University of Strathclyde

Research on Key Techniques of Flexible Workflow based Approach to Supporting Dynamic Engineering Design Process

By

Dongbo Wang

Thesis submitted to the University of Strathclyde, Glasgow for the Degree of

Doctor of Philosophy

Department of Design, Manufacture and Engineering Management

University of Strathclyde, Glasgow, Scotland, UK

November 2013

Signed: Dongbo Wang

Date: Feb 6th, 2016

Acknowledgements

In the deep night of Shanghai city in China, when I nearly complete my thesis, I can’t help thinking back the tough but fulfilling every day and night pursuing my PhD, I am always feeling so thankful to different people who care and help me during my PhD study sincerely.

I do like to express my extremely gratitude to my respected supervisor Prof. Xiu-Tian Yan for giving me this golden opportunity to study for split PhD training in University of Strathclyde, for his numerous profound suggestions and careful modifications on my research work and paper published, especially for his encouragement when I nearly want to give up my PhD study. I cannot forget every discussion on my papers and thesis whether in his office, by email or by telephone over thousands miles from UK to China, especially when Prof. Xiu-Tian Yan helped me over night to modify my paper published in International Journal of Design Engineering. Great thanks to Professor Jonathan Corney, Professor Jiju Antony and Dr Iain Boyle for their insight understanding and encouragement during the research reviews these years. And I also want to thank Mrs Paula Harvie for her continued and important help during my PhD study. Sincere thanks also go to University of Strathclyde for giving me such excellent research environment.

I do also like to thank my co-supervisor Prof. Runxiao Wang, Vice President of Northwestern Polytechnical University (NPU) in China, for his carefully teaching and serious guidance. Both his active and open academic thinking and responsible attitude toward research gave me a great help. I really appreciate the help from Prof. Xiansheng Qin, in Northwestern Polytechnical University, who give me tremendous help in both my PhD study and my work. I do also like to thank open source code project NickLee.Framework for useful help to the thesis.

Thank my Parents, brother and all my family members, thank for your endless support with my growth. Finally I would like to express my deep gratitude to all the people who care and help me.

Abstract

Engineering design process (EDP) is a highly dynamic and creative process, and the capability in managing an EDP is considered as a major differentiating factor between competing enterprises. The most important prerequisite to establish an engineering design process excellence is a proper management of all the design process activities and the associated information. The most important impact in recent years on the EDP and on the activities of designers has come from computer-based data processing. Workflow, the automation of a business processes in whole or part, is a useful tool for modelling and managing a business process which can be reprensented by a workflow model (computerized process definition). By considering the dynamic characteristics of EDP, an EDP management system must be flexible enough to support the creative and dynamic EDP.

After the introduction of engineering design process and its new trend, as well as flexible workflow technology, reviews of both engineering design process and its supporting flexible workflow technology shows that there is a need for a holistic framework to automate and coordinate design activities in the creative and dynamic EDP, and the flexible workflow technology should also be improved comprehensively in flexibility and intelligence in order to support better engineering design management.

By introducing the relations between the EDP and flexible workflow, a virtual workflow and an autonomic flexible workflow built upon autonomic computing is investigated, and an innovative engineering design process management framework based on multi-autonomic objects flexible workflow is proposed. For the flexible workflow modelling in the framework, a dynamic

instance-based flexible workflow modelling method is proposed for multi-autonomic objects flexible workflow. In order to improve the intelligence of flexible workflow, after examining the principle of flexible workflow intelligence in flexible workflow, a new flexible workflow autonomic object intelligence algorithm based on both extended Mamdani fuzzy reasoning and neural network is proposed, weighted fuzzy reasoning algorithm, as well as precise and fuzzy hybrid knowledge reasoning algorithm is designed; a bionic flexible workflow adaptation algorithm is proposed to improve the intelligence of autonomic object flexible workflow further. According to the characteristic of EDP, such as cross-enterprises and geographical distribution, and in order to realize the flexible execution of distributed flexible workflow engine, a distributed flexible workflow engine architecture based on web service is proposed and a flexible workflow model description method based on extended WSDL (Web Service Description Language) and BPEL4WS (Business Process Execution Language for Web Services) is proposed. A flexible workflow prototype system supporting engineering design process is implemented according to the proposed EDP management framework in Microsoft VS.Net 2005 environment. The framework is demonstrated by the application in an EDP of a MTO company, and it shows that the proposed framework can support the creative and dynamic process in an efficient way. Finally, the strengths and weakness of the framework as well as the prototype system is discussed based on the results of the evaluation, and the proposed areas of future work are given.

Table of Contents

Chapter 1 Introduction

Design is an engineering activity that affects almost all areas of human life. It uses the laws and insight of science; builds upon special experience and provides the prerequisites for the physical realization of solution ideas (Pahl and Beitz 2003). Engineering design is a purposeful activity directed towards the goal of fulfilling human needs in the form of a PRODUCT, particularly those which can be met by technological factors of our civilisation (Rehman, Duffy, and Yan 2008). The design and development department is of central importance in any company. Designers determine the properties of every product in terms of function, safety, ergonomics, production, transport, operation, maintenance, recycling and disposal. In addition, designers have a large influence on production and operating costs, on quality and on production lead times.

Design as a verb is also interpreted as 'Design Process', it is a process consisting of problem solving activities in which design solutions are generated to satisfy customer needs". There are mainly four phases which are outlined below in Engineering Design Process (EDP): planning and clarifying the task, conceptual design, embodiment design and detail design (Pahl and Beitz 2003). EDP is a highly dynamic process for the following reasons: firstly, the EDP is an evolving process, and then the design evolves as it progresses from an initial concept to a more detailed design; secondly, there are many uncertainties which can happen in the engineering design process; thirdly, the creative design processes are hardly predictable and can only be pre-planned on a coarse level; fourthly, the engineering design process is an intensive decision making process and is heavily depended on the design context which may change regularly with new decisions; finally but most importantly, with the advances of new design strategies, such as concurrent engineering and collaborative design, the engineering design process becomes more complex with more relations among design activities to manage, and designers always work in a team environment which add another dimension of management complexity (Germani, Mandolini, Mengoni, Peruzzini 2010).

Some research found that almost 75% of a designer's work consists of seeking, organizing, modifying and translating information (Hurst, 1999), often unrelated to his own personal discipline. The capability in managing an EDP is considered as a major differentiating factor between competing enterprises (Marquardt and Nagl 2004). The most important prerequisite to establish a design process excellence is a proper management of all the design process activities and the associated information. So the efficient management of the engineering design process is very important.

The most important impact in recent years on the design process, and on the activities of designers, has come from computer-based data processing. Not only can computer-aided design influence design method, the computer-based system can reengineer a new design process and provide efficient means of supporting engineering design process management, such as popular Product Data Management (PDM) system. Workflow as a specialist research field is concerned with the automation of a business process in whole or part, and is a useful tool for modelling and managing a business process. It can enforce tightly coupled co-operation between team members according to a pre-defined logical process between activities (Zhuge 2003).The workflow can management and coordinate the EDP in an efficient way and can automate the EDP, and send the suitable information to the relevant designers in the design process. The workflow technology has been widely applied in the PDMs (Li 2014).

But considering the dynamic characteristic of EDP, the conventional workflow technology cannot satisfy the requirements of the creative and complex EDP very well due to its fixed logic approach. An EDP management system must be flexible enough to support the dynamic EDP. How to improve the flexibility and intelligence of the design workflow to be able to adapt to the changes of the EDP has become an important research topic in both the engineering design and workflow research area (Muller, Greiner and Rahm 2012). The flexible workflow is commonly defined as the workflow which can respond and satisfy the changes quickly without the redefinition of the workflow model in the condition of environment, condition and process status change. The flexibility is mainly realized by the change of workflow node functions, the change of the workflow logic and dynamic extension of the workflow. The flexible workflow can then support the change of design function, the change of the design process and the dynamic joining and exit of the design activity in EDP, so that the flexible workflow can support the creative and dynamic engineering design process satisfactorily. The key to this is the generation of a flexible and dynamic computer-based EDP workflow management system (Lindow, Woll and Stark 2012).

In a dynamic EDP, all the design process activities and associated information should be structured in a purposeful way that is in a clear sequence of main phases and individual working steps, and the design process should be controlled and coordinated simultaneously. The EDP management framework should not only support the dynamic changes in creative and complex EDP but also adapt the changes intelligently. This requires that decision making activities in an EDP should also be fully supported, and then the EDP can be evolved by new decision making. With the workflow execution and the above arguments, the EDP management framework based on flexible workflow can be implemented in a computer-based system which can automate and coordinate design activities in the creative and dynamic EDP. This thesis proposes an EDP management framework based on flexible and intelligent workflow concept, aiming that the proposed framework can support better the modelling, coordinating, and controlling of dynamic and creative engineering design process. The framework will provide a new approach to supporting integrated design information and design activity management using workflow technology, can provide team designers with more effective tools to cope with rapid changes, iteration and evolution in EDP and exhibit the required intelligence for design managers in managing all the design activities.

1.1 Research Motivation, Aim, Objectives and Methodology

The efficient support to the dynamic and creative engineering design process, the management of coordination of inter-related design activities, and the decision making with the evolution of engineering design process are the key to the dynamic engineering design process management framework. The understanding of relations between the engineering design process and workflow can be important premise to construct the proposed framework based on flexible workflow technology. The suitable engineering design process definition or modelling in the proposed framework requires the methods to model evolving flexible workflow. Furthermore the flexible workflow should facilitate the complex decision making in the design process, and guide designers on how the workflow-based framework can be implemented and applied in the engineering design process. All these present a demand to research further in the field of flexible workflow supporting dynamic engineering design process.

Finding a more efficient approach and proposing a useful framework based on a flexible workflow to support specifically and effectively dynamic and creative engineering design process is the primary motivation that the author has decided to undertake this PhD research project. This motivation naturally leads to the following research aim definition and structured objectives of the project.

1.1.1 Research Aim and Objectives

The overall research aim is to understand better the links between the engineering design process and workflow, and to propose a computational framework for supporting better dynamic engineering design process management. This support will be devised by proposing, implementing and validating a flexible and intelligent workflow management system. By executing such a workflow system, the engineering design process management software based on the computational framework can provide better support in conducting all the engineering design activities, and satisfying the needs for better design decision making.

To achieve the above aim, the following detailed specific objectives have been specified in this research:

- To understand the characteristics of engineering design process and its relations with the flexible workflow;
- To undertake a critical review of existing methods, associated framework and software systems which support the engineering design process management, as well as flexible workflow supporting engineering design process, in order to identify their deficiencies/shortcoming;

- To propose an engineering design process management framework based on flexible and intelligent workflow, understand the detailed framework composing structure, the principle of workflow flexibility and intelligence;
- To study the flexible workflow modelling method according to the characteristics and demands of engineering design process, and give a meta-model of flexible workflow. Based on the meta-model, define the flexible workflow model by its activities, control links and data links and their graphical model expressions;
- To understand the role of decision making in the dynamic engineering design process, undertake the fundamental principle of flexible workflow intelligence, formalize the expression of workflow, propose an suitable innovative intelligence algorithm in proposed flexible workflow and its collaborative reasoning algorithm;
- To apply the web service technology in engineering design process management framework based on flexible workflow, detail the framework of distributed flexible workflow and describe the flexible workflow model based on web service;
- To design and implement a flexible workflow management prototype system supporting engineering design process, design the architecture and modules of the prototype system, as well as database design of the prototype system, and finally to evaluate and demonstrate the developed approach in a customer truck design process;
- To discuss and suggest future research directions and research areas.

1.1.2 Research Methodology

In order to have a profound understanding of the research, different methods are used to achieve different objectives and they form an integrated research methodology detailed below:

- Literature review: literature review aims to review the critical research problem from the related literature. And it can bring unbiased and comprehensive view of the previous research work. In this thesis, literature review on the current approaches to the engineering design process management, as well as the supporting flexible workflow key technologies is undertaken;

- Research problem and challenge investigation: investigation is a good way to have deep and broad understanding on what is the research question, and an investigation in chosen design teams or companies at the early stage of research is conducted;

- Problem identification: problem identification in this research is to propose and confirm challenging research problem in dynamic engineering design process management framework based on key techniques of flexible workflow, and the implementation of computer-based system;

- Research Problem definition: the research problem definition is a process of formally specifying the statement mainly on how to use flexible workflow to support the dynamic engineering design process, and how to make flexible workflow intelligent enough to satisfy the requirement of decision making in engineering design process;

- Solution generation: the solution generation of the research is to solve the research problem by developing an approach to and proposing a flexible workflow approach deploying modern computer technologies and advanced algorithms in deriving an engineering design process management framework based on flexible workflow;

- Modelling: modelling is an important means to analysing and describing the inner mechanism and relations of the research problem, in which the mathematic and graphical modelling method is adopted to describe the structure of both framework and its supporting flexible workflow;
- Generating algorithms: A suitable intelligent algorithm is the key to supporting the decision making in the engineering design process. A decision making algorithm in the framework will be proposed to suggest the suitable action of flexible workflow when change happens in engineering design process;
- Simulation: simulation is kind of useful means to demonstrating and undertaking what-if studies of the effect of proposed model and algorithm. Suitable simulation tools will be identified in this thesis to testify the related decision making algorithm in engineering design process;
- Case study and solution evaluation: the evaluation of the developed solution is to be performed using case studies through the proposed computer-based system. Based on the evaluation results, the strengths and weaknesses of the system in supporting engineering design process management will be identified and recommendations is suggested;
- Thesis writing: the result of the research is to be documented in this thesis and in some published papers.

1.2 Structure of the Thesis

To serve the above research aim and objectives of the research, the thesis is structured as follows and shown in figure 1-1:

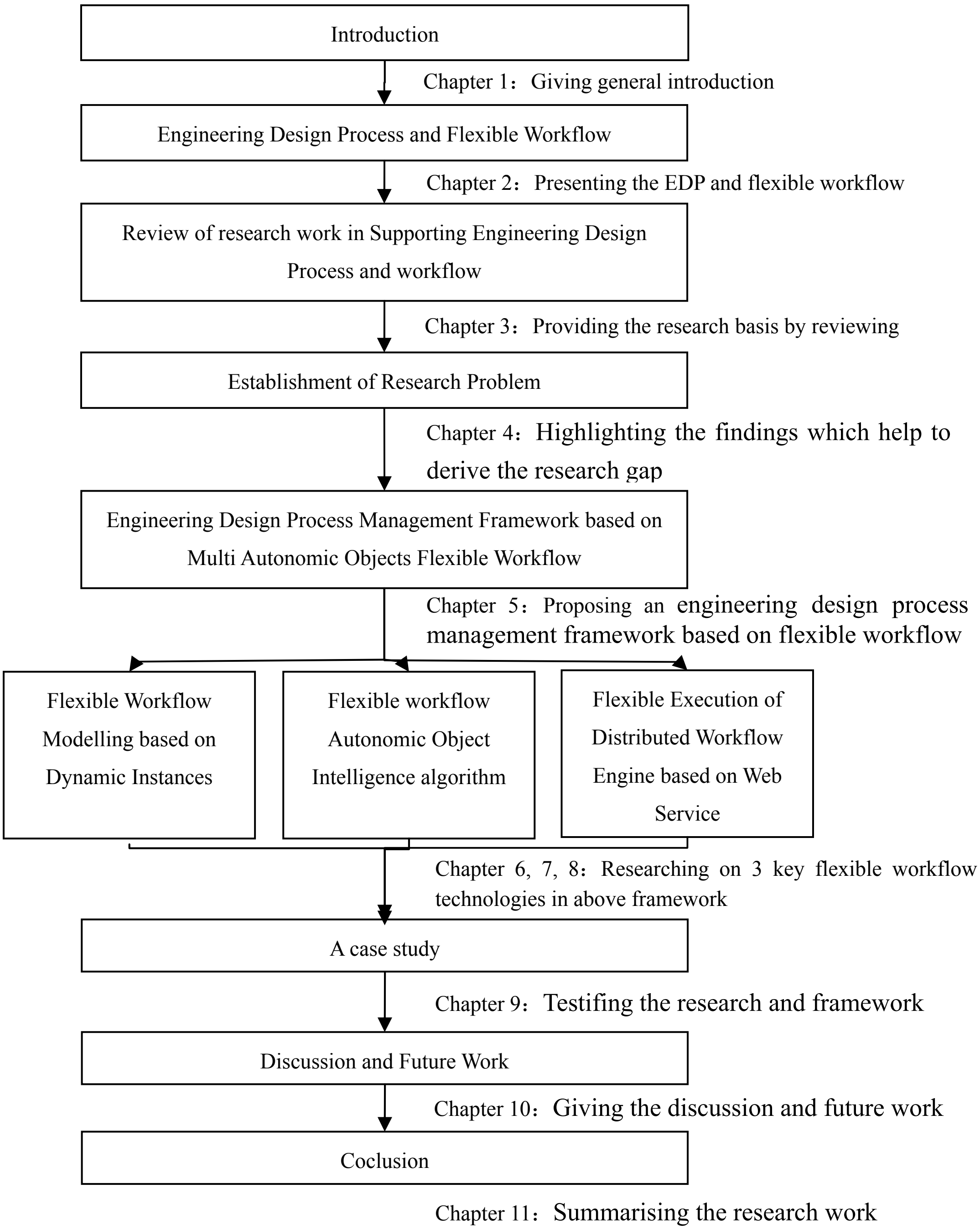


Figure 1-1 Thesis Structure

- Chapter 1 presents the motivation, aim and objectives of the research. The motivation of the research is based on the current design problem considering the engineering design process management supported by workflow technologies;
- Chapter 2 introduces the engineering design in general and discusses the engineering design process with focus, including the engineering design process model, characteristics of engineering design process, then introduces the basic idea of flexible workflow and its management system, gives the in-depth analysis of flexible workflow technologies, all these help to give sound background and basis for further research;
- Chapter 3 mainly reviews the framework of engineering design process supporting by different methods at first, then reviews the flexible workflow and its enabling technology, such as modelling technology, decision making algorithm, and web service technology all of which can support the flexibility and intelligence of workflow;
- Chapter 4 highlights the finding of review, these findings can help to derive the research problem formulation and form the research questions, and the research contents is also outlined to direct the follow-on research work;
- Chapter 5 firstly introduces the relations between the dynamic engineering design process and flexible workflow based on the description and analysis of engineering design process and its management. The virtual workflow and flexible workflow based on autonomic object mainly built upon autonomic computing is given, and the engineering design process management framework based on multi-autonomic objects is proposed, the modelling methods of the framework is investigated further;

- Contrast to the traditional two stage workflow modelling methods, Chapter 6 describes two new workflow modelling methods, namely evolutional modelling and dynamic instance modelling. Based on dynamic instance modelling, a systematic flexible workflow modelling approach is proposed, including its meta-model, flexible workflow activity, autonomic object, control logic, data logic, and hierarchical activity model which are formally defined in detail. Finally the inner structure of flexible workflow activity and graphical expression of workflow model is given;

- Chapter 7 explains the principle of flexible workflow intelligence in flexible workflow change expression, autonomic object monitoring and execution, autonomic object reasoning, and so on. The flexible workflow autonomic object intelligence algorithm based on both extended Mamdani fuzzy reasoning and neural network is proposed. The Decision Making Trial and Evaluation Laboratory method is introduced in the proposed algorithm to compute the independence weight of autonomic object knowledge. The weighted fuzzy reasoning algorithm, as well as precise and fuzzy hybrid knowledge reasoning algorithm is designed.

- Chapter 8 proposes distributed flexible workflow engine architecture based on web service, and its realization method by using SOAP (Simple Object Access Protocol) and WSDL (Web Service Description Language) is given. By considering the characteristics of flexible workflow model and web service, the WSDL and BPEL4WS specification is extended respectively. The flexible workflow model description method based on extended WSDL and BPEL4WS is also described. Finally flexible workflow execution algorithm and dynamic extension algorithm are also given and illustrated.

- Chapter 9 details the implementation of a flexible workflow prototype system supporting integrated product development in Microsoft VS.Net 2005 environment, and the architecture, main modules and the database design of the prototype system are described and illustrated in detail. Finally an application of the intelligent and flexible EDP framework in an EDP of a Make to Order (MTO) Company testifies that the framework can support the creative and dynamic design process in an efficient way.
- Chapter 10 discusses the strengths and weakness of the approach as well as of the prototype system based on the results of the evaluation, and lists the areas of future work in the selected research field.
- Chapter 11 concludes the thesis by summarising the work done in the research.

1.3 Chapter Summary

This chapter presents the brief introduction of the thesis. After introducing the research motivation, aim, objectives and research methodology, the structure of the thesis is give in detail.

Chapter 2 Engineering Design Process and Flexible Workflow

The aim of this chapter is to provide a review on the engineering design and engineering design process, analysing the characteristics of the engineering design process, and reviewing the flexible workflow which potentially can support the engineering design process. In the first section, the engineering design is briefly introduced; the engineering design process and its characteristics is described and analysed in the second section; the flexible workflow and its key techniques is discussed especially for understanding its support of engineering design in the third section.

2.1 Engineering Design

The Oxford Dictionary defines the word ‘Design’ as:

- “A preliminary plan or sketch for the making or production of a building, machine, garment, etc.
- The art of producing a building, machine, garment, etc.
- A general arrangement or layout of a product”.

The aforesaid meanings of ‘Design’ are interpreted differently in different contexts and therefore can be considered context dependent. While engineering design is a purposeful activity directed towards the goal of fulfilling human needs in the form of a product, particularly those which can be met by technological factors of our civilization (Rehman, Duffy and Yan 2008). Design is an engineering activity that affects almost all areas of human life, uses the laws and insight of science; builds upon special experience and provides the prerequisites for the physical realization of solution ideas (Pahl and Beitz 2003).

The activities of designers can be roughly classified into four common distinguished types of direct and indirect design activities as illustrated in figure 2-1:

- Conceptualising: The conceptual design provides abstract solutions and may sometimes result in incomplete solutions that are expected to satisfy the user requirements;
- Embodying: engineering a solution principle by determining the general arrangement and preliminary shape and materials of all components;
- Detailing: finalising production and operating details;
- Computing: representing and information collecting, these occur during all phases of the design process (Pahl and Beitz 2003).

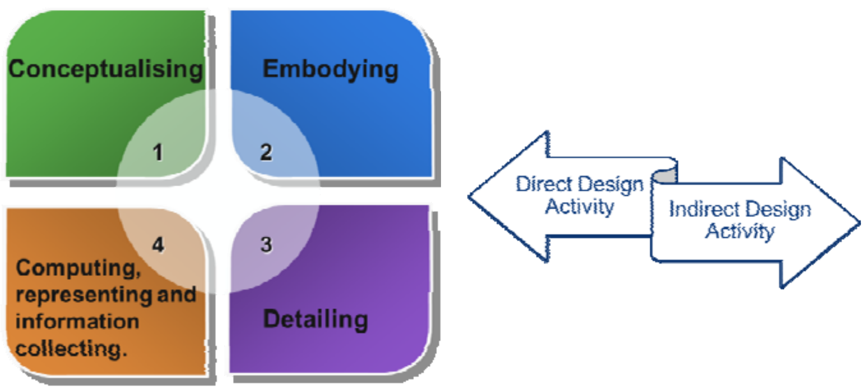


Figure 2-1 Design activity classification

Besides the above four common classifications of direct design activities, there are still indirect design activities, such as collecting processing information, coordinating staff, etc., all of which are important factors in engineering design process management. The design is of an essential part of the product life cycle, because designers determine the properties of every product in terms of function, safety, ergonomics, production, and so on. Another important reason is that production and assembly depend fundamentally on information from product planning, design and development.

2.2 Engineering design process

2.2.1 Engineering Design Process Model

In view of the central responsibility of designers for the technical and economic properties of a product, and the commercial importance of timely and efficient product development, it is important to have a defined design procedure to generate good solutions. Typically the procedure includes a plan of actions that link working steps and design phases according to content and organisation.

An engineering design process (EDP) is a process used by engineers to help develop products, and the engineering design process in its simplest form is a general problem solving process which can be applied to any number of classes of problems. The design process constitutes of all the related activities carried out by the team members while they work on the design problem (Westerberg and Subrahmanian et al 1997), the required design activities have to be structured in a purposeful way that forms a clear sequence of main phases and individual working steps, so that the flow of work can be properly planned and controlled. Association of German Engineers proposes a generic approach to the design of technical systems and products, emphasising the general applicability of the approach in the fields of mechanical, precision, control, software and process engineering as shown in figure 2-2 (Pahl and Beitz 2003).

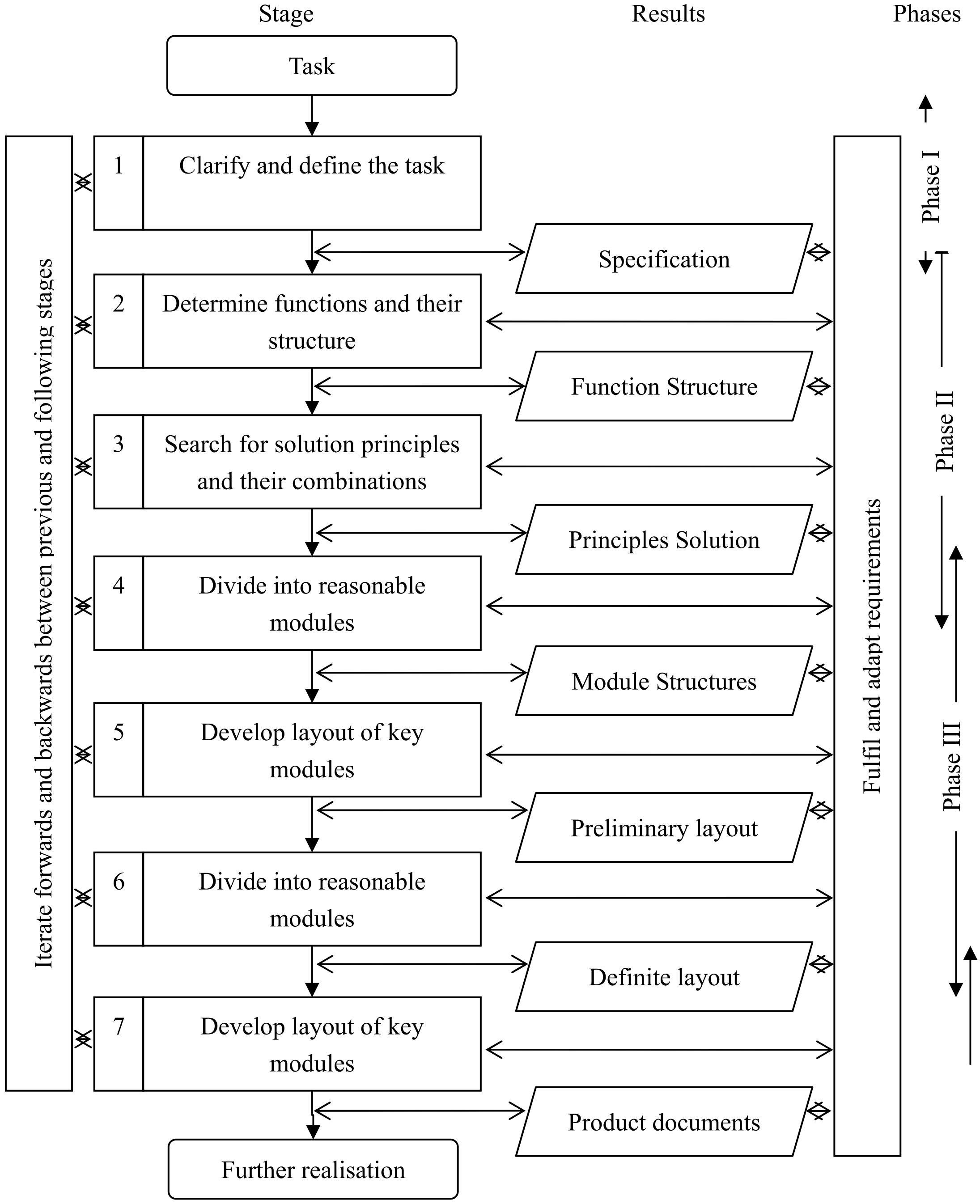


Figure 2-2 General approach to design (Pahl and Beitz, 2003)

The approach in figure 2-2 includes seven basic working steps that accord with the fundamentals of technical systems and a company strategy. Because the aim is for general applicability, the design process has been only roughly structured, thus permitting product-specific and company-specific variations. The above approach should therefore be regarded as a guideline to which detailed working procedures can be assigned. Special emphasis is placed on the iterative nature of the approach and the sequence of the steps must not be considered rigid. Some steps might be omitted, and others repeated frequently. Such flexibility is in accordance with practical design experience and is very important for the application of all design methods.

In addition to the above Pahl and Beitz's model, due to the complexity of the design activity, several researchers have developed different design process models to completely describe different stages of the design process and three such models have been developed and are shown in figure 2-3. All these contributions have led to the development of a stronger theoretical background and use of a more concise and systematic approach to engineering design. These design models detail the subdivision of the overall design problem into sub-problems emphasizing the iteration and interaction within and between the stages of the engineering design process while maintaining the progression of the engineering design process. Despite the vocabulary differences in these approaches and associated models, there are four common distinctive phases appearing in all models, i.e. Task Clarification, Conceptual Design, Embodiment Design and Detail Design in the design process. A brief description of these design phases is discussed in the following sections.

- Clarification of Tasks: the design process starts by the recognition of some needs for a new product and with the clarification of the tasks phase. The outcome of this phase is a well-documented and identified understanding of all problems called a Product Design Specification (PDS). Large, complex and diverse problems are broken down into smaller manageable sections. Although it is desirable that a fully defined PDS is written before starting the design process, it is practically impossible, because the design process is iterative and the PDS is regarded as a dynamic document which evolves alongside the design process.

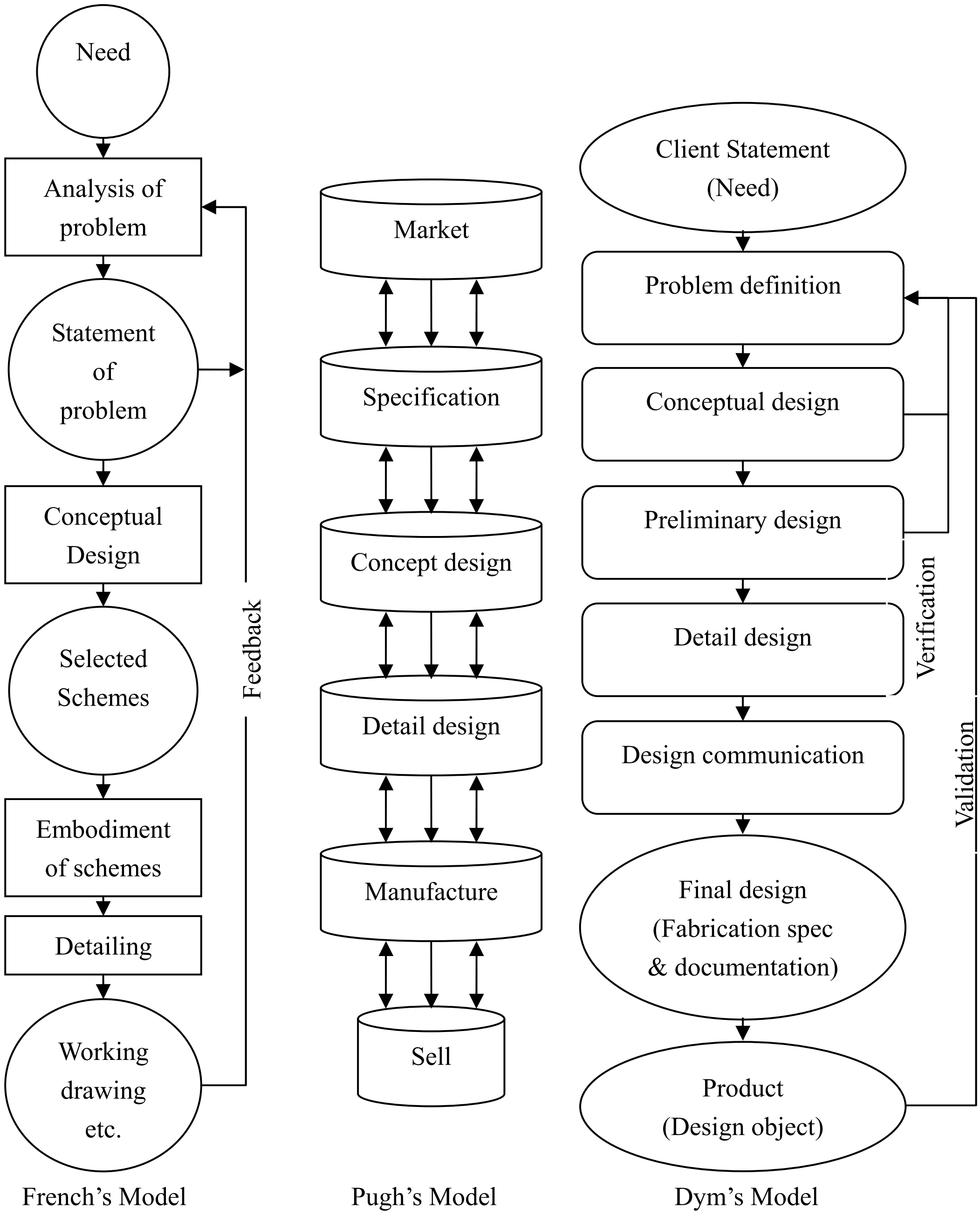


Figure 2-3 Different design process models

- Conceptual Design: At this stage, the conceptual design results in abstract solutions and may sometimes result in incomplete solutions that are expected to satisfy the user requirements considering all viewpoints e.g. function, economy, technology, service etc. The output of the conceptual design stage is the development of one or more new design concepts that can be used as the basis for the embodiment and detail design.

- The Embodiment Design: At the embodiment design phase a designer quantifies less abstract concepts into more concrete proposals i.e. the definitive layout. A definitive layout is worked out by incorporating preliminary form design, which includes shape, principal dimensions, materials and surface qualities of individual parts in the solution and the layout design, which includes determining the spatial relations between the parts in the solution, in accordance with technical and economic considerations.

- Detail Design: The definitive layout selected during the embodiment design phase is further refined to completely specify the structure of the solution, the shapes, dimensions, tolerances, surface qualities and materials of all the individual parts used in the solution and to document them in assembly, detail drawings and part lists. After producing the product documents, the design process phase of the product life cycle finishes and the next phase i.e. production/manufacturing can be realised based on the product documents.

Management of the "Engineering Design Process" involves an approach where an organization makes decisions related to the engineering design process in customer/market oriented fashion as well as optimizing the design process. Design Management further provides a link between "Design Process", "Technology", "Tools & Methods" and "Management" at different levels of an organization (Rehman, Duffy and Yan 2008). Normally design management includes following activities: planning, scheduling and control; process modelling; task management; rework and iteration control; decision making; resource management, et., al.

2.2.2Characteristics of Engineering Design Process

The EDP is an evolving and creative process, during which most of the information is incomplete, especially at the early stage of the EDP. For example, at the conceptual design stage, the design then evolves as it progresses from an initial concept to a more detailed design that satisfies such requirements. Besides there exist many uncertainties in the EDP, some of which come from the EDP itself and other from the dynamic market environment. Complicated multi-objective decision-making processes under uncertainty found in the EDP are difficult to define or predict.

The main characteristics of engineering design are trans-disciplinary, highly complex, and iterative (Rehman, Duffy and Yan 2008). EDP by nature is a complex and often creative process. A well-coordinated and executed process can be a key differentiating factor among competing enterprises. The key to such a process is a proper management of all the design activities within EDP and design information generated during design. Based on customers' requirements and market research, EDP typically starts from the conceptual design with incomplete design information, followed by the detailed design with concrete design solutions, during which many design alternatives are explored and evaluated through both unexpected and planned iterations. Designers typically need to make thousands of design decisions before a final solution is derived and it is clear this is a complex process. The design activities are of an inventive nature and designers do not just apply existing solutions to the problem, instead they innovate and generate new ideas. These creative design processes are hardly predictable and therefore can only be pre-planned on a coarse level. The lack of precise planning even on a medium-grained level inevitably results in highly dynamic and unpredictable design processes (Heller et al 2004).

The dynamic design processes are also characterised by the difficulty in design activity planning and time allocations and incomplete information at the early design stage. The changes often happen in the design process. For example, the design requirements may change very often owing to the new customer demands, the market competition and so on. Companies can liquidate, relocate, and expand. Teams are constructed and disbanded as initiated and terminated. Designers' performance is variable due to illness, the workload and their domain knowledge and experience. These result in difficulties in planning precisely and allocating accurate time for all design activities. In addition, as design progresses, more information is generated and incomplete design information becomes complete for the design problem. The management of this dynamically expanding and vast design information with many versions can be challenging.

The design process is also an iterative process, and many design activities are repeated many times for refinement. Additional information generated in this process always gives rise to new insights either to address a problem which has not yet been recognised or to exploit an identified potential for improving an existing solution or even to evolve the design requirements. Designers can learn and therefore increase and improve their capabilities in their design solution refinements.

There are many uncertainties which can happen in the engineering design process. Some of The design process is unpredictable and heavily depended on the design context, the uncertainties come from the engineering design process itself, some others from the dynamic market environment. It is clear that complicated multi-objective decision-making processes under uncertainty found in the EDP are difficult to define or predict. Along with the above-mentioned characteristics, the EDPs in manufacturing organisations are more distributed in terms of organization and knowledge intensiveness in terms of design sophistication. The design process itself can be considered as a decision-making process (Roozenburg and Eekels 1995). This is true indeed as argued below.

Engineering design process is a decision-making process (Habhouba, Cherkaoui and Desrochers 2011), in which the basic sciences, mathematics, and engineering sciences are applied to convert resources optimally to meet a stated objective. Typically designers need to make thousands of design decisions before a final solution is derived from many evaluated solutions. These design decisions can have significant impact on product design and development as well as on the subsequent life cycle of the product (Borg, Yan and Juster 2000). The design process is characterised by making complex and interdependent decisions at each stage of the EDP that lead to optimal design solutions. Because the information in the design processes is typically incomplete, vague and imprecise, dynamic changes often happens and decision making is throughout the whole design process, the decision making requires extended support at each stage of the design process. In another word well-coordinated design activities and readily available design information through appropriate information management become two essential requirements in modern product development. More precisely, within a given task, the type of intervention on the product was unpredictable and heavily depended on the design context, leading the participants to define specific goals for their action locally (Mantripragada, Kinzel and Altan 1996).

2.3 New trends in engineering design process

The most important impact in recent years on the engineering design process, and on the activities of designers, has come from computer-based data processing (Pahl and Beitz 2003). This can be in the form of design context knowledge and their application using a computer support system such as ConDes (Computer Based Design). CAD is influencing design methods, organizational structures, the division of work, e.g. between conceptual designers and detail designers, as well as the creativity and thought process of individual designers.

Recently modern engineering design is practiced as an interdisciplinary endeavour requiring individual engineers to work as part of a team that involves a range of specialists. Concurrent Engineering (CE) has become increasingly important for product development (Xu, Li et al 2007). CE is a systematic approach to integrate the design of products with related manufacturing processes using some software packages and computing techniques in a computer environment. CE draws together team working and cooperation, with the aim of reducing the need for costly design modifications in the later stages of design and product development (Lees, Branki and Aird 2001). Co-design is another increasingly important philosophy used in modern manufacturing corporations to collocate a multidisciplinary design team to carry out a complex design task through effective communication and collaboration (Li, Fuh and Wong 2004). CE and co-design are complementary in functions since the former emphases a vertically seamless linkage between the upstream design and the downstream manufacturing processes through the creation of intelligent strategies for effective information interchange, while the latter focuses more on the horizontally interpersonal aspects of group work in the upstream design phases.

Among the features of concurrent engineering is the notion of distributed design, and the ability to communicate design changes to multidisciplinary teams. Engineering design is a complex activity. Differences in system architectures and information structures, and co-ordination requirements tend to reduce the effectiveness of distributed design (Chao et al 2002). Complex engineering design projects generally require the cooperation of multidisciplinary design teams and the readily accessibility to various engineering tools (such as CAD, FEA (Finite Element Analysis), as well as dynamic and kinematics analysis, simulation, and optimization software systems), databases and knowledge bases. In order to coordinate multiple engineering design activities in a design project and to guarantee the integration of different engineering tools, it is very important to have an efficient collaborative engineering environment (Hao et al 2006).

With the trend for global competition and the rapid advances of the Internet technologies, both of them are moving towards supporting distributed applications, in which geographically dispersed users, systems and resources can be integrated in an Internet/Intranet environment beyond the traditional boundaries of physical and time zones (Li, Fuh and Wong 2004). Globalization and rapid evolving of Internet and Web-based technologies have revolutionized the product development process. Engineering a product is a complex process involving the integration of distributed resources, such as human beings, engineering tools, and a large variety of product-related information systems (Hao et al 2006). Especially in CE, it seeks to enable multi-disciplinary teams, sometimes working at different sites to work cooperatively. The successful design of large and complex made-to-order (MTO) products such as ships, offshore oil platforms and airplanes, require the collaboration of multidisciplinary design teams. These teams use specialized and often distributed computer systems to aid their design process (Chao et al 2002; Damien et al 2014).

Design can also be considered to be an information process or an information transformation process (Hicks and Culley et al 2002). Recently, modern engineering design has evolved as an interdisciplinary endeavour requiring individual engineers to work as part of a team that involves a range of specialists (MacGillivray and Domina et al., 2007). An increasingly high number of participants from various fields are now clearly involved in the design process (Jean-Francois and Pascal 2002). For all but the simplest of projects, the process of design is undertaken by a group or groups of heterogeneous designers. Each of these individuals or groups will normally work on distinct but some-times overlapping parts of the overall design, and will operate from different perspectives (Lees Branki and Aird 2001). These require careful task decomposition and allocation among team members and appropriate tools to support such a distributed and complex operation which involves many designers working on many different yet closely linked project activities.

The development of knowledge-based systems will increase the ease with which information can be retrieved about the discipline specific knowledge as well as knowledge about past and existing products. The issue of co-operation in design is becoming crucial. New organizations, based on concurrent engineering principles, after many years of experimentation within various companies and industrial domains, still suffer from a lack of efficiency. An increasingly high number of participants from various fields are now clearly involved in the design process, each of them having their own design conceptual worlds made of design tasks, solution representations, language for description of solutions and tools to support analyses (Jean-Francois and Pascal 2002).

Design also involves cooperation among team members from different disciplines and potentially multiple enterprises, causing additional difficulties concerning the coordination of the overall design process (Heller et al 2004). The design process constitutes of all the related activities carried out by the team members while they work on the design problem (Marquardt and Nagl 2004). With the advances of Concurrent Engineering and Collaborative Design, the EDP includes not only the traditional inner EDP in an enterprise, but also the cross-enterprise EDP in the collaborative design. Figure 2-4 shows a practical and simplified EDP in a Customized Truck company proposed and used in this research as a case study, this design process start with “Conceptual Design”, then the design process is split parallel into “Carriage Design”, “Chassis Design” and “Fluid Driver Design” which followed by “Intensity Verification”, afterwards three split activities was merged into “Assembly Design ”.

It assumes that designers will increasingly communicate electronically with design colleagues and become more dispersed geographically. We also expect that larger percentages of each designer’s tasks will become computer based. Thus, their computing environment needs to include such facilities that designers find it effective to use throughout a design project (Mittleman, 2009).

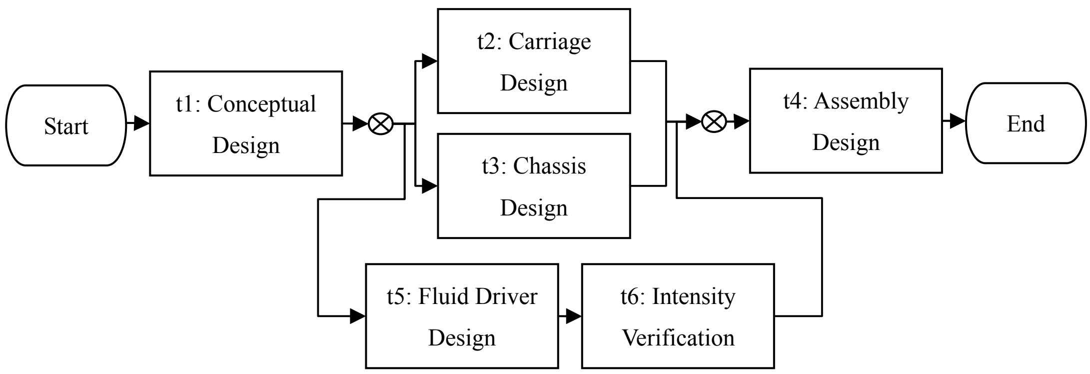


Figure 2-4 A typical engineering design process

For all the above reasons, the engineering design process should be automated to bring real benefits to industrial practice. This means that research should be undertaken to ensure that right and suitable information should be send to the right person in the design process at right time through right communication means. The engineering design process should be managed and coordinated in an efficient way. Workflows, as the automation of the business procedures where documents, information or tasks are passed according to a defined set of rules, have the potential to support and manage the engineering design process.

As described above, the EDP is a creative process and there exist many uncertainties in the EDP. Complicated multi-objective decision-making processes under uncertainty found in the EDP are difficult to define or predict. The conventional workflow technology cannot satisfy the requirements of the creative and complex EDP very well for its fixed logic. An EDP management system must be flexible enough to support the dynamic EDP.

2.4 Workflow and its potential applications in engineering design

2.4.1 Workflow and Workflow Management

The Concept of Workflow is derived from Production and Office Automation (OA) domain, which was proposed mainly for the fixed work procedures in the daily work. Workflow provides better means to realize the organization and optimization of the process. Workflow realize the integration of material flow, capital flow, information flow and related work process as well as applications, and help implement the business process integration, business process automation and business process management.

