net protocol, the most widely used cooperation and coordination method in agent-based systems, was adopted in this research as the agent coordination strategy.
- The implementation of the prototype agent-based collaborative product development system was presented in Chapter 3. A simple sheet-metal part (with a single wall and two holes) was selected for the case study to demonstrate the performance of the prototype agent-based system based on the integrated design and analysis methodology.
- To answer Question 5 (What design optimization methods are suited for agent-based sheet-metal product development? How do agents collaborate to solve optimization problems?), in Chapter 4, the prototype system was extended by adding a group of optimization agents to the supplier agent communities to study how agent coordination can facilitate the optimization process of tooling design for sheet-metal forming. A cyclic data flow framework based on an A-Teams approach (Talukdar et al., 1996) was employed to organize those optimization agents. Optimization agents coordinated by sharing getting initial points from and posting optimization results to a shared database controlled by an optimization management agent in the prototype system. Three traditional optimization algorithms, simulated annealing, genetic algorithm and the downhill

simplex method, were adopted for the optimization. A popular objective function, deviation of thickness from unique average and two sets of typical design variables, blankholder forces and drawbead restraining forces were used for case studies. Thickness constraints and formability constraints were included in the optimization model. To facilitate the implementation of simulated annealing and genetic algorithm approaches, a static penalty function method was used to transform the constrained optimization problem to a new constrained optimization model with only constraints of ranges of design variables. A deep drawn rectangular cup and an automobile front door panel were selected for case studies.

- To investigate Question 6 (How should the performance of the agent-based sheet-metal product development systems be evaluated?), a systematic evaluation methodology was presented in Chapter 5 for the architecture modeling and performance evaluation of agent-based collaborative systems especially for sheet-metal product development or other related applications. This methodology is an extension to Lee and Huang (2004)'s evaluation approach. Both static and dynamic attributes of such collaborative product development systems were evaluated for the purpose of the performance prediction in the preliminary system design stage and the performance validation in the system implementation stage. Real-time system performance was statistically studied based on the data collected by the visualiser agent generated with the ZEUS agent building toolkit. Case studies were designed to investigate the applicability of the proposed performance evaluation strategies.

6.2 Conclusions

Based on the forgoing summary of this dissertation, the following conclusions have been drawn from this research.

Agent-Based Collaborative Sheet-Metal Product Development

- Generally, the production of sheet-metal parts has the requirements of large-batch sizes and short lead times. Life-cycle activities in sheet-metal product development are usually implemented by different industrial entities with different expertise. Feedback from low-level (e.g. process planning, die design and manufacturing, stamping production) to the high level design (e.g. part design) usually leads to re-designs of part and/or die.
- A collaborative product development environment is needed to reduce lead times and costs, and to increase product performance and accuracy of sheet-metal products.
- The agent-based approach can provide solutions for the following requirements: low cost, high quality, short time-to-market, integration of legacy systems and confidentiality requirement of participants, and is appropriate for facilitating the communication and coordination among participants.

Role Allocation in Agent-Based Collaborative System for Sheet-Metal Parts

- Based on the initial application analysis, developers determine what responsibilities agents should fulfill. Responsibilities undertaken by participants in the supply chain of sheet-metal parts should be assigned to specific agents to ensure that agents in the collaborative system can work autonomously.

- Meanwhile, agents should be equipped with the ability to communicate and coordinate with other agents in the system.

Agent-Based System Architecture and the Internal Structure of Each Agent

- The federated architecture is suitable to develop open, scalable multi-agent system architectures and was proposed (in Chapter 3) for the construction and performance study of a prototype system to demonstrate the functionality of such a system.
- For the fast development of agent-based systems, ZEUS agent building toolkit was selected for the construction and performance study of the prototype systems in this research. It was found that ZEUS toolkit is very suitable for developing light-weight collaborative design systems for real applications.
- Utility agents, i.e. facilitator agents and nameserver agents, are easily created with the ZEUS toolkit to provide the service of storing the abilities of agents, receiving and responding to queries from agents about the abilities or addresses of other agents, and maintaining a society-wide clock.
- In this research, agents generated with the ZEUS toolkit included the following components: the communication mechanism, the coordination engine, the planner, the internal event model, and the connection mechanisms to external (legacy) systems. The comprehensive internal structure of agents is applicable for real complex applications.

Agent Communication

- The FIPA ACL specification has proved to be a satisfactory standard in this research.
- TCP/IP, the network communication protocol used in this research, has proved very suitable for the construction of light-weight agent-based systems.

Agent Coordination

- The contract-net protocol used in this research is suitable for developing open, complex and dynamic systems with a large number of agents due to the flexibility requirements of such systems.

Construction and Performance Study of the Prototype Agent-Based Collaborative System

- The construction and performance studies of the prototype system (presented in Chapter 3) demonstrated that the development techniques and tools, such as ZEUS toolkit, Autodesk Mechanical Desktop and ObjectARX, COM, MySQL and its APIs, are suitable for developing light-weight collaborative design systems for real applications.
- The performance of the prototype system has demonstrated that the agent-based approach has the ability to seamlessly integrate activities involved in the product development process of sheet-metal parts.

Agent-Based Optimization

- It was found that the performance of each agent (and optimization technique) depended strongly on the complexity of the problem. More interestingly, for a given amount of computational effort, a network of collaborating agents using different optimization techniques always outperformed agents using a single technique in terms of both the best solution found and the variance of the collection of best solutions.

- The cyclic data flow framework has been demonstrated to be a natural way for data sharing among optimization agents.
- Result analysis has demonstrated that the agent-based optimization approach works well with both simple and relatively complex process optimization problems.
- The modeling techniques of case studies and performance evaluation approaches (presented in Chapter 4), such as the distribution of best points and the comparison of impacts of optimization agents, can provide significant fundamental basis for the deployment of agents and the selection criteria of the termination criteria of optimization algorithms and other optimization algorithms that may be applicable to such optimization problems.

System Performance Evaluation

- The proposed performance evaluation method, which combines the evaluation of static and dynamic attributes and the real-time system performance, makes it possible for the performance evaluation to be performed at the preliminary system design stage and the performance validation in the system implementation stage, and can be used as a reference approach for architecture modeling and performance evaluation of general agent-based collaborative design and manufacturing systems.

- Taking into consideration the complexity and dynamics of sheet-metal product development, a hybrid system architecture, which combines federated, hierarchical and autonomous system architectures, is suitable for developing prototype or real systems for sheet-metal product development in this research.

6.3 Recommendations on Future Research

The following recommendations are offered for future research:

- Optimization of the deployment of agents, especially for agent-based optimization cases, is recommended for future research. Performance evaluation and agent impact analysis based on the system performance evaluation strategies proposed in Chapters 4 and 5 can provide the fundamental basis for this research.
- Static coordination strategies were applied in this research. Termination criteria were set based on random tests. Further studies are recommended to develop an “adaptive” coordination strategy to adjust the termination criteria of specific optimization algorithms involved in agent-based optimization based on real-time performance evaluation results.
- Results of agent-based process optimization have shown that simulated annealing and genetic algorithms involved in this research are computationally expensive for complex optimization problems. Further research on the selection and combination of optimization algorithms is recommended. Gradient-based optimization algorithms can be recommended for agent-based optimization and approximation-based optimization methods, e.g. Response Surface Methodology (RSM) (Redhe et al., 2002), can be used for computation-intensive problems.

- Incremental FEA solvers should be used for cases that are complex and have high requirement on the accuracy of simulations. Incremental FEA solvers can also be used for validation of process design in the late stage of product design and process design.

Due to the time and cost limitation of this research, some simplifying assumptions were made and some other important issues were not studied for the development of the prototype agent-based collaborative systems. There is room to improve the prototype system before implementing it in industry, although those improvements are not expected to affect the conclusions of this research. More features need to be added to make the system to be able to handle general sheet-metal parts and more knowledge for part and tooling design is needed for real applications. Security and learning issues are not studied in this research, but are important for real applications. Security is a common issue for Web-based and agent-based applications. Learning can be used to enhance the performance of agent-based systems.

Chapter 7. References

A-Teams, 2007, http://www.cs.cmu.edu/afs/cs/project/edrc-22/project/ateams/WWW/ (accessed on November 16, 2007).

Autodesk Inc., 2004, "ObjectARX Software Development Kit", http://objectarx.autodesk.com/ (accessed on October 14, 2004).

Ayed, L.B., Delameziere A., Batoz, J.L. and Knopf-Lenoir, C., 2005, "Optimization of the blankholder force with application to Numisheet'99 front door panel", VIII International Conference on Computational Plasticity, E. Onate and D. R. J. Owen (Eds), Barcelona.