The Workflow Management Coalition (WfMC) defines the workflow as: "… the automation of the business procedures where documents, information or tasks are passed between participants according to a defined set of rules to achieve, or contributed to, an overall business goal". While a workflow model is computerised representation of a process that includes the manual definition and workflow definition. (Hollingsworth 1995)

The typical characteristic of the workflow is that the workflow supports the separation between the process logic and the information system, and realizes the separation between the application and the process as well. It can implement the process management and decision support of the design process by modifying or re-defining the process model to improve the performance without modifying the concrete application in the design process.

The workflow technology is widely applied in different field, such as manufacturing filed. In order to better support the business process modelling, analysis and implementation, and satisfy and standardize the different workflow product, a general workflow reference model is proposed by WfMC as shown in figure 2-5.

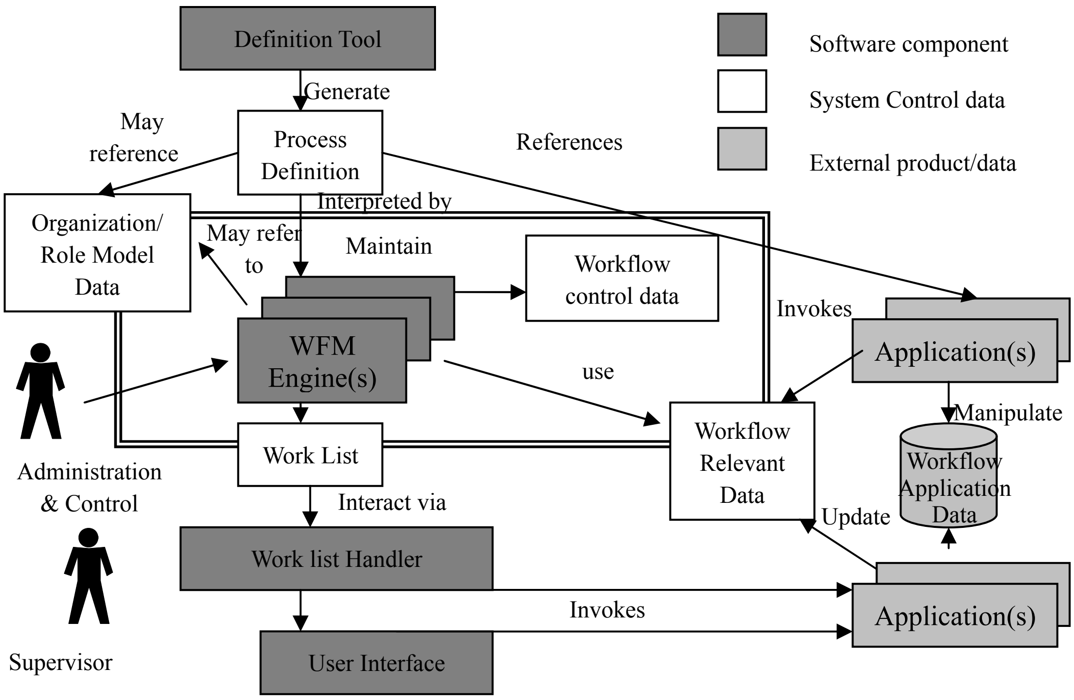


Figure 2-5 Workflow Reference Model

The workflow is managed and executed by workflow management systems, which is a software system. Such a system fulfils the definition and management of the workflow model, and executes the workflow in a standalone computer or networked system according to the logic of a pre-defined workflow model.

All of the workflow management systems typically support the following three functions:

- Establishment of Workflow: this is mainly to fulfil the definition and modelling of the workflow process and related activities, realize the change from a general process, specifically engineering design process in this thesis, to the computer-executed formal definition.
- Execution and Control: This function is to create an instance of the workflow model and execute the model. The workflow service software therefore acts as the workflow engine.
- Interaction: This is to realize the interaction between the user and an IT application in a process.

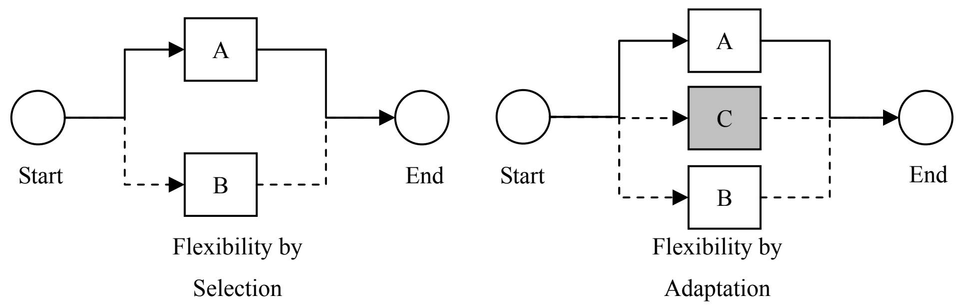
The workflow based approach has potential to manage and coordinate the design process in an efficient way, and can automate the design process, send the suitable information to the suitable person in the design process, so the workflow is potentially a useful way to manage the design process. But as mentioned above, the design process is a creative process, most of the information is not complete, especially in the early stage of the engineering design, for example, in the conceptual design stage, besides there are many uncertainties which can happen in the design process. Some of the uncertainties come from the design process itself, some of them coming from the dynamic market environment. Complicated multi-objective decision making processes under uncertainty are required to be incorporated in the design. A specific design process for a project is difficult to be defined or predicted completely at the beginning of the design process, instead it can only be determined during the execution of a project with new design choices made, new design demand identified, and design decisions made. It is clear that there is a potential to apply workflow concept in the engineering design process management.

2.4.2 Flexible Workflow Concept and Analysis

The traditional workflow working method is mainly divided into two stage, modelling stage and running stage. Objectively, by providing the separation of logic definition and process execution, the running instance can be changed by executing the modified model. Compared with re-developing traditional management system, the workflow can support flexibility to some extent. This two stage working method is suitable to support the business process in which the logic can be determined in advance. But the ordinary workflow technology can not satisfy the requirements of the creative and complex engineering design process very well. The management system for the engineering design process should support the flexible definition and management of the design process. How to improve the flexibility of the workflow to adapt to the change of the design process has become a popular research topic in both the engineering design and workflow research area. The flexibility means that the computer system can respond to the changes happened during the design process when there are changes in the environment, condition and status of the design.

There are already many definitions on flexibility of workflow, such as flexible workflow, adaptive workflow and dynamic workflow and so on (Lau et al 2003). Taking consideration of creative and complex design process and based on the above definitions for flexibility and workflow, this research defines the concept of Flexible Workflow as follows in this thesis. Flexible workflow is a design procedure/process in which design information is continuously generated by participating design engineers in a flexible organisational structure and decomposed design tasks are completed by them using the information received from other participants at suitable yet flexible time according to a predefined design process model to generate or contribute to a satisfactory design solution, The above process can be modified to respond to changes of work environment, work condition or states without redefining the workflow model. Flexible workflow normally can be classified into Flexibility by Selection and Flexibility by Adaptation (Heinl 1999), while Flexibility by Selection means change can only be realized by selecting pre-defined choices when Flexibility by Adaptation means change can be realized by reacting to the different situation.

As shown in figure 2-6 Flexibility by Selection means there is an alternative design process path and every path is feasible and correct. Only the path selection in the process is different in different situation. Flexibility by Selection can only process Pre-defined execution path while Flexibility of Adaptation can change agilely according to the real-time and practical situation. One or more execution path which hasn't been foreseen at the planning stage can be added in certain situations. Flexibility by Adaptation is therefore to provide a mechanism enabling workflow adapt all the unknown situations, and the Flexibility of Adaptation is also called 'ad-hoc' workflow.



A, B, C are workflow activities, dashed line is dynamic flow, while shadow box is dynamic added activity

Figure 2-6 Selection Flexibility and Adaptation Flexibility

The main demands of workflow flexibility for EDP are as follows:

1) In the EDP, the process cannot be pre-defined in most situations, and it changes very frequently according to the execution of the EDP including the changes of design activity, design process and

design resources;

2) According to the demand of market and the continuous development of enterprises, especially in new product development, the EDP should also change accordingly;

3) The design software, designer and design resources which are interacted by workflow may also change with the change of enterprise operation environment;

4) In the running of EDP, there are often unexpected situations arising from new circumstance which cannot be predicted in the workflow modelling;

5) The traditional router (control logic) is defined beforehand, but in EDP, incomplete design information becomes complete as design progresses for the design problem. This requires that the control logic of EDP should be determined dynamically.

It can be imagined that, in modelling flexible workflow, if the workflow model is flexible enough, directed by the workflow enactment service, the workflow management system can change the execution of workflow instance according to its running environment and execution status without modifying the workflow model. Based on the review of engineering design process and the potentials offered by the workflow research, the flexibility of managing design workflow can be proposed based on the literature review in following aspects:

1) The flexible definition of workflow model: the flexible workflow model should express complex, and flexible business process, and the dynamic modification and extension of workflow model in the runtime should be supported;

2) The flexible execution of workflow: the workflow management system should support the

flexible execution of workflow model,

3) The flexible extension of workflow: the dynamic adding or deleting of workflow node should be supported;

4) The workflow instance adaptation: a workflow model can be mapped into different workflow instance according to different requirements;

5）The activity task dynamic allocation and resources dynamic calling, the actor of workflow activity can be dynamically determined, and workflow resources can be called in an optimized way.

2.5 Chapter Summary

The Engineering design is introduced at first, and then the engineering design process is introduced with emphasis on characteristics of engineering design process and new trends in engineering design process. This review reveals that challenging engineering design process is of complex and creative nature, which requires significant support by providing right information at the right design stage in right presentation in order to facilitate engineering designers to generate and evaluate satisfactory design solutions to address increasingly challenging demands from varied and sophisticated customers. Workflow based research has potentials to provide a logical and timely support for typically unstructured design process. According to the requirements of engineering design process, the flexible workflow, which can support the dynamic engineering design process compared with traditional workflow, is described in detail. The illustration of Engineering Design Process and flexible workflow shows that the flexible workflow provides a meaningful way to support the engineering design process.

Chapter 3 Review of research work in Supporting Engineering Design Process and workflow

3.1 Review of research work in Engineering Design Process

The capability in managing an EDP is considered as a major differentiating factor between competing enterprises (Marquardt and Nagl 2004). The complexities arising in the EDP requires the formal analysis and computational support, and managerial tools are required in addressing the coordination of design processes (Chao et al 2002; Hao et al 2006). The most important prerequisite to establish a design process excellence is a proper management of all the design process activities and the associated information.

With advances of computer technologies, many product development life-cycle processes have been automated by the introduction of computer-based systems, such as Computer-Aided Design (CAD), Computer-Aided Process Planning (CAPP), Computer-Aided Manufacturing (CAM),and so on (Xue and Yang 2004).

Traditional mechanical CAD systems aid design processes only by providing shape-modelling capabilities. Many advanced computer-based tools have been developed to assist engineering designers and design managers. But these tend to be focused on the product rather than the process, assist only with the latter stages of the design process, and be 'island solutions' rather than integrated into an overall support environment (Wallace and Burgess 1995).

Many distributed systems have also been developed to support product development life-cycle activities conducted at different locations and associate these distributed activities into an integrated environment using multi-agent and Web technologies. These research works manage and support the EDP, and according to their different objectives, they can be classified into and reviewed in the following categories:

3.1.1 Review of research work in Agents-oriented EDP supporting systems

Hao (Hao et al 2006) proposed an intelligent computational support for concurrent engineering by a set of interacting autonomous intelligent agents, and the authors claimed that this system is particularly suitable for the cohesion and efficient operation in computational design support, but it only provide the agent-architecture to support the product development process, and it haven't give the exact method how the agent can response to the changes in the process. An agent in computer science normally means an autonomous entity which observes and acts upon an environment changes and directs its activity towards achieving goals. An agent-based architecture was used in a service-oriented framework and validated its feasibility and advantages for application in the automotive blow moulded parts design and optimization in (Heller et al 2004), but this paper mainly provides a delegation-based approach for coordinating distributed design processes in a coarse level. Agents provide a flexible and dynamic approach to distributed/multidisciplinary design team which can reduce redundant design activities, and improve co-ordination as reported in the work by. Chao (Chao et al 2002), it research the transmission of the change in the design process, but didn't answer how to react to the changes, especially within the flexible workflow. Madhusudan (Madhusudan 2005) proposed an agent-based Process Coordination framework for distributed design process management; this approach embeds autonomous agents in a workflow-based distributed systems infrastructure. The agents are very useful in the application of the management of the design process, but how the agents and the workflow are connected together to support the EDP better should be studied further. It was proposed in (Lees, Branki and Aird 2001) that an appropriate model for engineering design is in the form of a set of interacting autonomous intelligent agents, possessing different problem-solving capabilities and differing degrees of intelligence, several intelligent mechanism such as rule-based inference, constraint satisfaction, case-based reasoning and neural network, as well as machine learning is mentioned, it provide a sound bases for the intelligent research in this thesis, but the detailed intelligence realization method should be studied to enable the EDP intelligence.

For agent-oriented EDP supporting system, the evolutional EDP cannot be supported in a satisfied way by current approaches although the intelligence or decision making in EDP is addressed. For Web and Internet-based information systems, the EDP can be controlled or managed in such a system, but similar with conventional workflow systems, the EDP cannot be supported very well due to their fix logic and deficiency in intelligence commonly used. For EDP modelling and project management systems, the EDP is only modelled or be managed instead of executed or supported. The flexibility and intelligence support in the EDP evolution should be improved.

Based on this review, it is concluded that in creative engineering design, the current manual and semi-automated information management methods have many weaknesses, namely lack of intelligent support due to inadequate use of artificial intelligence such as agents; lack of dynamic support for the evolving changes during the whole design process.

Similarly, in industrial practice, a large variety of commercial systems are being used for the management of design processes. These systems include systems for product design management, workflow management, and project management, each of which plays an important role in current proactive, however they still show deficiencies in many applications, which are reviewed, summarised and discussed below.

3.1.2 Review of research work in Web and Internet-based EDP systems

A system entitled AHEAD has been developed to equally cover products, activities, and resources and therefore it offers more comprehensive support than pure project or workflow management systems (Heller et al 2004), but it haven't introduce the dynamic changing mechanism clearly, the changes it support should be developed further. Another Internet-enabled system has been developed to support collaborative and concurrent engineering design, based on the state-of-the-art java and Web technologies (Li et al 2004), this system should be improved in supporting the change. Similarly Nagl (Nagl et al 2003) presented a management system by supporting the coordination of dynamic design processes, but it haven't clarify the usage of the workflow in the design process, how the workflow management system be constructed and how the dynamic property of the design be supported haven't been solved. An integrated design system that incorporates the design history, information and intent to support an improved design was proposed (Lee and Lee, 2001), but only object-oriented technology is used to support the automation of product design, the creative and dynamic EDP can't be supported in a satisfactory way.

There have also been noticeable works to help planning the design process with a view to improve the efficiency of the design process and make the design process more smoothly, but it works more in view of product design planning instead of product design process supporting (Heer and Wörzberger 2011).

3.1.3 Review of research work in EDP modelling and project management

In addition, research efforts have also been attracted onto the modelling of the project management of an EDP. For example, the main contribution claimed by Gonnet (Gonnet et al 2007) is to present a Collaborative Model for capturing and representing the engineering design process. Moreau and Back (Moreau and Back 2000) gave a comprehensive review of the use of various tools and techniques used in an EDP. These include the conventional design process planning such as the Gantt chart and PERT chart, then the Design Structure Matrix, the task sequence diagram, the design state network, and finally a method to quantify the time and cost impacts on engineers, and construction of event-driven process chain (EPC) of projects, resulting from the changes in the information management driven process to the design process, but all these manage the EDP in the project view. Eckert et al reported work on tools for planning and managing dynamic design processes based on the parameter values exchanged between tasks, and developing visualization techniques for the design process that makes these dependencies and connections salient(Eckert et al 2001).

The above review reveals that conventional project tools support the management of design processes only to a limited extent. The current design planning methods cannot integrate the information with the design activities and cannot support and coordinate these activities within the design process. Especially during the runtime of engineering design process, since there are regular changes and decision making happened in the engineering design process, the conventional project management tools show significant limitations and cannot manage the dynamic changing process. As a result, the engineering design process could be terminated with new engineering design process model to be constructed. The engineering design process needs a framework to support the dynamic change and decision making, and requires the support and management of the design activities and information in a satisfactory way.

3.1.4. Review of research work in Workflow-based supporting systems

The workflow technology coming from the Office Automation can potentially provide an advanced method for the organization and management of the design process. Work-flow management systems (WfMS) differ from project management systems since they address fairly detailed project focused task execution support rather than only design activity focused and high-level design process and planning. In an ideal world, the design process should be managed and the design process should be supported by the computer systems automatically. The workflow based approaches have been proposed as a useful way to manage and coordinate the design process.

Marquardt et al introduced a workflow and information centred support of design processes which is implemented by a workflow modelling system, satisfying the needs of highly dynamic and creative engineering design processes (Marquardt and Nagl 2002), it mention the workflow in the design process, but it haven’t clarify the usage of the workflow in the design process, how the workflow management system be constructed should be studied. Another framework was proposed by Madhusudan, which embeds agents in workflow management systems architecture to interleave both manual and automated design tasks (Madhusudan 2005). The agent is very useful in the application of the management of the design process. But how the agent can respond to the changes and how the agents and the workflow are connected together have not been explained. Hao et al. presented the results of an industrial case study in the development of a collaborative e-Engineering environment for mechanical product design engineering by applying intelligent software agents, Internet/Web, workflow, and database technologies (Hao et al 2006), but the intelligent problem solving capabilities is not explained in this paper. In order to facilitate the product design and realisation processes, an Internet-enabled system has been developed to support collaborative and concurrent engineering design by seamlessly integrating three functional modules, i.e., co-design, Web-based visualisation and manufacturing analysis in (Li et al 2004; Mavrikios et al 2011), but this paper mainly pay attention to the design process support by integrating the different modules, it cannot support the change modification in the EDP.

The EDP is a creative process, during which most of the information is incomplete, especially at the early stage of the EDP. Besides there exist many uncertainties in the EDP, some of which come from the EDP itself and other from the dynamic market environment. Complicated multi objective decision-making processes under uncertainty found in the EDP are difficult to define or predict. The conventional workflow technology cannot satisfy the requirements of the creative and complex EDP very well for its fixed logic. An EDP management system must be flexible enough to support the dynamic EDP. How to improve the flexibility and intelligence of the workflow to be able to adapt to the changes of the EDP has become an important research topic in both the engineering design and workflow research area (Lees et al 2001; Marquardt and Nagl 2004).

From the above review of related work, it is clear that most of traditional workflow based approaches have a fixed procedure and are incapable of dealing with dynamic changes during an

EDP. In addition, the EDP is highly dynamic and creative and the intelligence requirement in the EDP management is not well addressed by traditional approaches.

3.2 Review of research work on Flexible Workflow

3.2.1 Workflow Modelling

A model is a kind of simplified expression of objects by using of words, diagrams, symbols and relations. Workflow is the abstract expression of office, business or other processes. The main function of workflow in its build time is to implement the computerised definition called workflow model. Normally a workflow model also includes an organization model, resource model and so on, and they are integrated with process model to describe fully the business process. Due to time constraint, only the process model of workflow will be mainly considered and described in this research. In the remaining thesis, a workflow model always means a process model of workflow if there is no special indication.

There are mainly two ways of describing workflow model, formal and informal ones. Informal description is to describe the activities and their relations of a workflow model by graphical symbols, whereas a workflow description language is used in formal description, i.e. a set of language symbols used to describe all kinds of elements, typically XML Process Definition Language (XPDL) by WfMC (WfMC 2006). Once a workflow model is defined by workflow modelling tool, the workflow instance can be generated according to the needed initial conditions and execution parameters, managed, monitored and scheduled by workflow enactment service. A workflow instance is a specific business or in this research design enactment process, and is the projection of the workflow model in real time.

A Meta model is a high level model which is normally used to describe workflow models, Meta models can generate abstract representation of the basic elements and rules in workflow models, and describe workflow models as guidance for workflow modelling. The workflow meta-model is defined by WfMC to describe inner relations of workflow models and to describe every objects, the relations between these objects and attributes of these objects in workflow models. Based on workflow meta-model, workflow model and instance relation is given in figure 3-1. Besides supporting entities or elements needed by defining workflow models, a fine workflow model should satisfy all requirements in workflow modelling and execution, such as complete expression capability, formal semantics, graphical features, and so on.

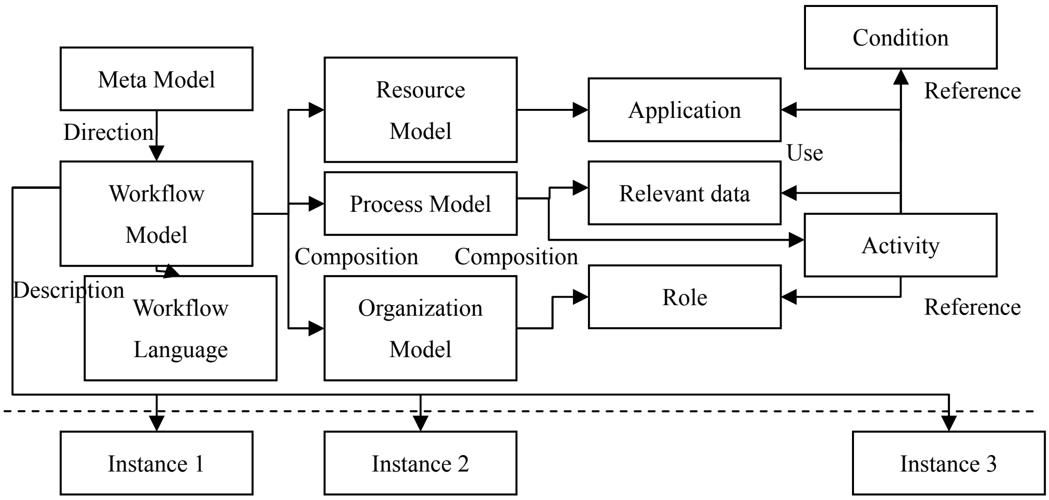


Figure 3-1 Workflow model and instance relation

Recently workflow modelling methods have been developed rapidly. There are many workflow modelling systems available in the market. Scholars also proposed some modelling methods according to business process characteristics and requirements in specific domain. These can be summarised as follows.

1) Activity Network based modelling method: This is a typical activity-based Input Process Output (IPO) model. It's mainly advantage is simple, intuitionistic and easy understanding, and it

is suitable for fixed logic workflow. Its disadvantage is that its diagram/network structure cannot describe real and complex process efficiently with its low flexibility. This will cause particularly problems for design process modelling (Wang et al 2006).

2) Petri-network based modelling method: Petri-network is a graphical and mathematical modelling method suitable for many different systems. Its main advantage is that it has high description capability and powerful analysis means, and has strict semantic and graphical language, whereas its disadvantage is its complicated model structure, which makes it difficult to be applied in the engineering design process (Fung et al 2003).

3) Object-oriented modelling method: It mainly pays attention to the description of all entities in processes, typically modelled by tools such as UML, and it is used as modelling tool for workflow modelling (Eynard et al 2004; Bruning 2011).

4) Role-based modelling method: the workflow is described as a system which is constructed based on the interoperation of roles. This method is more suitable to sudden and random business processes while are also suitable to fix business processes. Its disadvantage is that it is hard to describe complicated route logic, and control execution details (Zhang 2010).

5) Event-driven Process Chain modelling method: the main entities of EPC are functions and events. A function is triggered by an event, and an event can be or is produced by a function. Its main advantage is its good description and easy reading capability (Aalst and Hofstede 2005; Wang et al 2006).

6) Compositional workflow modelling method: This is a method in which a number of above workflow modelling methods are used to address the modelling needs of complex processes. The

above modelling methods have their own characteristics and applicability for different processes. In some cases, it is very hard to describe a business process completely by one single modelling method. As a result, the present research works pay more attention to the integration of different modelling methods.

3.2.2 Literature review of Flexible Workflow and Intelligence

Engineering Design Process is a highly dynamic process. In principle a workflow system can be used as a key technology to manage Engineering Design Process. The workflow system however is required to be flexible enough to satisfy the demands of the dynamic engineering design process. Most of the traditional workflow modelling method is based on fixed process model and lacks the support to a dynamic changing process. The flexible workflow modelling method and intelligence has become an important research area in workflow research, and the following summarises the main research work identified in the literature as typical representations of each of the categories:

1) Traditional flexible workflow: Workflow modelling languages based on Petri nets have been used to add mechanisms to allow for a more direct and intuitive support of the workflow (Aalst and Hofstede 2005). Similar to other traditional flexible workflow approaches, they lack intelligent behaviour and have other deficiencies discussed later.

2) Rule-based flexible workflow: Muller (Muller et al 2004) described an ECA (Event Condition Action) rule model to automatically detect logical failures and to determine the necessary workflow adaptations. This work showed great advantage of using such automated fault detection.

3) Flexible workflow based on dynamic definition: A relatively fixed model is defined in this approach, and then other suitable activities can be selected at the runtime. The flexible workflow

is defined in workflow execution (Orlowska et al., 2005);

4) Flexible workflow based on dynamic modification: In contrast to the dynamic definition approach, the workflow model is modified at the runtime to improve the flexibility of the approach (Crowe et al 2001);

5) Service-based flexible workflow: A service layer is added between workflow modelling and its execution, and in this approach the dynamic and flexible workflow is realized by the separation and the dynamic combination of events in these two activity groups (Crowe et al 2001);

6) Object-oriented flexible workflow: This approach uses object-oriented programming philosophy in workflow modelling to provide required modelling simplification, object encapsulation and other associated advantages. A chosen object-oriented approach and the use of UML diagrams for the modelling and integration of product, process, and resource data is detailed for a turboprop aircraft project (Eynard et al 2004);

7) Knowledge-based flexible workflow: The elements in flexible workflow model, such as activity, router, and so on, is expressed by knowledge entity, and the knowledge base of properties and rules is established to realize the flexibility (Chung et al 2003);

8) Flexible workflow based on coordination and control: A meta-workflow as a control and coordination mechanism for exception handling is proposed to control over base workflow (Kumar et al 2005);

In addition to the above methods, research works have also been reported on flexible workflow in the view of organization modelling, task dynamic allocation, or workflow exception handling. Some of typical workflow is shown in figure 3-2 and illustrated as follows:

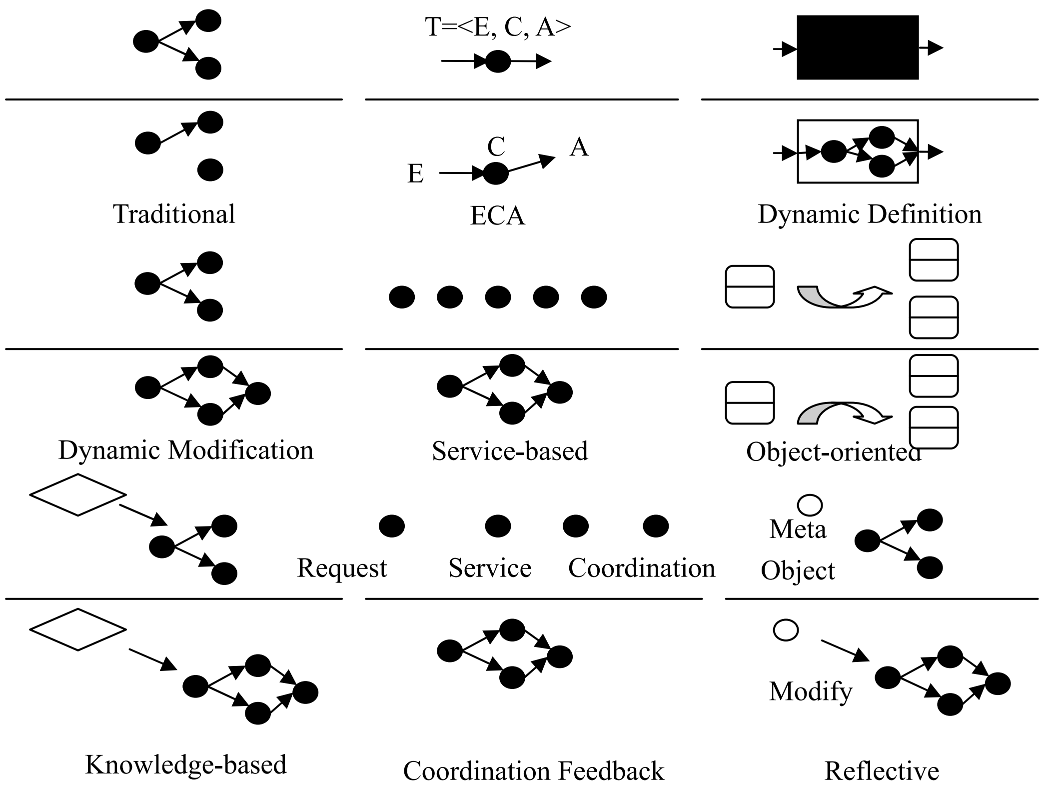


Figure 3-2 Typical flexible workflow modelling methods

1) The traditional flexible workflow means the flexibility is realized by selecting pre-defined logic;

2) ECA based flexible workflow means the flexibility is realized by the ECA rules, the flexible decides how it changes according to the result of ECA;

3) Dynamic definition based flexible workflow means the modelling of flexible workflow is defined dynamically in run-time, the flexible workflow will change according to the dynamic model;

4) Dynamic modification based flexible workflow means the flexible workflow can change itself by dynamically modifying itself;

5) Service-based flexible workflow means the flexible workflow is realized by combining

different services;

6) Object-oriented flexible workflow means the flexible workflow is realized by modifying the properties and actions of the object, or combining different objects;

7) Knowledge-based flexible workflow means the flexible workflow is realized according to the knowledge stored in the workflow, the changing of the flexible workflow can be very complicated with the decision;

8) Coordination feedback based flexible workflow means that the flexible workflow is based on coordination and feedback theory, and is realized by the request, service, coordination and instantiation;

9) Reflective flexible workflow means that the flexible workflow is realized by modifying its functions;

The above review of the research works on flexible workflow reveals that their approaches improve flexibility and intelligence of workflow model from their different perspectives. But in order to support the dynamic engineering design process, there are still some deficiencies in these flexible workflow research based approaches and these include:

1) Above flexible workflow approaches are all based on a two stage definition-execution modelling method. These methods not only cannot describe the actual process fully, but also neglect the inner relationships between the workflow model and its instance. The strict separation of modelling and execution stages hinders heavily the improvement of workflow flexibility. The essence of flexible workflow modelling research is therefore to break through the two stage workflow modelling method.

2) It is clear that a flexible workflow method and model structure which matches the requirements of engineering design process should be proposed and implemented in order to enhance the flexibility and intelligence required during engineering design.

3.2.3 Review on Workflow Execution

In workflow management system, the workflow model created by workflow modelling tool is executed by a workflow enactment service to create workflow instance and monitor the execution process of workflow instance (Hollingsworth 2005). The core of workflow enactment service is the workflow engine, whose main function includes the initiation of workflow model, scheduling of and monitoring on the execution of each activity in a workflow instance, and the interoperation of an application software and operator. The execution of workflow engine is based on interpreting the model, then workflow engine can acquire detailed execution information, and the workflow instance can be instantiated automatically and directed by the workflow model. Mainly two aspects need to be studied in determining the implementation of workflow engine (Hollingsworth 2005):

1) Firstly, the underlying communication infrastructure of the workflow engine should be determined. The underlying communication infrastructure determines the inter-connection of components in the workflow engine, and is the basis of distributed application of the workflow engine;

2) Secondly, the collaboration process of components in a workflow engine, including the model submission, model execution, and the collaboration between components, should also be determined. The interoperation between components realizes the execution of a workflow

management system.

Concerning the flexible workflow technology supporting dynamic engineering design process, the flexible workflow system must be a distributed system, because the information infrastructure, organization structure or collaboration environment of dynamic engineering design process all are distributed geographically for many engineering design organisations. The distributed workflow engine is a group of workflow engines distributed at different workflow nodes, which work collaboratively to implement the execution of the overall workflow activities. Each distributed engine implements part of the workflow instance, the collaboration of different distributed workflow engines is realized by a reliable communication mechanism (Blake and Gomaa 2005; Xu and Wang, 2002).

With the increasing application of workflow management system, the departmental workflow applications have gradually expanded to the cross-enterprise workflow applications. Traditional distributed computing model, such as CORBA (Common Object Request Broker Architecture) and DCOM(Distributed Component Object Model),cannot support the client and server communication on Internet very well for the exclusive agreement and interface (Booth and Haas et al 2005). CORBA and DCOM all are based on strict management environment, and the interoperation between these two agreements is difficult to realize. Besides, CORBA and DCOM are all suitable to communication between servers, but there is severe deficiency in communication between clients and servers, and there is normally firewall restriction in the application of CORBA and DCOM.

For the cross-enterprise distributed workflow, Web Service as a new distributed computing model based on open Internet standard, realizes the cross-platform, cross-programming interoperability, and provides reliable communication infrastructure for cross-enterprise and Internet distributed workflow application. There are already research efforts on distributed workflow based on Web Service. Wang et al. (Wang et al 2006) introduced an agent-based workflow approach, where the business-to-business and enterprise workflow coordination is realized by an agent while the Web Service realizes the infrastructure of the information transformation and service description, but the flexible workflow cannot be supported by Web Service mechanism. Hu (Hu et al 2003) proposed a three layer flexible workflow management system, in which a service layer is added into the definition and execution layer, and the dynamic binding of workflow definition and execution is realized by service layer, but to some extend the service layer can only provide the mean for services combination, the change of workflow node cannot be described by web service. A workflow management system based on Web Service, including the Web Service flow description based on BPEL4WS, message transformation between services has also been reported (Xu and Wang 2005), the flexible workflow description by BPEL4WS is not researched yet. Gong (Gong 2004) described a distributed workflow management system component model and Architecture, not only realizing the distribution of components and the distributed execution of workflow engine, but also realizing higher layer distribution of workflow process model, similarly but the web service supporting for flexible workflow engine and flexible workflow model should be described.

The above research efforts have been directed to the distributed workflow based on Web Service in different view. After careful analysis, the research work on flexible workflow, especially the flexible workflow based on Web Service still requires further investigation in particular in following two aspects:

1) The current research work on workflow based on Web Service pays much attention on the Web Service description and realization of workflow and its execution process. There are few detailed research documentations on the realization of distributed workflow engine based on the Web Service;

2) The current description of cross-enterprise flexible workflow is based on the Web Service, including message call based on SOAP（Simple Object Access Protocol）, workflow activity description based on WSDL（Web Service Description Language） and workflow process description based on BPEL4WS. All these are based on a fixed workflow model, but there are few descriptions of dynamic flexible workflow model.

It is therefore necessary that across-enterprise distributed workflow engine architecture based on Web Service should be proposed, and the distributed flexible workflow engine based on Web Service and flexible workflow description based on Web Service should be further studied.

3.2.4 Review of workflow management systems

Recently workflow and its application have been investigated, especially in the engineering design area, according to the dynamic and creative characteristics of EDP. How to improve the flexibility of workflow has become a popular research topic in the workflow field (Ha and Suh 2008). The following summarises the key findings in these fields.

1) Flexible workflow modelling: Muller (Muller et al 2004) proposed and developed a rule-based adaptive workflow management system entitled AgentWork, the rule-based system has limited

intelligence for some specific problem, for example the fuzzy decision in the EDP. The concept of Service Flow was also proposed in (Wetzela and Klischewski 2004), and the service process is modelled and designed with the service as the centre, this concept provides improved flexible service configuration for the service flow, but the service flow can only realize the flexibility by the combination of services.. The WFCP-net (Workflow-net based on Colored Petri Net) is used to model a family of workflow processes with similar process routes and logic rules in (Liu et al 2002), the Petri Net is good at describing the workflow, but how to describe the flexible workflow also should be studied further.

2) Flexible workflow execution: the service-based cross-organization workflow was researched based on agent-oriented composition (Blake and Gomaa 2005), but the service composition cannot support the inner modification of flexible workflow. The exception handling and coordination mechanism was studied (Kumar et al 2005), it provide a reference for the intelligence of flexible workflow. Biegus (Biegus and Branki 2004) proposed an approach to provide support for workflow interoperability, which would allow for linking business partners without the necessity of a major redesign of their workflows, as mentioned before, it cannot resolves the inner modification of flexible workflow nodes.

3) Flexible workflow task and resources allocation: (Li et al 2004) proposed an innovative approach with corresponding algorithms to the checking of resources consistency for a workflow specification. A novel authorization model that incorporates authorization rules was then proposed (Liu et al 2004) to support the planning of assigning tasks to roles/users and the run-time activation of tasks. An efficient workflow task allocation method is proposed an efficient workflow task allocation method in a distributed workflow system (Son et al 2003), which is

based on the locality principle. Above researches provides a sound foundation for the flexible workflow task allocation.

4) Flexible workflow intelligence: Muller (Muller et al 2004) presented the concept and implementation of AgentWork, which is a workflow management system supporting automated workflow adaptation in a comprehensive way. This approach also uses a rule-based approach followed to specify exceptions and necessary workflow adaptations. The rule-based intelligence has its limitation in fixed rule decision. A service-based dynamically reconfigurable system framework for supporting future self-managed software system was also proposed (Cao et al 2004), the reconfiguration method also can realize the modification of flexible workflow, but how to modify the flexible workflow should be studied. (Hwang and Tang 2004) proposed an architecture model which can deals with both expected and unexpected exceptions in the context of workflow management, expected exceptions and their handling approaches are specified by ECA rules, while cases of unexpected exceptions are characterized by their features and resolution approaches, similarly how to react to the exception should researched, especially for unexpected exceptions. Chunga (Chunga et al 2003) investigated the use of ontology, agents and knowledge based planning techniques to provide support for adaptive workflow or flexible workflow management, especially in the area of new product development within the chemical industries, while the intelligence algorithm is not explicitly introduced for the ontology or agent.

5) Flexible workflow management system: A workflow tracking system was designed (Ong and Foo 2004) to improve the performance and productivity of a production line by reducing non-value added costs and activities within the manufacturing and assembly process. A re-configurable and customized flexible workflow management system which can be integrated in

industry was also proposed (Lau et al 2003). Both of them should be studied further for its supporting to the flexible workflow application in EDP.

The EDP management is concerned with actually the management of the activities in EDP, and is typically a workflow process. The above literature review shows that the application of flexible workflow in EDP not only can realize the management of activities in EDP and EDP automation, but also has the potential to support the dynamic and creative EDP by its flexibility. The flexible workflow management system can implement the information and process integration in EDP, help call related application, and support the management and coordination of an EDP. As a core technology of process modelling and management, as well as process automation, flexible workflow offers advanced means for the process management and flow optimization of an EDP.

It can be seen that some research work on workflow application in EDP management has been undertaken, but there are still areas of research needing more effort for improvement namely:

1) The current EDP management system based on workflow is built on static workflow technology. The structures of these systems tend to be rigid and inflexible. The workflow model is therefore required to be modified in order to support dynamic characteristics of an EDP. Because the flexible workflow can respond to the changes in an EDP dynamically, it provides potentially a good scope to re-configure the EDP management system rapidly. The EDP management system based on flexible workflow should therefore be studied further;

2) When the workflow is applied in the management of an EDP, it normally is applied in the inner EDP system in the enterprise. But for the collaborative and concurrent engineering design, there is a recruitment of normally involving the interoperation of a multi-workflow management system.

This requires that a general framework of cross-platform and cross-system EDP management should be proposed and developed, meanwhile, the sub or inner EDP should change according to the general or parent EDP;

3) For the current EDP management systems, they are normally developed by certain computer programming language according to the detailed design of the system. The EDP model cannot be mapped directly onto an EDP management system, which restricts the flexibility and interoperability of these systems. By using flexible workflow, it is possible that an EDP management system can be implemented directly by the execution of a workflow model.

In addition, the literature review reveals that there are few flexible workflow applications particularly intended for and used in EDP management systems, although there are already many general improvements in flexible workflow. Specifically following research questions in the flexible workflow areas require further investigation and improvements, especially in its application in the dynamic and creative EDP.

1) Flexible workflow technology doesn't provide enough support for cross-enterprise businesses process in collaborative or concurrent engineering design. How to resolve the complex flexible workflow for an EDP still needs to be studied further;

2) The flexibility of workflow model should be improved. On one hand, the two stage work method of workflow cannot reflect the practical process and meet the requirements; on the other hand, the inner relation between a workflow model and its running instance is neglected. The strict separation of workflow model and an instance hinders heavily the workflow flexibility. New methods and tools to overcome these limitations should be investigated. The key to improving the

flexibility is to break this two stage work method and related flexible expression of workflow model and develop a new one.

3) The current research work on generic flexible workflow mainly adopts a passive response to any change, but in the dynamic and creative EDP, fine and responsive adaptability is demanded for the different changes and uncertainties. The intelligence of current flexible workflow should therefore be improved. Due to the fact that he information required during the conceptual stage is incomplete, imprecise, and fuzzy, the intelligent capability in uncertain conditions becomes the key to improve the flexible workflow intelligence.

4) Current EDP management systems actually are loosely-coupled. As a new distributed computing model in open Internet standard, Web Service can satisfy the dynamic and loose-coupled demand of an EDP. The flexible workflow model and workflow engine based on Web Service needs to be improved;

5) How to improve the flexibility of workflow is a systematic work, especially in the EDP management, and it is imperative to take a systematic approach to improving the flexibility comprehensively.

3.3Chapter Summary

This chapter presents a literature review on work supporting flexible workflow technology of engineering design process. The review work in engineering design process is given in agents-oriented EDP supporting system, Web and Internet-based EDP management system, EDP modelling and project management, as well as workflow-based supporting system. The design process should be managed and the design process should be supported by the computer automatically, Workflow-based EDP supporting systems can support the engineering design process better since they address fairly detailed execution support rather than only high-level planning. Then the supporting flexible workflow technology is reviewed in workflow modelling, flexible workflow intelligence, and workflow execution and workflow management systems. The advantages and disadvantages of flexible workflow research are analysed. Based on these review and analyse, research questions are defined and it is concluded from this review that the flexible workflow technology has potential for EDP and should be further improved to support the dynamic engineering design process.

Chapter 4 Establishment of Research Problem

This chapter discusses the research problem formulated on the basis of discussion and outcomes of the literature review presented in the preceding chapters. The main findings of the review of the related methodologies are classified into findings of supporting engineering design process and findings of flexible workflow in section 4.1. Based on these findings, the research gap is formulated and established in section 4.2. Finally the research boundary of the research work is defined, so that the focus and areas of research work are clearly identified.

4.1 Findings of Review of Methods and Tools Done

Chapter 3 discusses the review work in detail from the two perspectives i.e. engineering design process and flexible workflow. The key findings in terms of deficiencies and shortcomings identified in this research review are further explained and structured in the following sections in order to formulate research problem definitions.

4.1.1 Findings of Review of Supporting Engineering Design Process

Section 3.1 describes the research work in engineering design process, especially in workflow-based supporting work. Although the engineering design process can be supported and managed in a satisfactorily way, but for current complex and dynamic engineering, the support of engineering design process should be improved in following way:

1) The current workflow-based engineering design system mainly constructed on static workflow technologies. The changing of engineering design process can only by realized by modifying

workflow, so the engineering design process cannot supported in a dynamic way. The flexible workflow technology can be applied in supporting dynamic engineering design process, because the flexible workflows can response to the process change dynamically. The rapid re-construction of engineering design process supporting system can be realized. The research on flexible workflow supporting dynamic engineering design process should be further conducted.

2) The former application of workflow for engineering design management normally is limited in the inner engineering design management in an enterprise. But for current global cooperation and Internet-based engineering design process, engineering a product is a complex process involving the integration of distributed resources and the complex flexible workflow inter-operation are normally involved in engineering design process. The complex engineering design process should be supported by an integrated flexible management framework which can support the hierarchical engineering design process. Meanwhile the inner system of engineering design process should change according to the overall change of the overall engineering design process, the dynamic inner engineering design process should also be supported.

3) For the development of engineering design process management system, it should be developed according to the engineering design process model while there are direct mapping mechanism from engineering design process model to engineering design process supporting system, i.e., the engineering design process supporting system is hard to be developed directly. But for the flexible workflow-based engineering design process supporting system, the engineering design process management system can be generated rapidly by the execution of engineering design process model, which is modelled by flexible workflow.

4.1.2 Findings of Review of Flexible Workflow

From the review of research work on flexible workflow and its support for engineering design process, it is clear that there exist few applications of flexible workflow technology in engineering design process management. The flexibility of workflow can support the dynamic engineering design process in a satisfactory way. Recently, how to improve the flexibility of workflow has become a popular research topic. But in supporting dynamic engineering design process, the flexibility of workflow should be further improved in following aspects:

1) Flexible workflow technology cannot support the inter-enterprise engineering design process and the integration of different engineering design processes. How to solve the complex workflow inter-operation is very important to flexible workflow supporting engineering design process;

2) The flexibility of workflow model should be improved further, the conventional two-stage modelling method on side cannot reflect the actual business process, on another side ignore the inner relation between the workflow mode and workflow running. The strict separation of workflow modelling and instance executing severely hinder the flexible workflow. Breaking through the two stage workflow working method and bettering the flexible expression can improve the flexibility of workflow effectively;

3) The current flexible workflow research reflects more on passive response to changing. But in dynamic engineering design process, the excellent adaptation is needed in the dynamic engineering design process for its change and uncertainty characteristics, the intelligence of flexible workflow should also be improved. Especially in the engineering design process, the design information and knowledge is fuzzy, the intelligent processing capability in uncertainty

become the key to improve the flexible workflow intelligence.