Aytug, H., Khouja, M. and Vergara, F. E., 2003, "Use of genetic algorithms to solve production and operations management problems: a review", International Journal of Production Research, Vol. 41, No.17, pp. 3955 - 4009.

Batoz, J. L., Guo, Y. Q. and Mercier, F., 1995, "The inverse approach including bending effects for the analysis an design of sheet metal forming parts", NUMIFORM'95, A. A. Balkema, Rotterdam, Netherlands.

Bauer, B., 1999, "FIPA TCC Agent Communication - Extending UML for the specification for the Agent Interaction Protocols", FIPA/OMG.

Brinbaum, L., Hammond, K., Marlow, C. and Yang, H., 1999, "Integrating Diverse Information Resources in a Case-Based Design Environment", Engineering Applications of Artificial Intelligent, 12: 705–716.

Brown, D.C., Dunskus, B., Grecu, D.L., Berker, I., 1995, "SINE: support for single function agents", Proceedings of the Applications of AI in Engineering, Udine, Italy.

Camacho, David and Aler, Ricardo, 2005, "Software and Performance Measures for Evaluating Multi-Agent Frameworks", Applied Artificial Intelligence, 19: 645-657.

Campbell, M.I., Cagan, J. and Kotovsky, K., 1999, "A-Design: An Agent-Based Approach to Conceptual Design in a Dynamic environment", Research in Engineering Design, Vol. 11, pp. 172-192.

Cao, Junwei, Kerbyson, Darren J. and Nudd, Graham R., 2001, "High Performance Service Discovery in Large-Scale Multi-Agent and Mobile-Agent Systems", International Journal of Software Engineering and Knowledge Engineering, 11(5): pp. 621-641.

Cheok, B. T. and Nee, A. Y. C., 1998, "Trends and developments in the automation of design and manufacture of tools for metal stampings", Journal of Materials Processing Technology, 75(1-3), pp. 240-252.

CHI 1997, 2007, http://www.acm.org/sigchi/chi97/proceedings/panel/jrm.htm (accessed on November 16, 2007).

Chin, Kwai-Sang and Tang, Dunbing, 2002, "Web-Based Concurrent Stamping Part and Die Development", Concurrent Engineering: Research and Applications, Vol. 10, No. 3, pp. 213-228.

Chung, J.S. and Hwang, S.M., 1997, "Application of a Genetic Algorithm to the Optimal Design of the Die Shape in Extrusion", Journal of Materials Processing Technology, 72, pp. 69-77.

Collis, Jaron and Ndumu, Divine, 1999, ZEUS Technical Manual, Intelligent Systems Research Group, BT Labs.

Cutkosky, M. R., Englemore, R. S., Fikes, R. E., Gruber, T. R., Genesereth, M. R., Mark, W. S., Tenenbaum, J. M. and Weber. J. C., 1993, ''PACT: An experiment in integrating concurrent engineering systems.'' IEEE Computer, Vol. 26, No. 1, pp. 28-37.

Cutkosky, M. R., Tenenbaum, J. M. and Glicksman, J., 1996, "Madefast: Collaborative Engineering over the Internet", Communication of the ACM, 39(9), pp. 78-87.

Davidsson, P., Johansson, S. J., Persson, J. A. and Wernstedt, F., 2003, "Agent-based Approaches and Classical Optimization Techniques for Dynamic Distributed Resource Allocation", 2nd International Joint Conference on Autonomous Agents and Multiagent Systems Workshop - Representations and approaches for time-critical decentralized resource/role/task allocation, Melbourne, Australia.

Delameziere, 2002, "On the Optimization of Process Parameters in Sheet Metal Stamping Using a Simplified Analysis Simulator", APOMAT Working Group on Optimization Methodologies, Budapest, November 28-29.

Deshpande, S., and J. Cagan, 2004, "An Agent Based Optimization Approach to Manufacturing Process Planning", ASME Journal of Mechanical Design, Vol. 126, No. 1, pp. 46-55.

Durfee, E.H., Lesser, V.R. and Corkill, D.D., 1989, "Trends in Cooperative Distributed Problem Solving", IEEE Transactions on Knowledge and Data Engineering, KDE-1(1), pp. 63-83.

FIPA, 2005, http://www.fipa.org (accessed on August 14, 2005).

Fruchter, R., Reiner, K.A., Toye, G. and Leifer, L.J., 1996, "Collaborative Mechatronic System Design", Concurrent Engineering: Research and Applications, Vol. 4, No. 4, pp. 401-412.

FTI, 2002, http://www.forming.com (accessed on February 12, 2002)

FTI, 2004, http://www.forming.com (accessed on August 14, 2004).

Gaines, B., Norrie, D. and Lapsley, A., 1995, "Mediator: an intelligent information system supporting the virtual manufacturing enterprise", Proceedings of 1995 IEEE International Conference on Systems, Man and Cybernetics, New York, pp. 964-969.

Genesereth, M. R. and Ketchpel, S. P., 1994, Software agents, Communications of the ACM, 37(7):48-53.

Goodwin, G. M., 1968, "Application of Strain Analysis to Sheet Metal Forming Problems in the Press Shop," La Metallurgica, Vol. 60, pp. 767-774.

Green, S., Hurst, L., Nangle, B., Cunningham, P., Somers, F. and Evans, R., 1997, "Software Agents: A review", Dublin, Intelligent Agents Group Trinity College Dublin.

Guo, Y. Q., Batoz, J. L., Naceur, H., Bouabdallah, S., Mercier, F. and Barlet, O., 2000, "Recent Developments on the Analysis and Optimum Design of Sheet Metal Forming Parts Using a Simplified Inverse Approach", Computers and Structures, 78, pp. 133-148.

Hao, Q., Shen, W., Zhang, Z., Park. S.W. and Lee, J.K., 2006, "Agent-Based Collaborative Product Design Engineering: An Industrial Case Study", Computers in Industry, 57, pp. 26-38.

Hindman, C. and Ousterhout, K. B., 1998, "A Virtual Design System for Sheet Metal Forming", Journal of Engineering Materials and Technology, Vol. 84, pp.107-116.

Holland, J.H., 1975, "Adaption in Natural and Artificial Systems", MIT Press (1992 Edition).

Homaiffar A., Qi C. and Lai S., 1994, "Constrained Optimization Via Genetic Algorithms", Simulation, 62:4, pp. 242-254.

Hu, Zunnian, 2001, "Optimization of Blanks for Sheet Metal Forming Using the Simulated Annealing Method", Master Thesis, McMaster University.

Hu, W. and Poli, C., 1999, "To Injection Mold, to Stamp, or to Assemble? A DFM Cost Perspective", Journal of Mechanical Design, 121: 461-471.

Huh, H. and Kim, S. H., 2001, "Optimum Process Design in Sheet-Metal Forming with Finite Element Analysis", Journal of Engineering Materials and Technology, Vol. 123, No. 476, pp.476-481.

JADE, 2007, http://jade.tilab.com (accessed on November 16, 2007).

Jain, P., Fenyes, P. and Richter, R., 1992, “Optimal blank nesting using simulated annealing”, Journal of Mechanical Design, 114 (1), pp.160--165.

JATLite, 2005, http://java.stanford.edu (accessed on August 14, 2005).

Jennings, N. R., 1996, “Coordination Techniques for Distributed Artificial Intelligence”, in: Foundations of Distributed Artificial Intelligence. Wiley, pp. 187-210.

JESS, 2007, http://herzberg.ca.sandia.gov/jess/ (accessed on November 16, 2007).

Keeler, S. P., 1965, "Determination of Forming Limits in Automotive Stampings," Sheet Metal Industries, Vol. 42, pp. 683-691.

Khuri, A. I. and Cornell, J. A., 1996, Response Surfaces: designs and analyses (Statistics Textbooks and Monographs Series, Vol 152).

Kirkpatrick, S., Gelatt, C. D. Jr., Vecchi, M. P., 1983, “Optimization by Simulated Annealing”, Science, Vol. 220, pp. 671-680.

Kobayashi, S. and Kim, J. H., 1978, “Deformation analysis of axisymmetric sheet metal forming processes by the rigid-plastic finite element method”, Mechanics of Sheet Metal Forming, Plenum Press, New York, pp. 341-363.

Kok, S. and Stander, N., 1999, “Optimization of a sheet metal forming process using successive multipoint approximations”, Structural Optimization, Vol. 18, Issue 4, pp. 277-295.

KQML, 2007, http://www.cs.umbc.edu/kqml/ (accessed on November 16, 2007).

Kuri-Morales, A. F. and Gutiérrez-Garcia, J., 2001, "Penalty functions methods for constrained optimization with genetic algorithms: a statistical analysis", Proc. 2nd Mexican International Conference on Artificial Intelligence, Springer-Verlag, Heidelberg, Germany, pp. 108–117.