4) The inter-enterprise engineering design process is loose-couple process, Web Service, as a new distributed computing model based on open Internet standard, can satisfy the agile and loose cooperation requirements of engineering design process, the research on flexible workflow based on web service, especially in realization of distributed workflow engine based on Web Service, as well as flexible workflow model description based on Web Service.

5) The flexibility of workflow reflects in many aspects, such as flexible workflow model, flexible workflow execution. Improving the flexibility of workflow is systematic engineering, especially in supporting dynamic engineering design process, a systematic flexible workflow theory and method should be researched for better supporting dynamic engineering design process.

4.2 Research Gap

Based on the shortcomings identified in the previous sections and review presented in Chapter 3, this PhD research argues that there is a need for a new framework which can support the dynamic engineering design process management with an innovative intelligent and flexible workflow technology. The new framework can adapt to the changes of engineering design process, and can support the engineering design process real-time running, design activities, information exchange and decision making, as well as managing the engineering design process smoothly. The research problem is presented as follows:

It is necessary and better for engineering design process to investigate and propose a computation framework to support dynamic engineering design process in an inter- and inner-enterprise engineering design process for a complex mechanical product, for example airplane or truck component. The framework is supported by flexible workflow technology, the engineering design process is described by flexible workflow model which can be modelled by a dynamic instance-based manner, and the model can adjust itself according to the change of engineering design process by its dynamically changed structure and execution by a flexible workflow engine. For any change in the engineering design, can the framework not only enable the change of engineering design process, but also know how the engineering design process should change. Especially for the incomplete and imprecise information in engineering design process, the framework should have a mechanism to determine how the change should happen. Besides, the framework should be executed by to realize the application of engineering design process management, the distributed flexible workflow execution and flexible workflow expression based on Web Service should be studied. Finally, an engineering design process management system is developed based on the proposed framework in order to demonstrate the framework.

The research problem presented above raises a number of questions, which are addressed in this research:

- How the flexible workflow can support the engineering design process? In chapter 5, a framework of engineering design process management based on Multiple Autonomic objects flexible workfoow is proposed to support the dynamic and innovative engineering design process.

- How the workflow model can be flexible enough? In chapter 6, a flexible workflow modelling method based on dynamic instances is proposed to realize the high level of flexibility of workflow by a formal and graphical expression.

- How the flexible workflow chould be intelligent to determine the change of engineering design process? In chapter 7, a flexible workflow autonomic object intelligence algorithm, especially fuzzy reasoning algorithm in uncertain condition, is proposed to realize the intelligence of flexible workflow.

- How the flexible workflow engine can be realized by web service and how the flexible workflow can be described by web service? In chapter 8, a flexible execution of distributed workflow engine based on web service is proposed and flexible workflow is described by web service as well to trealize the application of flexible workflow in dynamic engineering design process.

- How the flexible workflow management prototype system can support the engineering design process? In chapter 9, A flexible workflow management prototype system supporting engineering design process management implementation developed by Microsoft.Net technology is introduced, and the application of the prototype system incates the the dynamic and creative EDP management can be supported satisfactorily.

4.3 Research Boundary

In order to focus on the research work, this thesis bounds the established research problem in following way:

- Only the engineering design process is considered, i.e. the detailed design activity is outside of this research work.

- Both of the inner- and inter-engineering design process is considered, while the complex team-based engineering design process is emphasized.

- Only mechanical design domain is selected, with a particular focus on airplane components.
- Mainly the incomplete and imprecise knowledge and decision making is considered in this research.

4.4Chapter Summary

This chapter establishes the research problem by explaining the shortcomings of the current methodologies identified from the review of literature in the previous chapter. The dynamic engineering design process can be supported by flexible workflow technology by activity and information coordination, as well as process automation. Based on the above, research questions are raised which will be addressed during this research and the research boundary is set in order to focus the work on the identified research problem. The next chapter proposed multi autonomic objects flexible workflow to support an engineering design process management framework.

Chapter 5 Engineering Design Process Management Framework based on Multi Autonomic Objects Flexible Workflow

Based on the literature review of current technologies available and the challenging problems faced by many multiple enterprises working in collaboration in their product development, this chapter presents a structured way of firstly introducing Autonomic computing, followed by a set of vigorous definitions of Autonomic Objects, which lays the foundation in terms of model representation and modelling for autonomy, monitoring and so forth in this research.

5.1 Engineering Design Process based on Virtual and Flexible Workflow

5.1.1 Enterprise Engineering Design Process based on Virtual Workflow

In a typical EDP, especially in complicated product design, there is often several design teams working together to cooperate to complete the whole design tasks. Each individual design team has their own workflow system to manage their individual lower level EDP. It is therefore important to note that the EDP can only be implemented by the coordination of design teams. This phenomenon will become more complicated when cross-enterprise or cross-discipline design is required for a design, because different enterprises have their different design processes which are normally supported by different heterogeneous, non-compatible workflow management systems. In such an environment, the realization of workflow management interoperation will be the key to implement the cross-enterprise EDP. The interoperation of workflow has always been a difficult and popular area. The main research areas include interoperation of workflow model and workflow management system (Chung et al 2003; Wetzela and Klischewski 2004; Liu et al 2005;

Karsten A. Schulz and Orlowska 2004; Biegus and Branki 2004). The key to workflow model interoperation is standardization and mutual understanding of workflow models. More specifically, the key to workflow interoperation is the connection and call between heterogeneous workflow systems. There are mainly four different interoperations, namely, chain, nested, parallel synchronized and peer-to-peer, as explained in figure 5-1.

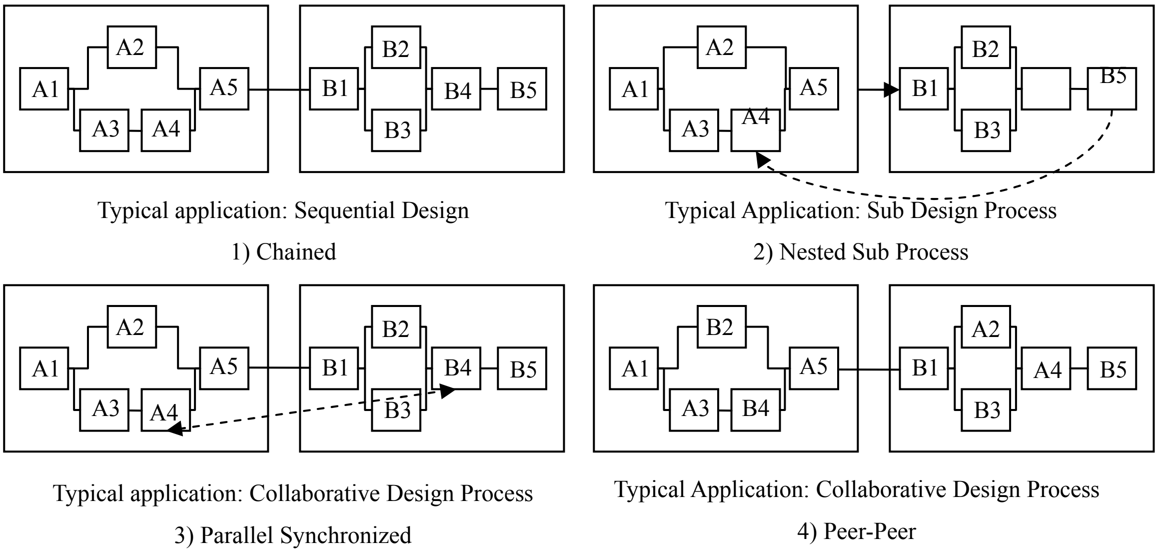


Figure 5-1 Four different workflow interoperation

- Chain interoperations: this model allows a connection point within process A to connect to another point within process B, the typical application of chain interoperation is sequential design;
- Nested interoperations: this allows a process executed in a particular workflow domain to be completely encapsulated as a single task within a (superior) process executed in a different workflow domain, the typical application of nested interoperation is sub design process;
- Parallel synchronized interoperation: This model allows two processes to operate essentially independently, possibly across separate enactment services, but requires that synchronisation points exist between the two processes. Synchronisation requires that once the processes each

reach a predefined point within their respective execution sequences, a common event is generated. The typical application of parallel synchronized interoperation is Collaborative Design Process;

- Peer-to-peer interoperation: this model allows a fully mixed environment. The typical application of peer-to-peer interoperation is collaborative design process;

The flexible workflow can be applied in the inner EDP of enterprise/design group to improve the efficiency of design work. But for cross enterprise EDP, the flexible workflow cannot be applied directly into whole cross enterprise EDP. But in cross enterprise engineering design work, its both loose coupled and tightly cooperated design process is a highly dynamic process which demands the overall agility. On the other hand, it is important to emphasize that the cross enterprise should be optimized in the overall context, so an overall model should be adopted to coordinate running of cross enterprise EDP.

It can be noted that the interoperation of heterogeneous workflow management system can be considered as procedure call, or data exchange between the systems. These calls can be local call or distributed Remote Procedure Call (RPC). WfMC defines the interface for heterogeneous workflow information interaction as interface 4, i.e. the workflow interoperability interface of enactment service. The enactment services can be interoperated by WAPI (Workflow APIs) which mainly complete the connection establishment, process control and activity management functions. The workflow interoperation in this dissertation is mainly realized by web services technology which is explained in in this Chapter in detail.

It can be concluded according to above discussion that actually the interoperation of workflow in cross-enterprise EDP is the call and transfer of control and data information based on common understanding, including the activation of activity and sub-activity, attributes and status of process and activity instance, transmission of workflow relevant application data, and reading/writing of workflow models. In cross enterprise EDP the control and data information between workflow enactment services can be invoked through a Workflow Application Programming Interface (WPAI). If WAPI invocation and model definition (including Naming mechanism, workflow relevant data and control) can be supported by different running environments, the information can be transmitted directly between workflow engines; otherwise the information in different environments should be mapped beforehand by gateway service, as explained in figure 5-2, for example, the design application in heterogeneous system can only be mapped before they can connect to and understand each other.

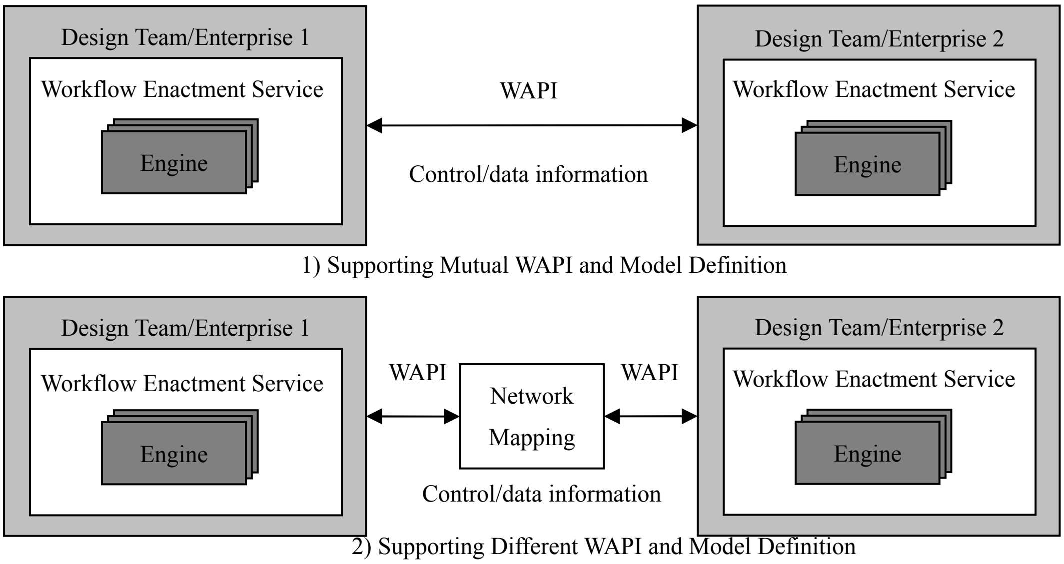


Figure 5-2 the nature of heterogonous workflow interoperation

Based on the inference and requirements discussed above, the concept of virtual workflow is proposed in this research as follows: the cross enterprise EDP can be considered as a business process composed of workflow in a virtual or geographically distributed virtual environment enabled by modern computer networking infrastructure, instead of the complicated business process composed of different heterogonous workflows and their interoperation. The enterprise design activity in the cross enterprise EDP will become a node of such a virtual workflow, the interoperation between inner heterogonous workflow systems in cross enterprise EDP will be changed into the transmission of control and data between virtual workflow nodes. The proposed virtual workflow change the complicated workflow interoperation into cross enterprise EDP overall modelling into the information transmission between workflow nodes, then the cross enterprise EDP can be described overall by a workflow model. On the other hand, the virtual workflow model can also benefit packaging the inner design process and application, and then only the general model and interface should be considered. The example of general description of cross enterprise EDP by virtual workflow and flexible workflow is illustrated in figure 5-3. As illustrated from figure 5-2, the complicated interoperation in EDP become the exchange of control and data information from two workflow system, while the network mapping mechanism should be used when the two workflow system adopt different WAPI and Model Definition.

As a typical illustration of an approach to the workflow, the global civil aviation industry is forming the globalization of design, manufacturing and market, especially for the development of Boeing 787 airplanes. The design of Boeing 787 is jointly implemented by the companies in American, Japan, Russia and Italy. For example, the wing box of Boeing 787 is jointly designed by Boeing, Mitsubishi and Fuji in Japan, the engineering design process is described for the collaborative wing box design process as shown as an example in figure 5-3. In this figure, by the packaging of virtual workflow, the collaborative design process can be considered as a virtual workflow starting from wing general design to assembly design.

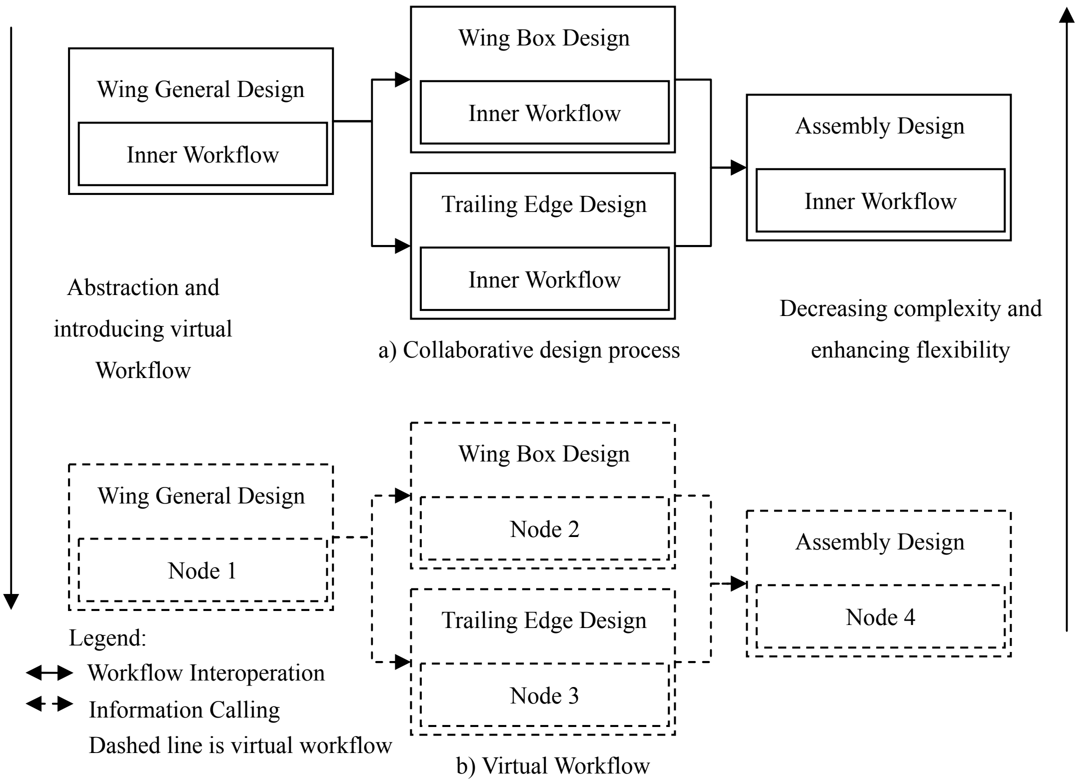


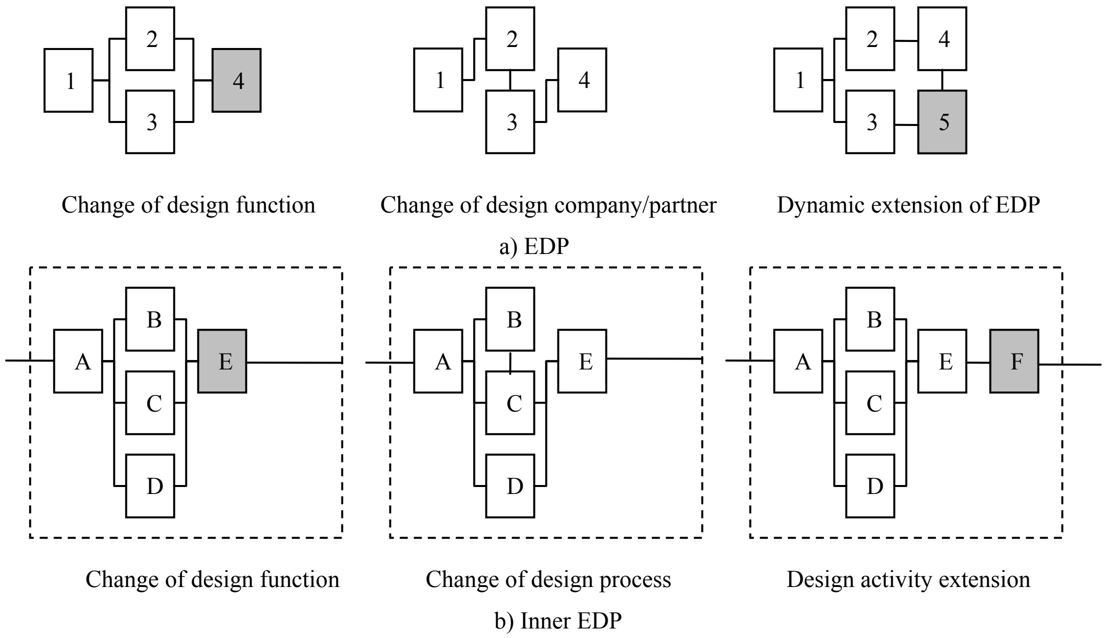
Figure 5-3 the description of collaborative design process based on virtual workflow

5.1.2 Team Engineering Design Process based on Flexible Workflow

The introduction of virtual workflow can improve the flexibility of EDP in general. For collaborative EDP, its inner engineering design process (Inner EDP) is also managed and supported by workflow management system; The Inner EDP fulfils the design functions of a design activity in virtual workflow. The dynamic characteristics of EDP not only demands that the general EDP can response to the dynamical competition, but also its Inner EDP can adjust itself to satisfy the general change of EDP. For example, the wing box design company must adjust its EDP according to the design requirement of Boeing Company and its design capability.

The dynamic reconfiguration and adjustment of EDP mainly is mainly reflected in the change of design activity function, the design process logic, and the adding or deleting of design activities.

All these changes require the reconfiguration and adjustment of the design activity’s Inner EDP. Using these features and flexibility associated with reconfiguration and adjustment embedded in an approach, the flexible workflow can be realized by the change of workflow node function, workflow control logic and the dynamic extension of workflow. The support of flexible workflow to the EDP is shown in figure 5-4.



Legend: C1, C2 means different company/team in EDP; A, B means the different activity in Inner EDP; Shadow box means the added design company/team or design activity.

Figure 5-4 EDP and Inner EDP supported by flexible workflow

In figure 5-4, a) denotes three changes happened in reconstructing and adjusting EDP, with example in figure 5-3, the function of design company 4 is changed from assembly design to the intensity verification for the deficiency in the compositional material intensity used in a complex product. The EDP process can also change in that the design company C3 design according to the design result of design company C2 instead of result of design company C1 or a new design company can be added in the EDP of the complex product. The b) denotes three changes happened in the Inner EDP, for example, the structure of wing box design can be changed, the Inner EDP of wing box can be changed or a new design activity can be added according to the change of EDP. So the reconstruction and adjustment of both EDP and its Inner EDP can be supported by flexible workflow for its node function change, control logic change and dynamic extension.

5.2 Flexible Workflow based on Multi Autonomic Objects

5.2.1 Introduction to Autonomic Computing

The proposed virtual workflow in section 5.1.1 provides the theoretical basis for how virtual workflow concept can be applied into each level of cross enterprise EDP modelling. The entity/node of EDP normally has certain objects and decision making capability and can be run in an autonomic or semi-automatic way. All these enteritis and their behaviours are coupled and dependent on each other to certain extend. How to enable the EDP have intelligence and adapt to changes with least intervention has become an urgent requirement of improving the efficiency of EDP. The key requirement is to identify a method to improve the intelligence decision capability of flexible workflow.

Autonomic Computing was first proposed by IBM in Agenda conference in October 2001. The term autonomic was derived from human biology. The autonomic nervous system monitors a person’s heartbeat, checks his blood sugar level and keeps his body temperature close to 98.6° F without any conscious effort on your part (IBM 2005). In much the same way, self-managing autonomic capabilities anticipate IT system requirements and resolve problems with minimal human intervention. Autonomic computing enables the IT system the ability to manage it and dynamically adapt to change in accordance with business policies and objectives. Self-managing environments can perform such activities based on situations they observe or sense in the IT environment rather than requiring IT professionals to initiate the task. These environments are self-configuring, self-healing, self-optimizing, and self-protecting. However, there is an important distinction between autonomic activity in the human body and autonomic activities in IT systems. Many of the decisions made by autonomic capabilities in the body are involuntary. In contrast, self-managing autonomic capabilities in computer systems perform tasks that IT professionals choose to delegate to the technology according to policies. An autonomic computing system has four typical attributes: Self-configuring, Self-healing, Self-optimizing, Self-optimizing.

Self-managing capabilities in a system accomplish their functions by taking an appropriate action based on one or more situations that they sense in the environment. Autonomic Computing considers it's managed IT system as Managed Resources. Its autonomy is realized by one or several autonomic Manager (AM). Every AM monitor its managed resources by Sensor and analyse it's found information at first, then AM plan and execute its needed action, the action of AM is realized by Effector. AM interacts with its managed resources by Touchpoint which is composed of Sensor and Effector. The above mentioned functions of AM constitute the Control Loop of AM (Sterritt et al 2005; Barrett et al 2005):

1) Monitor: The monitor function collects the details from the managed resources, via Touchpoints, and correlates them into symptoms that can be analyzed.

2) Analyze: The analyze function provides the mechanisms to observe and analyze situations to determine if some change needs to be made.

3) Plan: The plan function creates or selects a procedure to enact a desired alteration in the managed resource.

4) Execute: The execute function provides the mechanism to schedule and perform the necessary changes to the system. Part of the execution of the change plan could involve updating the knowledge that is used by the autonomic manager.

AM and its managed resources is shown in figure 5-5. The AM owned knowledge and its acquisition and use is very important to realize the autonomic computing. In an autonomic system, knowledge consists of particular types of data with architected syntax and semantics, this knowledge can be stored in a knowledge source so that it can be shared among autonomic managers. Knowledge can be obtained in following three manners:

1) The knowledge is passed to the autonomic manager.

2) The knowledge is retrieved from an external knowledge source.

3) The autonomic manager itself creates the knowledge.

Normally there is a Manual Manager in the AM which is an implementation of the user interface that enables an IT professional to perform some management function manually.

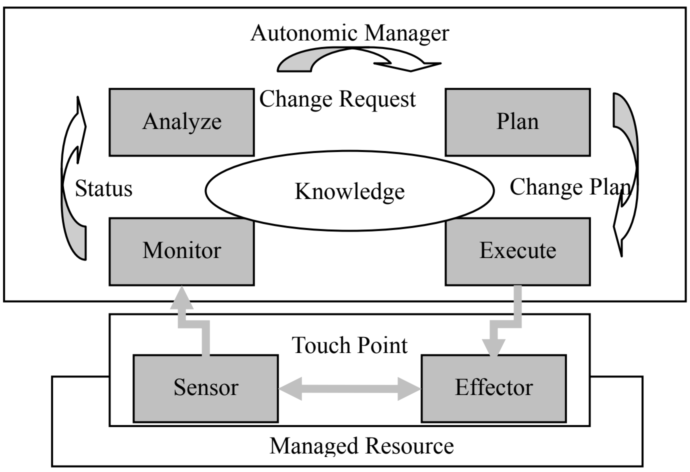


Figure 5-5 Autonomic Manager Structure

5.2.2 Flexible Workflow based on Multi Autonomic Objects

Although autonomic computing is mainly used in the IT system dynamic management, the basic idea of autonomic computing actually is very suitable to the management of flexible workflow system. The characteristics of autonomic computing enable the flexible workflow to find the changes in its running environment dynamically, analyse the changes and plan corresponding actions according to its knowledge, and finally adjust or reconstruct a flexible workflow by applying intelligent software agents, Internet/Web, workflow, and database technologies.

The application of Multi-Agent System has wide application in EDP, as in work reported. (Chao et al 2002) Agents provide a flexible and dynamic approach to distributed/multidisciplinary design team which can reduce redundant design activities and improve coordination. Available software techniques such as intelligent software agents, Internet/Web infrastructure, and database technologies in software prototype system had been implemented in a FIPA-compliant agent platform, with a wheel–axle assembly being used as a test case for system validation (Hao et al 2006).

In multi-agent system, agent has the characteristics, such as autonomy, sociality, reactivity and activity, EDP become a coordinated agent network with certain autonomy, each execute one or more functions and cooperate with other agents. Complex Adaptive System (CAS) is a dynamic network of many agents (which may represent cells, species, individuals, firms, nations) acting in parallel, constantly acting and reacting to what the other agents are doing.EDP is a typical complex adaptive system, distributed autonomic decision and coordination between agents is emphasized in a multi-agent system, it is exactly in line with the characteristics of EDP.

The introduction and application of agent technology in EDP which is constructed by flexible workflow can improve the intelligence of EDP. But the current research based on multi-agent technology is on how to improve the intelligence generally (Blake and Gomaa 2005). Research on integration between multi-agent technology and flexible workflow, as well as the realization of multi agent intelligence in workflow are still inadequate. The concept of agent and autonomic computing is different, unlike agent take imitating human thinking as the main target autonomic computing emphasize on a kind of system capability to adopt the dynamic changing environment. But many knowledge and concept in artificial intelligence, such as agent theory, knowledge expression and reasoning methodology, can benefit the realization of autonomic computing. Based on agent theory, a Multi Autonomic Objects (AO) flexible workflow representation method is proposed for this research to provide modelling foundation for this research in this paper, the definition of AO is given at first in this research as follows.

Definition 5-1 [AO]: AO is an intelligent entity based on autonomic computing, and embedded in a flexible workflow activity,

$AO = (Monitor, Analyzer, Planner, Executive, Knowledge, Touchpoint)$, among them,

1) $Monitor$ 、 $Analyzer$ 、 $Planner$ and $Executive$ is monitoring, analysing, planning and executing algorithm;

2) $knowledge$ is AO knowledge, which is the basis of AO intelligence;

3) $Touchpoint$ is the touching manager, $Touchpoint$ can be defined as a four-tuple, $Touchpoint = (Sensor, Effector, Orchestrator, Manual)$, where a $Sensor$ mainly realizes the status detection and information collection of AO managed resources, and sends collected data

to a $Monitor$,an $Effector$ mainly realizes the operation to managed resources, an $Orchestrator$ mainly realizes the communication and coordination with other AOs, and a $Manual$ mainly enables the interoperation with human when multi-AO failes to make decision.

The AO knowledge is very important to realize autonomy, monitoring, analysing, planning and executing algorithms of which all relies heavily on the expression of knowledge.

Definition 5-2 [AO Knowledge]: $knowledge = (BK, RK, CK)$, among them,

1) $BK = \{bk_i \mid i = 1,...,n\}$ is a body of basic knowledge which mainly describes some facts of workflow activity. For example, if the activity is structure design in EDP, then the bk_i stores basic knowledge relating to structural design, such as the material intensity, structure type and so on;

2) $RK = \{rk_i \mid i = 1,...,n\}$, is another body of reasoning knowledge which is the core of AO knowledge, AO plan suitable action according to RK ;

3) $CK = \{ck_i \mid i = 1,...,n\}$, contains control knowledge which mainly describes how to use the RK ;

Based on the above concept of AO, the Multi-AOs flexible workflow is proposed.

Definition 5-3 [Multi AOs-based Flexible Workflow]: Multi AOs-based Flexible Workflow (AO_FW) can be defined as a six-tuple, $AO\_FW = (T, AO, L, D, O, R)$, where

1) $T = \{t_i \mid i = 1,...,n\}$ is a set of AO_FW activities, $T \neq \Phi$;

2) $AO = \{ao_i \mid i = 1,...,n\}$ is a set of AO embedded in every flexible workflow activity, an ao_i is corresponding with t_i and ao_i exist based on t_i, denoted as $ao_i \xrightarrow{1:1} t_i$;

3) $L = \{l_j \mid j = 1,\ldots,m\}$ is the control relation between activities, named as control arc, $L \subset T \times T$;

4) D is a data arc, O represents an organizational model, and R is a resource model.

The intelligence of multi-AO based flexible workflow can then be realized by AOs embedded in each flexible workflow activity and its cooperation. An AO can detect the working status of a flexible workflow instance, and response to the change by detecting, analysing, planning and executing a control loop. An AO can and will also undertake decision making with uncertain information in EDP. A single AO can only cope with the intelligence requirements of a single flexible workflow activity, whereas the intelligence of whole system flexible workflow is realized by the collaboration of multiple AOs. In a flexible workflow instance, each individual AO works according to the control logic while it adjusts its embedded flexible workflow activity by its intelligence. If there is no change between the workflow running instance and expected instance, the AO doesn't start yet. The workflow runs according to its scheduled model. If there is any change, the AO is started and takes the following actions: detecting flexible workflow status via a Sensor, coordinating with other AOs, transferring acquired information to a Analyzer, analysing these information with its knowledge and choosing suitable actions which can be used to adjust flexible workflow activity functions or control logic, executing planned actions by a Effector. For some complicated decision making problems, AO can collaborate with other AOs by a Coordinator, or resolve domain problems by its Manual Manager. Chapter seven will describe the Multi AO intelligence algorithm in detail. The architecture of AO and Multi-AOs based flexible workflow is proposed and detailed in figure 5-6.

The detailed structure of AO is given in right part of figure 5-6. In Touchpoint of AO, Coordinator and Manual Manager is added besides the original Sensor and Effector. The detailed structure of multi-AO based flexible workflow is given in the left part of figure 5.6 In Multi-AOs based flexible workflow, an AO monitors running status of each AO, and takes suitable actions to operate or adjust workflow model by its control loop (according the actual application, the four steps in the control loop can be omitted or merged partly), then the flexible workflow instance is generated by a workflow engine. The detailed definition of flexible workflow model will be described in chapter 6. The proposition of Multi-AOs based flexible workflow improves the intelligence of workflow further based on flexible workflow, the flexibility of workflow become a kind of flexibility with autonomy and intelligence, with the capability to find change actively and take suitable actions to response it.

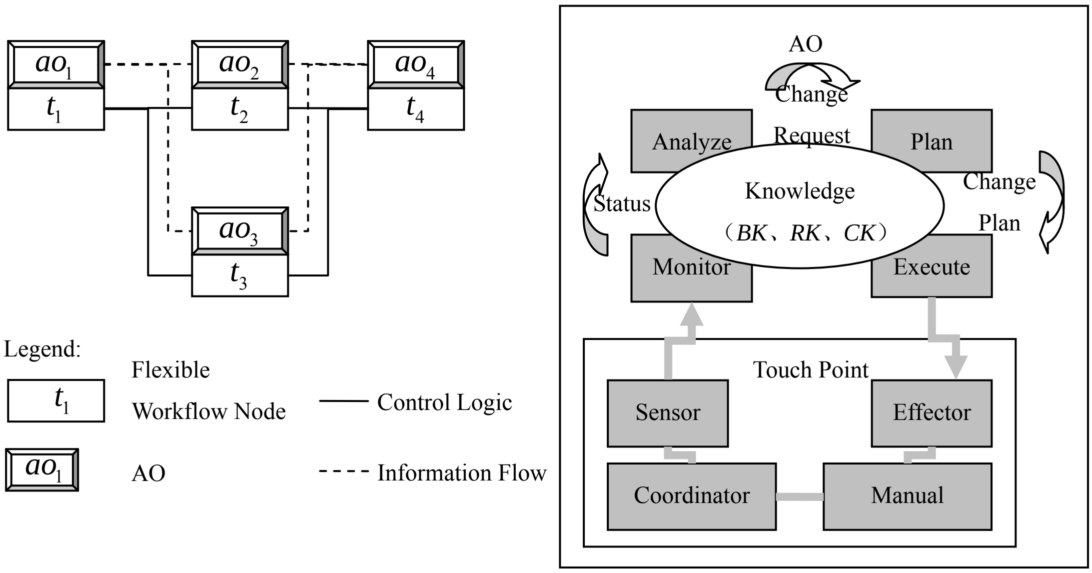


Figure 5-6 the architecture of AO and Multi-AOs based flexible workflow

It is worth noting that, although both AO and workflow engine operate a flexible workflow model, but they achieve this in different ways. An AO mainly detects the workflow instance and decides how to change workflow instances according to the change happened in workflow running instance and its environment. The intelligence of flexible change is realized; while the workflow engine mainly executes the flexible workflow model adjusted by the AO. The workflow engine is the tool to generate flexible workflow instances. The flexible workflow model is the basis for both of them.

5.3 Engineering Design Process Management based on Multi Autonomic Objects Flexible Workflow

Based on proposed virtual and flexible workflow, as well as the multi-autonomic objects flexible workflow, the framework of engineering design process management based on a multiple Autonomic object flexible workflow is proposed. The components and functions of the framework are introduced, and related modelling methods of engineering design process management framework are given in this section.

5.3.1 Framework of Engineering Design Process Management based on Multiple Autonomic Objects Flexible Workflow

The establishment of an EDP management system is the key to manage and support EDP. With the multi-AOs flexible workflow as the core, an EDP management framework is proposed in this thesis. Multi-AOs flexible workflow can describe the EDP management framework, and then the framework can be mapped into the EDP management system. By the intelligence of multi AOs and the flexibility of workflow, the rapid reconstruction and dynamic recombination of EDP can be enabled.

An EDP is a typical hierarchical structure, and a virtual workflow constitutes the overall level of

EDP framework, then the different design company/teams will establish their individual inner EDP according to the design activity of the virtual workflow. The inner EDP is also composed of and constituted by a flexible workflow model. According to the complexity of inner EDP, the inner EDP can also be decomposed into next level sub EDP (or compositional EDP) until the sub-EDP cannot be decomposed further at atomic EDP level. The whole EDP is a nested hierarchical structure constituted by multiple processes with different levels and granularity. The node in upper EDP is assembled by its sub-process. The level and granularity of the EDP framework is decided by the complexity of an engineering design process. An instance of nested hierarchical EDP for an aircraft design is given in figure 5-7. Figure 5-7 give an example of typical hierarchical EDP, it shows how a engineering design activity can be decomposed into sub-design process or even further, it can be modelled by virtual workflow and inner workflow, and the modelling of the EDP instance is the basis for the proper management of the EDP.

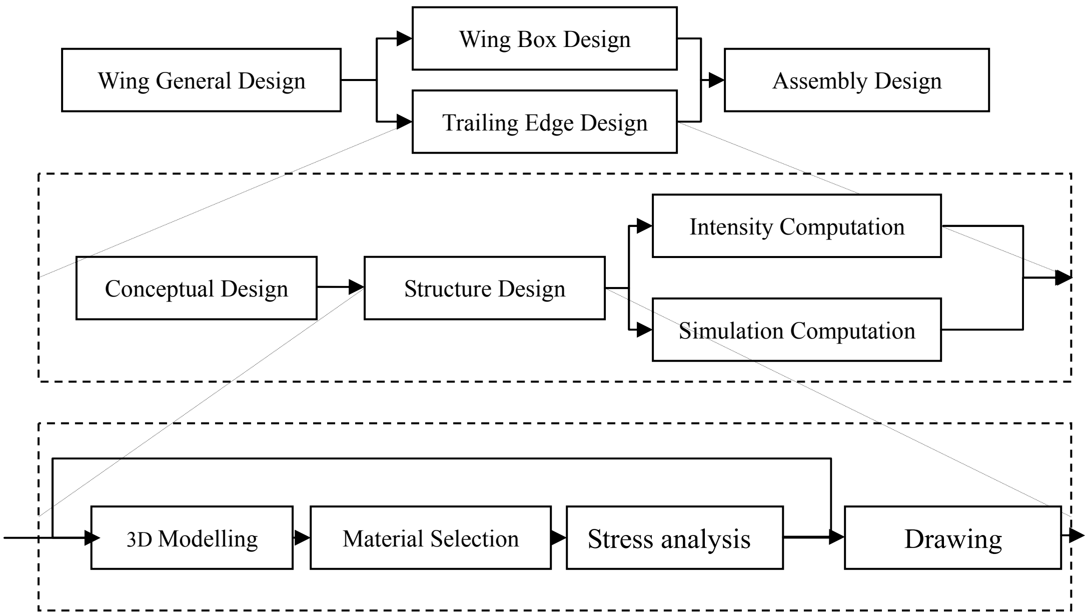


Figure 5-7 Instance of nested hierarchical EDP

By studying carefully the hierarchical structure of an EDP, every level of the EDP is a typical workflow process, i.e., the EDP can be constituted by multi-level nested workflow. The top of the EDP is virtual workflow which is responsible for the general control and coordination of collaborative or cross-enterprise EDP and assignment of design task to the next level workflow. The upper level workflow activity may be either a simple task node or a complex task node which can be decomposed further. Technically, the nest structure can be decomposed endlessly, but in the actual EDP, the decomposition of EDP is moderate. On one hand, excessive level will increase the control complexity, and on the other hand, the EDP can be a process constituted totally by simple and manageable task nodes after several times decomposition in the practical EDP.

The nested hierarchical structure of an EDP not only reflects the practical running of EDP, but also improves the flexibility and adaptability of EDP in the following three aspects: 1) the nested structure improves the re-construction capability of the system. The re-construction can be realized at different level and with different granularity. Every workflow activity can be an independent sub-system, so it can add or exit the EDP with least effect to other activities in the EDP; 2) the hierarchical structure improves system capability of re-combination, the EDP management framework can be constituted by workflow modules with different granularity and level, and the workflow modules can be re-used with any granularity; 3) the re-construction and re-combination capability of EDP management framework also implies the extensible capability of the system, and the system can be extended easily by adding the level of workflow or modules in any level. The framework of EDP process management is given in figure 5-8.

As shown in figure 5-8, the multi-AOs flexible workflow is the core of an EDP management framework, and all of the activities in this EDP are described by a flexible workflow model. Corresponding with the EDP, the model of multi-AOs flexible workflow is also a typical

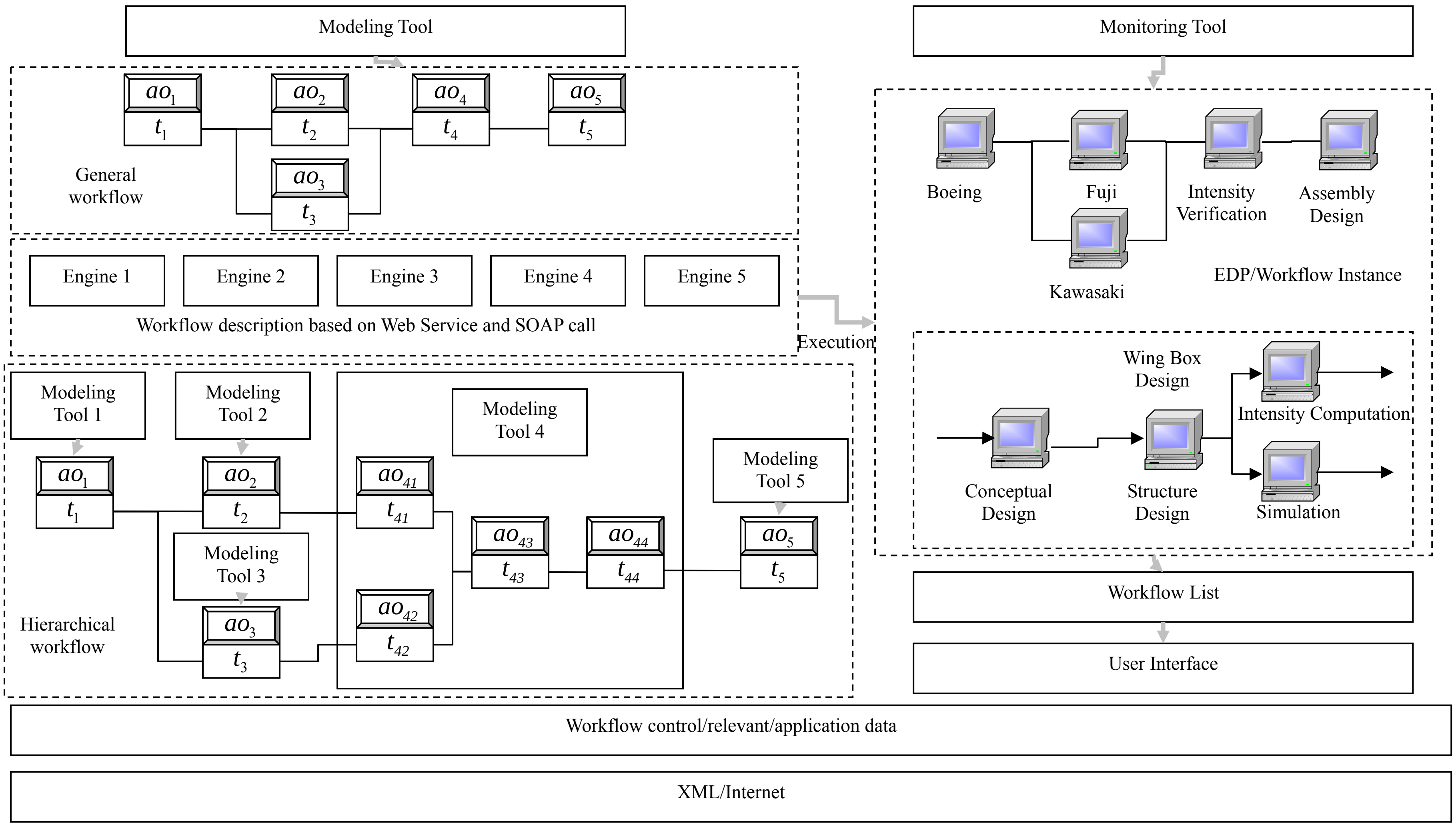


Figure 5-8 The proposed framework of EDP process management

hierarchical structure. The application of multi-AOs flexible workflow in the EDP management is exemplified by a two level flexible workflow model, and the general model and hierarchical model of activity t_4 in general model is given respectively. The intelligence of the EDP management framework is realized by the AOs embedded in flexible workflow activities and their collaboration. The AO can also be decomposed according to the decomposition of flexible workflow activities, the formalized definition of AO decomposition, and the flexible workflow activities introduced in detail in Chapter 6.

Modelling tool is one of the key modules in EDP management framework, which can implement the workflow model definition. The general model of EDP is actually defined as a virtual workflow by general modelling tool; and the decomposed flexible workflow is defined individually by modelling tool in each design company/teams. The principle of flexible workflow decomposition is that the decomposed flexible workflow model can realize the functions of upper node regardless of the concrete realization method, whereas the decomposed lower level flexible workflow should inherit the interface and input/output of upper flexible workflow node. In a hierarchical flexible workflow model, when a upper level workflow node is triggered, the input information is transferred into its correspondent lower level flexible workflow nodes, and the lower level flexible workflow is triggered. When the lower level workflow is finished, the lower level workflow node sends its output information to the parent node as its output flow.

The framework of EDP management based multi-AOs flexible workflow is a typical distributed workflow and each node in general or virtual workflow is executed by independent workflow engine. Similar with the hierarchical structure of flexible workflow, the inner flexible workflow model in design company/teams can also be executed by multi workflow engines according to the complexity of inner EDP. The collaborative EDP is normally cross-enterprise, distributed and heterogeneous, the EDP management framework should be built on an open Internet standard. In EDP management framework based on multi-AOs flexible workflow, the distributed workflow engine is described by Web Service (Booth and Haas et al 2005) technology, the collaboration of distributed workflow engine is realized by the SOAP call. Executed by the workflow engine, the flexible workflow instance, i.e., EDP process can be created. The right part in figure 5-8 is the

EDP instance created according to the multi-AOs flexible workflow model, this instance is the general EDP in the general view; the decomposed flexible workflow instance can implement the concrete function of design activities in EDP. The instance can interact with user interface or application program with workflow list. The characteristics of EDP management framework based on multi-AOs flexible workflow, and their meanings are listed in Table 5-1.

Table 5-1 The characteristics of an EDP management framework

ID	Characteristics	Explanation
1	Hierarchical	The EDP framework is hierarchical structure constituted by flexible workflow
2	Nested	The flexible workflow node in EDP management framework can be decomposed into lower flexible workflow
3	Modular	Each flexible workflow node can be a workflow model in next level
4	Abstract	The upper flexible workflow node in EDP management framework is the abstraction of its next level workflow model
5	Black Box	The flexible workflow node in EDP management framework is a black box, its inner structure can be neglected in modelling
6	Inheritance	The lower flexible workflow inherit the interface and input/output of its upper flexible workflow node
7	Similarity	Each level in the hierarchical EDP management framework is constituted by flexible workflow
8	Autonomic	The autonomic object is embedded into flexible workflow in EDP management framework

5.3.2ModellingMethods of Engineering Design Process Management Framework

For the EDP management framework based on multi-AOs flexible workflow, it is necessary to study the modelling methods of the EDP management. The traditional system modelling methods can be classified into top-down and bottom-up approaches. The top-down approach has the merits of clear modelling objective, high intelligibility however it is hard to determine the system outer or inner border in the initial stage. In contrast, the bottom-up approach has the merits of object-oriented, easy-construction while its disadvantage is too much attention to detail.