Lazaro, A.S., Engquist, D.T. and Edwards, D.B., 1993, "An Intelligent Design for Manufacturability System for Sheet-Metal Parts", Concurrent Engineering: Research and Applications, 1: 117–123.

Lee, C. H. and Cao, J., 2001, "Shell element formulation of multi-step inverse analysis for the design of axisymmetric deep drawing process", International Journal for Numerical Methods in Engineering Eng., 50, pp. 681-706.

Lee, L. C., Nwana, H S, Ndumu, D T and De Wilde, P., 1998, "The Stability, Scalability and Performance of Multi-Agent Systems", BT Technology Journal, Vol 16 No 3.

Lee, S. K. and Hwang, C. S., 2004, "Architecture Modeling and Evaluation for Design of Agent-Based System", Journal of Systems and Software, Vol. 72, pp. 195-208.

Lee, R.S., Chuang, L.C., Yu, T.T. and Wu, M.T., 1995, "Development of an Assessment System for Sheet Metal Forming", Proceedings of the International Conference on Precision Engineering, pp. 515–518, Singapore.

Li, Caiyuan, Li, Jianjun, Wen, Jianyong and Xiao, Xiangzhi, 2001, "HPRODIE: using feature modelling and feature mapping to speed up progressive die design", International Journal Production Research, Vol. 39, No. 18, pp. 4133-4151.

Ling, Y.E., Lee, H.P. and Cheok, B.T., 2001, "Web-Based Metal Forming Advisory System for the Finite Element Analysis of Sheet Metal Bending", Proceedings of the Institution of Mechanical Engineers. Part B, Journal of Engineering Manufacture, 216: 39–45.

Liu, G., Lin, Z. and Bao, Y., 2002, "Optimization Design of Drawbead in Drawing Tools of Autobody Cover Panel", Journal of Engineering Materials and Technology, Vol. 124, pp. 278-285.

Mackerle, J., 2004, "Finite Element Analyses and Simulations of Sheet Metal Forming Processes", Engineering Computations, 21 (7-8), pp. 891-940.

Mantripragada, M., Kinzel, G. and Altan, T., 1996, "A computer-aided engineering system for feature-based design of box-type sheet metal parts", Journal of Materials Processing Technology, Vol. 59, pp. 241–248.

Marciniak Z. and Duncan, J. L., 1992, Mechanics of Sheet Metal Forming, Edward Arnold, London.

Marciniak Z. and Kuczynski, K., 1967, "Limit strains in processing of stretch forming sheet metal", International Journal of Mechanical Sciences, Vol. 9, pp. 609–620.

Maturana, F. and Norrie, D., 1996, "Multi-Agent Mediator Architecture for Distributed manufacturing", Journal of Intelligent Manufacturing, Vol. 7, pp. 257-270

Maturana, F., Shen, W. and Norrie, D. H., 1999, "MetaMorph: an adaptive agent-based architecture for intelligent manufacturing", international Journal of Production Research, Vol. 37, No. 10, pp. 2159-2173.

Metropolis, N. and Rosenbluth, A. and Rosenbluth, R. and Teller, A. and Teller, E., 1953, Equation of state calculations by fast computing machines J. Chem. Phys. 21:1087-1092

Microsoft Corp., 2007, "Component Object Model technologies", http://www.microsoft.com/Com/default.asp/ (accessed on November 16, 2007).

Mitchell M., 1996, "An Introduction to Genetic Algorithms", MIT Press, Cambridge, MA.

Mouatassim, M. El, Jameux, J. –P., Thomas, B. and Pasquale, E. Di, 1995, "An industrial finite element code for one-step simulation of sheet metal forming", NUMIFORM'95, A. A. Balkema, Rotterdam, Netherlands.

MySQL AB, 2003, MySQL Reference Manual, http://www.mysql.com/doc/en/index.html (accessed on 09 May, 2003).

Naceur, H., Delaméziere, A., Batoz, J.L., Guo, Y.Q. and Knopf-Lenoir C., 2004, "Some improvements on the optimum process design in deep drawing using the inverse approach", Journal of Materials Processing Technology, 146, pp. 250-262.

Nelder, J. A. and Mead, R., 1965, "A simplex method for function minimization", Computer Journal, Vol. 7, pp. 308-313.

Numisheet'99, 1999, http://www.numisheet99.univ-fcomte.fr.

Numisheet'99 benchmarks, 1999, Numerical Simulation of 3D Sheet Forming Processes, Editors: Gelin, J. C., Picart, P., Volume 2, Burs Edition, Besancon, France.

Nwana, H., Lee, L. and Jennings, N., 1996, "Coordination in Software Agent Systems", BTTechnology Journal 14(4): 79-88.

Ohata, T., Nakamura, Y., Katayama, T., Nakamachi, E. and Nakano, K., 1996, "Development of Optimum Process Design System by Numerical Simulation", Journal of Materials Processing Technology, 60, pp. 543-548.

Ohata, T., Nakamura, Y., Katayama, T., Nakamachi, E. and Omaori, N., 1998, "Improvement of Optimum Process Design System by Numerical Simulation", Journal of Materials Processing Technology, 80-81, pp. 635-641.

Oliveira, E., Fischer, K. and Stepankova, O., 1999, "Multi-agent systems: which research for which applications.", Robotics and Autonomous Systems, 27: 91-106.

Parunak, H. V., 1998, "What can Agents do in Industry, and Why? An Overview of Industrially-Oriented R&D at CEC", CIA'98.

Parunak, H. V. D., Suter, J., Fleischer, M., Ward, A., 1999, "The RAPPID Project: Symbiosis between Industrial Requirements and MAS Research", Autonomous Agents and Multi-Agent Systems, 2, 111-140.

Poli, C., Dastidar, P., Mahajan, P. and Graves, R.J., 1993, "Design for Stamping – Analysis of Part Attributes that Impact Die Construction Costs for Metal Stampings", Journal of Mechanical Design, 115: 735–743.

Rai, Rahu and Allada, Venkat, "Modular product family design: agent-based Pareto-optimization and quality loss function-based post-optimal analysis", International Journal Production Research, 2003, Vol. 41, No. 17, 4075-4098.

Redhe, M., Forsberg, J., Jansson, T., Marklund, P.-O. and Nilsson, L., 2002, "Using the response surface methodology and the D-optimality criterion in crashworthiness related problems- an analysis of the surface approximation error versus the number of function evaluations", Structural Multidiscipline Optimization, 24, 185–194.

Roos, N., 2006, "Agent Communication Languages", http://www.fdaw.unimaas.nl/education/4.1MAS/Slides/ACL1.pdf (accessed on June 12, 2006)

Roy, S., Ghosh, S. and Shivpuri, R., 1997, "A new approach to optimal design of multi-stage metal forming processes with micro genetic algorithms", International Journal of Machining Tools and Manufacturing, Vol. 37, No. 1, pp. 29-44.

Sandia National Laboratories, 2003, http://herzberg.ca.sandia.gov/jess/index.shtml (accessed on 09 May, 2003).

Serenko, A. and Detlor, B., 2002, "Agent toolkits: a general review of the market and an assessment of Instructor satisfaction with utilizing toolkits in the classroom", Working Paper #455, July, 2002.

Shen, W. and Barthes, J.-P., 1995, "DIDE: A Multi-agent Environment for Engineering Design", Proc. First Int'l Conf. Multiagent Systems, AAAI Press, Menlo Park, Calif., pp. 344-351.

Shen, W., Kremer, R., Ulieru, M. and Norrie, D., 2003, "A Collaborative Agent-Based Infrastructure for Internet-Enabled Collaborative Enterprises", International Journal of Production Research, Vol. 41, No. 8, pp. 1621-1638.

Shen, W. and Norrie, D.H., 1999, "Agent-Based Systems for Intelligent Manufacturing: A State-of-the-Art Survey", Knowledge and Information Systems, an International Journal, 1(2), 129-156.

Shen, W. and Wang, L., 2003, "Web-based and agent-based approaches for collaborative product design: An overview", International Journal of Computer Applications in Technology, Vol. 16, Nos. 2/3, pp. 103-112.

Shen,W., Xue, D. and Norrie, D. H., 1998, "An agent-based manufacturing enterprise infrastructure for distributed integrated intelligent manufacturing systems", Proceedings of the Third International Conference on the Practical Application of Intelligent Agents and Multi-Agents (PAAM'98), London, UK, pp. 533-548.

Shneiderman, B., 1981, "Direct manipulation: A step beyond programming languages", Proceedings of the joint conference on Easier and more productive use of computer systems. (Part - II): Human interface and the user interface - Volume 1981.

Siirola, J. D., Hauan, S., Westerberg, A. W., 2003, "Toward Agent-Based Process Systems Engineering: Proposed Framework and Application to Non-Convex Optimization", Computers & Chemical Engineering, 27, pp. 1801-1811.