As described before, the EDP management framework can be a nested hierarchical flexible workflow model constituted by a flexible workflow in general. In the EDP, the general EDP is considered at first, the general model of EDP management framework is defined. In practice, the EDP modelling method is required to support each design company/team by enabling and facilitating their individual EDP management system, i.e., workflow system, according to the design task; for example, a design company can select suitable design partners to form the general EDP. The inner EDP management system will decompose the design activity and work according to decomposed flexible workflow model of the inner EDP.

Considering the merits of both top-down and bottom-up approaches, as well as the characteristics of the EDP management, a “general negotiations design, individual coordination assembly” modelling method of the EDP management framework is proposed as follows. Firstly, the EDP management framework is modelled mainly by the core leading or main contract enterprise. Each supporting or sub-contract company/teams in the EDP then negotiates with the core company or each other to determine the virtual flexible workflow and interface of their corresponding nodes in a virtual flexible workflow. Secondly, each design company/team models their inner EDP by a flexible workflow model based on the general demands and interface of the design activity. Thirdly, according to the complexity of the inner EDP, the inner EDP is decomposed to model the detailed flexible workflow model in the inner activities undertaken by the corresponding department of the design company/team. The above three procedures define the state of “general negotiations design” of the proposed modelling method. According to the defined flexible workflow model in each level of the EDP management framework, it is necessary to assemble these models into a general model. Firstly, from the bottom of the EDP management framework, each model is assembled according to the interface of their upper node, and by adjusting or re-constructing the lower model, the upper model is determined and constructed, Secondly, according to the same method, the even upper level model is constructed and determined until the flexible workflow model in each design company/team is determined. Finally the core enterprise will check if each design company/team can realize their design tasks, input/output and the interfaces, then coordinate with other design companies to adjust or reconstruct related models. After that, the core enterprise implements and links the whole the EDP management framework. The proposed “general negotiations design, individual coordination assembly” modelling method is illustrated in figure 5-9.

The modelling method in figure5-9 is actually a stepwise refinement and gradual integration process. The “general negotiations design” process is to establish the flexible workflow model in each level in the general view, whereas the “individual coordination assembly” is to coordinate the interface and input/output of each lower flexible workflow model, then to assemble the models from the bottom from the view of practical business process. The “general negotiations design, individual coordination assembly” method combines the merit of top-down and bottom-up method, not only can model the EDP management framework in general but also incorporate and consider the detail of the EDP management framework.

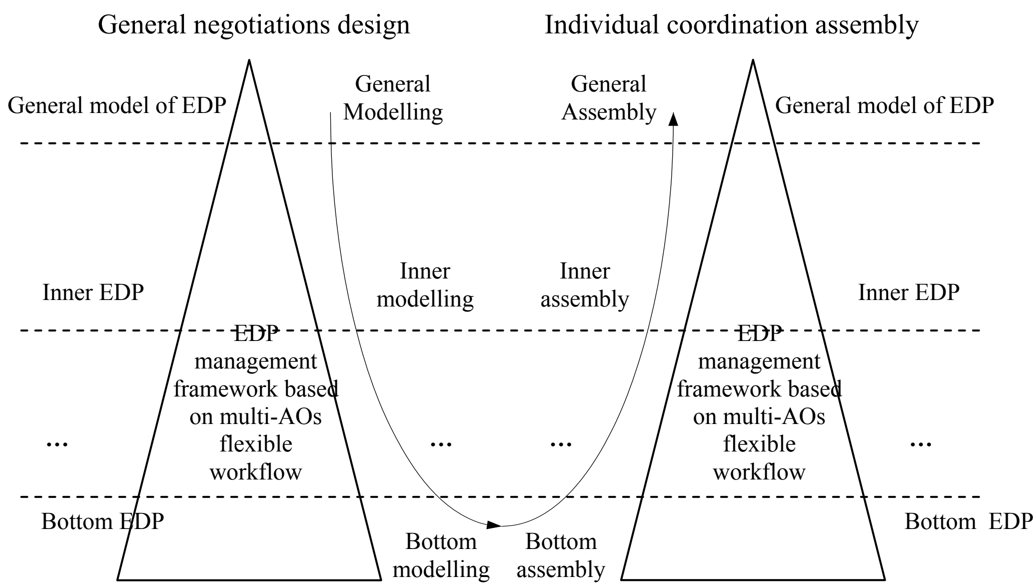


Figure 5-9 A “general negotiations design, individual coordination assembly” modelling method

In practice, in the modelling of an EDP management framework, most of the design companies/groups have their individual workflow management system to manage the EDP. There are two different situation in transferring the workflow model between companies, one of which is that the design company already has its workflow management system, and the EDP management framework only transfer the function demand and interface of the lower model to the company, The other is that the design company hasn’t its workflow management system, then the company can run the workflow management system in the EDP framework directly. In both situations, the interface of flexible workflow of the design company in the EDP should all meet the requirements of the general model.

Based on above discussion, four system architectures of the EDP management framework is proposed, it can help the designers how to choose their EDP management framework.

1) Case-based Architecture: enterprises in a EDP share the same workflow model;

2) Loosely coupled architecture: enterprises in EDP only have their own corresponding sub-workflow model;

3) Model interface share architecture: this architecture similar with the loose coupled architecture, the main difference is that the design companies need not to have their own workflow model; instead only need to follow the demand of interface, the design company can determine their own EDP;

4) Zero model architecture: this architecture is very suitable to the design company which only has simple EDP. Its EDP management is embedded and realized by the flexible workflow in the core design Company. This small design company can take a part in the collaboration in the EDP without heavy investment in a related IT system. A typical example of Zero model can be applied in company with which its design partner can log on its IT system to cooperate.

5.4 Chapter Summary

Aiming at the complex workflow interoperation in an engineering design process, the concept of virtual workflow is given at first. The flexible workflow can be applied in overall modelling of the dynamic engineering design process by a virtual workflow, and the flexible workflow can be used in every level of engineering design process. In order to improve the intelligence of dynamic engineering design process, based on autonomic computing technology, the autonomic object embedded in flexible workflow nodes is proposed and given first. Then four architectures of multi-autonomic objects flexible workflow are proposed to support identified scenarios. The workflow intelligent flexibility is realized by autonomic objects among their collaborations. Based on the above virtual workflow and multi autonomic objects flexible workflow, an engineering design process management framework based on multi-autonomic objects flexible workflow is proposed to support the dynamic engineering design process. Finally a “general negotiations design, individual coordination assembly” modelling method of EDP management framework is proposed.

Chapter 6 Flexible Workflow Modelling based on Dynamic Instances

The flexible workflow based on multi-AOs is the core of the Dynamic Engineering Design Process Management Framework. The flexibility of workflow is mainly realized by the flexible workflow model and its flexible execution. A workflow model with high flexibility and formal expression is the key and basis of improving workflow flexibility. The flexible workflow modelling technology supporting Dynamic Engineering Design Process is introduced and discussed in this chapter.

6.1 Flexible Workflow Modelling Methods

6.1.1 Traditional Workflow Modelling Methods

According to WfMC reference model, it can be found that all workflow management system has following 3 similar characteristics, as shown in figure 6-1:

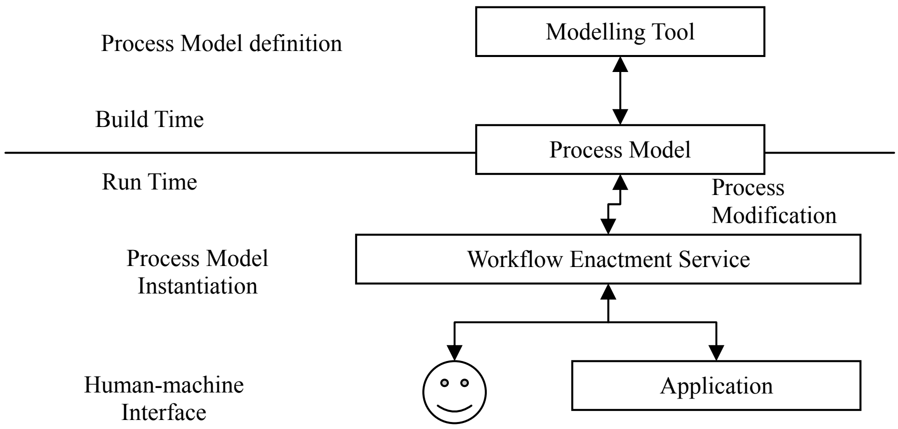


Figure 6-1 the traditional modelling method of flexible workflow management system

1) Build-time function: defining the workflow model;

2) Run-time function: executing workflow model;

3) human-machine interoperation in run-time: realizing the interoperation between human and application;

Broadly speaking, the execution and interoperation can be defined as run-time function, so the traditional workflow work method can be divided as two inter related stages: modelling and running stage. Especially in engineering design process, on one hand, the workflow model cannot be specified at the very beginning of engineering design process; on the other hand, the two stage work method neglect the inner relation between the defining and executing of workflow model, and is lack of the feedback from the workflow running instance.

6.1.2 Evolutional Workflow Modelling Methods

Dynamic engineering design process is in a continuous changing environment, the engineering design process should evolve continuously to keep up with the changing environment. This demands that flexible workflow should have excellent adaptation capability to dynamic changing running environment. Darwin proposed evolution theory in 1859, evolution is change of species in the interaction with its living environment, and the change can often improve the adaptability of specifies to the environment. Adaptation is the evolutionary process whereby an organism becomes better able to live in its habitat or habitats by any change of organic structure, physiological process and behaviour. The concept of evolution includes both biological evolution and non-biology evolution.

Inspired by Darwin's theory of evolution and similar to biology evolution, the workflow system proposed in this work is designed to be an evolution system as well in order to improve its adaptation capability to environment. It is designed to evolve higher with the interoperation between outer environment change and inner adjustment. The evolution of workflow is a process in which workflow modify its inner structure and function to adapt to environment. In order to define these workflow processes precisely, a formal method is used to describe the workflow modelling method in this dissertation, and the definition of workflow model and instance is given first.

Definition 6-1 Workflow Model: by referencing Definition 5-3, the workflow model is defined in this research as $W_i = (T, AO, L, D)$, among them, T means the set of Activities, AO is the set of Autonomic Objects, L is the set of control logics between activities, and D is the set of data links between activities, while i means the evolution version number of workflow model.

Definition 6-2 Workflow Instance: the workflow instance WIi can be defined as a 5 tuple, $WI_i = \left(TI_i, AOI_i, LI_i, DI_i, E_i \mid i = 1, 2, ..., N\right)$, among them, TI means the set of workflow activity instances, AOI is the set of autonomic objects instance, LI is the instances of control logic, DI is the instances of data link, E is the set of workflow running environment parameters, while i means the evolution version number of workflow instances. The mapping from W_i to WI_i is realized by the execution of a workflow engine described in Chapter 8.

Non-stabilizing factor is a concept from evolution theory and is very important for a system to evolve, the underlying causes of the non-stability is so-called fluctuation. A system can improve its self-adaptation capability to adapt the development and change of environment by interaction of fluctuation and environment selection. Corresponding with non-steady state, the steady state can be considered as system normal state, while the system evolution can be considered as a non-reversible movement tracing steady state from one to another. In an engineering design process, the workflow execution process is the interaction process between a workflow instance and workflow running environment. The change happened in the engineering design process (corresponding to workflow instance) and its running environment is considered to be a disturbance. If a workflow instance is deviated from a steady state to another state because of the existing of fluctuation, then the workflow system adjusts its micro structure continuously in terms of the change of models and instance structures, in order to adapt to changing engineering design process and its running environment. The fluctuation can produce a new steady state of workflow instance, and then new workflow instance is bound to be disturbed to a new non-steady state. This demands that a workflow model is modified to achieve a new balance. It is clear that a workflow model continuously improves its adaptation capability in this evolution process. The formal definition of evolutionary workflow modelling method is given as follows:

Definition 6-3 Workflow Model Defining Function: The workflow model defining function is defined as $W_1 = Define\_Model\left(R\right)$, where W_1 is the initial workflow model, while R is the actual business process. When a workflow modelling function is read in a input document of

EDP management system, it means a workflow model will be defined.

Definition 6-4 Workflow Model Executing Function: The workflow model executing function is defined as $WI_i = Execute\_Model(W_i, E), i = 1, \ldots, N$, among them, W_i is a workflow model, E is the workflow environment parameters set, the workflow instance is produced by execution of workflow model.

Definition 6-5 Workflow Instance Disturbing Function: The workflow instance disturbing function is defined as $S_{unstable} = Disturb\_Workflow(WI_i,\ D_{out}, D_{in}, S_{stable}), i = 1, ..., N$, where WI_i is a workflow instance, D_{out} is outer disturbance set, D_{in} is inner disturbance set, the state of workflow instance is changed from stable state S_{stable} to instable state $S_{unstable}$ by the affection of workflow instance disturbing function.

Definition 6-6 Workflow Model Modifying Function: The workflow model modifying function is defined as $W_{i+1} = Modify\_Workflow(W_i, WI_i, D_{out}, D_{in}, S_{unstable}), i = 1, ..., N$, where W_i is a workflow model before evolution, WI_i is a disturbed workflow instance, W_{i+1} is a workflow model after evolution. The instable workflow instance is modified, the evolution of workflow model can be described as $W_i \xrightarrow{evolve} W_{i+1}, i = 1, ..., N$. The state of workflow instance is changed to stable when the new instance is produced from the newly modified workflow model.

Considering the running of an evolved workflow model will affect the current running workflow instance after the workflow model is modified. Three processing strategies developed by Robert Muller, et al., 2004 have been adopted in this research, which consists of:

- Restart Strategy stop the running workflow instance, then restart its instance, the strategy is not suitable to the case that the former affection cannot be diminished;
- Continue Strategy keep the instance running according to formal model until it finish, start the new instance according to new model, this strategy is not suitable to the long term process modification;
- Migrate Strategy migrate all or part of active instance according to new workflow model, i.e., migrating original model W_i to W_{i+1}, then generate instance WI_{i+1} .

In migration strategy, the state of workflow instance and workflow instance modification method is considered, hence the migration method has its obvious advantage, i.e., the evolution of workflow model can be reflected in a workflow instance as soon as possible. This flexibility of updating workflow model can be improved greatly; the formal description of Migration Strategy is given as follows:

Definition 6-7 Executing Point: executing point $T_e \in TI_i, i = 1,...,N$ is the current executing node of workflow instance.

Definition 6-8 Modified Set: modified set is modified elements set in a workflow model after disturbance, as $M = (M_1, M_2, ..., M_m) \subseteq T_j \cap L_j, j = 1,...,K$.

Definition 6-9 Sequence Function: Sequence function means the login position of any element in Modified Set related to executing point, as

$$Sequence\left(T_e, M_j\right) = \begin{cases} -1 & if\ T_e\ earlier\ than\ M_j; \\ 0 & if\ T_e\ equal\ to\ M_j; \\ 1 & if\ T_e\ latter\ than\ M_j; \end{cases} \quad , \forall M_j \in M, j = 1,...,K .$$

Based on sequence function, Modified Set can be described as $M = M_o \cup M_e \cup M_l$:

Definition 6-10 Executed Set: $M_o = \left\{ M_j \mid Sequence(T_e, M_j) = -1, j = 1,...,K \right\}$;

Definition 6-11 Executing Set: $M_e = \left\{ M_j \mid Sequence(T_e, M_j) = 0, j = 1,...,K \right\}$;

Definition 6-12 Unexecuted Set: $M_{ue} = M_j \mid Sequence(T_e, M_j) = -1, j = 1,...,K$;

The migration strategy of a workflow evolutional modelling method is that when workflow model evolves from Wi to Wi+1, the current workflow instance is WIi, evolution target is WIi+1, then the migration function can be defined as:

$$Transfer(WI_i,\ WI_{i+1}) = Execute\_Model\left(M_{o\,i}, M_{e\,k}, M_{l\,i+1}\right), i = 1,...,N \qquad ,$$

while $k = \begin{cases} i & if\ Status(M_e) = 'Active' \\ i+1 & if\ Status(M_e) = 'Ready' \end{cases}$, $Status(M_e) = ('Ready', 'Active')$ is the State Function of a workflow model.

This strategy means that, for executed set, the workflow model is migrated until the new instance is started; for unexecuted set, the modification of instance takes effect instantly; for executing set, the executing state is considered to determine a migration strategy. The evolutional workflow modelling method establishes the link and feedback between workflow model and its running environment, and the above evolutional workflow modelling method is illustrated in figure 6-2.

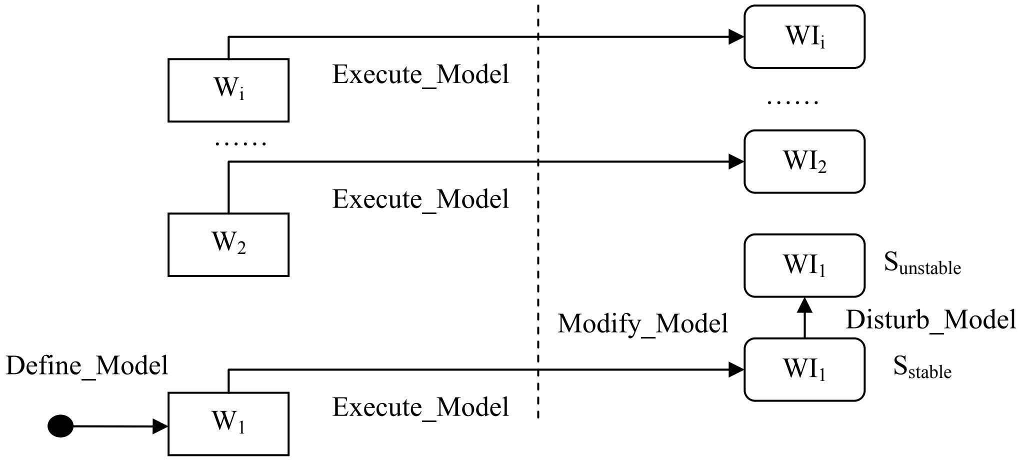


Figure 6-2 Evolutional workflow modelling method

6.1.3 Dynamic Instance-based Workflow Modelling Methods

Evolutional workflow modelling breaks through the two stage workflow modelling method, improve the workflow flexibility through the continuous adaptation of the workflow model to dynamic changing environment. One step further, similar with biological evolution, the adaptation process of evolutional modelling method is a gradual changing process; the workflow model should be modified first, and then is instantiated to adapt to its running environment. But for the dynamic engineering design process, it requires a capability for an immediate response.

Thinking in biological evolution, creatures can adapt to natural environment by evolution. After several generations, creatures have adaptable capability by accumulating knowledge in evolution. A typical example is chameleon in forest; it can change its skin colour with the change of outside environment, then to protect itself from the attack of its natural enemies. So if a workflow system has this capability, it can avoid the modification and migration of workflow model and can take suitable method to adapt to its running environment.

Similar with creature in natural environment, the workflow is to be put into concrete running environment to investigate its adaptation capability. At first, a workflow model is considered as a creature, the workflow instance then can be considered as one time action of workflow model in workflow running environment. Analogized with creature adaptation, it is not hard to find out that if one workflow model can be initiated to different workflow instances in different running environment that is workflow model have different forms in running environment, then the workflow model will have high flexibility.

Based on above description and previous research work, a dynamic instance based workflow modelling method is proposed. In workflow running process, the workflow instance will change and be modified by the execution of workflow engine instead that the workflow model be changed with the change of the workflow running environment. It means that every different workflow instance is derived from the same workflow model. The difference between the evolutional workflow modelling method and dynamic instance based workflow modelling method is that the modification of workflow model is changed to the change of workflow model based on workflow model. Because it does not involve the change of a workflow model, the migration of a different workflow mode is not needed to be considered in the workflow instance, even with the re-execution of modified workflow model. So the workflow model is considered as a creature, the model is static at its definition stage, so the workflow model is denoted as static model. But in the workflow running environment, the workflow instance is the running form of a workflow mode, and the workflow instance is denoted as a dynamic instance. The definition of a static model is same as definition 6-1.

Definition 6-13 Workflow Running Environment: A workflow running environment E_i is defined as a 2-tuple, $E_i = \left(EP_i, EN_i\right), i = 1,2,...,N$, while i means different running environment, EP_i the set of state parameters of environment, and EN_i is the description parameters of environment, which describes the changes in different environment separately.

Definition 6-14 Workflow Dynamic Instance: a workflow dynamic instance WI' can be defined as a 7-tuple, $WI_i^{'} = \left(TI_i^{'}, AOI_i^{'}, LI_i^{'}, DI_i^{'}, E_i \mid i = 1,2,...,N\right)$, while i means different running environment, $WI_i^{'}$ means dynamic instance of static model in i environment, $TI_i^{'}$ means activity sets of dynamic instance in i environment, same to remains elements in the definition.

Definition 6-15 Workflow Dynamic Instance Mapping Function: workflow dynamic instance mapping function maps the workflow static model to workflow dynamic instance, $WI_i^{'} = Map\_Instance(W,E_i),i = 1,...,N$,there are one to many mapping relation between workflow static model and workflow dynamic instance, i.e., $\forall E_i, \exists W \xrightarrow{Map} WI_i^{'}, i = 1,..., N$,, it means that there are specific running environment, all workflow static model can generate corresponding dynamic instance. Dynamic instance workflow modelling method is explained in figure 6-3.

The proposition of dynamic instance workflow modelling method puts more demands on flexible workflow modelling, and the workflow is required to response and adapt to the change without the workflow model modification. The dynamic instance workflow modelling method requests that on one hand the model should have flexible structure and workflow enactment service should execute workflow model flexibly, on other hand the workflow system should be intelligent enough to adjust workflow instance to adapt to complex running environment.

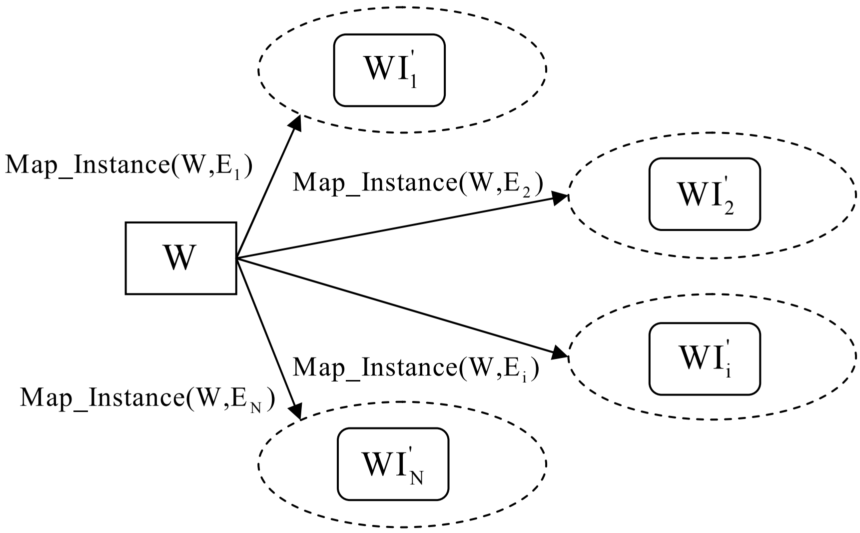


Figure 6-3 Dynamic instance-based workflow modelling

6.2 Flexible Workflow Modelling based on Dynamic Instances

6.2.1 Basic Elements of Workflow Flexibility

The dynamic instance based workflow modelling method requires that workflow model can be initiated as different workflow instances to adapt to its outside environment changes without modifying its model, Adjustment is often needed in this adaptation. This method requests that a workflow model has fine flexible structure to satisfy the change of a workflow instance. The fundamental flexible element in workflow model is explained in figure 6-4.

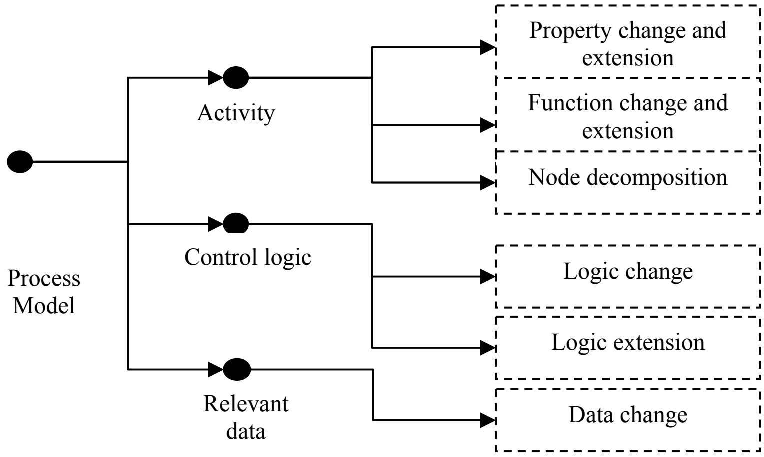


Figure 6-4 Flexible elements of a workflow model

It can be seen from Figure 6-4, the flexible change of workflow models is reflected in the change of model composition and its relations. With activities in a workflow model as an example, activity is the fundamental unit in workflow model which is mainly described by its properties and functions. The properties and functions of traditional workflow models are normally fixed, but in dynamic engineering design, the function of a specific design activity, such as a design task, the related designer, is always changeable and extensible. The change of workflow instances is realized by the change and extension of properties and functions.

For a workflow model, the Control Logic means the execution sequence and relation of activities in workflow model. The control logic is always fixed in a traditional workflow model, a follow-up activity of a specific activity is determined in model definition. But in the proposed dynamic engineering design process, a follow-up activity is often needed to be changed according to real time running state of workflow instance and its running environment. For example, a process planning engineer design activity should be added after a part has been designed.

Based on the above discussion, several principles to implement flexible workflow are given as follows;

1) Principle of Object-Oriented technology: by referencing object-oriented technology, the

property and function of a flexible workflow activity is defined and described, and the property and function of workflow activity is accessed or modified in workflow running. The execution of every workflow activity then will be decided by the property and function of workflow activity; then there are little dependence between activities because of the independence of objects; finally because the object is dynamically linked, the activities can be linked dynamically according to the execution result of the activity and state of running environment, the dynamic control logic can be realized;

2) Principle of Loose Coupling: In order to realize the dynamic control logic between activities, the composition of workflow should be decomposed first. Based on that, the elements of flexible activity should be loosely coupled, and then the change of activity property and function dose not affects the property and function of follow-up function;

3） Principle of Extensibility: in addition to the change of workflow activity, another obvious change of workflow instance is the extensibility of workflow models, including the extension of activity property and function, extension of activity (adding or deleting of activity), the workflow model supports the above requirements;

4) Principle of Data-centric: by analysing above three principles in detail, it reveals that above changes are all actually realized by the changes of property, function and rule. It can be said that the change of a management function can always be converted to the change of management parameters by decomposition and analysis. In view of data process, the flexible workflow model can be considered as assemblies of workflow control data and workflow application data. So basically the changes of flexible workflow are based on the changes of related data in workflow model.

Above fundamental elements and principles describe how to realize the flexible structure of workflow models, and how to change the structure by the autonomic objects embedded in activity of flexible workflow models. The fundamental flexible workflow principle, typically principle of data-cantered, provides the theoretical basis for the instance operation by autonomic objects. The flexible change of workflow is therefore realized by the operation and changes of workflow instance data.

6.2.2 Meta-Model of Flexible Workflow

The flexible workflow model in dynamic engineering design process is constructed by using the formal expression of flexible workflow model. Here a Meta Model is used, which is a kind of model to define the structure and rules of semantic model, and is referred in this study as model expressing model language. A Meta Model is used in this study to describe the each elements, element property and relations between each element inside the workflow model. It can determine the modelling method and formal expression of workflow model, and it can define fundamental objects in an EDP workflow model. More new objects can be added based on the Meta Model defined by users for a design process. The Meta model is the basis of dynamic instance of a design process workflow model, and the Meta model is described in this study by using Unified Modelling Language (UML) as shown in figure 6-5. The Meta model for dynamic instance workflow model is design base on the four principles defined in section 6.2.1, and it has following characteristics:

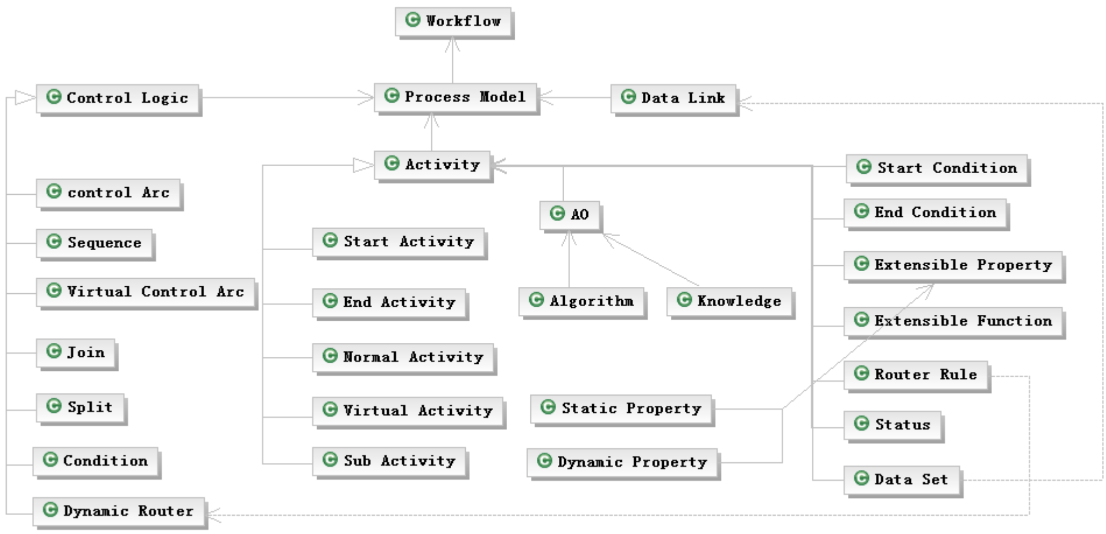


Figure 6-5 A Meta model for a dynamic instance flexible workflow model

1) The extensible property assembly and extensible function assembly is enabled and proposed to support the dynamic change of flexible workflow model;

2) The router rule in a workflow activity is proposed to support the dynamic control logic in a workflow instance;

3) The virtual task and virtual control logic is proposed to help modelling flexible workflow

models;

4) The autonomic objects embedded in an activity are modelled to support the intelligence of a flexible workflow;

Directed by a Meta model, the definition and formal expression of a flexible workflow model can be given in detail in following sections.

6.2.3 Activities of Flexible Workflow

6.2.3.1 Definition of Flexible Activity

To facilitate a precise and accurate definition of activity modelling of workflow, this study has formulated a set of definitions and they are introduced next.

Definition 6-17 Activity: Every node in workflow model is an activity, An activity is an abstract form of a workflow task, task $t_i \in T$ can be defined as a 13-tuple,

$$t_i = \left(ID, Type, D_{in}, D_{out}, Extend\_A, Extend\_F, S, Con_{start}, Con_{end}, Router, AO_i, O_t, R_t\right):$$

1) ID is unique identification of task t_i, it should be described in the form like $Virtual\_Workflow\_ID.Activity\_ID$;

2) $Type$ defines the type of activity;

3) D_{in} is input data set of activity, it comes from the data inputs of its all front activities, while D_{out} is output data of activity, it include all data outputs pointing to following activities;

4) $Extend\_A$ is extensible property set, while $Extend\_F$ is extensible function set;

5) S is the state set of activity which define all possible state in workflow instance;

6) Con_{start} is the start condition while Con_{end} is the end condition;

7) $Router$ is the dynamic router library. It can describe the dynamic router which can be determined according to the running instance. It can be supposed that the default control logic is

the first rule in dynamic router rule library;

8) AO_i is autonomic object model which is embedded into task t_i, AO_i and t_i have one to one relation;

9) O_t is organization model set which is called by activity, while R_t is resource model called by activity;

In order to better describe the detail meaning of activity elements, the definition of previous activity and subsequent activity is given below:

Definition 6-18 Previous activity: for one specific activity $t_i \in T$, its previous activity t_front is the activities which have direct control relation with t_i and lie in the beginning point of control relation. The set of previous activity can be defined as $T\_front_i = \{t\_end \mid \forall t_i : (t\_end, t_i) \in L\}$. $T\_front_i = \{t\_front_j \mid j = 1,2,...,M\}$ is the set of t_i previous activities in figure 6-6.

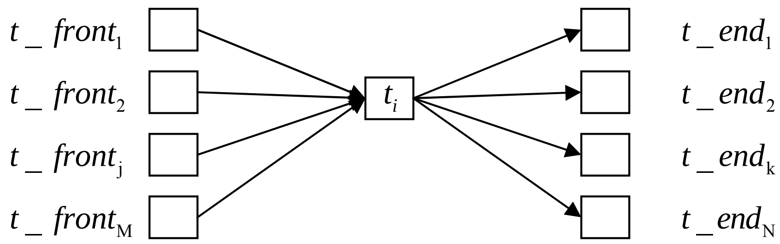


Figure 6-6 Previous and subsequent activity of workflow activity

Definition 6-19 Subsequent activity: for any specific activity $t_i \in T$, the subsequent activity of t_i is the activity which has direct control relation with t_i and lies at the end point of control logic. The subsequent activity set can be defined as

$T\_end_i = \{t\_end \mid \forall t_i : (t\_end, t_i) \in L\}$. $T\_end_i = \{t\_end_k \mid k = 1,2,...,N\}$ is subsequent activity set of t_i.

6.2.3.2 Activity Type

To define fully a flexible workflow meta model, five kind of activities are further defined:

Definition 6-20: the type of activity is further defined to enable refined modelling and is defined as a set of Enumeration variable, and it can be defined as $Type = \{Start, End, Normal, Virtual, Sub\}$, among which,

1) $t_{(Type=Start)}$ is a start activity, $\left|T_{(Type=Start)}\right| = 1$, meaning that a workflow model has and only has one start activity. $T\_front_{(Type=Start)} = \Phi$, means the previous activity of start activity is a null set;

2) $t_{(Type=End)}$ is an end activity. $\left|T_{(Type=End)}\right| = 1$ means that a workflow model has and only has one end activity. $T\_end_{(Type=End)} = \Phi$, means the subsequent activity of end activity is a null set;

3) $t_{(Type=Normal)}$ is an ordinary activity in workflow model. $t_{(Type=Normal)}$ is the main part of task in a workflow model, and the task of the workflow model is mainly implemented by an ordinary activity. The flexibility is realized by $Extend\_A$, $Extend\_F$ and $Router$ in activity qualifications;

4) $t_{(Type=Virtual)}$ is a virtual activity in workflow model, and the function of a virtual activity is similar with function of an ordinary activity. The main difference is that virtual activity is used to express the activity which is generated dynamically in a workflow instance rather than that it is defined at the very beginning of a workflow model;

5) $t_{(Type=Sub)}$ is a decomposable activity in workflow model. A decomposable activity not only supports hierarchical workflow model, but also can support dynamic decomposition and dynamic composition of sub=activity in workflow instances, This provide flexibility in determining the type of an activity as an activity which cannot be determined at the initial definition stage, can be determined at runtime.

6.2.3.3 Activity State

A State is used in this study as in system science to refer normally those model behavioural situation and characteristics, which can be observed and identified in a design process or a system. The running of a workflow instance is highly inter-related with the state of workflow node, and the change of workflow instance is jointly determined by the inside and outside running environment as well as the state of a workflow instance.

Definition 6-21: activity state is a very important property of defining the dynamic behaviour and transition of an activity in its life cycle, which can be a value of a state set $S = \{Initial, \mathrm{Re}ady, Executing, Suspended, Ter\min ated, Finished, Aborted, Unexecuted\}$.

Finite Automation (FA) is a formal language recognition model, The FA is adopted and used to describe formally the state change relation of an activity.

Finite Automation Machine is expressed as: $M = (Q, \Sigma, \delta, q_0, F)$, where, Q is no-empty and limited set of state, q is one sate of M ; Σ is an input alphabet, or input string belonging to Σ; δ is a state transfer function, $\delta : Q \times \Sigma \to Q$; q_0 is start state of M, $q_0 \in Q$; F is end start of M , $F \subseteq Q$ 。 A finite automation state transfer diagram is given in figure 6-7.

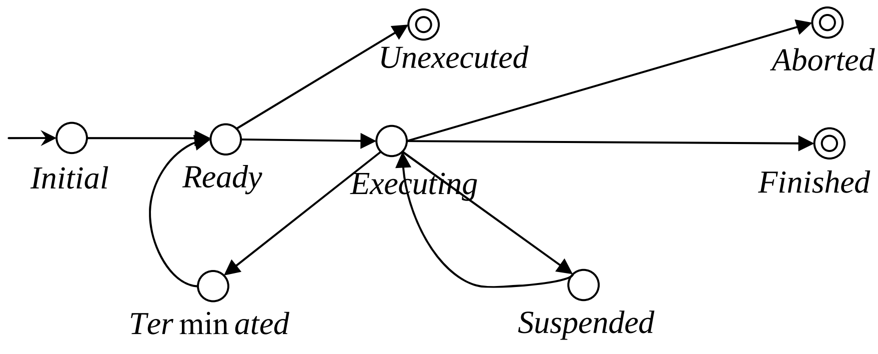


Figure 6-7 A finite automation state transfer diagram

Definition 6-22 activity state operation set: activity state operation set is defined as: $Op\_S = \{Start, Execute, Ter\min ate, Suspend, Succeed, Abort, Skip, \mathrm{Re}start, \mathrm{Re}sume\}$, while $if\ \ Con_{start} = True\ then\ Start$, $if\ \ Con_{end} = True\ then\ Succeed$.

With the above definitions, it is possible to define activities of a model more accurately and this will enable flexible and intelligent modelling by programming in chapter 9. By the formalized definition in this thesis it can provide a sound basis for unified language to exchange workflow model in EDP, and by its one-one relation to the graphical expression of workflow model, it can help to create the EDP to its graphical expression.

6.2.3.4 Activity extensibility property and function

Properties and functions are two types of attribute elements in an object. A property describes the features of class, whereas a function or also refereed to method describes the operation executed by a task. According to Principle of Extensibility and Principle of Data-cantered, an activity in a workflow model should have multiple properties and methods to achieve selection flexibility. Moreover, according to system science, any system is generated, run, continued and evolved in a certain environment. The structure, state, property and behaviour of the system are all related to the environment, and this is called environmental dependency. Considering the environmental dependency, the property and method of activity should be changed and extended to satisfy the demand of dynamic instances in different environments.

Regarding the properties of any object, some of the properties are inherent and their values are always fixed regardless of the changes, but other properties are changeable according to the change> Those former propertied are called static properties, and the later ones are called dynamic properties. The concept of static and dynamic property is introduced in the extensible feature of a workflow activity, so that the dynamism of dynamic properties becomes the main elements in determining the flexibility of a workflow model.

Definition 6-23 Activity extensibility: the extensibility property of activity is defined by an expression of $Extend\_A = \left(Extend\_A_{static}, Extend\_A_{dynamic}\right)$, where $Extend\_A_{static}$ is static extensibility property set, and $Extend\_A_{dynamic}$ is dynamic extensible property set.

Definition 6-24 static extensibility property: static extensibility property set is defined as

$Extend\_A_{static} = \{(Name\_Static\_A_i, Value\_Static\_A_i) \mid i = 1,...,N\}$, where i is a natural number, and the value of i can be decreased or increased, i.e., the property set can be dynamically extended. But for any static property, its value is fixed.

Definition 6-25 dynamic extensible property: dynamic extensible property set is expressed as $Extend\_A_{dynamic} = \{(Name\_Dynamic\_A_i, Value\_Dynamic\_A_i) \mid i = 1,...,N\}$, and i is a natural variable number, . But for any dynamic property, it value can be changed.

By using the above static and dynamic extensible property definitions, the necessary basic features of a workflow activity can be described. In a dynamic engineering design process, static definitions can be used to describe the fixed properties of an activity such as the names of design tasks, design product names; whereas the dynamic extensible property definitions can be used to define some of the changeable information, such as the design tasks. The static and dynamic extensible property set actually is constructed as a property table, which facilitates the change and extension of activities by using the table operations of the property table.

Predicate logic is a formal language, and so far is the most precise language to express human thinking. Predicate logic not only is suitable to express factual thinking, such as state and properties of an object, but also can be used to describe actions. By referring standard SQL language, the predicate logic is used in this research to define the extensible property set of an activity.

Definition 6-26 Operation set of extensibility property: the operation set of extensibility property is defined as:

$$Operate\_A = \begin{cases} \{Select(x,y), Insert(x,y), Delete(x,y)\} & 1 \\ \{Select(x,y), Insert(x,y), Delete(x,y), Update(x,y)\} & 2 \end{cases}$$

$$1, x = Name\_Static\_A_i, y = Value\_Static\_A_i \; ;$$
$$2, x = Name\_Dynamic\_A_i, y = Value\_Dynamic\_A_i$$

Definition 6-27 Extensibility Method of an activity: A Method describes a function of an activity, and the method here means the actual operation executed by activity, $Extend\_F = \{F\_Name_i, Parameter_i, Operator_i \mid i = 1,....,N\}$, where

1) $F\_Name_i = \{F\_Name_{ij} \mid j = 1,...,M\}$ is the operation set defined by the first order predicate, for example, $F\_Name_i = \{Conceptual\ Design, Drawing, Computing, ...\}$ means this operation set includes conceptual design, engineering diagram drawing and so on;

2) $Parameter_i$ means a specific parameter within the parameter set of a specific operation, and this set is made up from relevant workflow definition attributes;

3) $Operator_i$ is a combination operator on an activity, and its main function is to combine simple operations into a complex operation such as $Operator_i = \{Null, \wedge, \vee, \neg, ...\}$, where $Null$ is null operator. When an operator is expressed as $Operator_i = \{Null\}$, it means that the operation is a simple operation, $\wedge$ is AND operator, $\vee$ is OR operator, more complex operator can be defined further based on these logic basic operators.

Definition6-28 Operation Set of Extensibility Method: similar to the operation set of extensibility property, operation set of extensibility method can be described as $Operate\_F = \{Select(x), Insert(x), Delete(x), Update(x) \mid x = Extend\_F_i\}$, and it can be used to select, insert, delete and update extensibility method in an activity;

The extensibility property and method, as well as the graphical representation of their operation set are illustrated in figure 6-8.

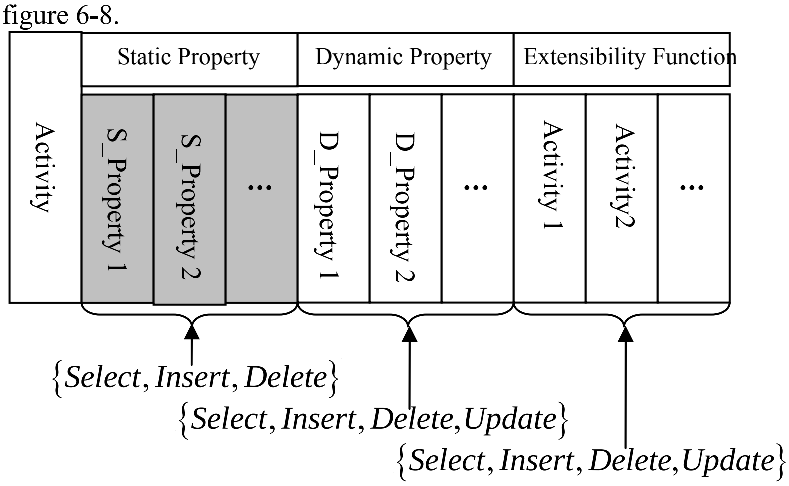


Figure 6-8 A graphical representation of extensibility property and method operations

The autonomic object in an activity is given in detail in section 5.2.2 and it is not repeated here.

6.2.4 Control Links of Flexible Workflow

6.2.4.1 Definition of Control Logic

In a workflow model, L is a control relation between activities, and is called control logic arc. The concept of control relation is derived from Directed Graph (DG), and the directed arc between activities is used to point from previous activity to subsequent activity. The meaning of control relation is that the subsequent activity linked by control relation is not allowed to be executed until the task of previous activity is implemented. Control relation reflects the critical definition of control and time sequence for all activities concerned. The dynamic router among activities can be realized by the dynamic change of a control relation. In the following section, the formal description of control relation is given at first, and the realization of dynamic router is introduced thereafter.

Definition 6-30 Control Relation: the control relation of a workflow mode is defined as $L = (Start\_A, End\_A, Con, Type)$, where $Start\_A$ is the start activity of a control relation, End_A is the end activity of a control relation. $Start\_A, End\_A \in T$, Con is the control condition of control relation, $Type$ is the control point type set of control relation.

According to the definition of control relation, the sequence of activities in a workflow model can be defined as $\exists L, \forall Start\_A, End\_A : Start\_A > End\_A$.

Knowing from the definition 6-30, the function of a control relation is jointly determined by the control point type and condition on it. The condition is Boolean function which is used to determine the actual point direction in workflow dynamic instance according to the type of control point. The definition of a control variable is given below at first.

Definition 6-31 Variable: Variable V_i is the abstract form of property, state, data and so on related to a workflow model and instance, $V_i = \{v_i \mid i = 1,\ldots,N\}$;

Definition 6-32: condition expression; condition expression is the relationship among variables, $C = \{c_i \mid i = 1,...,N\}$, the detailed definitions are as follows:

1) Setting computing operator $cal = \{+,-,*,/,()\}$, then $\forall v_i, v_j \in V, v_i \ cal \ v_j$ is computing expression ce,

2) $\forall ce_i, ce_j \in CE, ce_i \ cal \ ce_j$ is also computing expression.