Smith,R., 1980, "The Contract Net Protocol: High Level Communication and Control in Distributed Problem Solver," IEEE Transactions on Computers, Vol. 29, No. 12, pp. 1104-1113.

Talukdar S., Baerentzen L., Gove A., de Souza P., 1996, "Asynchronous Teams: Organizations for Algorithmic Computation", EDRC Tech-Report 18-56-96, EDRC Carnegie Mellon University, Pittsburgh, PA.

Tang, D.-B., 2004, "An agent-based collaborative design system to facilitate active die-maker involvement in stamping part design", Computers in Industry, Vol. 54, Issue 3, pp. 253-271.

Tang, D.-B. and Gao, B., 2007a, "Feature-based metal stamping part and process design. Part I: stampability evaluation", International Journal of Production Research, Vol. 45, No. 12, pp. 2673-2695.

Tang, D.-B. and Gao, B., 2007b, "Feature-based metal stamping part and process design. Part II: stamping process planning", International Journal of Production Research, Vol. 45, No. 13, pp. 2997-3015.

Tang, D.-B., Zheng, L. and Li, Z.-Z., 2001, "An Intelligent Feature-Based Design for Stamping System", International Journal of Advanced Manufacturing Technology, 18(3): 193–200.

Tekkaya, A. Erman, 2000, "State-of-the-art of simulation of sheet metal forming", Journal of Materials Processing Technology, 103, pp. 14-22.

Thiyagarajan, N. and Grandhi, R.V., 2005, "Multi-Level Design Process for 3-D Preform Shape Optimization in Metal Forming", Journal of Materials Processing Technology, 170, pp. 421-429.

Toye, G., Cutkosky, M., Leifer, L., Tenenbaum, J. and Glicksman, J., 1993, "SHARE: A methodology and environment for collaborative product development", Proceedings of 2nd

Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprise, IEEE Computer Society Press, pp. 33-47.

Vamanu, Carmen, 2003, "Drawbead and Binder Force Optimization in Draw Die Design", Master Thesis, McMaster University.

Wang, N. –M. and Wenner, M. L., 1978, "Elastic-viscoplastic analysis of simple stretch forming problems", Mechanics of Sheet Metal Froming, Plenum Press, New York, pp. 367-398.

Weiss, G., 2000, "Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence", MIT Press; 1st edition.

Wiederhold, G., 1992, "Mediators in the architecture of future information systems", IEEE Computer, 25(3), pp. 38-49.

Wooldridge, M. and Jennings, N. R., 1995, "Intelligent agents: theory and practice", The Knowledge Engineering Review, 10, pp. 115-152.

Xie, S.Q., Tu, P.L., Aitchison, D., Dunlop, R. and Zhou, Z.D., 2001, "A WWW-Based Integrated Product Development Platform for Sheet Metal Parts Intelligent Concurrent Design and Manufacturing", International Journal Production Research, 39(17), pp. 3829–3852.

Yamamoto, G. and Nakamura, Y., 1999, "Architecture and Performance Evaluation for a Massive Multi-agent System", Proceedings of Autonomous Agents 99, pp. 319-325.

Zhang, Y., 1999, "A Distributed Object-Oriented Blackboard-Based Framework for Multiagent Systems", Master's Thesis, McMaster University.

Appendix A ZEUS Agent Building Toolkit

ZEUS agent components, agent architecture, communication mechanism, ontology definition and knowledge representation are discussed as follows:

ZEUS Components

The generic ZEUS agent includes a set of components as shown in Figure A-1. Components of the generic ZEUS agent include:

- the communication mechanism,
- the co-ordination engine,
- the planner,
- the internal event model, and
- the connection mechanisms to external (legacy) systems.

ZEUS toolkit provides comprehensive graphical user interfaces including an ontology editor, an agent definition editor, a task description editor, an organization editor and a co-ordination editor. These tools speed up the development of the agent-based systems.