3) if $vl \in V$, then operator $op \in \{=, \neq, <, >, \leq, \geq, \Pr edicative\}$, among them $Predicative$ is the set of predicate logic, so op can express more complicated logic relation;

4) Defining logic relation operator $lp = \{\neg, \wedge, \vee\}$, then $\exists lp \in \{\wedge, \vee\}, \forall c_i, c_j, c_i \ lp \ c_j$ is a condition expression.

As mentioned earlier, ECA can be used to describe flexible workflow model, ECA based workflow model can describe complex logic relationships. The ECA rule is used to describe control logic of flexible workflow model, and then the dynamic router of activities is constructed.

Definition 6-33 ECA rule: ECA rule $R_{ECA} = (E, C, A)$, while E is the event triggering rule, C is condition expression of a rule, A is the action of a rule. In describing a control relation, normally E is the event when the previous state of an activity is changing to end, A indicates the action for subsequent execution when the condition is satisfied.

The key of ECA rule is the monitoring of event. The basic theory of AO monitoring will be given in Chapter 7. The event is composed of atomic and composite event, $E = (E_{Atomic}, E_{Composite})$. The composite event $E_{Composite} = (E_{Atomic}, Op\_E)$ can be defined by the operation of atomic event, while composite operator $Op\_E = \{AND, OR, NOT, ANY, ALL\}$.

6.2.4.2 The classification of control relation

The semantic of control relation is very important to the workflow model definition. There are 20 workflow control modes (control logic) proposed by i2 Corporation in 2005.Because flexibility is the focus of this dissertation, only 6 relevant and commonly used workflow control logic modes proposed by WfMC are studied in this dissertation.

Definition 6-34: the control point type in control relation can be an enumeration set, $Type = \{Seq, AND\_Split, AND\_Join, OR\_Split, OR\_Join\}$, where Seq is the set of sequence control point, AND_Split is the set of and split control point, AND_Join is set of and join control point, OR_Split is the set of or split control point, OR_Join is set of or join control point, and the iterative point type can be composed of above control point types. The type of control point and join or split type joint determines the control logic of a workflow model.

1) Sequence control point: Seq is a basic logic relation of workflow model, meaning that the subsequent activity END_A executes its task after the task of its previous $Start\_A$ is finished. It is mainly used to construct the successive steps in workflow model, the condition expression can also be defined in sequence control point, and its default condition expression is $True$;

2) AND SPLIT control point: is mainly used to describe the split structure pointing from one activity to many activities in parallel. This means that when the task of previous activity $Start\_A$ is finished, there are many subsequent activities $End\_A_i, i = 1,...,N$ can be started, and split control point is described in figure 6-9 where e_i is trigger event and c_i is condition.

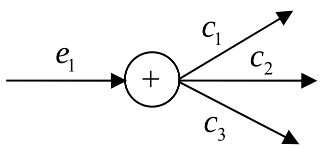


Figure 6-9 And split control point diagram

The AND SPLIT control point is jointly expressed by the control identifier $\oplus$ and split structure, if there are $n \in N$ branch starting from control identifier, there are one condition c_i in each branch, when the event e_i is trigged , all of c_i should be judged, $\forall c_i, if\ c_i = true\ then\ End\_A_i.enabled$, while $i = 1,...,n, n \in N$. The default situation is

$\forall c_i,\ c_i = true$, all of the subsequent activities from the split control point should be executed.

3) A AND JOIN control point mainly describes the workflow control logic structure in which multiple activities join into a one activity, and all of the parallel activities is synchronized. It means that subsequent activity End_A is started after its all previous activities $Start\_A_i$ are finished, the AND JOIN control point is described in figure 6-10.

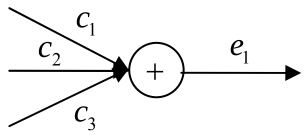


Figure 6-10 AND JOIN control point diagram

The AND JOIN control point is jointly expressed by control identifier $\oplus$ and join structure, if there are $n \in N$ branch pointing to control identifier, there are one condition expression in each branch, all of c_i should be judged, defining condition expression $C_j = (c_1, c_2, ..., c_m) \subseteq C_i = (c_1, c_2, ..., c_n)$, where $i = 1,...,n,\ j = 1,...,m, n \geq m$, i.e., C_j is subset of C_i. The necessary and sufficient conditions of triggering subsequent activity is $\forall c_j, (c_1 \wedge c_2 \wedge ... \wedge c_m) = true\ then\ End\_A.enabled$. The default situation is $C_j \equiv C_i$, meaning the subsequent can only be started when all of the conditions in joining branch is true.

4) OR SPLIT control point is used to describe the workflow control logic structure in which several branch started from one activity, and only one branch can be selected. Thist means that $End\_A_k \in \{Start\_A_i, i = 1,...,N\}$ is triggered after the previous activity $Start\_A$ is finished, as explained in figure 6-11.

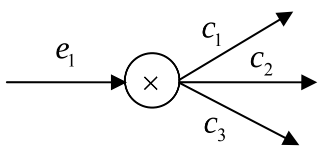


Figure 6-11 or split control point diagram

The OR SPLIT control point diagram is expressed jointly by control identifier $\otimes$ and or split structure. After the event e_i is triggered, all of the condition c_i,

$\forall c_i, \exists c_k = true, c_k \in \{c_i \mid i = 1, \ldots, n\}$ and $\forall c_i, \exists c_j = false, j = 1, 2, k-1, k+1, \ldots, n$ should be evaluated. If the condition c_k is satisfied, then the End_A_k is started.

5) OR JOIN control point describes the workflow control structure in which several branch joining into one activity, when any one of its previous activities is finished, the subsequent activity End_A can be triggered, as explained in figure 6-12.

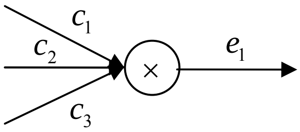
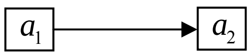
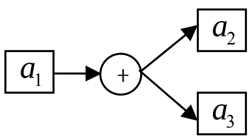
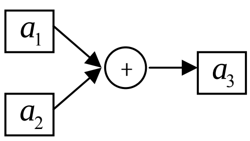
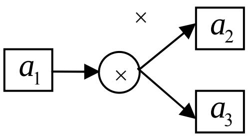
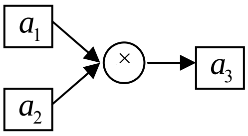


Figure 6-12 or link control point diagram

The OR JOIN control point is expressed by jointly by control identifier $\otimes$ and join structure, and if all of the conditions $\exists c_k \in \{c_i \mid i = 1, \ldots, n\} = true$, then End_A is started. The classification of control relationships and examples of related ECA rules are given in table 6-1.

Table 6-1 Classification of control relation and ECA rule example

Classification	Example		ECA rule
Sequence		a1: Conceptual Design a2: Embodiment Design	$End(a_1)$ $c_1 = true \quad Enable(a_2)$
And Split		a1: Wing General Design a2: Wing Box Design a3: Trailing Edge Design	$End(a_1)\ c_1 = true\ Enable(a_2)$ $End(a_1)\ c_2 = true\ Enable(a_3)$
And Join		a1: Intensity Computation a2: Simulation	$End(a_1)\ AND\ End(a_2)$ $c_1 = true\ AND\ c_2 = true$ $Enable(a_3)$

		a3: Drawing	
Or Split		a1: Structure Design a2: Designer A a3: Designer B	$End(a_1)\ c_1 = true\ Enable(a_2)$ $End(a_1)\ c_2 = true\ Enable(a_3)$ c_1 and c_2 is exclusive conditions
Or Join		a1: Concept 1 a2: Concept 2 a3: Layout Design	$End(a_1)\ c_1 = true\ Enable(a_3)$ $End(a_2)\ c_2 = true\ Enable(a_3)$

*: $End(a_i)$ denotes the end event of a_i, c_i denotes the condition of control logic, $Enable(a_i)$ denotes the activity is enabled.

6.2.4.3 Dynamic control relation and activity dynamic router

In dynamic execution of flexible workflow, in addition to the above control relationships and fixed control logic composed by control relations, the control logic of workflow instance should be often modified or extended dynamically. Some of these control logic relationships can be determined at the beginning of workflow modelling while some others can only be determined at runtime. The dynamic control logic is described in this section, and the static control arc and dynamic control arc is given at first.

Definition 6-35 Static control arc: the static control relation L is the control relation defined in workflow modelling, and is the same as the control relation in definition 6-30.

Definition 6-36 Dynamic control arc: the dynamic control arc is the control relation which is determined in runtime of workflow instance as: $L' = (Start\_A', End\_A', Con', Type)$, where $Start\_A'$ and End_A' are the previous and subsequent dynamic activities of a dynamic control arc, and Con' is a dynamic condition. It must be emphasized that the dynamic control arc is relative dynamic, although the dynamic control arc cannot be determined in workflow modelling, but for any specific workflow instance, the dynamic arc can be determined, i.e., for any specific workflow instance, dynamic arc actually becomes a static arc after such an instantiation or determination. This is called the static characteristic of dynamic control arc, $\exists WI_i^{'}, L' \rightarrow L$.

According to the static characteristics of a dynamic control arc, the dynamic control arcs can be expressed in the same way used in representing a static control arc. So the dynamic arc can be represented by a virtual control arc and control point, it can also be expressed by ECA rules formally. By analyzing table 6-1, it can be seen that the control relation of a workflow actually is the pointing relation after the condition generated from previous activities judged by subsequent activity. Specifically it includes two parts, one part is the condition for subsequent activities, and another part is the judgment of the condition generated from the previous activity. The dynamic control arc can be described jointly by control rules of previous activity and subsequent activity.

Definition 6-37 Dynamic Router: the dynamic router $D\_R = \left(DR\_Front, DR\_End\right)$ is used to describe the rule set of workflow dynamic control arc, DR_Front is the set of previous router rule, while DR_End is the set of subsequent router rule.

1) for any specific previous router rule, it is expressed by the subsequent rule library of each activity t_i, $DR\_Front_i = \left\{DR\_Front_{ij} \mid j = 1,...m\right\}$, every rule in the rule library defines a control relation, DR_Front_{i1} is static control arc.

2) for any specific subsequent router rule, it is expressed by the previous rule library of each activity t_i, $DR\_End_i = \left\{DR\_End_{il} \mid l = 1,...n\right\}$, DR_End_{i1} is default static link arc.

The first order predicate can also be used to define the operation set of dynamic router library, $Operate\_R = \left\{Select\left(x\right), Insert(x), Delete(x), Update(x) \mid x = DR\_Front_{ij}\right\}$. The dynamic control link relation between activities can be realized by dynamically choosing or extending router rule. In Definition 6-37, only one previous activity to multi-subsequent activities relation and multi previous activity to one subsequent activities is considered, but actually the control logics in a workflow model is a complex multiple to multiple network structure, typically as shown later in figure 6-15.

Within a workflow dynamic instance, especially for dynamic route, there are often exists the situation where the execution results affect the control relation of subsequent activities. For this situation, it is hard not only to express complex network logic, but also to distinguish the router rule. For example, if activity a_3 and a_4 are subsequent activities, then a_1、a_2 are the common previous activities of a_3、a_4. This logic can be denoted as $f_1(a_1, a_2) \to f_2(a_3, a_4)$, and it is hard to distinguish the subsequent router rule library. Confusion for router rule arises and this is called logic uncertainty of a dynamic router. The way to solve logic uncertainty of a dynamic router is the introduction of null node a_0, as shown in (b) of figure 6-13. By introducing the null node, the complex rule can be changed into the combination of two simple rules.

Then the logic uncertainty of dynamic router can be solved in following way $f_1(a_1, a_2) \to f_2(a_3, a_4) \Leftrightarrow f_1(a_1, a_2) \to f_0(a_0)\ AND\ f_0(a_0) \to f_2(a_3, a_4)$, the subsequent router rule of a_1, a_2 is changed to the previous router rule of a_0, the previous router rule of a_3, a_4 is changed to the subsequent rule of a_0.

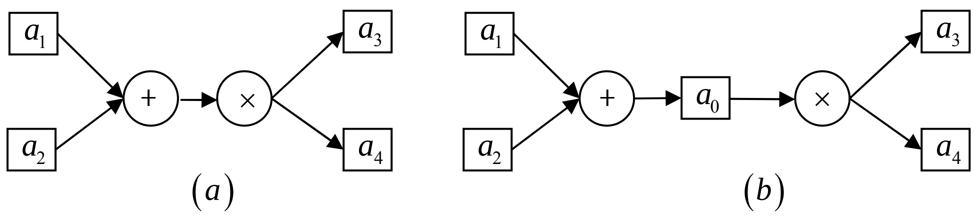


Figure 6-13 The logic uncertainty of dynamic router

These elements, such as dynamic extensible property, dynamic extensible method and dynamic router, are called basic elements used in this study to realize workflow instance flexibility.

6.2.5 Data Links of Flexible Workflow

Not only are there control relations between workflow activities, but also are there data relations between activities with the need of workflow execution. The data relation is called data arc. According to the reference model of WfMC, there are three kinds of data in workflow adopted in this study:

1) Workflow Control Data: The state of workflow model or instance is identified by workflow enactment using workflow control data;

2） Workflow Relevant data: The workflow relevant data is used to determine the change condition of instance states, and these data can be accessed and modified by workflow application;

3) Workflow Application data: The workflow application data is the data operated by application, for example, it includes design part name, design method in dynamic engineering design process;

In workflow instance, although the workflow data are not directly operated by workflow enactment service, but the workflow application data are still needed and transferred by the workflow enactment. The data link arc can be defined as:

Definition 6-38 Data Link Arc: the data link arc of workflow mode can be defined as $DL = \left(Start\_D_{out}, End\_D_{in}, L, D\right)$, $Start\_D_{out}$ is the output data set of start activity in data link arc, End_D_{in} is the input data set of end activity in data link arc, L is the control arc corresponding with DL, D is the data in L.

Several characteristics can be found from the definition of data link arc:

Data link arcs actually reflect the mapping relation between input data set and output data set, the data are actually transferred by data link arc according to control link arc.

Definition 6-39 Data Set: data set is composed of workflow relevant data and application data, $D = \left(D\_R, D\_A\right)$, while $D\_R = \left(d\_r_i \mid i = 1,...,N\right)$ is set of relevant data, $D\_A = \left(d\_a_i \mid i = 1,...,N\right)$ is set of application data.

Definition 6-40 Data Type: data type $Dtype = \left\{S\_type, C\_type, O\_type\right\}$ can be defined as the set of simple data, complex data and entity data, while,

1) Simple data $S\_type = \left\{String, Date, Time, Number, ...\right\}$ can be defined as string, data, time and number and so on;

2) Complex data $S\_type = \{String, Date, Time, Number, ...\}$ can be defined as complex data type composing by simple data, such as $C\_type_1 = \{String, Date, String, Number\}$;

3) Entity data $S\_type = \{File, Database, Object\}$ can be defined as set of file, database, object and so on.

6.2.6 Hierarchical Model of Flexible Workflow Activity

Dynamic engineering design process is a typical hierarchical structure; the dynamic engineering design process is composed of hierarchical embedded workflows. In a workflow model, the hierarchical structure is realized by decomposable activities.

Definition 6-41:decomposable activity: decomposable activity $t_{i(Type=Sub)} = \left(T_{i\_start}, T_{i\_end}, T_i, L_i, DL_i\right)$, where $T_{i\_start} = \{t_{i\_start}\}$ is the only start activity set, $T_{i\_end} = \{t_{i\_end}\}$ is only end activity set, T_i is the set of discomposing sub activities, and L_i is the se of control link arc. The decomposable activity is explained in figure 6-14, and the composition of activity should meet following rules:

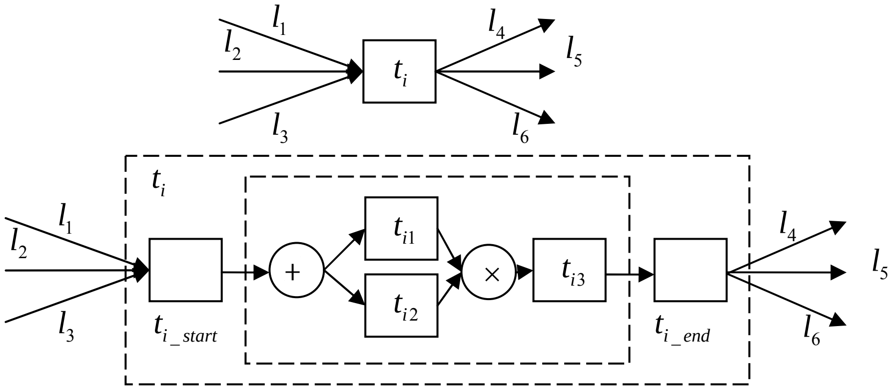


Figure 6-14 The activity hierarchical model

1) The previous activity set is equal to the only start activity set, $T\_front_i = T\_front_{i\_start}$, similarly $T\_end_i = T\_end_{i\_end}$;

2) The input data set of decomposable activity is equal to input data set of start activity in

decomposable activity, $D_{in} = D\_front_{in}$, similarly $D_{out} = D\_end_{out}$;

3) The start condition of decomposable activity is equal to the start condition of start activity in decomposable activity, similarly $Con_{end} = con\_end_{end}$;

4) The set of discomposing sub activity started from only start activity, ended in only end activity, $t_{i\_start} < t_i \in T_i < t_{i\_end}$, the control relations of decomposing sub activity is unrelated with decomposable activity;

5) The previous router library of decomposable activity is equal to the previous router library of start activity in decomposable activity, $DR\_Front = DR\_Front_{front}$, similarly $DR\_End = DR\_End_{End}$;

6) The only start activity and only end activity of decomposable activity cannot be decomposed further.

The AO are also need to be decomposed according the decomposition of activity, the AO of decomposed activity can be decomposed into Control AO and set of sub-AO embedded in sub activity.

Definition 6-42 AO Decomposition: the ao_i embedded in t_i can be decomposed into $ao_i = \left(ao\_control_i, \left\{ao_{ij} \mid j = 1,...,m\right\}\right)$, while $ao\_control_i$ is Control AO after decomposition which is in charging of overall autonomic computation, ao_{ij} is AO embedded in sub activity which is in charging of autonomic computation of sub-activity.

6.3 Flexible Workflow Graphical modelling

The graphical feature of workflow demands flexible workflow model is a useful feature for visualisation in graphical expression. Based on the formal definition of workflow, the detailed structure of activity in flexible workflow model is given at first, and then the graphical expression of flexible workflow mode is given further. In different workflow modelling method, the directed graph is simple and direct and easy understanding. It can express clearly the flexible elements in workflow model; the graphical workflow model is constructed based on direct graph.

6.3.1 Structure of Flexible Workflow Activity

The activity structure of a flexible workflow is illustrated in figure 6-15. The flexible workflow activity is composed of AO embedded in activity, previous router rule library, subsequent router library, start condition, end condition, extensible properties, extensible methods, input data set, and output data set. The function flexibility is realized by extensibility properties and methods, while the logic flexibility is realized by the previous router rule library and subsequent router library.

All previous link arcs are evaluated by previous router library of activities. Among the previous links, the broken line link arc means the dynamic link arc generated in a workflow dynamic instance. When the previous router is satisfied, the activity is started and the condition is assessed. If the condition is satisfied, then the activity is started.

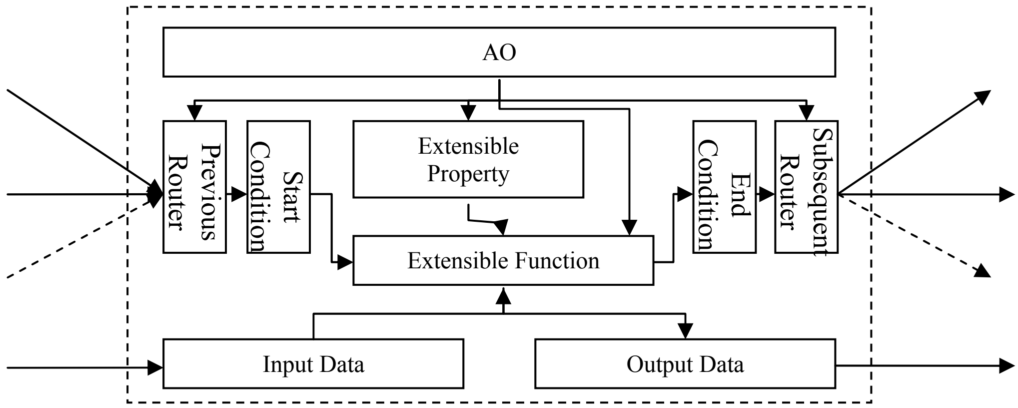


Figure 6-15 Detailed structure of flexible workflow activity

The data link arc is expressed by directed lines, and according to the definition of data link arcs, its data connections are always transferred on the control relation, if the control relation corresponding with data logic is connected, the workflow relevant data and application data in connected control arc is transferred into input data set.

The default method in extensible methods is executed to implement the function of the activity. The extensible method uses the input data set, generates output data set, then accesses and operates the extensible properties, when the task of the activity is finished, then the end condition is judged, and the related subsequent control link arc is triggered according to the current router rule in the subsequent router rule library if the condition is satisfied.

The AO is triggered into active state when there are previous link arc is satisfied, the AO then can be decided if work on flexible elements in activities according to the state of workflow instance, then the dynamic instance of workflow instance can be realized by the execution of modified flexible elements by the workflow engine introduced in Chapter 8.

6.3.2 Graphical Expression of Flexible Workflow Model

Based on the formal expression of flexible workflow model and extension of directed graph, the graphical expression of flexible workflow model can be defined in table 6-2. The elements of flexible workflow model are strictly correspondent with the formal expression workflow model, and the graphical expression can help express and understand the formal expression of workflow model. In table 6-2, the graphical expression of flexible workflow model is illustrated by the activities.

Table 6-2 The graphical expression of dynamic instance based flexible workflow model

Categories	Elements	Graphic	Description
activity	Start Activity	$\boxed{S}$	Start activity, the start condition can be defined in start activity
	End Activity	$\boxed{E}$	End activity, the end condition can be defined in end activity
	Ordinary Activity	$\boxed{}$	The extensible property, extensible method and dynamic router can be defined in ordinary activity
	Virtual Activity		Virtual activity is generated in dynamic instance
	Decomposed Activity	$\boxed{Sub}$	The decomposition activity is the activity which is can be decomposed in dynamic instance

Control relation	Control Link Arc	⟶	Control link arc means express fixed control logic, the condition can be defined in control arc
	Virtual Link Arc	- - - - →	Virtual link arc means the control logic which is determined in dynamic instance
	And control Point	⊕	And control point can be combined with control logic into And Join Link, and And Split Link
	Or Control Point	⊗	Or control point can be combined with control logic into Or Join Link, and Or Split Link
Data relation	Data link Arc	─□─→	Data Link Arc means the data transfer relation, and can be combined with control Link Arc in graphical expression
	Virtual Data Link Arc	- -□- -→	Virtual Data Link is correspondent with virtual control link arc, means the virtual data transfer relation

6.4 Chapter Summary

Aiming at the current two stage flexible workflow modelling method, and overcoming its disadvantage in dynamic engineering design process, the evolutionary workflow method is given based on evolution theory. One step further, the dynamic instance-based workflow modelling is proposed. By analysing the characteristics of flexible workflow, the four principles of flexible workflow are illustrated, and a flexible workflow Meta model in dynamic instance-based modelling method is constructed. Based on the proposed Meta model, the flexible workflow activities and autonomic objects, flexible control logic, data logic, as well as hierarchical model of flexible workflow activities are defined formally. Finally the detail structure of flexible workflow activities and graphical expression of flexible workflow is given. With these extensive and rigorous definitions, the work now has a solid foundation to model a work flow flexibly.

Chapter 7 Flexible workflow Autonomic Object Intelligence algorithm

The dynamic instance based flexible workflow modelling method consisting of definitions found in Chapter 6 can be used to define a highly flexible workflow model. In dynamic engineering process management framework based on multi-autonomic objects flexible workflow, the flexibility of workflow is realized by the autonomic object operation on flexible elements in flexible workflow instance. The autonomic object intelligence algorithm is the key to improve the intelligence of flexible workflow. However, the process of designing is very complex. Besides, the information required during the conceptual stage is incomplete, imprecise, and fuzzy. In responding to these early design features, fuzzy set theory is proposed to be used to handle linguistic ambiguity and other associated problems at this stage. The autonomic object intelligent algorithm, especially fuzzy reasoning algorithm in uncertain condition, is described in this chapter.

7.1 Formalized Expression of Flexible Workflow Intelligence Algorithm

7.1.1 Expression of Flexible Workflow Change

The flexibility of workflow is actually the rapid adaptation capability of workflow to dynamic change, this change is composed of change of workflow running environment and change of workflow itself. The change processing method of workflow can be studied in the view of exception handling. Broadly speaking, the exception not only includes the error in workflow, but also most importantly includes the offset between the dynamic instance and default instance. Because the research focus of this work is on flexibility, the error handling is not studied in this dissertation, the change processing, especially the intelligent processing for uncertain change, is studied emphatically. According to the definition 6-14, different dynamic instance can be generated in different running environment from one static workflow mode.

Flexible workflow change γ is defined as the offset between the dynamic instances WI_i

generated from static model in workflow execution and expected instance WI_0 from static model. According to the formal expression of flexible workflow model, $\gamma = \{\gamma e, \gamma a, \gamma f, \gamma s, \gamma c, \gamma d, \gamma t\}$ or $\gamma = WI_i - WI_0$, the '-'is the offset or difference between instances instead of minus. When WI_i is totally equal to WI_0, $\gamma = \Phi$.

1) $\gamma e = \phi \cup \{\gamma e_i \mid i = 1,...,n\}$ is the set of changed environment parameters;

2) $\gamma a = \phi \cup \{\gamma a_i \mid i = 1,...,n\}$ is the set of changed extensible properties;

3) $\gamma f = \phi \cup \{\gamma f_i \mid i = 1,...,n\}$ is the set of changed extensible methods;

4) $\gamma s = \phi \cup \{\gamma s_i \mid i = 1,...,n\}$ is the set of changed workflow states;

5) $\gamma c = \phi \cup \{\gamma c_i \mid i = 1,...,n\}$ is the set of changed control logics;

6) $\gamma d = \phi \cup \{\gamma d_i \mid i = 1,...,n\}$ is the set of changed workflow data;

7) $\gamma t = \phi \cup \{\gamma t_i \mid i = 1,...,n\}$ is the activity extension in workflow instance, specifically means the activity inserting and deleting.

Above ϕ means there is no changed happened in workflow instance.

The change of flexible workflow is happened in the execution of workflow instance, the change is inherent characteristic of workflow dynamic instance, the flexible workflow model has the agile structure to satisfy and adapt to change.

In workflow execution, if $\gamma = \Phi$, the embedded AO doesn't trigger any action, the workflow instance is executed according to the original workflow model; but when the AO detects that $\gamma \neq \Phi$, the AO is triggered, and it can acquire related information by $Monitor$, execute autonomic computation, operate on the workflow instance by $Effector$.

Above AO trigger method is called automatic trigger, i.e., the embedded ao_i in t_i detects the variables in instance and start the analysis and judgment on workflow instance. Besides the flexible workflow system can also set with trigger methods, such as information trigger, default trigger and manual trigger. Information trigger works on that basis that the ao_i is triggered by a piece of information from another AO ao_j . Default trigger is that ao_i is always started before t_i no matter whether $\gamma \neq \Phi$ or not. The manual trigger is that the AO doesn't monitor the instance variables of activity, but is triggered by a user. The default trigger method is an automatic trigger.

7.1.2 Theory of Flexible Workflow Intelligence

After the AO is triggered, the values of variables in flexible workflow instance are collected continuously by the AO. Once the offset is detected, the AO is triggered, the variables in dynamic instance WI_i are acquired by the $Monitor$ of AO. The input variables is matched and reasoned with AO knowledge. The concrete procedures are:

1) AO analyzes the input variables;

2) AO plans the concrete actions according to the above analysis;

3) AO operates the flexible workflow instance by $Effector$ to realize the intelligent change.

In the AO computation process, the current acquired variables maybe is not enough for the reasoning, i.e., this means evidence is not adequate. This means that AO embedded in one activity need instance variables from multi related activities, i.e., the change of activity in an instance is jointly determined by the states of multiple related activities. Meanwhile in a flexible workflow instance, the running of activity t_i is related with its previous activity t_front , and the running of its subsequent activity is partly determined by its running result, so the running of activities in instance is inter-related. The instance change of activity t_i is jointly determined by multi AOs including ao_i . In this situation, multi AOs need to collaborate to reason with their knowledge.

7.1.2.1 Instance variables and monitoring principles of AO

For intelligent reasoning of AO, its evidence is coming from the instance variables of flexible workflow dynamic instance. These variables are the input of AO intelligent reasoning, the instance variables is acquired from a $Monitor$ of AO.

Here an instance variable σ is the set of running parameters acquired from current workflow instance WI_i by $Monitor$. Instance variables are the input of AO,

$$\sigma = \sigma_e \cup \left(\bigcup_{i=1,..n} \sigma_{ti} \right) = (\sigma_k \mid k = 1,...m), \quad m = |\sigma_e| + \sum_{i=1,...,n} |\sigma_{ti}|,$$

1) σ_e is the set of environment variables, $\sigma_e = (\sigma_{ej} \mid j = 1,...m)$, m is the number of environment variables;

2) σ_{ti} is the set of instance variables correspondent with activity t_i, σ_{ti} is made up of elements in flexible workflow, $\sigma_{ti} = \sigma_{ti}^a \cup \sigma_{ti}^f \cup \sigma_{ti}^s \cup \sigma_{ti}^c \cup \sigma_{ti}^d$, while $\sigma_{ti}^a = (\sigma_{ti}^{aj} \mid j = 1,...m)$ is the set of extensible properties in activity t_i, $\sigma_{ti}^f = (\sigma_{ti}^{fj} \mid j = 1,...,m)$ is the set of extensible methods in activity t_i, σ_{ti}^s is the current state of t_i, $\sigma_{ti}^c = (\sigma_{ti}^{cj} \mid j = 1,...,m)$ is the set of router rules of t_i, $\sigma_{ti}^d = (\sigma_{ti}^{dj} \mid j = 1,...,m)$ is data set of t_i;

Not all of the instance variable σ is the input of AO, for any specific AO ao_i, set $\sigma_t = \sigma_e \cup \sigma_{ti}$ is the main instance variable of ao_i, is the main evidence of AO reasoning.

7.1.2.2 AO reasoning process

The analysing and planning process of AO is a typical reasoning process in which an AO finds new fact by using AO knowledge and finally operates the flexible elements of a dynamic flexible workflow instance. In autonomic computation, the analysing and planning process are two interrelated and distinct stage. For changes in an intelligent flexible workflow, the change is often obvious, it can only be omitted in autonomic computation, meanwhile, these two states are often be executed in parallel. For brevity, these two processes are combined into one process in the research of reported this dissertation.

Classified by the certainty of the used knowledge, the reasoning can be divided into certain reasoning and uncertain reasoning. Most of the objects and phenomenon in real work is not strict and accurate; many concepts are vague and don't have clear classification and boundary. It is hard to express and process by accurate mathematical model. Fuzziness is a type of deterministic uncertainty. It describes the event class ambiguity. Fuzziness measures the degree to which an event occurs, not whether it occurs.

Flexible workflow system is a dynamic complex system, and in dynamic engineering design process, the instance variable and autonomic object in the knowledge domain of engineering design are often uncertain, such as "the intensity of wing should be increased", "pressure is too high", "if stress concentration then increase the thickness of wing box". The higher is the complexity of flexible workflow system, and the lower is the system precision, and the higher the ambiguity of the system. The classical logic reasoning has could not satisfy the demand of AO reasoning in flexible workflow. Meanwhile human are often think and reason in the situation while knowledge is incomplete and inaccurate. In order to realize the intelligent change of flexible workflow instance, the AO must have the capability to reason in uncertain condition. The fuzzy reasoning of AO will be studied in this chapter.

7.1.2.3 The working principle of the proposed AO

The final reasoning result of AO is the operation on flexible workflow instance WI_i when $\gamma \neq \Phi$. The concrete procedures are executed by the $Effector$ on correspondent flexible workflow activity. The operation set on extensible property and method of flexible workflow activity, based on this the operation set ρ on flexible workflow elements is given in Table 7-1.

Table 7-1 The operation on elements in flexible workflow

Elements	Operation Set ρ
Extensible	$\left\{Select\left(t_i, x, y\right), Insert(t_i, x, y), Delete(t_i, x, y), Update(t_i, x, y)\right\}$

Property	t_i is activity, x is the name of specific property in activity, y is the value of x
Extensible Method	$\left\{\begin{matrix} Select\left(t_i, x, y\right), Insert(t_i, x, y), Delete(t_i, x, y), Update(t_i, x, y), \\ Setdefault\left(t_i, x\right) \end{matrix}\right\}$ x is the name of specific property in actvity, y is the value of x, $Setdefault\left(t_i, x\right)$ is the default method of t_i
State	$\left\{SetStatus\left(t_i, x\right)\right\}$, x is the any value in state set S
Control Logic	$\left\{\begin{matrix} Select\left(t_i, \varphi, x, y\right), Insert(t_i, \varphi, x, y), Delete(t_i, \varphi, x, y), \\ Update(t_i, \varphi, x, y), Setdefault\left(t_i, \varphi, x\right) \end{matrix}\right\}$ $\varphi = \left(F, A\right)$ is identifier, F and A is the operation on previous and subsequent router rules separately, x is the name of router rule, y is the name of x, $Setdefault\left(t_i, x\right)$ is the default router rule of t_i
Data set	$\left\{Select\left(t_i, x, y\right), Insert(t_i, x, y), Delete(t_i, x, y), Update(t_i, x, y)\right\}$ x is a data name in data set, y is the value of this data
Activity Extension	$\left\{Insert(WI_i, x), Delete(WI_i, x)\right\}$, x is activity

The final reasoning result actually is above operation and the composition of above operations. It is worth noting that the operation set of Flexible Workflow does not include operation on workflow environment.

7.1.2.4 Proposed AO intelligence in flexible workflow

The AO intelligence is derived from an intelligent reasoning process in which AO makes knowledge matching and fuzzy reasoning according to the input instance variables, and get the operation on flexible workflow elements. The implementation of AO intelligence in flexible workflow is given by flow charts in figure 7-1.

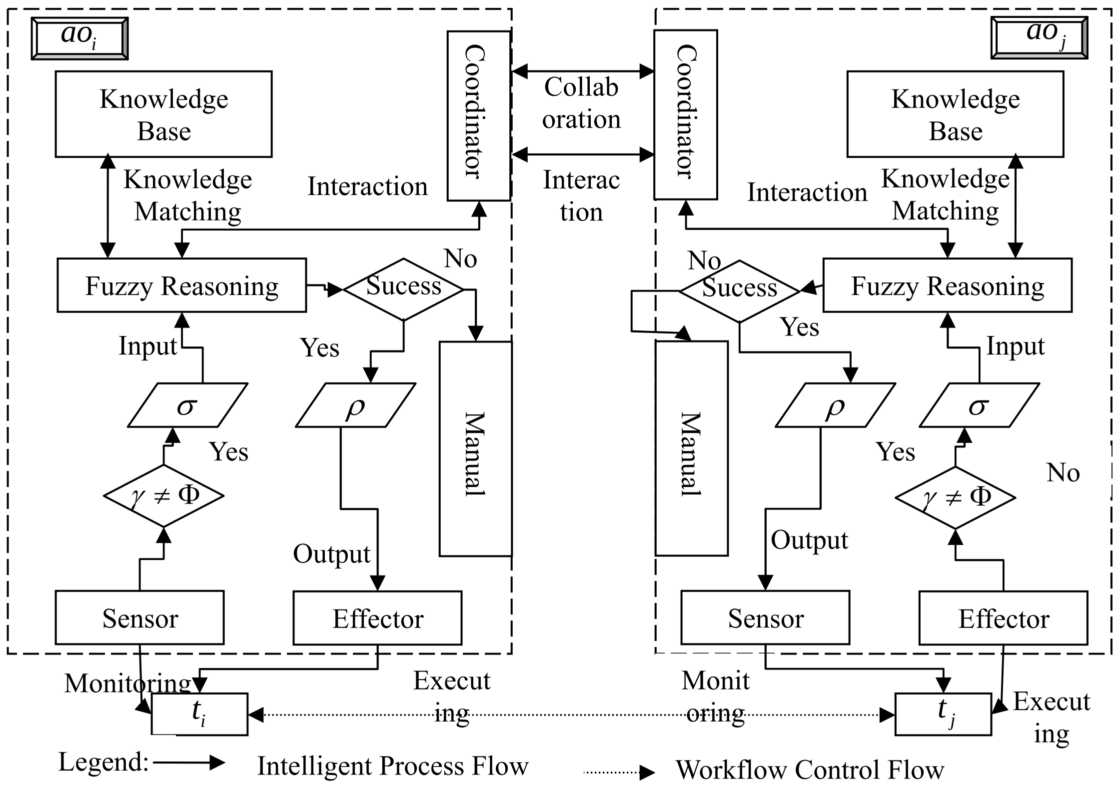


Figure 7-1 Principle of AO intelligence in flexible workflow

Taking two activities of flexible workflow in figure 7-1, the mechanism of AO intelligence is explained as follows:

1) An *Monitor* monitors the running of workflow activity, the instance variable σ is inputted into an *Analyzer* when the change is detected;

2) *Analyzer* matches the input variable with knowledge in knowledge library, and maybe cooperates with other AOs;

3) When the reasoning is successful, the reasoning results, i.e., the operation set ρ is got from the reasoning process, *Effector* operates on the flexible workflow activity according to the reasoning result;

4） When the reasoning is successful, an AO cooperates with *Manual*;

The intelligent principle of mutli-AOs is extended based on above principle;

The knowledge expression, fuzzy reasoning algorithm and multi-AOs cooperation reasoning algorithm is the core of above AO principle. The state space of flexible workflow can be defined to express the intelligent principle.

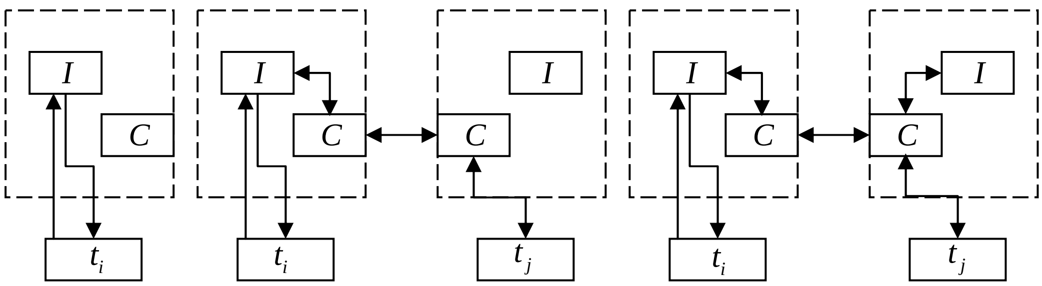
Supposing the intelligent reasoning process is P, then $P=\left(S_0,o,S_k\right), S_0=\left(S_0^{t1},S_0^{t2},...,S_0^{tn}\right)$ is the initial state of flexible workflow activities, o is operation set, $S_k=\left(S_k^{t1},S_k^{t2},...,S_k^{tn}\right)$ is the target state of flexible workflow activities. The intelligent reasoning process of AO is to find suitable ρ to change flexible workflow from initial state to target state, and to realize the flexible change of workflow instance.

7.1.2.5 Intelligent principle of Multi-AOs

Viewing form the scope of AO intelligence, the intelligence of AO can be divided to local intelligence, half-cooperation intelligence and cooperation intelligence.

- the local intelligence means that the AO can reasons and get ρ independently without the cooperation other AOs;
- half-cooperation intelligence means that the AO need to cooperate with AOs by acquiring their instance variables when the input instance variables is not enough for the needed reasoning fact; half –cooperation is suitable for the situation where a activity state is jointly determined by other related activities;
- cooperation intelligence means that AO need to jointly reasons with other AOs according to their separate knowledge;

The different AO intelligence is explained in figure 7-2.



a) Local Intelligence b) Half collaboration intelligence c) Collaboration Intelligence

Legend: I is reasoning machine, C is coordinator

Figure 7-2 Intelligent principle of Multi-AOs

7.2 Autonomic Object Intelligence Algorithm based on Weighted Fuzzy Reasoning

7.2.1 Extended Mamdani Fuzzy Reasoning System of Autonomic Object

Reasoning is a kind of thinking process in which a new judgment is derived from a known judgment according to a specific strategy. Fuzzy reasoning is an uncertain reasoning by using of fuzzy knowledge. Fuzzy logic is a form of multi-valued logic derived from fuzzy set theory to deal with reasoning that is approximate rather than precise.

Flexible workflow is highly dynamic system, especially when the flexible workflow is applied in supporting dynamic engineering design process. In dynamic engineering design process, there are much uncertain fuzzy information, and fuzzy knowledge also exists in the interdependent relations between different event. The traditional precise reasoning can solve the certain knowledge reasoning well, but it can't solve the uncertain reasoning caused by fuzzy information. The precise mathematic model is not needed in fuzzy logic reasoning, instead the knowledge is acquired like the way natural language works, then the knowledge and reasoning conclusion which cannot expressed by precise information can be expressed by using of membership function and fuzzy rules.

Fuzzy reasoning system is an advanced computation system based on fuzzy sets theory, fuzzy "if-then" rules and fuzzy reasoning method, the framework of basic fuzzy reasoning system is illustrated in figure 7-3. The fuzzy reasoning system of AO realizes the non-linear mapping from instance variable σ to operation set ρ on flexible elements in flexible workflow. The inputted fuzzy set and related membership function can be defined according to the different domain of σ and ρ in dynamic engineering design.

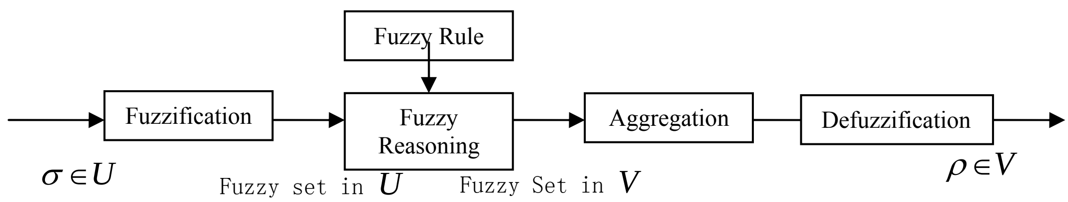


Figure 7-3 Basic fuzzy reasoning system

Mamdani fuzzy reasoning system is a typical fuzzy reasoning system; it works according to a set of precise input variables and a group of control rules getting from experienced domain expert: at first the fuzzy reasoning is conducted according to input membership function, then multiple fuzzy conclusion is aggregated by aggregation operator, after defuzzification, the precise conclusion is outputted, finally the mapping between input and output. But in the flexible workflow application in dynamic engineering design process, not only are there many fuzzy reasoning processes, but also are there precise reasoning processes. If there is only single fuzzy reasoning system, simple question is bound to be complicated for precise reasoning. The AO should conduct not only fuzzy reasoning but also precise reasoning. Based on Mamdani fuzzy reasoning system, an Extensible Madmani (EM) fuzzy reasoning system is proposed, the EM can realize both fuzzy and precise reasoning, and the hybrid fuzzy and precise reasoning, then the reasoning conclusion of EM fuzzy reasoning system can be more realistic by a human expert alike thinking mode. The structure of EM fuzzy reasoning system is given in figure7-4.

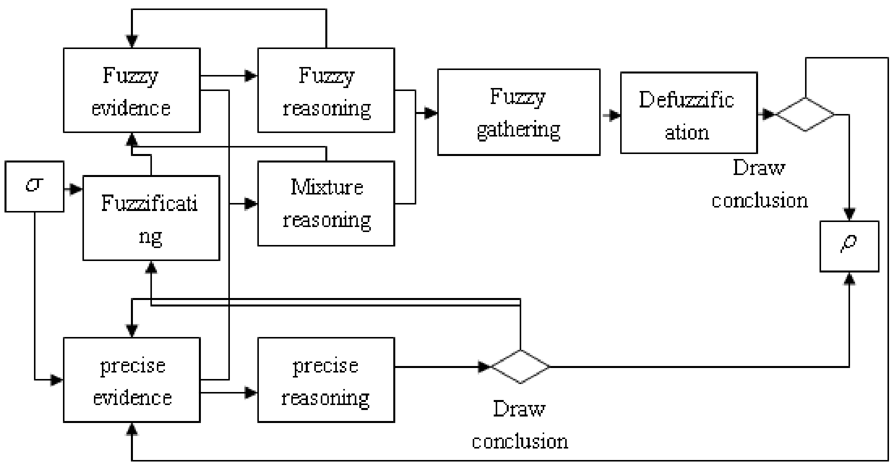


Figure 7-4 EM fuzzy reasoning system

In figure7-4, the instance variable σ is the input of the EM fuzzy reasoning system, according to different requirements, it can be precise evidence for reasoning, or be fuzzy evidence after fuzzifying; the rule matching is conducted, there are three kind of rule matching, namely, fuzzy rule matching, precise rule matching and hybrid rule matching; according to these three different rule matching methods, the precise reasoning can be conducted by precise evidence and rules, the fuzzy reasoning can be conducted by fuzzy evidence and rules, and the hybrid reasoning can be conducted by fuzzy or precise evidence with hybrid rules; the precise result can be got by precise reasoning, fuzzy result can be got by both fuzzy reasoning and hybrid reasoning; when the precise reasoning result is got, the result is judged if it is the final conclusion, the operation set ρ, if the result belong to ρ, then the result is output; otherwise, the precise reasoning result is added into precise evidence as middle fact, and meanwhile defuzzified precise reasoning result is added into fuzzy evidence for the further reasoning. In order to better illustrate the figure 4.4, the symbol 1, 2, 3 mean the fuzzy evidence got respectively by defuzzication, fuzzy reasoning and hybrid reasoning, while symbol 4, 5, 6 mean the precise evidence got respectively by precise reasoning, fuzzification of fuzzy reasoning and hybrid reasoning result.