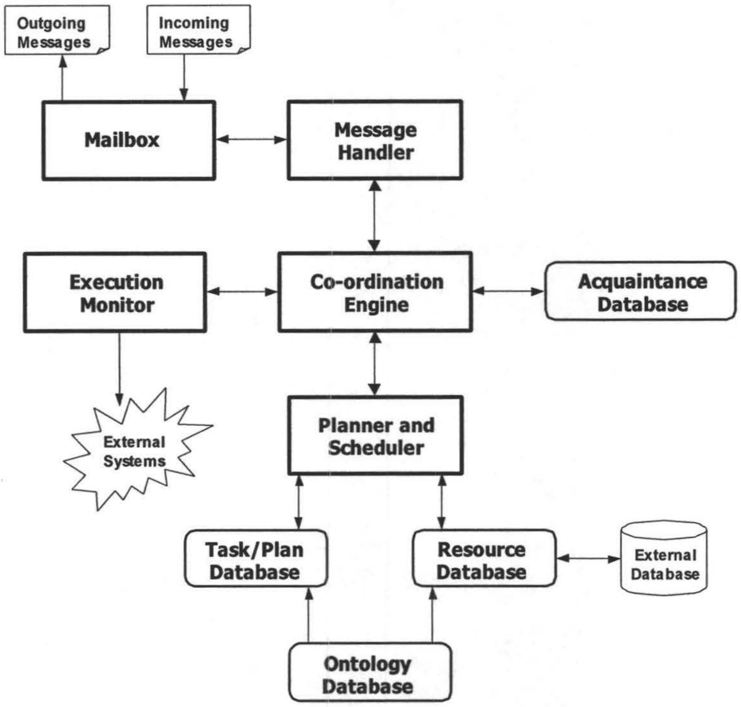


Figure A-1. Architecture of a generic ZEUS agent (Collis and Ndumu, 1999)

<u>ZEUS Utility Agents</u>

ZEUS toolkit provides an infrastructure of utility services. The ZEUS suite of utility agents consists of a name server and a facilitator agent that facilitate information discovery, and a visualiser agent for visualising or debugging societies of ZEUS agents. A ZEUS agent society may contain any number of these utility agents, with at least one name server agent. All three utility agents can be constructed using the basic components of the Agent Component Library, and are in fact simplifications of the generic ZEUS agent.

Responsibilities of facilitator agents include:

- To store the abilities of agents;
- To receive and respond to queries from agents about the abilities of other agents.

The Nameserver agent can play the following roles:

- To receive and respond to agents' requests for the addresses of other agents;
- To maintain a society-wide clock.

Visualiser agents can be used to view, analyze or debug societies of ZEUS agents. They function by querying other agents about their states and processes, and then collating and interpreting the replies to create an up-to-date model of the agents' collective behavior.

Agent Communication Languages

The need for collaboration between agents occurs for several reasons. Most are rooted in the problem of scarcity of resources – computing, information, know-how, etc. Since individual agents possess different resources and capabilities, a solution to a given problem may be beyond the capabilities of any one agent, requiring that a number of agents pool their resources and collaborate with one another in order to solve the problem (Collis and Ndumu, 1999).

An agent-independent inter-agent communicating language is needed for agent communication. All ZEUS agents communicate using messages that obey the FIPA 1997 ACL specification. The syntax of the FIPA ACL message is shown in Figure A-2.

```
Performative(
        type:           /* performative type, e.g. inform, cancel etc. */
        sender:                 /* name of agent sending message */
        receiver:               /* name of intended recipient agent */
        reply_with:     /* sender's conversation identification key */
        in_reply_to:    /* recipient's conversation key */
        content:                /* message content */
        language:       /* name of language in which content is expressed */
        address:                /* sender's address */
        send_time:      /* time at which message is sent */
        receive_time:   /* time when message is received */
        ⋮
        )
```

Figure A-2. Performative class (Collis and Ndumu, 1999)

<u>Ontology and Knowledge Representation</u>

Agents that communicate in a common language still have to use the same ontology or vocabularies for representing common domain concepts in order to understand one another. This can be achieved by creating general-purpose ontologies or by creating domain-specific ontologies and using inter-ontology translators to map between domain-specific ontologies (Collis and Ndumu, 1999).

ZEUS provides an ontology editor for defining the ontology items. ZEUS uses the term 'fact' to describe an individual domain concept. Fact objects are defined in terms of their attributes and the valid value ranges for each attribute. Attribute values can be primitive types, lists, other facts or constraint expressions that should ultimately resolve into a primitive type, list or fact.

For ZEUS agents, knowledge is stored in form of instances of 'ontology facts' using the Java Expert System Shell (JESS) (JESS, 2004).

Coordination Protocols and Strategies

Agent communication and coordination are the key problems of multi-agent system development. ZEUS toolkit provides pre-built coordination protocols, such as Contract-Net Protocol (Smith, 1980) and other auction protocols.