The EM fuzzy reasoning model can satisfy most requirements of all rules and evidence, its input variable can be either precise or fuzzy evidence, and it can realize the inter-conversion of precise variable and fuzzy variable in the reasoning process, and the hybrid reasoning. The reasoning method in the EM fuzzy reasoning system can be more agile, ensuring that AO can utilize both precise and fuzzy rules. The construction of EM fuzzy reasoning system includes following procedures in figure 7-4:

1) choosing related input variables and output variables, the input variable in this paper is the instance variable of flexible workflow instance, and the output variable is the elements in the operation set ρ on flexible workflow;

2) choosing suitable membership function of σ, according to domain and type of a specific σ;

3) choosing and classifying suitable reasoning rules, and determining membership function of fuzzy rules;

4) choosing different reasoning algorithm, while main point is on fuzzy reasoning and hybrid reasoning algorithm, including goodness-of-match calculation, conflict resolution and searching strategy;

5) Choosing defuzzification algorithm according to fuzzy result, and getting precise operation set ρ on flexible workflow;

6) According to the input and output variables, adjusting membership function, weight and searching strategy to get optimized reasoning algorithm;

It has been testified that, by selecting suitable membership function, fuzzification and defuzzification algorithm, and fuzzy reasoning algorithm, the fuzzy reasoning system can reach a given non-linear function in any precision. The fuzzy reasoning system is another important non-linear mapping model besides artificial neural network, and its characteristic is the fully and efficient utilization of language and knowledge information.

7.2.2 Expression of Autonomic Object Knowledge

7.2.2.1 The composition of AO knowledge

The knowledge of AOs is the core of the EM fuzzy reasoning system of an AO developed in this research, and the design of fuzzy reasoning algorithm is highly related with the composition of AO knowledge. The composition and expression of AOs knowledge also affect the performance and efficiency, so the composition and expression of AOs knowledge is the key to realize the intelligent computation of AOs.

In the flexible workflow application in dynamic engineering design process, because the execution process of flexible workflow is the embodiment of dynamic engineering design process, the instance variable σ of flexible workflow is related to the domain in engineering design. The domain of AO knowledge is closely related to and represents the domain of engineering design. In the above EDP process, the domain of engineering design can be further divided into concept design, embodiment design, or into layout design, computation, drawing and son.

Because of diversity of engineering design domains, the AO knowledge should be classified or expressed suitably. The typical AO knowledge includes Solution Topology Knowledge, Policy Knowledge, and Problem Determination Knowledge. But in reasoning system, in the view of knowledge effect, the knowledge can be classified into Fact Knowledge, Process Knowledge and Control Knowledge. Meanwhile in considering that the EM fuzzy reasoning process is a precise and fuzzy hybrid reasoning process, the AO knowledge can also be classified into precise and fuzzy knowledge. Synthetically considering the knowledge diversity, definition of AO knowledge and the objective requirements of EM fuzzy reasoning system, the AO knowledge is composed of basic knowledge base, rule base, fact base, middle facts base, case base and control rule base. These different compositions of AO knowledge definitely can be classified into precise and fuzzy knowledge, as illustrated in figure 7-5.

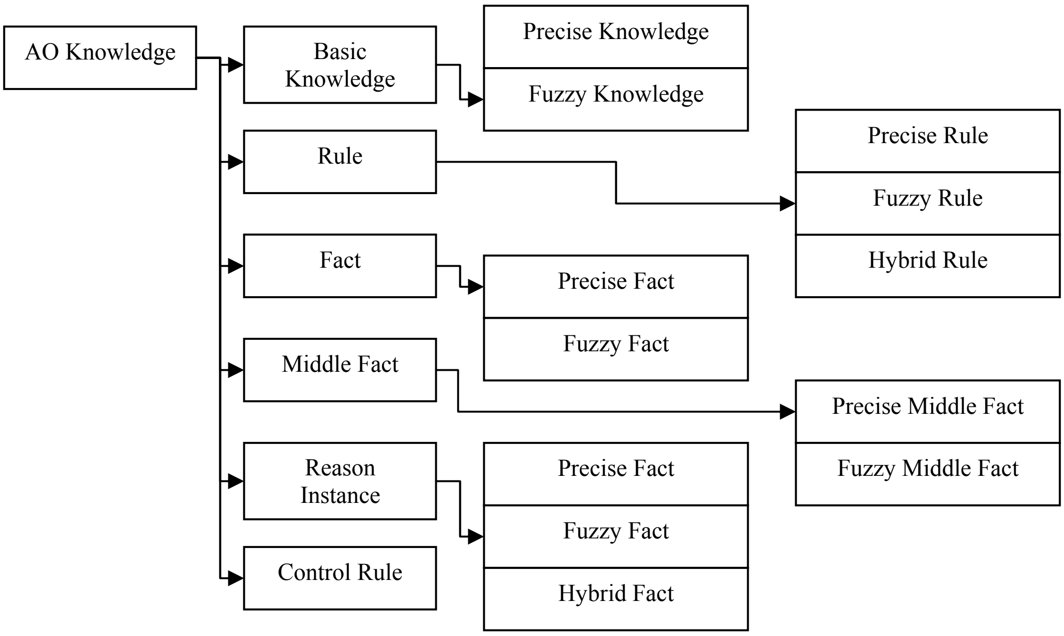


Figure 7-5 The composition of AO knowledge

The basic knowledge base of AO mainly includes different basic knowledge, common sense, property and status of a specific object in engineering design domain. Rule base mainly includes all process knowledge in AO, i.e., rules got from the comparison and analysis, the input evidence can be used to match these rules, and rule base is the core of AO knowledge. Fact base is used to store the middle result in the reasoning process as the foundation of further reasoning. Case base is mainly store the final successful case, while control rule mainly includes knowledge reasoning strategy, including Solving Strategies, searching strategy.

7.2.2.2 AO precise knowledge expression

Knowledge expression is a kind of description of knowledge, by concluding and summarizing the domain knowledge, then formalizing by suitable computer language and storage structure. Generally speaking, in choosing suitable knowledge expression method, the knowledge utilization, knowledge management, and the knowledge understand ability should be taken full consideration. Although the basic knowledge base and fact base belong to different logic category, they are similar in their expression methods because there are basic descriptions based on fact. In the composition of AO knowledge, rule base is the basis for logic reasoning and AO intelligence, case base is the reasoning result, and its expression method can be an extension of rule base. The knowledge expressions of AO fact and rule base are studied in this dissertation.

It can be learned from the figure7-5, the AO knowledge can be classified into precise knowledge and fuzzy knowledge, and their expression methods are quite different. Because predicate logic is very suitable to express the fact knowledge, such as status, property, concept and so on of an object, it is very suitable to express the precise basic knowledge of AO. When expressing the knowledge by predicate logic, the predicate should be defined at first, the exact definition of each predicate should be given, and related predicates are linked by conjunction to express integrated meaning, the general form of the predicate formula is given in formula 7-1:

$$P\left(x_1, x_2, \ldots x_i, \ldots, x_n\right) \qquad \textit{(formula7-1)}$$

Among them, P is the name of predicate, x_i is individual, and can be a constant, variable or function. By using of conjunction such as "$\neg, \vee, \wedge, \rightarrow$", the simple predicate logic can be linked to form complex predicate logic. For example, x is a design task, then $count\left(x \square 10\right)$ means the product number is 10. The predicate logic is used to express AO precise knowledge for its characteristics of Accuracy, tight and easy to implement.

According to the requirement of autonomic computing, the AO is required to support the uniform knowledge expression method, and then multi AOs in flexible workflow can share knowledge and do reasoning collaboratively. In this approach, a rule is the knowledge which can determine how the AO can change flexible elements in multi AOs flexible workflow. The rule is here used to define the knowledge expressing causal relationship. Considering the dynamic changing process of flexible workflow and the AO reasoning is often jointly determined by multi interrelated instance variables, the uncertain production rule based on weight factor is used in this dissertation to express the rule of AO knowledge, as illustrated in formula 7-2.

$$IF\ E_1\left(\omega_1\right)\ AND\ E_2\left(\omega_2\right)\ AND\ \ldots\ AND\ E_n\left(\omega_n\right)\ THEN\ H\left(CF\left(H, E\right), \lambda\right. \qquad \text{(formula7-2)}$$

Among them, E_i is production precondition expressed by the first order predicate, H is middle conclusion or operation on flexible workflow expressed by first order predicate, ω_i is the weight factor of E_i, ω_i can be higher value when its corresponding condition is more important for conclusion or have higher condition independence, $0 \le \omega_i \le 1$, and $\sum_{i=1}^{n} \omega_i = 1$; λ threshold value, the knowledge can be applied only the precondition $CF(E) \ge \lambda$ is satisfied, $0 \le \lambda \le 1$; for compositional condition, $CF(E) = \sum_{i=1}^{n} \omega_i \times CF(E_i)$, the credibility of conclusion can be got by $CF(H) = CF(H,E) \times CF(E)$, the production rule not considering uncertainty and weigh factor is given in formula 7-3:

$$IF\ E_1\ AND\ E_2\ AND\ ...\ AND\ E_n\ THEN\ H \qquad \text{(formula7-3)}$$

7.2.2.3 AO fuzzy knowledge expression

For AO fuzzy basic knowledge, its typical expression method is to describe a piece of knowledge by using fuzzy language value, in order to unify the AO knowledge representation. Combining the first order predicate, a dualistic first order predicate fuzzy knowledge expression is given in formula 7-4:

$$P(x, A), \text{typically, } x \quad is \quad A \qquad \text{(formula7-4)}$$

Among variables in the above expression, P is a predicate, meaning the concrete meaning of knowledge, x is the variable in domain defining the property of an object, $A = \int_{u \in U} u_A(u)/u$ is fuzzy concept or fuzzy value, which is expressed by the related fuzzy set or membership function.

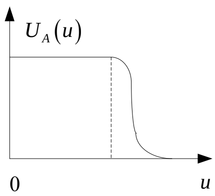
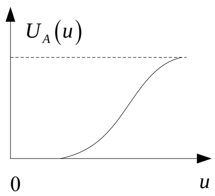
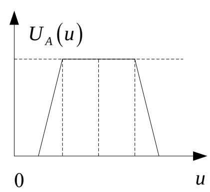
For an AO fuzzy rule, the relationship between fuzziness and uncertainty of the AO knowledge and instance variable should also been considered. A weight-based fuzzy production rule with credibility is used to express AO fuzzy rule as given in formula 7-5, $E_i : x_i\ is\ A_i\ CF_{i+1}$ is simple knowledge, x_i is the variable in domain U_i corresponding with instance variable in flexible workflow instance, A_i is fuzzy set in U_i, $H : y \ is \ B \ CF$ is conclusion, CF_i is the credibility of knowledge precondition, while CF is the credibility of knowledge conclusion, ω_i is weight.

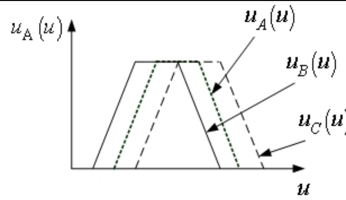
$$\begin{aligned}
&IF\ E_1(\omega_1)\ AND\ E_2(\omega_2)\ AND\ ...\ AND\ E_n(\omega_n)\ THEN\ H\ CF_1 \\
&\quad E_1 : x_1\ is\ A_1\ CF_2 \\
&\quad E_2 : x_2\ is\ A_2\ CF_2 \\
&\quad ... \\
&\quad E_n : x_n\ is\ A_n\ CF_{n+1} \\
&\qquad\qquad\qquad\qquad H : y\ is\ B\ CF
\end{aligned} \qquad \text{(formula7-5)}$$

For AO fuzzy knowledge, no matter it is the fuzzy basic knowledge or fuzzy rule, the key of the knowledge expression is the fuzziness degree of variable in different domain, i.e., its membership function. Membership function has close relation with the domain of knowledge, so the determination of membership function should relate to the concrete domain that means the determination of membership function should consider the concrete meaning of variable in engineering design process. How to define a suitable membership function is highly related with the reasoning quality. In EDP, it involves many domains such as mechanical, electronic, management, and so on, and it also can be divided into many stages such as conceptual design, embodiment design and so on. It causes the diversity of instance variable domain and its expression method.

In practical engineering application, the most important and used membership function is in the real number domain, the frequently-used membership function includes gauss membership function, sigmoid membership function, trapezoidal membership function [166] and so on. For the type of these membership functions can be classified into low fuzzy set, high fuzzy set and middle fuzzy set. The frequently-used membership function and their corresponding application in AO knowledge is given in table 7-2.

Table 7-2 AO typical knowledge and membership function

Type	Typical Membership Function			Typical Knowledge
Low Fuzzy Set	Reducing Half Normal Curve	Diagram		Example A: Design intensity is low; Supposing the low intensity is 4000mpa, K=2, $u_A(u)=e^{-2(u-1000)^2}, u>1000$.
	Formula	$u_A(u)=\begin{cases}1 & u \le a \\ e^{-k(u-a)^2} & u \succ a, k \succ 0\end{cases}$		
High Fuzzy Set	Increasing Half gauss Curve	Diagram		Example, A:Count of Bolt is larger than normal; Supposing that the normal bolt count for a part is 20, β=1, when $u>20$, $u_A(u)=\frac{20(u-20)}{1+20(u-20)}$
	Formula	$u_A(u)=\begin{cases}0 & u \le a \\ \frac{a(u-a)^{\beta}}{1+a(u-a)^{\beta}} & u \succ a, a \succ 0, \beta \succ 0\end{cases}$		
Middle Fuzzy Set	trapezoidal	Diagram		For example, A: the conclusion is that the activity “Intensity Verification” needed to be added, supposing that the possible scope is (4-6), impossible scope is (3-5), very possible scope is (5-7), the trapezoidal membership function can be established as follows:

The process of determining the AO knowledge membership function is given in figure 7-6. The approximate fuzzy distribution is searched to find the similar membership function according to the domain and exact means of AO knowledge. If the approximate fuzzy distribution can be found, the membership function is selected in the frequently-used membership functions; if not, the membership function is determined by data fitting. Then if the determined membership is satisfied, the membership function can be amended directly; if not, the determination of membership function is made through the expert judgment, and then be amended. If the amendment of membership function can reflect the fuzzy characteristics of AO knowledge, a series of fuzzy membership function can be generated by the extension of an acquired membership function. If not the membership function is assessed again. By the parameter adjustment or composition of the membership function, then by constantly amendment in practice, the membership function suitable to practice can be acquired.

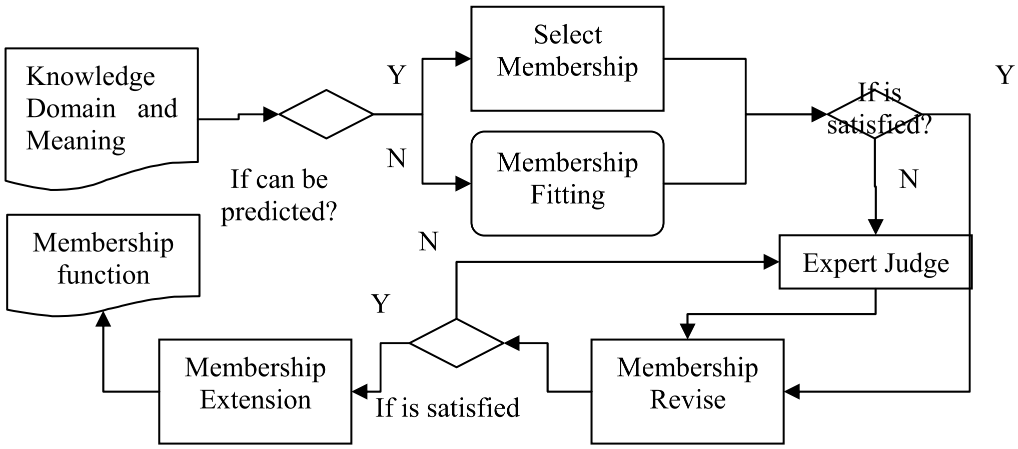
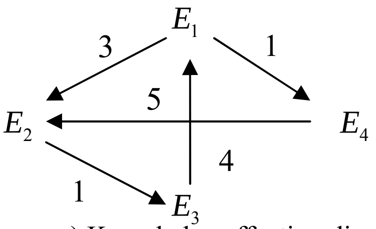


Figure 7-6 The determination process of AO fuzzy knowledge membership function

In an AO knowledge expression, regardless of the precise knowledge or fuzzy knowledge, one key question is to determine the weight ω of knowledge, and weight means the knowledge importance in compositional knowledge. In uncertain reasoning based on weight factor, it is a little bit hard to acquire weight. Analytic Hierarchy Process (AHP) is frequently –used method to acquire weight of knowledge, it main point is to compare the relative importance between the sub knowledge, but the independence of sub knowledge should be considered either. Sub knowledge should have higher weight if it has higher independence. A Decision Making Trial and Evaluation Laboratory (DEMATEL) is proposed to improve the AHP method. The DEMATEL method is introduced to compute the independence weight of AO knowledge, and then modify the weight got by AHP. Supposing the initial weight got by AHP is $\omega = (0.22, 0.11, 0.5, 0.17)$, then the weight computing method considering both the knowledge importance and independence illustrated in figure 7-7.



a) Knowledge affection directed graph

	1	2	3	4
1	0	0.6	0	0.2
2	0	0	0.2	0
3	0.8	0	0	0
4	0	1	0	0

b) Standardized relation matrix

	1	2	3	4
1	0.14679	0.91743	0.18349	0.22936
2	0.18349	0.14679	0.22936	0.036697
3	0.91743	0.73394	0.14679	0.18349
4	0.18349	1.1468	0.22936	0.036697

c) Compositional effect matrix

	1	2	3	4
1	0.3282	0.4719	0.1076	0.0923

d) Independence weight

	1	2	3	4
1	0.2633	0.2548	0.343	0.1389

e) Compositional weight

Figure 7-7 Example of AO knowledge weight computation

1) the effect directed graph is drew by Satty's 1 to 9 scale, then the relation matrix is got by this directed graph;

2) the standardized relation matrix $X = Z \bullet 1/\max\limits_{1 \le i \le n}\left(\sum_{j=1}^{n} z_{ij}\right)$ and compositional effect matrix $T = X\left(I - X\right)^{-1}$ is computed;

3) the independence weight $\omega t = \sum_{j=1}^{n} x_{ij}\omega j / \sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij}\omega j$ is computed, then the ωt is standardized, then the compositional weight $\omega a = \alpha\omega + \left(1-\alpha\right)\omega t$, then the compositional weight is standardized, α is 0.6 meaning considering independence of AO knowledge more.

7.2.3 Fuzzy Reasoning Algorithm

7.2.3.1 Fuzzification and Defuzzification algorithm

In EM fuzzy reasoning model, the matching process between input instance variable or middle fact and AO knowledge is the key of the fuzzy reasoning. Because the EM fuzzy reasoning is a hybrid reasoning process, the matching between precise fact and fuzzy knowledge, as well as fuzzy fact and precise knowledge should be solved, and these are realized respectively by fuzzification and defuzzification algorithms described below.

The effect of fuzzification is that the precise variable in real number is transferred into fuzzy variable with the form of membership function, which can be processed by fuzzy reasoning system. In hybrid reasoning system, the precondition of reasoning is that the precise and fuzzy variable should be in the same design process domain. The frequently-used fuzzification algorithm includes single-value fuzzification, gauss fuzzification and triangular fuzzification, the gauss fuzzification is mainly used in this thesis. The single-value fuzzification and gauss fuzzification is shown in formula 7-6 and 7-7.

Supposing x is the precise variable in domain U, A is fuzzy set in U, x is mapped into a fuzzy set A, the value of A membership function is 1 when x is equal to $x^{'}$, while the value is 0 when x is not equal to $x^{'}$:

$$u_A(x)=\begin{cases}1 & x=x^{'}\\ 0 & x\neq x^{'}\end{cases},\ x^{'} \text{ is inputted Precise Value;} \qquad \text{(formula7-6)}$$

The x is mapped into the fuzzy set A with gauss distribution by gauss fuzzification:

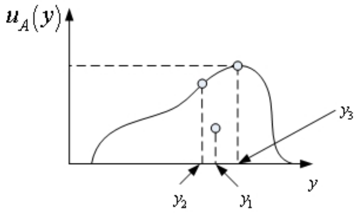
$$u_A(x)=e^{-\left(\frac{x-x^{'}}{a}\right)^2} \qquad \text{(formula7-7)}$$

For AO knowledge with any form of membership function, the fuzzy reasoning can be simplified by single-value fuzzification. And if the form of membership functions is other fuzzification such as gauss or triangular fuzzification, the AO knowledge is always pre-treated by the same membership to simplify the computing of the fuzzy reasoning. Besides all above fuzzification algorithm including single-value fuzzification, the precise variable can also be fuzzificated by discrete fuzzification in this research. Actually in discrete fuzzification, the each discrete variable is also determined by related membership function. Relatively speaking, the amount of computation is small in contrast with the large amount of computation of continuous fuzzification. Considering the example in table 7-2, Count of Bolt is larger than normal; supposing the input bolt count is 30, then $x^{'}=30$. The membership function is therefore a single-valued fuzzification: $u_A(x)=\begin{cases}1 & x=30\\ 0 & x\neq 30\end{cases}$, choosing 10, 20, 30, 40 as disperse sample, then the disperse membership is: $\{0/10, 0/20, 1/30, 0/40\}$.

The defuzzification algorithm is used to determine the precise value in a fuzzy set, the algorithm implemented in this research includes COA (center of area), ECOA (extended center of area) and MOM (middle of maximum) as illustrated in table 5-3. The different defuzzification method can be selected according to the specific applications. The precise value by defuzzification can be transformed further into new precise value according to the practical application.

Table 7-3 The defuzzification algorithm

Name	Formula	Explanation
Area Centre	$y_1=\frac{\int_y u_A(y)\,y\,dy}{\int_y u_A(y)\,dy}$	$u_A(y)$ is the membership function of output fuzzy set B, Area Centre method is to compute the precise value of fuzzy set area centre in domain B.
Area equal portion	$\int_{\alpha}^{y_2} u_A(y)\,dy=\int_{y_2}^{\beta} u_A(y)\,dy$	An area equal portion is to compute the precise value of area equal portions in domain B.
Great averaging	$y_3=\{y \mid u_A(y)=\max(u_A(y))\}$	Great averaging is to compute the precise value in domain B when $u_A(y)$ is max.

Example	

7.2.3.2 Knowledge matching algorithm

For precise knowledge matching, it is required to evaluate if the input instance variable is equal to the precondition of knowledge, and the conclusion can be drawn if the precondition is satisfied. For fuzzy knowledge matching, because the fuzzy set $A^{'}$ of input instance variable may not be totally same with the fuzzy set A of knowledge precondition, the fuzzy knowledge matching can be computed by closeness, semantic distance and similarity. In these cases, the Hamming distance is adopted to compute the matching degree by computing the semantic distance of fuzzy set. The Hamming distance of both discrete and continuous domains can be expressed respectively by formula 7-8 and 7-9:

$$d\left(A,B\right)=\frac{1}{n}\times\sum_{i=1}^{n}\left|u_A\left(u_i\right)-u_{A^{'}}\left(u_i\right)\right| \quad \text{(formula7-8)}$$

$$d\left(A,B\right)=\frac{1}{b-a}\int_{a}^{b}\left|u_A\left(u\right)-u_{A^{'}}\left(u\right)\right|du\,, \quad \text{(formula7-9)}$$

The domain of A and $A^{'}$ is a close interval[a, b] in real number field

Regardless of discrete or continuous domain, the matching degree of fuzzy set can be defined by formula 7-10:

$$\delta_{match}\left(A,B\right)=1-d\left(A,B\right) \quad \text{(formula7-10)}$$

If the matching degree is larger than the threshold of knowledge, the input variable can be said to match with the precondition of knowledge. It is notice to see that the matching degree is quite different by different matching method. The selection of threshold needs to reflect the matching method, because in EM fuzzy reasoning system, the matching result can be different according to the selected different matching method.

7.2.3.3 Fuzzy reasoning algorithm

If the input variable can be matching with the precondition of AO knowledge, then the fuzzy reasoning process or the problem solving process, of AO can be started. The forward reasoning is a reasoning process started from known fact, the basic idea of forward reasoning is that, starting from the initial fact, finding the suitable knowledge in knowledge base, constructing applicable knowledge set, then choosing one knowledge in knowledge set according to a kind of conflict resolution method, then adding the new fact got from the reasoning process as the known fact for further reasoning. This process is repeated until the solution is found or no more knowledge can be suitable for reasoning in the knowledge base. It can be known from the intelligence principle of flexible workflow in 7.1.2, the reasoning process of AO is to find the suitable operation in knowledge based according to the instance variable and AO knowledge, its reasoning process mainly is a forward reasoning process.

Although the forward reasoning process is relatively simple, and there are already many mature algorithm for fuzzy reasoning, ambiguity and randomness is the two main uncertain in the real world. For the flexible workflow application in dynamic engineering process, there are ambiguity and randomness in knowledge expression, and the relevance of compositional knowledge should also been considered for the close relevance in dynamic engineering process. The reasoning process should solve the weight-based compositional reasoning with credibility. Although there already many research on ambiguity and randomness, there is few research on weight-based fuzzy reasoning. For present weight-based fuzzy reasoning researches, most of them pay more attention on the condition that the fuzzy evidence is totally matched with the precondition of fuzzy knowledge, but for the condition that the fuzzy evidence is only similar with the precondition of fuzzy knowledge, how to find out the membership function of fuzzy conclusion is still needed to be studied. The weight-based fuzzy reasoning algorithm with creditability factor is given in this dissertation, the precise and fuzzy hybrid reasoning algorithm is given further.

In order to propose the weighted fuzzy reasoning algorithm with CF, the concept of simple fuzzy reasoning and multi dimensional fuzzy reasoning is introduced at first. The simple fuzzy reasoning is the reasoning where only simple condition without CF in the knowledge, as shown in formula

7-11.

Knowledge: $IF\ x\ is\ A\ THEN\ y\ is\ B$,

Evidence: $x\ is\ A^{'}$,and A and $A^{'}$ can be matched, then （formula7-11）

$B^{'} = A^{'} \circ R$

Simple fuzzy reasoning is the basis of further reasoning method, its key is to construct the fuzzy relation R. Several methods to construct R had been proposed by Zadeh and others [166], for example, R_m and R_a can be got by maximum and minimum rule and condition proposition arithmetic rule respectively, R_c can be got by minimum rule of condition proposition proposed by Mamdani, R_s, R_g, R_{sg} and R_{ss} can be got by multi value logic proposed by Mizumoto. It can be concluded that R_s, R_{sg} and R_{ss} are all fuzzy logic relation with good performance. These three relations is used to implement fuzzy reasoning in this dissertation as shown in formula 7-12 to 7-14.

$$R_s = \int_{U\times V} [u_A(u) \underset{s}{\rightarrow} u_B(u)] / (u,v) \quad ,$$

$$u_A(u) \underset{s}{\rightarrow} u_B(v) = \begin{cases} 1 & u_A(u) \le u_B(v) \\ 0 & u_A(u) > u_B(v) \end{cases}$$

（formula7-12）

$$R_{sg} = \left(A\times V \underset{s}{\Rightarrow} U \times B\right) \cap \left(\neg A\times V \underset{g}{\Rightarrow} U \times \neg B\right)$$
$$= \int_{U\times V} \left\{\left[u_A(u) \underset{s}{\rightarrow} u_B(v)\right] \wedge \left[\left(1-u_A(u)\right) \underset{g}{\rightarrow} \left(1-u_B(v)\right)\right]\right\} / (u,v)$$

（formula7-13）

$$R_{ss} = \left(A\times V \underset{s}{\Rightarrow} U \times B\right) \cap \left(\neg A\times V \underset{s}{\Rightarrow} U \times \neg B\right)$$
$$= \int_{U\times V} \left\{\left[u_A(u) \underset{s}{\rightarrow} u_B(v)\right] \wedge \left[\left(1-u_A(u)\right) \rightarrow \left(1-u_B(v)\right)\right]\right\} / (u,v)$$

（formula7-14）

Multi-dimensional reasoning is the reasoning where precondition of knowledge is compositional condition, and its general form is as formula 7-15. The conclusion $B^{'}$ can be got by Zadeh method as shown in formula 7-16.

Knowledge: （formula7-15）

$IF\ x_1\ is\ A_1\ AND\ x_2\ is\ A_2\ AND...AND\ x_n\ is\ A_n\ THEN\ y\ is\ B$

Evidence: $x_1\ is\ A_1^{'}, x_2\ is\ A_2^{'}, \ldots, x_n\ is\ A_n^{'}$

Conclusion: $y\ is\ B^{'}$

1) computing the intersection of knowledge precondition A,

$$A = A_1 \cap A_2 \cap \ldots \cap A_n$$
$$= \int_{U_1 \times U_2 \times \ldots \times U_n} u_{A_1}(u_1) \wedge u_{A_2}(u_2) \wedge \ldots \wedge u_{A_n}(u_n) / (u_1, u_2, \ldots, u_n)$$

(formula7-16)

2) computing the fuzzy relation $R(A,B)$ between A and B, and computing the intersection of evidence $A^{'}$

3) computing $B^{'} = A^{'} \circ R(A,B)$

The multi dimensional fuzzy reasoning solving is given in formula, the weighted fuzzy reasoning with CF will be studied base on formula. How to compute the membership function of conclusion $B^{'}$ according the knowledge and evidence and how to compute the CF of conclusion $B^{'}$ should be considered at first.

For the computation of membership function of conclusion $B^{'}$, it is similar with the precise reasoning with CF, the conclusion of reasoning is unrelated with CF, only the CF of final conclusion will be effected by the CF, the membership function is only related with the membership function and weight of knowledge and evidence. The introduction of weight actually introduces the idea of weighted average into the knowledge reasoning. According to formula 7-16, the weighted multi dimensional fuzzy reasoning is shown in following:

1) computing the intersection A of knowledge precondition;

Because of the introduction of weight, the fuzzy sets intersection of knowledge precondition is weighted averaged, the primary form of knowledge precondition intersection is $\omega_1 \times A_1 \cap \omega_2 \times A_2 \cap \ldots \cap \omega_n \times A_n$, this form is got by timing a coefficient ω_i with fuzzy set of each knowledge precondition; considering that the intersection operation of fuzzy set actually is the minimum of membership function, the intersection of knowledge precondition should be divided by average of weight $\sum_{i=1}^{n}\omega_i / n$, the intersection of knowledge precondition in weighted fuzzy reasoning is shown in formula 7-17:

$$A=\frac{n}{\sum_{i=1}^{n}\omega_i}\times\left(\omega_1\times A_1\cap\omega_2\times A_2\cap\ldots\cap\omega_n\times A_n\right),\text{when}\sum_{i=1}^{n}\omega_i=1,$$

（formula7-17）

$$A=n\times\left(\omega_1\times A_1\cap\omega_2\times A_2\cap\ldots\cap\omega_n\times A_n\right)$$
$$=\mathrm{n}\times\int_{U_1\times U_2\times\ldots\times U_n}\left(\omega_1\times u_{A_1}\left(u_1\right)\right)\wedge\ldots\wedge\left(\omega_{\mathrm{n}}\times u_{A_n}\left(u_n\right)\right)/\left(u_1,u_2,\ldots,u_n\right)$$

Considering the two dimensional fuzzy reasoning (without weight), it is actually a weighted fuzzy reasoning with same weight, i.e., $\omega_1=0.5$, $\omega_2=0.5$, $n=2$, according to the formula 7-17, $A=\int_{U_1\times U_2}u_{A_1}\left(u_1\right)\wedge u_{A_2}\left(u_2\right)/\left(u_1,u_2\right)$, so the 1) in formula 7-16 special case of formula 7-17 without considering the weight.

2) Constructing the fuzzy relation $R\left(A,B\right)$ between A and B according to R_s, R_{sg}, R_{ss};

3) Computing the intersection of evidence $A^{'}$ by formula 7-17;

4) computing $B^{'}$, $B^{'}=A^{'}\circ R\left(A,B\right)$;

The above computing procedure of conclusion $B^{'}$ is mainly suitable for the situation that precondition of knowledge A_i is not equal to the precondition of evidence $A_i^{'}$, when $A_i=A_i^{'}$, $B^{'}=B$, only the CF of conclusion need to be considered.

For the CF of conclusion $B^{'}$, the CF computation in weighted fuzzy reasoning with CF can be got based on the CF computation of multi dimensional fuzzy reasoning and CF computation of weighted uncertain reasoning as illustrated respectively in formula 7-18 and 7-19.

$$CF\left(E\right)=\sum_{i=1}^{n}\omega_i\times CF\left(E_i\right)$$ （formula7-18）

$$CF=\left(CF_1\wedge CF_2\wedge\ldots\wedge CF_n\right)$$ （formula7-19）

Because the weighted fuzzy reasoning with CF is based on the fuzzy computation, it should be constructed based on formula 7-19 as follows:

1) computing the general CF of the evidence

Considering the introduction of weight, the CF of evidence is weighted averaged, $\omega_1 CF_1 \wedge \omega_2 CF_2 \wedge \ldots \wedge \omega_n CF_n$, meanwhile considering the effects of weight, the weighted fuzzy CF is given in formula 7-20:

$$CF = \frac{n}{\sum_{i=1}^{n}\omega_i} \times \left(\omega_1 \times CF_1 \wedge \omega_2 \times CF_2 \wedge \ldots \wedge \omega_n \times CF_n\right) \text{,when } \sum_{i=1}^{n}\omega_i = 1,$$

（formula7-20）

$$CF = n \times \left(\omega_1 \times CF_1 \wedge \omega_2 \times CF_2 \wedge \ldots \wedge \omega_n \times CF_n\right)$$

2) computing the general matching degree of weighted fuzzy rules

Similar with the computing of general CF, the general matching degree of weighted fuzzy rules is given in formula 7-21:

$$\delta_{match} = \frac{n}{\sum_{i=1}^{n}\omega_i} \times min\left\{\left(\omega_1 \times \delta_{match}\left(A_1, A_1^{'}\right)\right), \ldots, \left(\omega_n \times \delta_{match}\left(A_n, A_n^{'}\right)\right)\right\} \text{,when}$$

$$\sum_{i=1}^{n}\omega_i = 1$$

, （formula7-21）

$$\delta_{match} = n \times min\left\{\left(\omega_1 \times \delta_{match}\left(A_1, A_1^{'}\right)\right), \ldots, \left(\omega_n \times \delta_{match}\left(A_n, A_n^{'}\right)\right)\right\}$$

3) similar with simple fuzzy reasoning, computing the CF of conclusion according to the general CF and general matching degree

The matching degree of evidence and knowledge precondition should be considered in computing the CF of conclusion, when the evidence and knowledge precondition is completely matching, i.e., $A_i = A_i^{'}$, the matching degree of evidence and knowledge needn't to be considered, supposing CF_A is the general CF of evidences by formula 7-20, CF_B is the CF of knowledge, the CF of conclusion is given in formula 7-22:

3) the credibility of conclusion can be got by general credibility and general matching degree according similar method of computing simple fuzzy reasoning's.

In computing the credibility of conclusion, the matching degree of evidence and knowledge precondition should be considered, when $A_i = A_i^{'}$, the matching degree can be neglected, supposing the CF_A is the general matching degree by formula7-20, CF_B is the credibility of conclusion, then the credibility can be got by formula7-22:

$$CF = min\{CF_A, CF_B\} \qquad \text{(formula7-22)}$$

When $A_i \neq A_i^{'}$, the matching degree of evidence and knowledge should be considered, the CF of conclusion is given in formula 7-23:

$$CF = \delta_{match} \times min\{CF_A, CF_B\} \qquad \text{(formula7-23)}$$

According to the weighted fuzzy reasoning with CF, the introduction of CF ensures that the knowledge can be applied only if the CF of evidence is larger than threshold without the completely matching. For the fuzzy knowledge in formula 7-5, when one of the evidence $A_i^{'}$ can be acquired, if the compositional CF of evidence $CF > \lambda$, the knowledge can be applied. Unable obtained evidence doesn't affect the application of knowledge.

In the practical application of EDP management, not only the simple precise or fuzzy reasoning is existed, but also hybrid reasoning of precise and fuzzy reasoning, such as "when the design task is hard and design cycle time is 12 days", there are few researches on such hybrid knowledge task, the compositional reasoning algorithm is given as follows:

1) for fuzzy knowledge, the matching degree of fuzzy knowledge is computed at first, for precise knowledge, computing if the evidence is equal to the knowledge, if there are equal, $\delta_{match} = 1$, otherwise $\delta_{match} = 0$;

2) for the precise knowledge in compositional knowledge, it is fuzzified by a specific membership function, the evidence is fuzzified in same way;

3) the compositional knowledge intersection A and $A^{'}$ of knowledge and evidence is computed respectively, then conclusion $B^{'} = A^{'} \circ R(A, B)$, the precise output of $B^{'}$ can be got by defuzzficated if needed;

4) the CF of evidence CF_A and the general matching degree of knowledge δ_{match} can be got respectively by formula7-20 and 7-21, $B^{'}$ can be got by formula 7-22;

The fuzzy reasoning process of AO is given in figure 7-8.

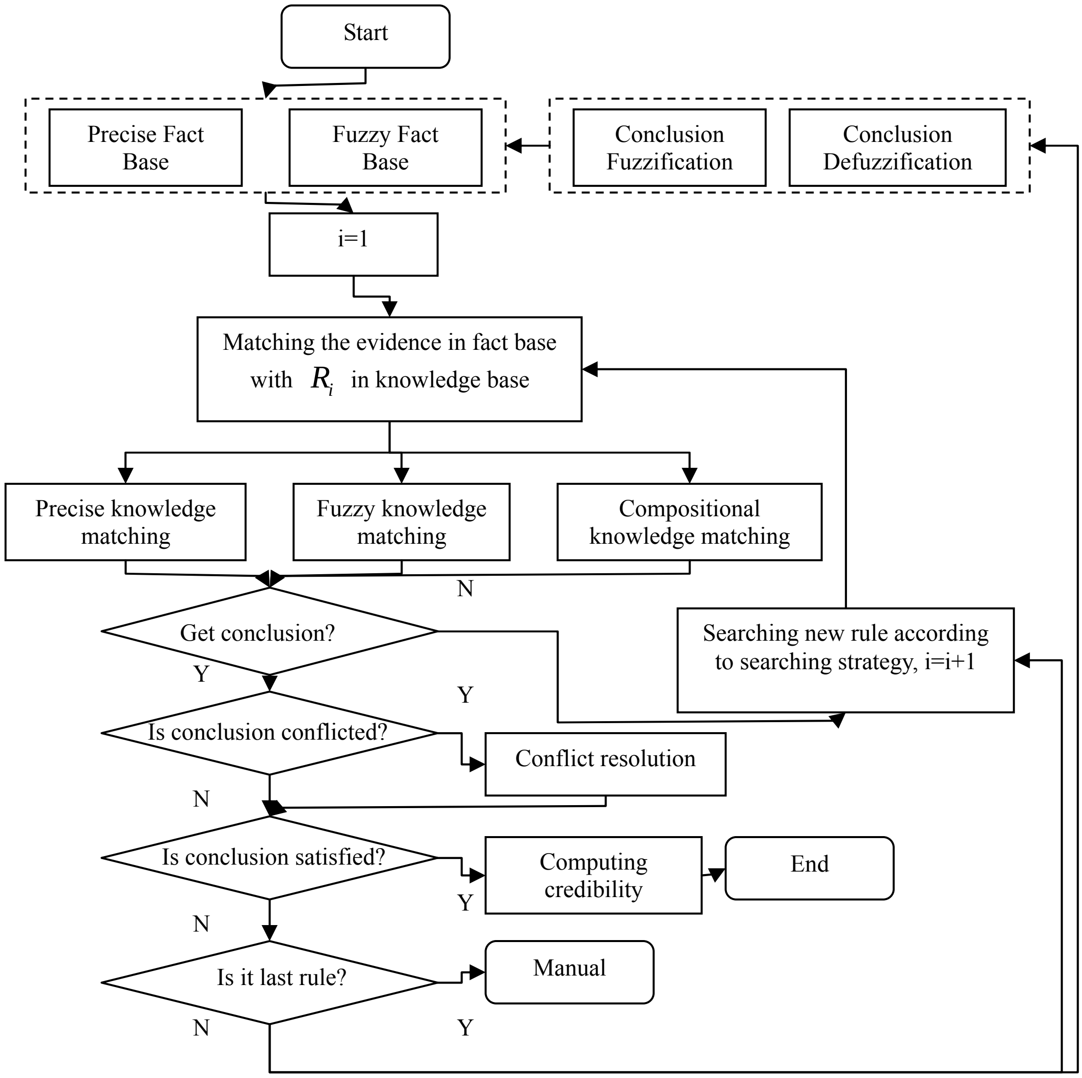


Figure 7-8 AO extended Mamdani fuzzy reasoning

7.2.3.4 Searching strategy and conflict resolution during reasoning

In intelligent reasoning, searching strategy determines how AO searches the usable knowledge, the performance and efficiency of reasoning system is affected directly by the searching strategy. Because the AO reasoning process is the process in which AO searches suitable operation on flexible workflow instance according to the instance variables, in this process, the derived conclusion is added into fact base as new piece of knowledge until the final operation is completed. Considering the completeness and efficiency of reasoning, the stimulating breadth-first searching strategy is adopted in AO reasoning, as shown in figure 7-9. AO select first rule according to the stimulating knowledge to form the root S_0 of the reasoning process; then starting from S_0 as intermediate fact, which is decomposed into second layer nodes, and exploring if there are target node in second layer, the node in second layer is not extended before the searching of these nodes is finished. Similarly after searching the nodes in i^{th} layer completely, a node in i_{th} layer is decomposed into $(i+1)^{th}$ layer, then it should be decided if the node should be decomposed further. In the searching process in the same layer, the stimulating knowledge is still used to determine if it needs to search nodes until the target node is found, as shown by node a_{34} in figure 7-9.

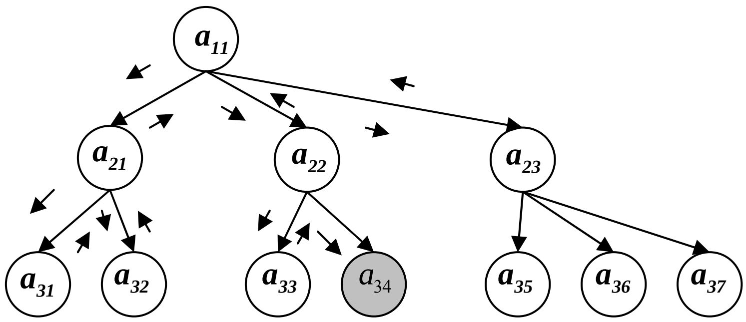


Figure 7-9 AO searching strategy

In the AO reasoning process, another important potential problem is conflicts among nodes and how the conflict resolution can be addressed. To resolve conflicts, a conflict resolution algorithm has been used in this thesis by sorting the knowledge. The conflict resolution strategy for AO reasoning is: 1) for precise reasoning, the evidence, i.e., the input instance variables is sorted by importance, that means the evidence has higher priority if it is main instance variable; 2) based on above, the knowledge pertinence is used to sort, the higher the pertinence, the higher the priority; 3) the freshness of knowledge is used afterwards if they have same importance and pertinence. As to the fuzzy reasoning, 1) the fuzzy matching degree of conflicted conclusions should be considered, if the conflict conclusions are in same domain and their matching degree is larger than or equal to certain threshold, the conflict conclusion combination is used, and the membership function and CF of combined conclusion is computed; 2) for the condition where the conflict conclusion is not in the same domain or their matching degree cannot satisfy the threshold, then the conflict resolution is similar with the method used in precise conflict resolution, the importance is used at first, the matching degree of fuzzy knowledge is used to sort.

7.3 Evaluation of Reasoning of Autonomic Object

7.3.1 A case study of AO based Reasoning

Based on the above extended Mamdani fuzzy algorithm for AO reasoning, a case study of EDP management in a medium-sized customised truck company (The Customised Trucks) has been undertaken. The company is a typical Make-to-Order (MTO) company, which manufactures products designed for a specific customer. The design process and the products are very complex while the batch size of the product is low. The design process is highly dynamic because it changes each time when there is a new order from the customer. The main products for the company include the different kinds of customised trucks, such as the heavy-duty dump trucks and flat platform trailers. Nowadays, the market competition becomes fiercer, so an efficient EDP is very important for the company to reduce the design cycle time and improve the quality to win the market competition. But the design processes for different trucks are different, so the company typically takes long time to reconstruct and manage its design processes. The EDP of a customized truck is given in figure 7-10, where (a) is the typical and traditional approach and process, (b) is the modified process determined by AO computation.

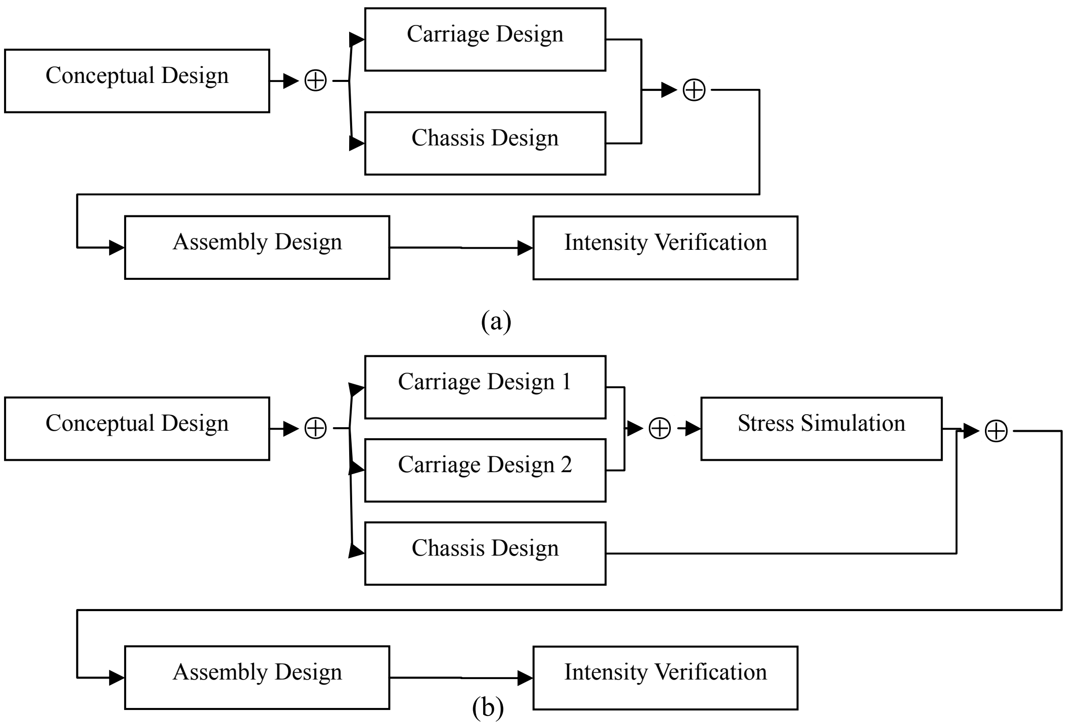


Figure 7-10 EDP in a customized truck company

The EDP in the figure 7-10 is part of the EDP of a flat platform trailer, the EDP is started from "conceptual design"; then is followed mainly by "carriage design" and "chassis design" in parallel, then by "assembly design" and "intensity verification". The "carriage design" is normally designed by one designer, and the material must be selected in "carriage design". Because the chassis of the flat platform trailer is mainly designed according to the original car, the design task of chassis is simpler than the carriage design. The AO of "conceptual design" can determined the subsequent activity in EDP, and the AO of "carriage design" will select the suitable material according to the design task.

By using the MATLAB fuzzy tool box of MathWork Company, the fuzzy reasoning system can be designed, established and tested. For the normal fuzzy reasoning, the reasoning instance is computed by the graphical fuzzy reasoning engine in MATLAB; but for the weighted fuzzy reasoning, it is computed by programs in appendix 1 programmed by MATLAB's basic computing functions, because the detail of min function and so on is not opened by MATLAB. In the prototype system, the AO computation is realized by calling of MATLAB program by C#.

7.3.2 Fuzzy reasoning of "conceptual design" AO

In order to support this particular engineering design process, a number of fuzzy rules have been generated and some examples are shown in table 7-4. These rules can be input into the system developed in this research through a graphical user interface developed based on MATLAB fuzzy toolbox and the rules in the system can be viewed using the user interface shown in Figure 7-11.

Table 7-4 Fuzzy reasoning rules of conceptual design AO

inputs	Design Difficulty D, including 3 membership functions, the type of membership function is triangle membership DMF1: the design is difficult, DMF1=trimf(X, [0.6, 0.8, 1.0]); DMF2: the design is normal, DMF2=trimf(X, [0.3, 0.5, 0.7]); DMF3: the design is easy, DMF3=trimf(X, [0.0, 0.2, 0.4]);
	Design Experience E, including 3 membership function, the type of membership function is gauss membership

	EMF1:the design experience is rich, EMF1=gaussmf(XB1,[0.2,0.6]); EMF2: the design experience is normal, EMF2=gaussmf(XB1,[0.2, 0.3]); EMF3: the design experience is poor, EMF3= FMF3=gaussmf(XB1,[0.2, 0.1]);
outputs	Design Cycle Time, including 3 membership functions, the type of membership function is triangle membership TMF1:design cycle time is long, TMF1= trimf(X,[10,16,22]); TMF2:design cycle time is slightly long, TMF2=trimf(X,[12,16,20]); TMF3: design cycle time is normal, TMF3=trimf(X,[5,8,11]); TMF4: design cycle time is slightly short TMF4=trimff(X,[-4,0,4]); TMF5: design cycle time is short TMF5=trimf(X,[-6,0,6]);
rules	Rule 1:If design is difficult AND design experience is poor Then design cycle time is long; Rule 2: If design is difficult AND design experience is normal Then design cycle time is slightly long; Rule 3: If design difficulty is difficult AND design experience is rich Then design cycle time is normal; Rule 4: If design difficulty is normal AND design experience is poor Then design cycle time is slightly long; Rule 5: If design difficulty is normal AND design experience is normal Then design cycle time is normal; Rule 6: If design difficulty is normal AND design experience is rich Then design cycle time is slightly short; Rule 7: If design difficulty is easy AND design experience is poor Then design cycle time is normal; Rule 8: If design difficulty is easy AND design experience is normal Then design cycle time is slightly short; Rule 9: If design difficulty is easy AND design experience is rich Then design cycle time is short;

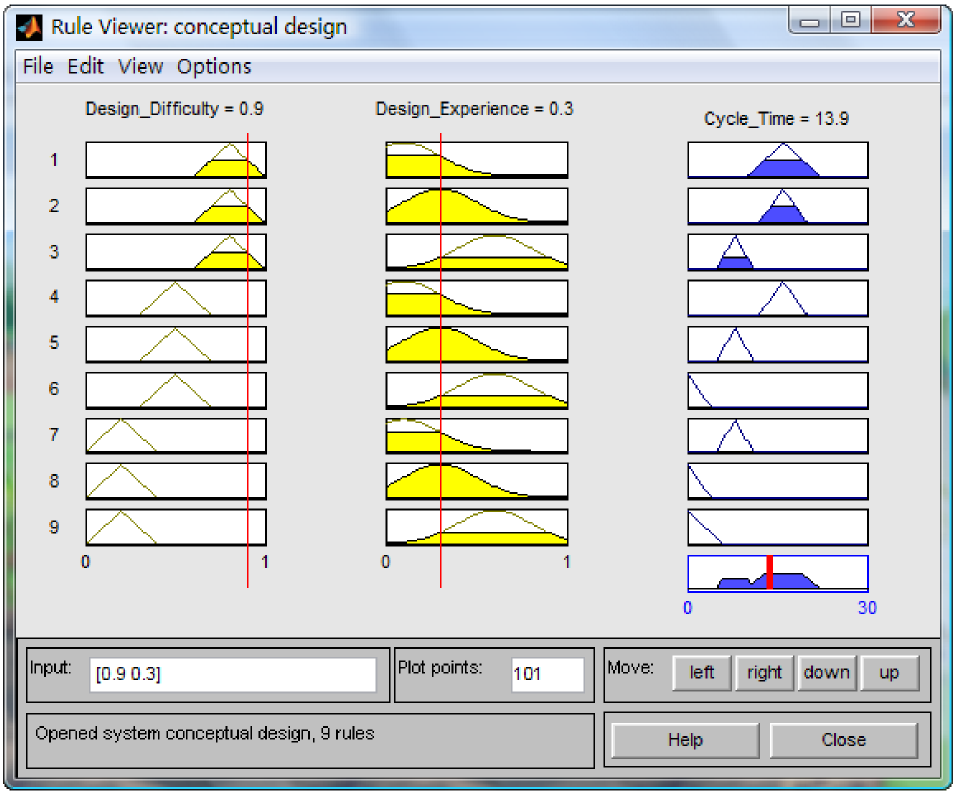


Figure 7-11 fuzzy reasoning result of conceptual design AO

The figure 7-11 shows the different reasoning results according to above 9 conceptual design AO reasoning rules in table 7-4, the Cycle_Time fuzzy outputs can be greatly different by these 9 rules in same Design_Difficulty and Design_Experience. Finally considering all these 9 rules the Cycle_Time can be 13.9 hours when the Design_Difficulty is 0.9 and Design_Experience is 0.3.

The fuzzy reasoning results of conceptual design AO are generated during the reasoning process and the distribution result of fuzzy reasoning can be inspected by a user as shown in an example in figure 7-12.

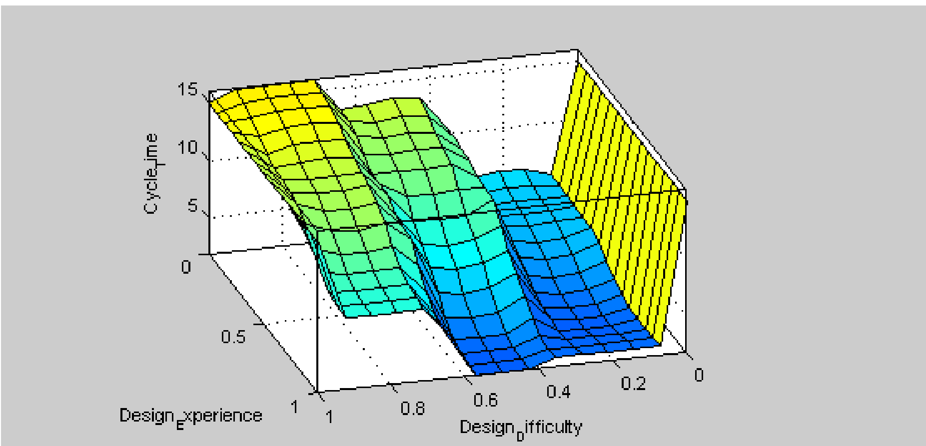


Figure 7-12 Distribution result of fuzzy reasoning

By searching the reasoning results in figure 7-12, the conceptual design AO can find out what is the Cycle_Time by different input of Design_Difficulty and Design_Experience.

7.3.3 Fuzzy reasoning of “carriage design” AO

The weighted fuzzy reasoning input, output and rules of carriage design AO is given in table 7-5. When the weight of the carriage is 3554 kg, the stress requirement is 560 Mpa. The weighted fuzzy reasoning in the discrete points of 50 can be evaluated by a MATLAB program function developed for the research.

Table 7-5 fuzzy reasoning rules of carriage design AO

input	Weight W, including three membership functions, the type of membership function is triangle membership WMF1:the weight is heavyWMF1=trimf(X,[0.6,1,1.4]), x is input; WMF2:the weight is normal OMF2=trimf(X,[0.2,0.5,0.8]); OMF3:the weight is light OMF3=trimf(X,[-0.4,0.0.4]);
	Intensity I, including three membership functions, the type of membership function is gauss membership IMF1: the intensity is highIMF1=gaussmf(XB1,[0.5,1]); IMF2: the intensity is low IMF2=gaussmf(XB1,[0.5,0]);
Output	strength-weight ratio, including two membership functions, the type of membership function is triangle membership function RMF1: ratio is high RMF1= trimf(X,[7,10,13]); RMF2: ratio is normal RMF2=trimf(X,[2,5,8]); RMF3: ratio is low RMF2=trimff(X,[-3,0,3]);
Rules	Rule 1: if the weight is heavy (0.4) AND intensity is low Then strength-weight ratio is low; Rule 2: if the weight is light (0.3) AND intensity is high Then strength-weight ratio is high; Rule 3:if intensity is normal Then strength-weight ratio is normal;

The weighted membership function is given in figure 7-13, and the reasoning result is given in figure 7-14. The weighted fuzzy reasoning program is designed according to the formula 7-16 and 7-17, the fuzzification is realized according to respective input membership function, the weighting of sub condition is realized by formula 7-17 and min function in MATLAB, and the fuzzy relation is realized by min function in MATLAB, while the synthesis of multi-condition is realized by max function and defuzzification is realized by centroid method. The complete source code can be refered to the Appendix 1.

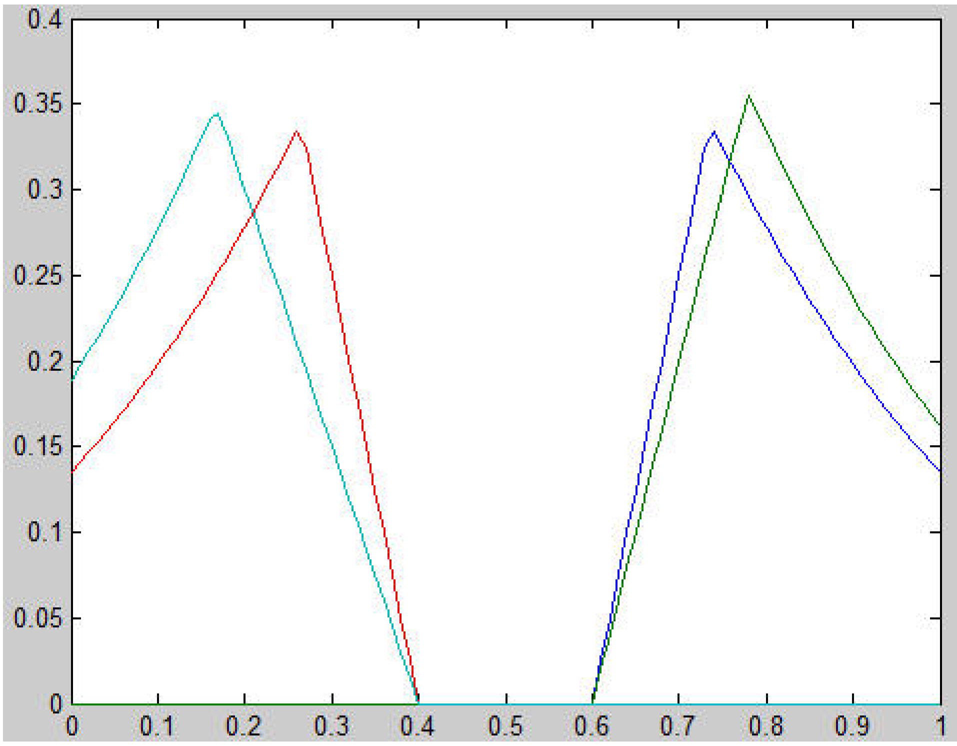


Figure. 7-13 the membership function of carriage design

In figure 7-13, it gives the comparisons between ordinary membership function and weighted membership function, and it can see very clearly the effect of weight.

By searching the reasoning results in figure 7-14, the “carriage design” AO can find out what is the strength-weight ratio quickly. Actually there are many mutual method to do the accurate computation in EDP, but for the dynamic EDP, the design context are always in change, and it hard to acquire accurate design parameters, fuzzy reasoning can provides usefully method to realize the intelligence of engineering design.

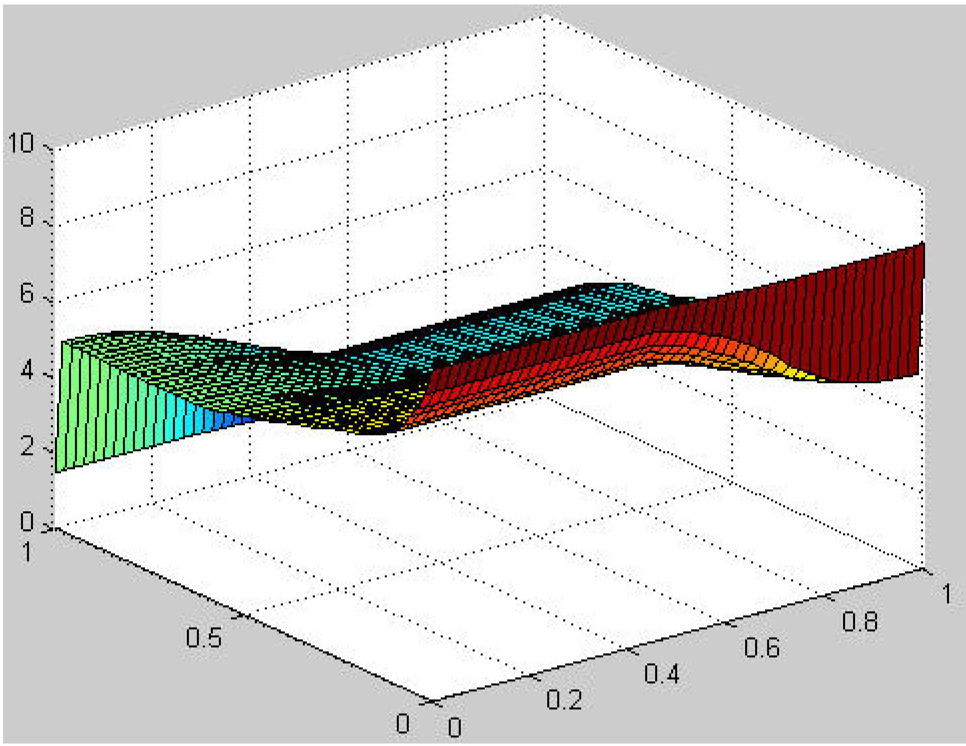


Figure 7-14 The reasoning results distribution of carriage design

7.4 Multi Autonomic Objects Collaborative Reasoning Algorithm

In multi AOs flexible workflow the intelligent collaboration between AOs is very important to the intelligent operation of a flexible workflow. The local and half-collaboration reasoning algorithm of AO has been given in 7.2. But when the AO is activated, the collaboration with other AOs is often needed because the limited knowledge or capability. The multi AOs collaboration can refer to the multi agent collaboration. The main collaboration method is described and analysed first, and the multi AOs collaboration reasoning algorithm is given later in this section to illustrate the scope of the research work.

7.4.1 Collaborative Mode of Multi Autonomic Object

It can be seen from the multi AOs flexible workflow model, the AO embedded in each flexible workflow node is the entity to realize the intelligent reasoning. After the decomposition of flexible workflow node, the control AO after decomposition is responsible for the coordination of each node's reasoning task. According to the architecture of AO in flexible workflow, the collaboration of multi AOs is classified into three methods in this research:

1) Horizontal collaboration: the horizontal collaboration only exists in the AOs in the same layer and with the same father AO, and the collaboration of AOs belonging to different father is conducted by their control AO;

2) Vertical collaboration: the vertical collaboration mainly deals with the collaboration between control AO and sub AOs. The vertical collaboration can be classified into upward collaboration and downward collaboration according to the initial AO; the upward collaboration is that the AO in sub-node cannot implement the general reasoning task, and sends collaboration requests to the control AO, while downward collaboration is that the control AO sends collaboration request to sub-AOs;

3) Hybrid collaboration: the hybrid collaboration is the combination of above two collaborations. For a specific complex reasoning task, the different collaboration is needed to realize the AO intelligence, one typical application of hybrid collaboration is the collaboration of AOs from different father nodes;

Based on the difference of collaboration, the multi AOs collaboration can be classified into task bearing and conclusion sharing. Task sharing means that the task T is decomposed into several sub tasks t_i, every AO bears a certain amount of sub tasks. Conclusion sharing means that AOs sharing their conclusions with other as the evidence for further reasoning. For the multi AOs flexible workflow, the solving of AO is $P = \left(S_0, o, S_k\right)$, because in the reasoning process multi AOs collaborate for one solving problem P more often, the collaboration of AO should mainly be the conclusion sharing collaboration.

The AO collaboration process is highly dynamic process, according to the dynamic changing of workflow instance. An AO changes its collaboration partner dynamically according to different reasoning solution. How to find suitable AO to collaborate is an important problem in multi AOs collaboration. Contract Net is currently most used collaboration mechanism of Multi-Agent System. A Manager is responsible for monitoring the execution and processing of task and a Worker is responsible for the execution. Every node is associated with the fixed role for the Manager or the Worker. A Contract Net is very suitable to realize the dynamic AOs intelligent collaboration.

Based on Contract Net, a AOs negotiation process based on conclusion sharing is given in figure 7-15. A multi AOs collaboration process is explained in a typical decomposed multi AOs flexible workflow structure.

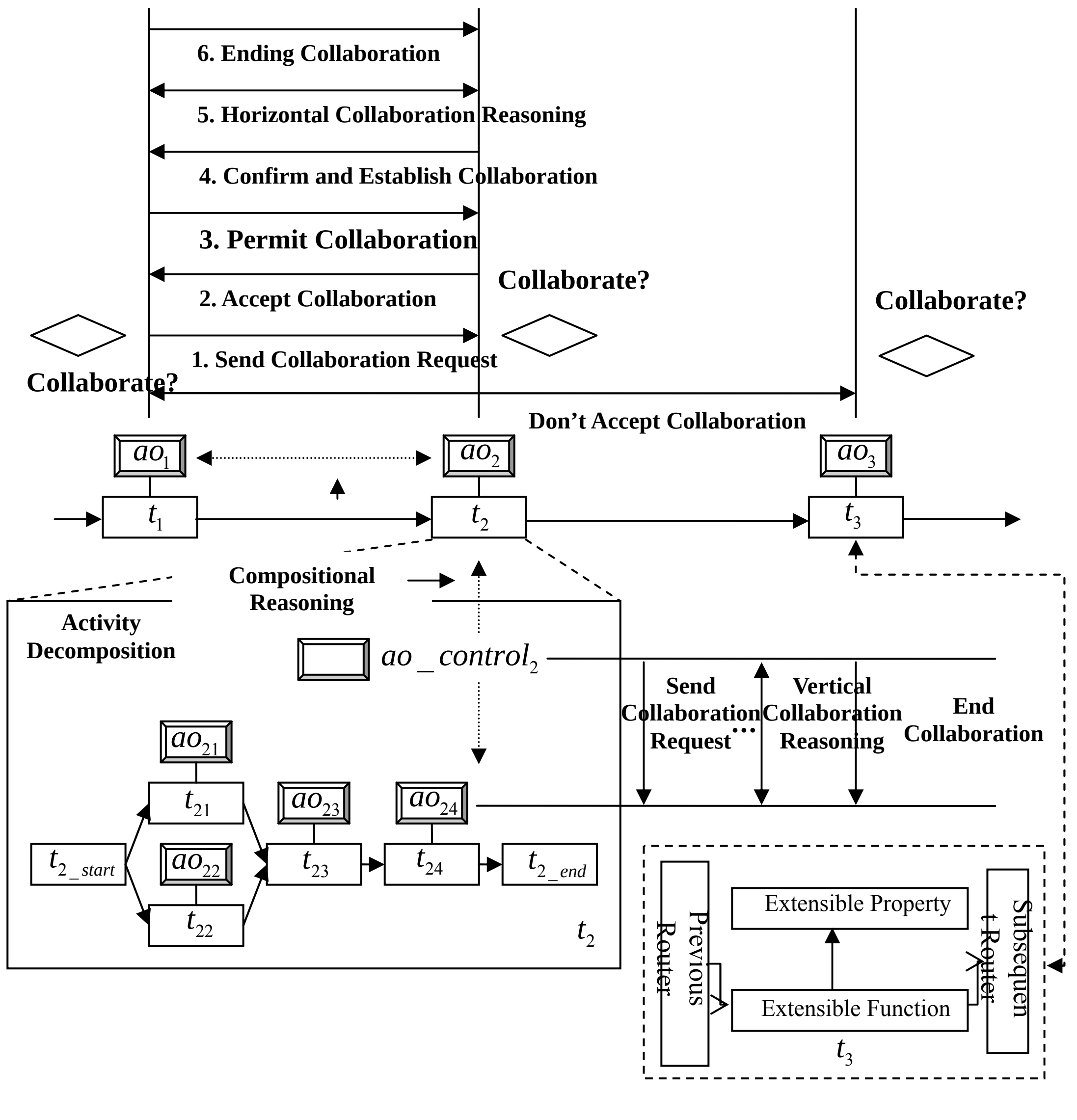


Figure 7-15 Collaboration of multi AOs

When AO ao_1 need collaboration, it sends collaboration request to AO ao_2 and ao_3 in other nodes, ao_2 accepts the request by its capability while ao_3 refuse the request; Once ao_1 receives the response from ao_2, it analyses if the response can satisfy the collaboration request, if it is satisfied, ao_1 sends collaboration permission to ao_2, the conclusion sharing collaboration is established if ao_2 confirms the permission, the collaboration is ending when collaboration finished. If the conclusion cannot be generated by collaboration, the procedure step 1 to step 6 in the figure is repeated and new collaboration partner is needed. The collaboration will turn to Manual in AO if the collaboration cannot be satisfied in required time.

For the AOs decomposing from the ao_2, ao_1 may need to collaborate with ao_{24} further, this collaboration is realized via a control AO $ao\_control_2$. It is a typical vertical collaboration as shown in the figure. The collaboration among ao_1, ao_2 and ao_{24} is typical hybrid collaboration process. The flexible elements operated by AO are illustrated by activity t_3 in the figure.

7.4.2 Collaborative Reasoning Algorithm of Multi Autonomic Object

Based on the multi AOs collaboration, the multi AOs collaboration reasoning algorithm based on conclusion sharing is illustrate in figure 7-16 with the collaboration of two AOs as example. It can be seen from section 7.2 that the intelligent of single AO is $\rho = (\sigma, K, P)$, then in the collaboration of ao_1 and ao_2 in figure 7-9, the intelligent reasoning of ao_1 is $\rho_1 = (\sigma_1, K_1, P_1)$ and the intelligent reasoning of ao_2. For the conclusion sharing collaboration, multi AOs have mutual middle facts and final conclusion, so the collaboration of ao_1 and ao_2 can be expressed as $\rho = (\sigma_1, \sigma_2, K_1, K_2, F, P_1, P_2)$, i.e., two AOs share input variables, reason according to each knowledge and reasoning mechanism, and share the intermediate facts and conclusion. The multi AOs collaboration process based on conclusion sharing is a process in which multi AOs exchange conclusions and solve conflicts continuously until the final conclusion is derived. The collaboration process is described in figure 7-16, the specific fuzzy reasoning algorithm including fuzzification, defuzzification, reasoning algorithm and conflict resolution uses the algorithms described in section 7.2. The conclusion sharing process includes the sharing of input instance variables, sharing of intermediate facts and sharing of conclusion. The multi AOs collaboration can be realized by the interaction of sharing of conclusion sharing and conflict resolution.

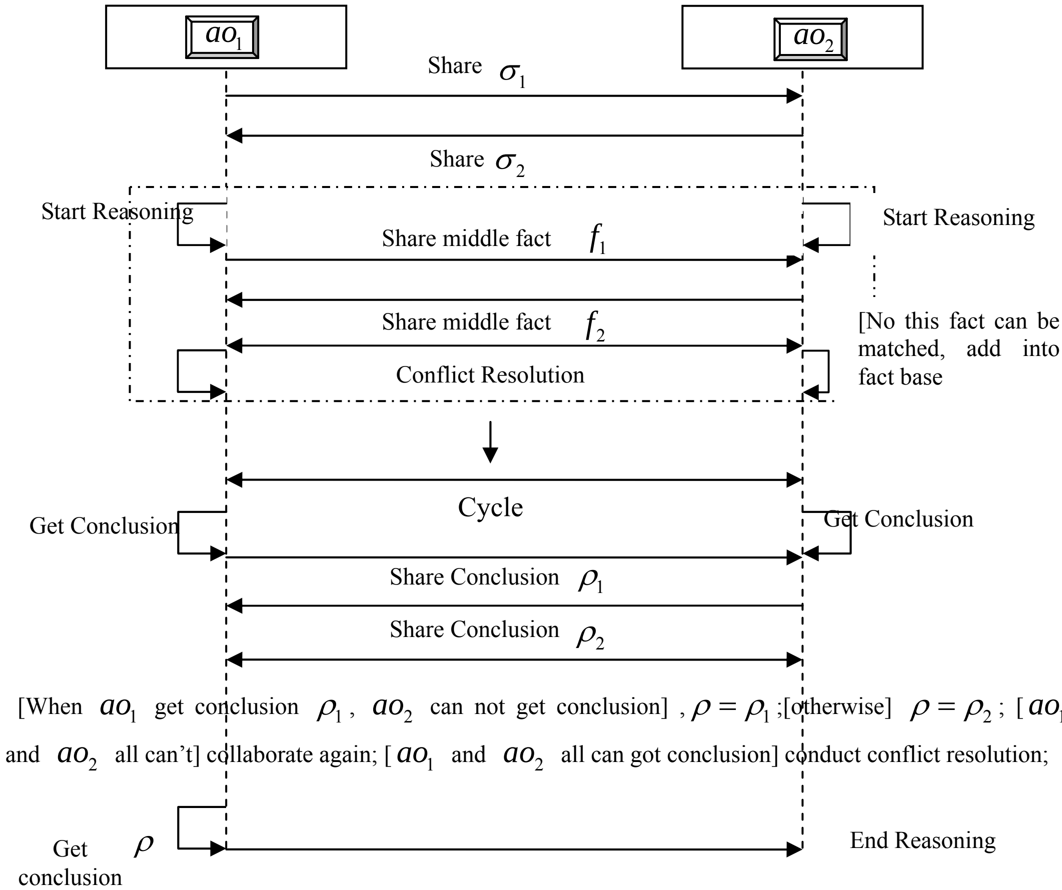


Figure 7-16 multi AOs collaboration reasoning algorithm

7.5 Bionic Flexible Workflow Adaptation intelligence Algorithm

AO intelligent algorithm based on fuzzy reasoning provides useful means to realize the intelligence of flexible workflow change. It can respond to outside change according to the judgement by the fuzzy knowledge, but sometime it can't respond the changes in a reasonable and satisfied time, especially concerning much complex fuzzy knowledge. Specifically, the workflow should have the capability to realize the change in a simpler and more direct way, as our human do by conditional reflection. A bionic flexible workflow adaptation algorithm is proposed to study the flexible workflow autonomic object intelligence further.

The capability that biological neural system can respond quickly to external changes provides a new perspective to flexible workflow intelligence. Creatures like humans can have quick and correct response to external stimulus by its neural network system in the form of conditional reflection. Meanwhile, the Artificial Neural Network (ANN) modelled from bionic neural system has good capability of self-adaptation and self-learning, and the computation of artificial neural network is a typical Non-programming computation. Considering above advantages of ANN, the developed ANN technology is applied in flexible workflow, and how the ANN can be integrated with flexible workflow to improve the ability of flexible workflow to external changes is studied.

7.5.1 Bionic neural network and flexible workflow

Creatures using their own neural system, can take appropriate response quickly to external stimulus in form of reflection arc. Biological neural systems have the effect of self-organization, memory, joint, and fault tolerant. If the flexible workflow can analyze the dynamic changing information by coordinating multi-neurons in flexible workflow neural network, and realize the automatic modification, the intelligent of flexible workflow responding to external changes will be greatly improved.

From the perspective of neural cells, the reflection arc is the assembly of simpler neural cells. By imitating the structure and function of biological neural cells, the biological reflection and its quick response can be realized. ANN is the simulation of biological nervous system in the view of nervous cells.

7.5.2 Engineering Design Process based on Bionic Flexible Workflow Model

7.5.2.1 Bionic Flexible Workflow Nodes neuron group model

Biological neural network consists of different nerves populations including many nerves cells, while engineering design process consists of different layer or group including many workflow nodes. Considering the structural similarity between biological neural network and engineering design process represented by workflow, neurons population model (node neurons population model) embedded in the design activity and neural network workflow model (layered neural network model) existed in every layer of engineering design process can be defined, the intelligence of engineering design process based on bionic flexible workflow is realized by the working of node neurons population model and layered neural network model embedded in flexible workflow.

The biological reflection arc receives the both inside and outside stimulatory signals by its sensor, the stimulatory signals is converted into nerve impulses by afferent nerve at first; then sent to nerve center where the nerve impulses is processed, finally the processed nerve information is sent to the effecor by efferent nerve to cause the movement of the body. The reflection arc finishes the response to the stimulations.

The node neurons population model is illustrated in figure 7-17 by imitating the structure and function of biological reflection arc, among them,

a) Sensor is equivalent to the receptor of reflex arc; it is embedded in the workflow node with the function to monitoring the dynamic change of workflow node. When there is change of workflow node parameter, status and data, the changing information is input into pre-processing unit.

b) Pre-processing unit is equivalent to the afferent nerve in reflex arc, it receives the changing information from sensor, and pre-processes and convert the changing information into digital information which can be identified by ANN, then key words extracted from changing information is matched with the ANN model and key words stored in knowledge, and processed information is transferred to processing unit group.

c) Processing unit group is equivalent to the nerve center in reflex arc, it consist of following three parts: input domain, middle domain and output domain, each of them including a number of artificial neurons. The artificial neurons from three domains integrate with each other and constitute a number of artificial neural network models. Different external change will trigger different ANN in the workflow. The connection of neurons in different ANN can be represented in a unified fully connected form. If two neurons are not connected, then the connection weight is set to 0. Processing unit group trigger the corresponding ANN model and its study mechanism, process the input information and send the processing result to post-processing unit.

d) Post-processing unit, its function like the “nerve impulse-power” conversion of effector in reflex arc. Post-processing unit receives and processes the input information from processing unit group (post processing is the inverse process of pre-process), then operate the workflow node according to the output information and store the processing information into knowledge set.

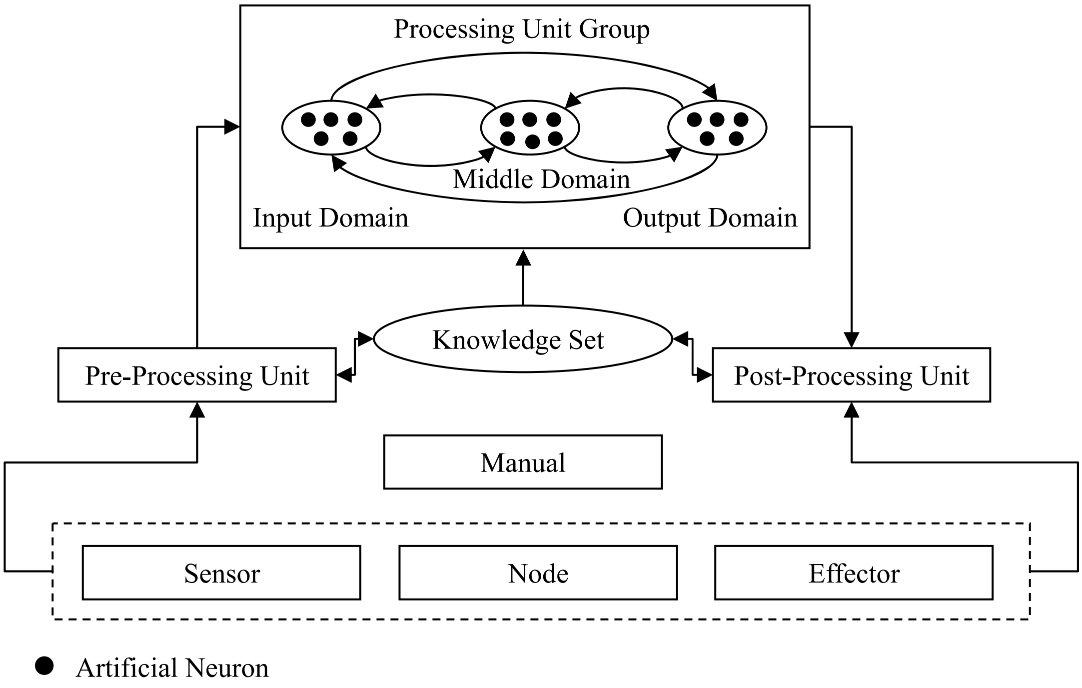


Figure 7-17 Neuron group model

e) Effector, its function is to trigger the action of body as effector do in reflex arc. Similar with the sensor, the effecor is embedded into the workflow node; it receives the output information from post-processing unit, sends related result to workflow node, and performs related operation on workflow.

f) Knowledge set stores the rules; ANN models type and study cases.

Normally there is a Manual in the node neurons population model, humans can interfere with specific changes or make decision by Manual. The node neurons population model only deals with inner changes in the workflow nodes, while layer neural network model deal with external changes concerning multi nodes in flexible workflow.

7.5.2.2 Engineering design process based on layer neural network workflow model

Engineering design process represented by workflow is a typical layered structure, the intelligence of each node in design process is realized by nodes neuron groups, correspondingly, the intelligence of each layer of flexible workflow is realized by building layer neural network model. The layer neural network model which is built on each layer of flexible workflow is shown as figure 7-18.

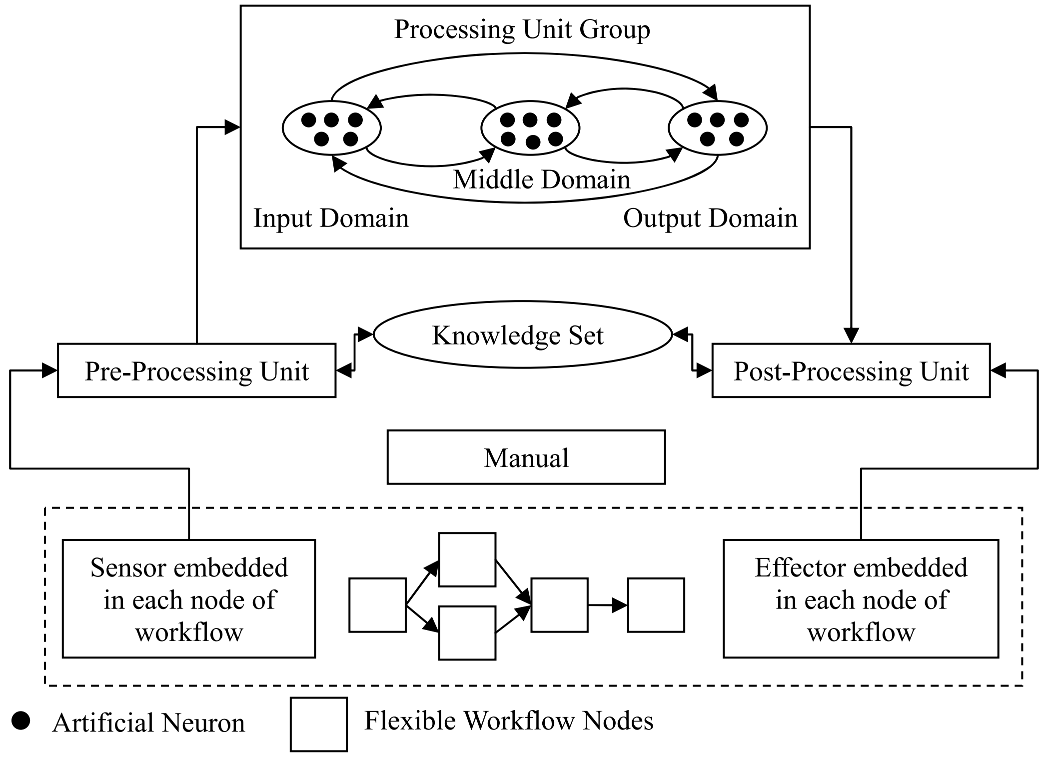


Figure 7-18 Layer neural network model

The nodes neuron group model and layer neural network model have similar structure. Their components is nearly same, this is decided by the unity of ANN. It also gives the convenience for the management of workflow system. Layer neural network is responsible for handling the dynamic change of multi-nodes in EDP, while nodes group model is responsible for handling the dynamic change of single node in EDP.

In layer network model, the sensor and effector embedded in every node is the sensor and effector embedded in single node of node neuron group model. The sensor in the single node can perceive the external change and send the information to the pre-processing unit, then the pre-processing unit match the external change with the ANN model in layer neural network mode and node neuron group model, the information is disposed by processing unit group.

7.5.2.3 The operations of neural network model flexible workflow in the different layers of the EDP

There are two different types of operations in different layers of EDP, up-down layer and adjacent layer operation. The operation process is illustrated in figure 7-19. If there is external change happens, up layer neural model start the computing. If computing result involves the modification of upper and adjacent layer workflow node, the up layer neural model will output the computing result into upper layer neural model, and decides if transfer the computing result to more upper layer. The more upper layer or adjacent layer neural network model will begin the computing, send the computing result into lower layer and store the computing result into knowledge set, the layer node execute the operation according the computing result.

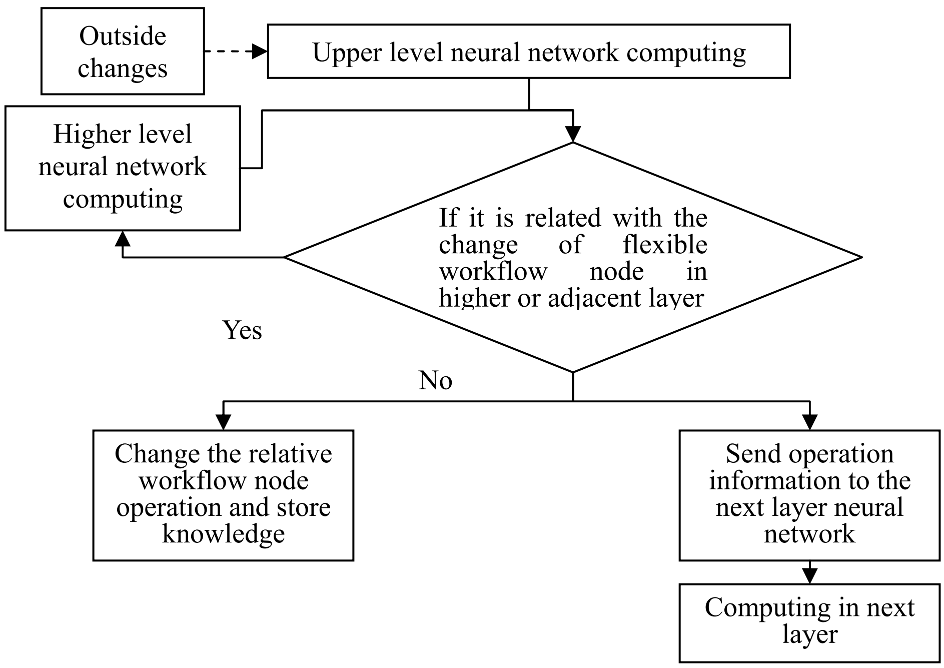


Figure 7-19 Operation of different layers of flexible workflow layer neural network model

The operations of neural network model flexible workflow in the different layers of the EDP are illustrated in figure 7-20. If there is external change in design activity in layer A, then the layer neural network model begin the computing, if the computing result is related with the modification of nodes in upper layer C or adjacent layer B, then the layer neural network model in layer A transfer the computing result to upper layer C for computing, and will decided if it is need to transfer the computing result into more upper layer; if the computing result is not transferred into layer C, the computing result will be transferred into layer A or adjacent layer B, the node in layer A or B will execute the operation according the computing result in the EDP. This can happens in a typical EDP, while assembly design results will determines the lower layer of detailed parts design as shown in figure 7-20.

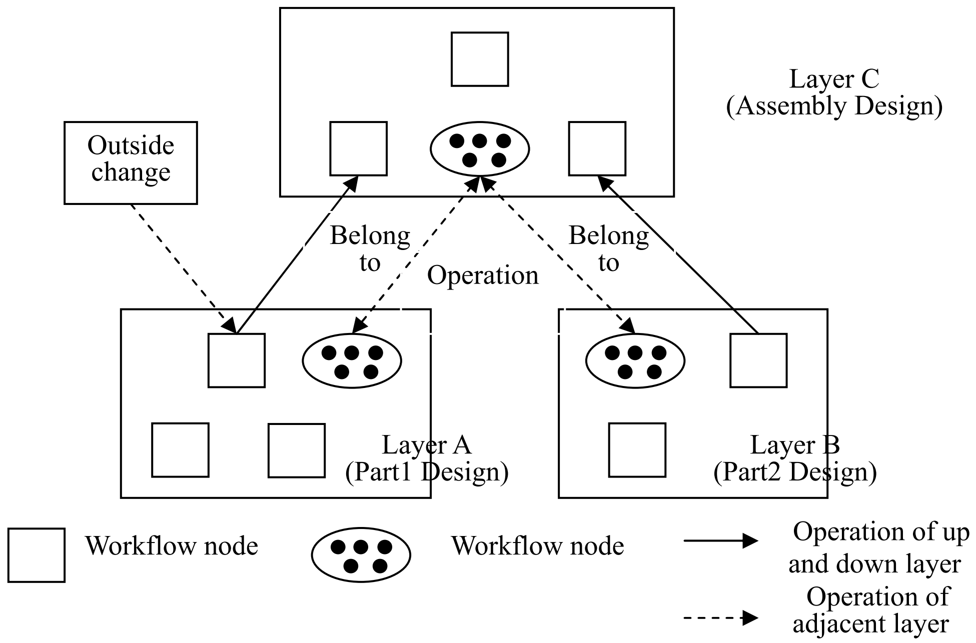


Figure 7-20 Operation of different layers of flexible workflow neural network model

It can be concluded that the communication of flexible neural network is in the way of up-down, down feedback, adjacent layers is communicated by their upper layer.

7.5.3 Bionic Adaptive algorithm of flexible workflow and its Application in EDP

Based on bionic flexible workflow model in EDP, a bionic flexible workflow adaptive algorithm is proposed, as illustrated in figure 7-21.

1) Extract data: according to WfMC, there are three different workflow data, i.e., workflow control data, workflow related data and workflow application data. The data in EDP will be classified as above three categories;

2) The classification of data: classify the extracted data after the extraction. For the example of application data in EDP, the workflow application data is classified by task or department, such as the data of CNC workshop includes the materials of parts, machine and the operation program;

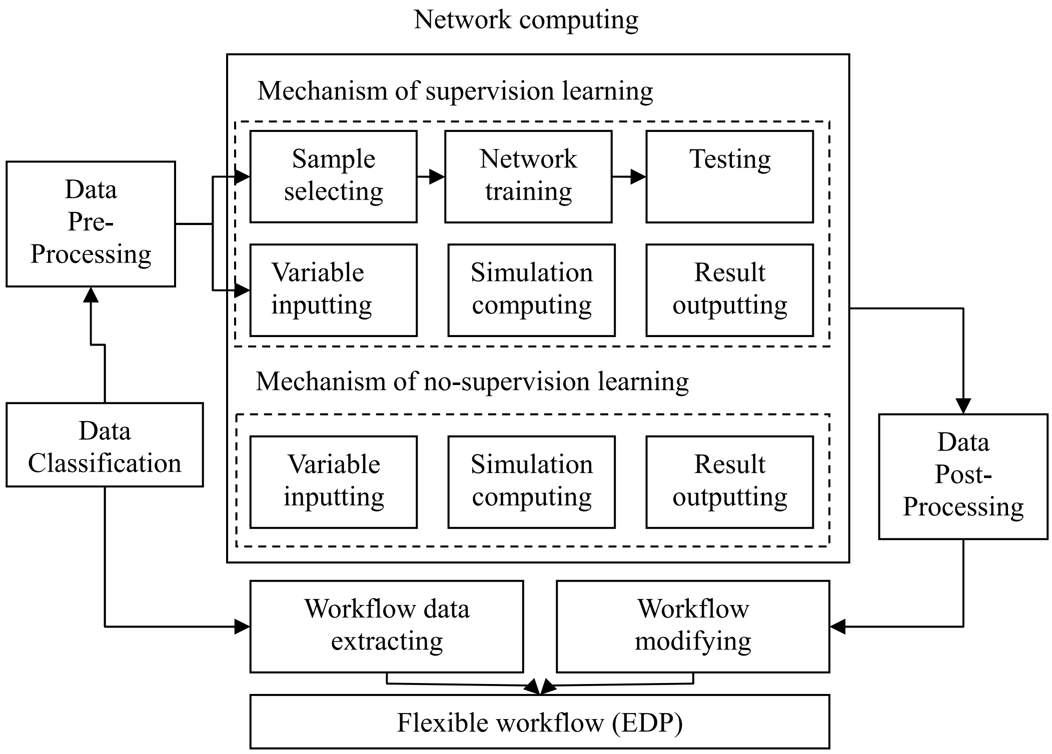


Figure 7-21 the framework of adaptive algorithm in bionic flexible workflow

3) Data pre-processing: transforming nonnumeric information to numeric one that can be accepted by neural network. The nonnumeric information includes the state information of activities, the converted conditions of activities and application data. For some determined discrete, such as state information, tool name, it value can be numbered from 1 to N. For some data such as machine performance, it can be instanced into real from 1 to 0 corresponding to very good and very bad;

4) Network computing: matching of the neural network type is conducted before network computing. According to the definition of ANN in this thesis, there are two types of matching, including $Net\text{-}Type$ and $Learn\text{-}Type$. The computing is conducted after choosing suitable ANN model;

5) The post-processing of data: restore the anti-normalized output data into true data, post-processing is inverse process of pre-processing;

6) The response of workflow node: according to the output information of neural network, executes operation in accordance with the workflow nodes.

Taking a work time computing process in EDP as an example, a part design nodes need ensure the norm of working hour according to the specific parameter in the part order. A bionic flexible workflow model will be build mainly on part design as example to testify the proposed bionic flexible workflow adaptation algorithm. The ANN model in this node neuron group is BP network. The BP network is one of the most widely used network model, which adopts back propagation algorithm. The model is based on trained BP network; therefore, introduce the establishment of the BP network used to disposed unit group firstly. The face grinding is selected to testify effect responding to the change of fixed work hour factors' changes. Table 7-6 is the standard fixed working hour table of face grinding.

Table7-6 Standard work hours fixed table of face grinding (min)

Rough-ness	Diameter (mm)	20	24	29	35	41	50	60	72	85	103	124	149	178	214	257	308
	Width (mm)	5	6	7	8	10	12	15	18	21	26	31	37	45	53	64	77
	6	0.5	0.6	0.7	0.8	0.9	1	1.1	1.3	1.5	1.8	2	2.3	2.7	3	3.5	4
	7	0.6	0.7	0.8	0.9	1	1.1	1.3	1.5	1.8	2	2.3	2.7	3	3.5	4	4.7
	8	0.7	0.8	0.9	1	1.1	1.3	1.5	1.8	2	2.3	2.7	3	3.5	4	4.7	5.4
	9	0.8	0.9	1	1.1	1.3	1.5	1.8	2	2.3	2.7	3	3.5	4	4.7	5.	6.2

It can be seen from table 7-6, the factors that are relative to the norm of working hour of parts include roughness, diameter, and width. Small parts of the fixed working hour data for a specific process of part machining can be obtained directly from table 7-6, while most of them can only be derived from the table roughly, these fixed working hour data is imprecise in this way. By using of the function of BP network which can approximate any multivariate nonlinear function value by multiple nonlinear node interpolation, the accuracy of computing the fixed working hour can be improved.

A three-layer BP neural network model is adopted in the Processing Unit Group, as shown in figure 7-22. The number of input, hidden and output layer neuron is 3, 9 and 1 respectively. The transmit function of input, hidden and output layer is tansig, logsig and purelin. Network learning function is learngdm, and the trained function trainlm.

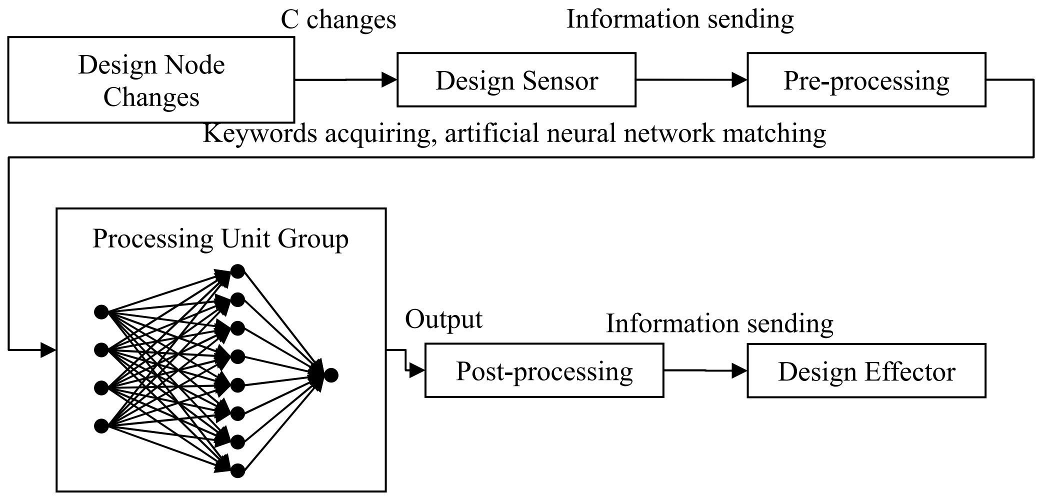


Figure 7-22 Case of bionic flexible workflow model in work time computing of EDP

The established BP network can be trained and simulated by data extracting table 7-6. The output of the trained BP network is shown in table 7-7.

Table 7-7 The BP network output result

Roughness	Diameter (mm)	Width (mm)	Output work hour (min)
6	30	10	0.97
6	35	6	0.50
6	41	15	1.35
6	55	20	1.68
7	30	5	0.55
7	24	10	1.11
7	30	20	1.95
7	35	10	1.02
7	41	10	0.97
7	72	18	1.49
7	220	53	3.53
7	257	80	3.99
7	308	60	5.05
8	60	30	2.56
8	90	18	1.67
8	80	21	1.90
8	100	26	2.23
8	210	45	3.91
8	214	53	4.01
8	308	77	5.42
9	30	5	0.99
9	55	6	1.10
9	55	20	2.14
9	60	8	1.27

The trained BP network is stored in the knowledge set of fixed working hour node, among them, $Net\text{-}Type$ is 1: 2: Labor time standard, 1 means the layer of fixed working hour node, 2 means the number of face grinding in various processing types. Since the mechanism of BP network is the learning supervision mechanism, the type of $Learn\text{-}Type$ is $Established$. The key word of the BP network is fixed working hour.

The instance of bionic flexible workflow is shown in figure 7-22, the change of flexible workflow node is the difference between two node in different instance, this difference can be represent by the attribute of nodes, the method, the environment, the active state, the control logic and the set of changes of node. Supposing the fixed working hour of the present face grinding is (roughness, d,w, Td) = (9, 50, 12, 1.5), when the diameter changes to 55mm and the width changes to 20mm, the part design node of bionic flexible workflow begin to change, the node data sets in the C change accordingly. Sensor can acquires the change information, and send the information to pre-processing unit.

Processing will extracts the changes information according to the problem types and nodes information. The extracted changing information is that processing type number is 2, change value of processing length. According to the match principle and key words, the joint searching for ANN type and key words is conducted, the BP network whose $Net\text{-}Type$ is 1: 2: Labour time standard is found in the knowledge set.

By the computing of Processing Unit Group, the output fixed working hour Td = 2.14min, the output result is send to Effector and the current running instance is modified to new instance according to the output result. Meanwhile this modification case is stored in the knowledge set. Above case can testify that the proposed bionic flexible workflow can respond to external change rapidly.

By above bionic adaptive algorithm of flexible workflow, without the complex reasoning, the EDP can rapidly find out the suitable changed result according the input change, and the new EDP instance can be created in a reasonable and satisfied time. This can provides another useful way to improve the intelligence of EDP.

7.5 Chapter Summary

Based on the proposed the Multi-AOs flexible workflow, the theory and architecture of AO intelligence is researched in detail. The monitoring and execution of AO is introduced. Aiming at the uncertainty of AO intelligence represented by fuzziness, the AO intelligence algorithm based on Extended Mamdani fuzzy reasoning is proposed, the architecture of AO fuzzy reasoning and AO knowledge expression is introduced first, then weighted fuzzy reasoning algorithm and precise and fuzzy hybrid reasoning is researched and implemented with emphasis, finally it is demonstrated that the proposed AO intelligence can be used to determine the flexible production process when order and production is dynamic by a practical sub-contractor production process. Further, aiming at the slow response of flexible workflow to external change, a bionic reflex mechanism is introduced to flexible workflow, a bionic flexible workflow model and its adaptation algorithm is proposed. The algorithm is implemented and testified by an example of formulating fixed working hour in a production workflow, showing the effectiveness of the algorithm.

Chapter 8 Flexible Execution of Distributed Workflow Engine based on Web Service

The flexible model must be executed by workflow engine to realize the application of flexible workflow in dynamic engineering design process. Considering the inherent characteristics of cross-enterprise and distribution, aiming at addressing the deficiency of current research on flexible workflow based web service, this chapter describes the work undertaken to investigate the distributed flexible workflow execution and flexible workflow expression for web environment and timely service using such an environment.

8.1 Web Service Environments

Web Service is a new distributed computing framework based on web based standards, such as XML. It is a self-contained, self-described and modular service, it can be published, located and called by web and is based on certain open standards and draft, such as WSDL, SOAP, UDDI (Universal Description Discovery and Integration) and so on. Web Service is platform-independent, supporting Object-Oriented application, and loose couple of client and server. Web Service can be described, published, and searched and called, in simple word, Web Service is a API or application program which is provided for outside and called over Internet, the application software in other companies can access and use this application service over Internet.

Web Service is an object deployed over Web, and it has the merit of Object technology. Because Web Service is based on XML-based and open Web technology, it has better openness than current Object technology and is new distributed application program platform for interoperation. Web Service defines how interoperability of application program is realized. The Web Service can be programmed by any language and in any platform. Web Service has follows advantages: good encapsulation, loose-couple, using standard protocol and norm, and high integration capability.

Web Service is based on Service Oriented Architecture, SOA, as illustrated in figure 8-1. There are altogether three roles in SOA:

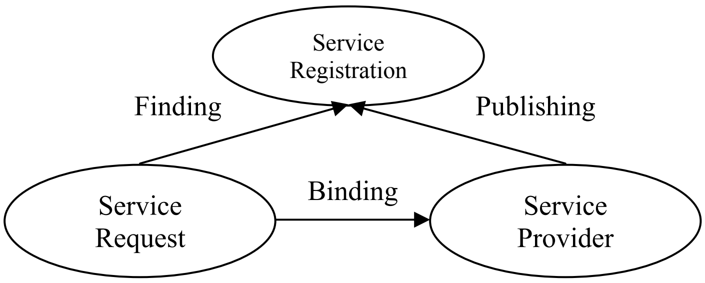


Figure 8-1 Service Oriented Architecture

1) Service Provider: publishing service, and responding with service request;

2) Service Registration Centre: registering published Web Service, classifying Web Service and providing searching service;

3) Service Requester: searching necessary service in service registering center, and using service;

The component in SOA architecture must have one or more above roles, these roles interoperates by three operations namely, publishing, searching and binding. In order to support above three operation, the standard Web Services protocol stack (IBM 2006), as illustrated in (a) of figure 8-2. The (b) in figure 8-2 describes the cross-organization workflow Web Service call principle based on Web Service protocol stack. As shown in figure 8-2, the communication basis of Web Service is HTTP, it is a non-deterministic transmission. HTTP has become the standard network protocol of Web Service in Internet for its universality.

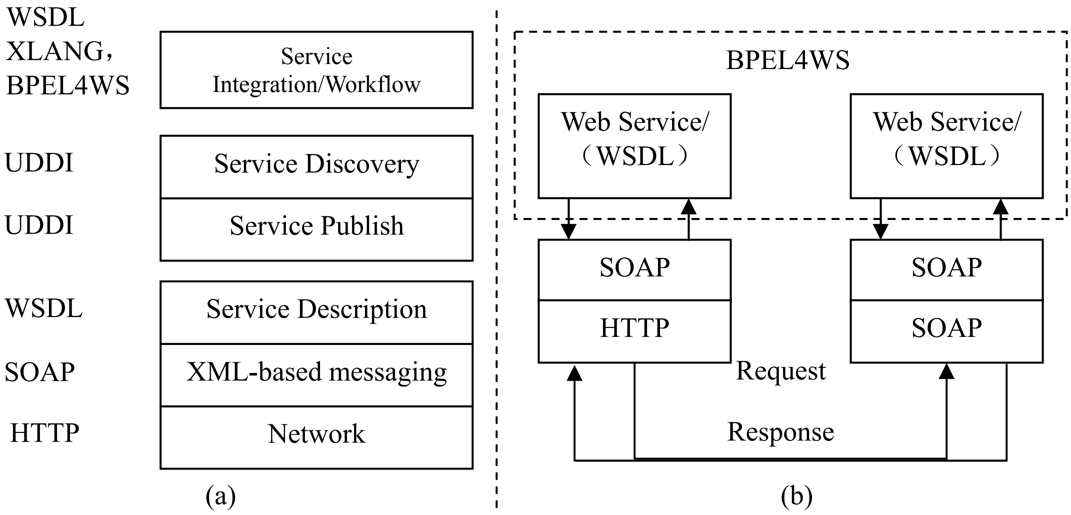


Figure 8-2 Web Services protocol stack and Web Service call principle (IBM Software Group, 2006)

XML (Bray et al., 2006) is data coding standard, the current industry standard of XML information transfer is SOAP. SOAP is based on XML and is a simple and light-weight mechanism to transfer structured data between application programs on Internet. SOAP provides a framework for cross-Internet call service in distributed and dispersive computing environment and provides a cross-platform integration mechanism independent of the programming language and underlying distributed object infrastructure. SOAP is a Remote Procedure Call (RPC), although SOAP is not bound by a specific transportation protocol, HTTP becomes the most popular protocol for SOAP. SOAP is better than DCOM、CORBA and RMI（Remote Method Invocation）in cross firewall for its running on HTTP.

Web Services Description Language, WSDL (Christensen and Curbera et al 2006) is XML-based Web Service Interface description language specifying how a program interoperates with Web Service. Web Services are defined as a set of access points or ports in WSDL document. The information in WSDL document is the abstract description of data while port type is the abstract set of operation.

Universal Description Discovery and Integration, UDDI (Microsoft 2006) is set of distributed Web Services information registry centre standard based on Web, and includes a set of access protocol standards. Enterprises can register their own Web Services and enable other enterprises to find and access these Web Services by UDDI. UDDI includes description of SOAP information XML Schema and UDDI API which jointly construct the basic information model and interaction framework for UDDI. The description and information interaction are mainly studied in this chapter, while the application of UDDI is not studied in detail.

Web Services not only provide the inter-operation technology between Web resources, but also provide the method to define work process by combining basic Web Services. There is no commonly recognized standard language for combining Web Service process, the commonly used include WSFL (Web Services Flow Language) of IBM, XLANG (XML LANGuage) and BPML in Microsoft. In recent years, IBM, Microsoft and BEA jointly published BPEL4WS (Liu et al 2004; Hu et al 2003; Andrews et al 2007), BPEL4WS can define business processes and their relation with Web Service.

8.2 Distributed Flexible Workflow Engine Design based on Web Service

8.2.1 Framework of Distributed Flexible Workflow based on Web Service

In dynamic engineering design process, the workflow should support the interoperation between collaborative enterprises. Because the collaboration between enterprises is often created dynamically, the dynamic engineering design process management system should support the cross-enterprise dynamic collaboration. Flexibility and cross-enterprise operation should be urgently addressed for flexible workflow. Web Service technology can support standard cross-enterprise, cross-platform application, and it has fine inter-operation and extensibility capability, and can implement complicated business operations by loose-coupling. So the Web Service technology naturally is a very suitable alternative to implement the underlying communication infrastructure of distributed flexible workflow engine supporting dynamic engineering design process and to satisfy the demand of flexible cross-enterprise inter-operation by dynamic engineering design process. Because the flexibility of flexible workflow is realized by dynamic instance-based workflow modelling, the execution of flexible workflow is highly related to the flexible workflow model. Because the multi AOs is integrated into a flexible workflow model proposed in chapter 6, a cross-enterprise distributed flexible workflow engine based on Web Service is proposed and prototyped in this chapter.

The engine of distributed flexible workflow supporting dynamic engineering design process should be a typical fully distributed structure composed by multi workflow engines. Integrating with the characteristics of flexible workflow model and Web Service, the framework of distributed flexible workflow engine based on Web Service is designed in the following way:

- Web Service is the communication infrastructure of distributed flexible workflow engine;
- Multi distributed flexible workflow is distributed in different enterprises in dynamic engineering design process;
- Each workflow engine is responsible for a part of workflow model, including the instance creation and execution as well as the navigation of workflow process;

- Each workflow engine is described as a Web Service;
- Workflow engines interact and transfer data by SOAP; meanwhile every node including the basic nodes after decomposition in flexible workflow model is described as a Web Service;
- Finally the process model of flexible workflow is described as a combined Web Service flow.

The execution of flexible workflow model by workflow engine is changed into the calling of Web Services. The introduction of Web Service not only implements the cross-enterprise distributed operation standard of workflow engine, but also realizes the dynamic router, dynamic binding of workflow calling and dynamic extension of workflow node, then the characteristics of flexible workflow can be implemented by the execution of flexible workflow instance. With a typical cross-enterprise workflow model as example, the Web Services-based Cross-Enterprise Distributed Workflow Engine, WSCE-DWE is given in figure 8-3, the WSCE-DWE is composed of four layers, namely, model layer, Web Service instance layer, workflow execution service layer and application layer.

1) the first layer is the flexible workflow model created by dynamic instance-based modelling;

2) the second layer is Web Service instance layer, this layer is formed by following two procedures, the node of flexible workflow model is described as Web Service by WSDL, the whole flexible workflow process model is described as a compositional Web Service flow; then the Web Service is called by workflow engine as instance, the cross-enterprise flexible workflow is executed;

3) the third layer is workflow enactment service layer, it is composed of cross-enterprise workflow engines, workflow engine is responsible for the calling and navigation of Web Service and compositional Web Service process, each workflow engine is described and implemented by Web Service, they interact with SOAP message. The function of workflow engine mainly includes instance initialization, activity execution, external interaction and data maintaining. The instance database records the running information of instance while workflow list records the work item of workflow instance. Workflow list is mainly responsible for the calling of workflow application interface and application procedure;

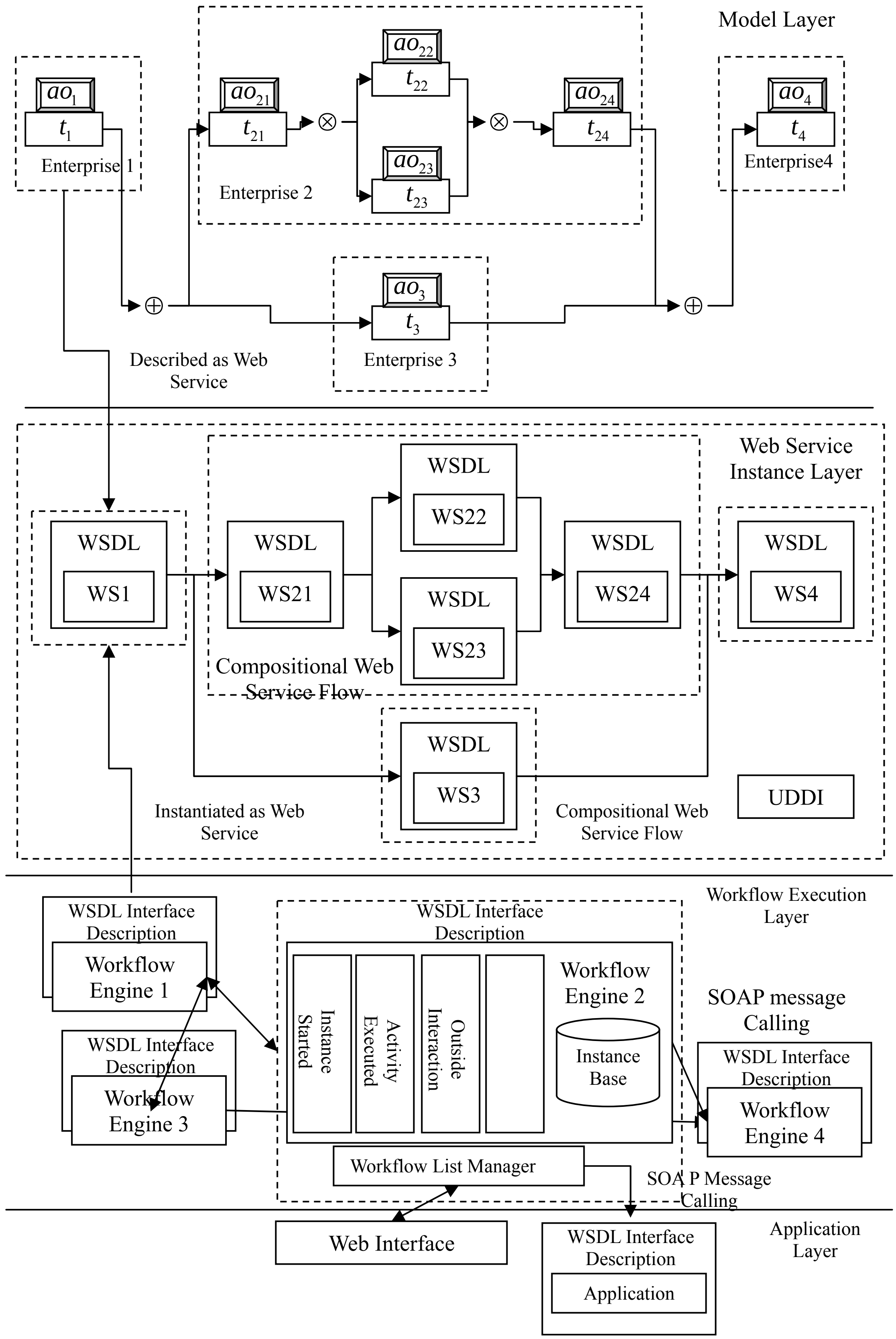


Figure 8-3 Framework of Distributed Flexible Workflow based on Web Service

4) the fourth layer is application layer, according to the difference of workflow interaction. The

application can be classified into Web interface operation and application program calling; the workflow user can call Web Service of workflow engine by Web interface, for the application program called by workflow engine, including the enterprise legacy system, it can be encapsulated and called by workflow to realize the integration of heterogeneous systems.

In WSCE-DWE, the granularity of a distributed workflow engine normally is configured as follows: a workflow engine is configured for a company, for large scale company, a workflow engine for a design group is configured while this engine is still described and implemented by Web Service. In WSCE-DWE, the SOAP message calling of distributed cross-enterprises workflow engines, Web Service interface description of workflow engine is studied.

8.2.2 SOAP Message Call of Distributed Flexible Workflow Engine

SOAP provides a simple, extensible and comprehensive XML information handling framework, it is can be used to define higher layer application program protocol, then to provide higher interoperability in distributed heterogeneous environment(Nilo Mitra, 2005).The core of SOAP is its information handling framework which defines a set of XML elements to encapsulate XML information.

In figure 8-3, the interoperation of encapsulated workflow is implemented by the SOAP information calling of Web Services. Four possible workflow interoperation model, chain service model, network sub-process model, end to end model and parallel synchronization model, is proposed by WfMC, and the interface of heterogeneous workflow information interoperation is defined as interface 4, interface 4 can be implemented by interoperation function, information calling and common gateway (Hollingsworth 2005). For any application of cross-enterprises workflow in dynamic engineering design process, because any nodes in every workflow can be interacting with outside, the distributed workflow engine should call each other, the interacting workflow engines should collaborate to implement whole calling process. The end to end model is very suitable to realize the design collaboration between design enterprises or groups, and it is consistent with the request/response manner of Web Service, the end to end model is the mainly manner of workflow interoperation.

Because the full interoperability is hard to achieve for workflow management system interoperation, the interoperation is mainly realized by process model sharing or transferring control, related and application data in different execution. In cross-enterprise engineering design process, according to the concept of virtual workflow, the workflow model corresponding with different workflow engine is distributed in different companies. So in the interoperation of distributed workflow engines based on Web Service, only the control and related information is need to be transferred to active corresponding workflow model and the model is not need to be shared instead. For workflow engine based on Web Service, the control information is transferred by encapsulated SOAP message.

The workflow interoperation based on SOAP message calling is based on the basis where the different workflow models can be understood by different workflow engines. But in the practical running of cross-enterprises engineering design process, the workflow models are often heterogeneous. Because the flexibility of the workflow is the research point, In order to simplify the complexity of the issue and highlight the emphasis of the research purposes, the study of this chapter is based on the following two assumptions:

1) the heterogeneous workflow model can be inter-converted by specific rules, although the model is heterogeneous;

2) the SOAP message of control and related data between distributed workflow engines can be inter-converted by XSLT（Extensible Style sheet Language Transformations）as illustrated in figure 8-4. It is assumed that the SOAP message between workflow engines can be understand by each other without conversion in this chapter.

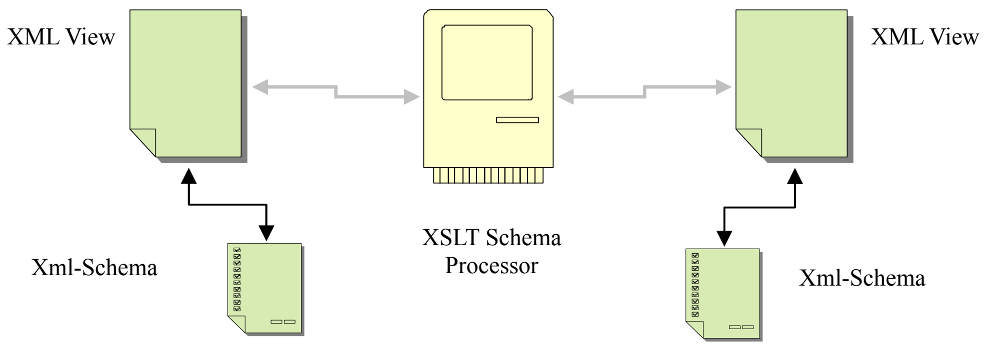


Figure 8-4 SOAP/XML document conversion

The above assumption is the foundation of Web Service-based distributed workflow engine interoperation. In fact, above assumption is established only if the workflow model in different enterprise is model following a common standard.

Request/response model is a commonly used SOAP application, sending workflow engine sends the SOAP document to receiving workflow engine, receiving machine will handle this request and generates response, then the response document is sent back to sending workflow engine. The distributed workflow interoperation is a complex business process in cross-enterprise design process, the workflow engine should connect with the corresponding workflow engine at first according to the router of flexible workflow model, sending SOAP message of control data on receiving response, then getting response and ending the connection. The interoperation process of cross-enterprises distributed workflow engines in cross-enterprise engineering design process is a conversation based on request/response model, as shown in figure 8-5.

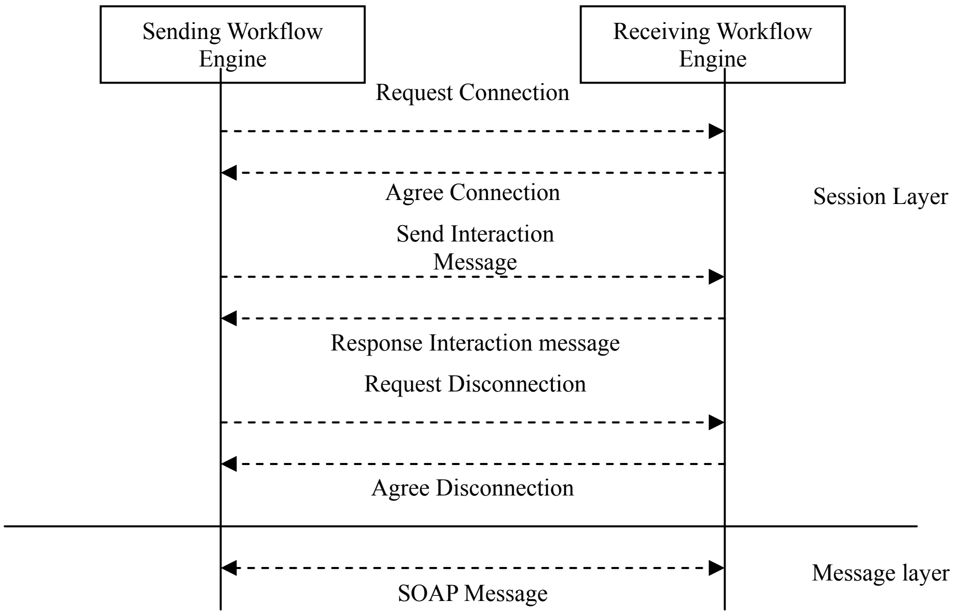


Figure 8-5 workflow engine SOAP message calling conversations

The conversation in figure 8-5 is realized by SOAP message, in which sending and responding message is the core. The information between workflow engines include the activation of activity or sub-process, the status of process/activity instance and workflow related and application data. A workflow engine is responsible for activating next workflow engine and sending related information after its corresponding activities is finished. The content of SOAP message between workflow engines is given in table 8-1.

Table 8-1 Workflow SOAP message

Envelope	SOAP Message Calling	
Header	Message ID, $Message\_ID$	
	Sending workflow engine ID, WEF_ID	
	Receiving workflow engine ID, WET_ID	
	Identification Information, $PassWord$	
Body	Operation of receiving workflow engine Web Service, $Operation\_Name$	
	Control Information	Sending instance status, S
		Sending the router information directing from sending instance to receiving instance, DR_Front_{ij}
	Related Information	Sending the extensible properties of instance, $Extend\_A$
	Application Information	Sending the data set, D

The SOAP message sending design information message from workflow engine WE1 to workflow engine WE2 is given in figure 8-6.

```
<!-- POST /WE_WebService/WE1_Service.asmx HTTP/1.1
Host: 192.168.9.7
Content-Type: text/xml; charset=utf-8
Content-Length: nnnn
SOAPAction: "http://tempuri.org/WE1_Webservice"
-->
<!--<?xml version="1.0" encoding="utf-8"?>-->
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="
http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Header>
        <S:Message_ID XMLns:S="http://tempuri.org">WE1063829WE21</S:Message_ID>
        <S:WEF_ID>WE1</S:WEF_ID>
        <S:WET_ID>WE21</S:WET_ID>
        <S:PassWord>SUPPLYORDER134986</S:PassWord>
    </soap:Header>
    <soap:Body>
        <T:WE21_INTER xmlns:T="http://tempuri.org/wengine">
            <T:Control_Info>
                <T:WE_S>Finished</T:WE_S>
                <T:DR_Front>T21</T:DR_Front>
            </T:Control_Info>
            <T:Relate_Info>
                <T:Order_ID>GEAE_5500018569</T:Order_ID>
                <T:Order_DATE>2006-03-09</T:Order_DATE>
                <T:Order_Comapny>XAC</T:Order_Comapny>
            </T:Relate_Info>
            <T:App_Info>
                <T:Part_Name>LK82670</T:Part_Name>
                <T:Part_Amount>10</T:Part_Amount>
                <T:Part_Price>345.72</T:Part_Price>
                <T:Part_Diagram>../GEAE/LK82670.DWG</T:Part_Diagram>
                <T:Part_Name>UL19468</T:Part_Name>
                <T:Part_Amount>22</T:Part_Amount>
                <T:Part_Price>63.24</T:Part_Price>
                <T:Part_Diagram>../GEAE/UL19468.DWG</T:Part_Diagram>
            </T:App_Info>
        </T:WE21_INTER>
    </soap:Body>
</soap:Envelope>
```

Figure 8-6 example of SOAP message calling

For the SOAP message in figure 8-6, the Header part of the SOAP message is normally fixed. But for the execution of flexible workflow, the extensible property and dynamic router is normally determined in runtime of workflow instance, the Body part of the SOAP message, such as the router in control information and the extensible property in related information, all should be variable according to the difference of workflow engine and workflow instance. So the SOAP message Schema of different workflow engine and instance should be extended based on specific standard.

The Schema of SOAP message is given in figure 8-7. For the reason of limited space, only the Schema in BODY part is given, and the Schema of control and application information is expanded either. It can be known from figure 8-7, the SOAP message of cross-enterprise flexible workflow is expanded dynamically based on this Schema, $T:NewControl\_Info$ in $T:Control\_Info$ means that the elements in SOAP message can be added or deleted dynamically, and type of elements can be changed. The simple type and complex type in SOAP message all are defined by Schema. For the entity data, such as document, it can be expressed by its link in the server. For the data set in database, it can be expressed by XML document.

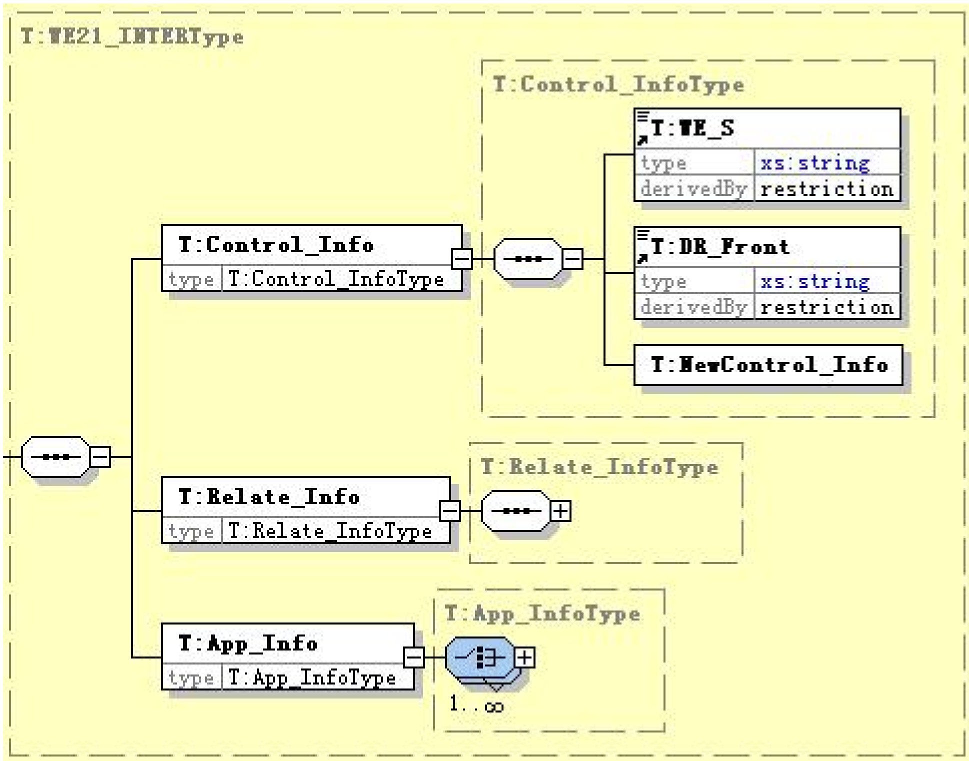


Figure.8-7 SOAP message Schema of workflow engine

8.2.3 WSDL Description of Distributed Flexible Workflow Engine

Based on the SOAP message calling, the cross-enterprise interoperation can be realized by workflow engine. In WSCE-DWE, the workflow is embodied by Web Service and it is described by WSDL, and specifically is programmed by .Net of Microsoft. The following tasks should be implemented by workflow engine (Hollingsworth 2005): initializing and executing the workflow process model, navigating for the execution of process and activity, interacting with outside resources and maintaining workflow control and related data.

The external form of above functions in workflow engine is realized by a set of interface. By the calling of these interfaces, the function of workflow engine can be implemented. A workflow application programming interface WAPI is given by WfMC, the WAPI normally can be defined by its logic function, the data type it operated (calling parameter) and referenced data structure. The function interface is introduced in following 8 aspects by WAPI: connection of workflow, process control, activity control, process status, workflow list management, workflow monitoring and management, and workflow application calling (WfMC 2005). In WSCE-DWE, the function interface of workflow engine should be mapped into the WSDL description, the elements composition of WSDL is given in figure 8-8.

Normally the elements in WSDL include Type, Message, Operation, PortType, Binding, Port and Service. The Message in WSDL document is the abstract description of communication data type. PortType is the abstract set of operation, a binding of PortType is formed by specific protocols and data formats, a Port can be defined by linking Web access address with reusable binding while the set of Port can be defined as Service.The service access point and abstract definition of information can be separated from specific service deployment and data binding by WSDL document. Type、Message、Operation and PortType describe abstract definition of Web Service, and there are unrelated with deployment detail. Service、Binding and Port is related with the deployment of specific service, as illustrated in the grey part of figure 8-8. The abstract definition can be reused, and the flexibility of workflow system can be improved further by the dynamic binding of service.

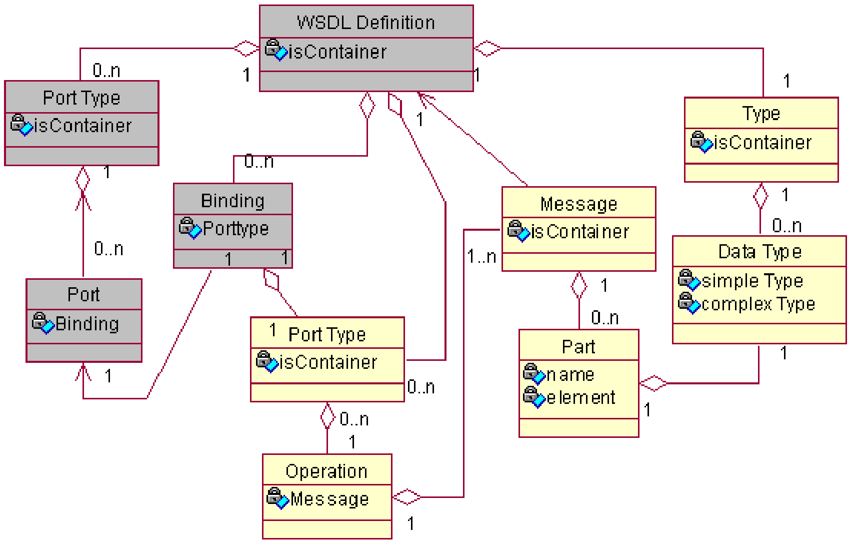


Figure 8-8 the elements composition of WSDL

According to the characteristic of WSDL, the design scenario of WSCE-DWE is: the workflow engine is described as Web Service, and the different functions in workflow engine is described as multi PortType; then the function of workflow engine is refined further, the concrete function is described as abstract operation and message; in deploying workflow engine, the description of workflow engine is bound with Web Service operation, the function of workflow engine can be implemented. The WSDL design of workflow engine is given in figure 8-9, only the main function is given in the figure.

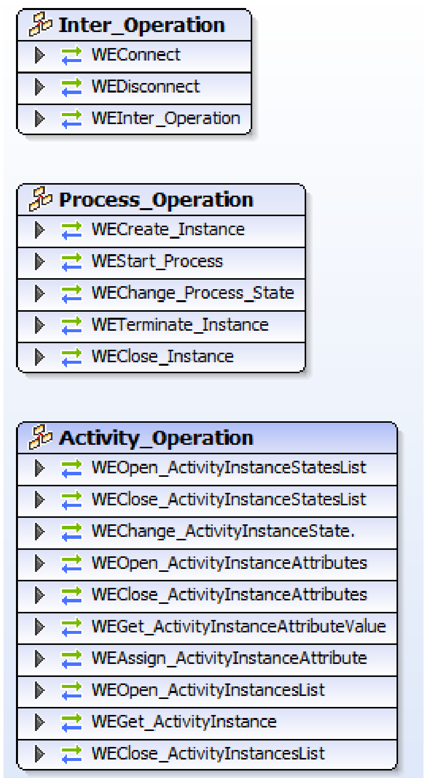


Figure 8-9 Web Service workflow engine PortType and abstract description

The abstract description of workflow engine is composed of PortType of interoperation, process control, activity control, workflow list management, management monitoring and application calling, each PortType is corresponding with a set of Opreation which can realize the function of workflow engine. The link line between the PortType and Operation shows their relation. The operation WEInter_Operation in PortType Inter_Operation is given in figure 8-10 while the Type is omitted, and the data type definition of Type is same with the data type in SOAP message. The Message part describes the input and output message, while input and output message all are SOAP message. The PortType describes the function of workflow engine, while Operation describes the name of operation. Binding describe the combination between "operation and message" and "protocol and data norm", and Service describes the deployment of the operation.

```
<message name="WEInter_OperationSoapIn">
    <part name="parameters" element="s0:WEInter_Operation"/>
</message>
<message name="WEInter_OperationSoapOut">
    <part name="parameters" element="s0:WEInter_OperationResponse"/>
</message>
<portType name="Inter_Operation">
    <operation name="WEInter_Operation">
        <input message="s0:WEInter_OperationSoapIn"/>
        <output message="s0:WEInter_OperationSoapOut"/>
    </operation>
</portType>
<binding name="Inter_Operation" type="s0:Inter_Operation">
 <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
    <operation name="WEInter_Operation">
    <soap:operation soapAction="http://tempuri.org/HelloWorld" style="document"/>
        <input>
            <soap:body use="literal"/>
        </input>
        <output>
            <soap:body use="literal"/>
         </output>
    </operation>
</binding>
    <service name="WE_SERVICE">
        <port name="Inter_Operation" binding="s0:Inter_Operation">
            <soap:address location="http://localhost/WE_WebService/WE1_Service.asmx"/>
        </port>
```

Figure 8-10 WSDL description of workflow engine

8.3 Description of Flexible Workflow based on Web Service

8.3.1 Service Description of Flexible Workflow

The Web Service-based distributed flexible workflow engine need to execute defined flexible workflow model, the formal and graphical expression of flexible workflow model have been given in chapter 6. If the workflow node in flexible workflow model can be encapsulated into Web

Service, the whole workflow model can be a Service process based on Web Service. The fine cross-platform and loose coupling of Web Service can benefit the application of flexible workflow in dynamic engineering design. There are already many researches on workflow description based on Web Service, but whether it is description of workflow node using WSDL or the description of workflow service flow by BPEL4WS (Business Process Execution Language for Web Services) there are normally conducted mainly on static workflow model, there are few researches on flexible workflow based on Web Service both at home and abroad.

The core of flexible workflow description based on Web Service is that the workflow node in the flexible workflow is described as Web Service, the whole flexible workflow is described as the Web Service process composed of Web Services of multi nodes in workflow. Based on WSDL and BPEL4WS, the description of flexible workflow node and flexible workflow service process are studied respectively. Because the flexibility of flexible workflow model is mainly embodied by flexible elements: extensible property, extensible function and dynamic router. And more importantly the flexible workflow model is defined in the dynamic instance-based workflow modelling methods, and the flexible workflow model description based on Web Service should reflect the characteristic of flexible elements and dynamic instance-based modelling method.

For the flexible workflow node description by WSDL, it mainly described for the node in flexible workflow, the extensible property and extensible function are mainly described in description of flexible workflow node. Meanwhile the dynamic router in flexible workflow is composed of previous router and subsequent router, every rule in the dynamic router corresponds with a dynamic control relation which is actually the hidden dependencies of process control relation, so the dynamic router is described by description of flexible workflow service flow. The dynamic instance-based modelling method is realized by the dynamic operation on description document of flexible workflow node and flexible workflow service flow.

In order to better describe the flexible workflow based on Web Service, an EDP instance of custom truck design is given in (b) of figure 7-10, a “carriage design 2” activity is added in parallel with the “carriage design 3” activity, a new activity is added after the above two activity.

As shown in figure 8-8, the Web Service description based on WSDL is mainly implemented by

PortType and Operation. For the Web Service description of flexible workflow node, the extensible function actually is the concrete function of flexible workflow node, it can be realized by PortType and Operation. But there no corresponding elements for extensible property in WSDL, the WSDL should be extended, the flexible workflow nodes description based on WSDL is given in table 8-2.

Table 8-2 The flexible workflow nodes description based on WSDL

ID	Elements	Web Service Description Method
1	Extensible property	Add a **eVariables** elements, including many **eVariable**elements, its type is corresponding with the **Part** elements in **Message** elements
2	Extensible Function	The extensible function can be described by a **PortType** element and many **Operation** elements, the property **DefaultOperation**can be defined in **Operation**, the value of property **DefaultOperation** is "default" means the Operation is default, there are one and only one default Operation
3	Autonomic Object	The autonomic object and flexible workflow activity is described in same WSDL document, the AO is described by a **PortType** element, the Sensor, Effector, Coordinator and Manual can be described individually by **Operation** elements
4	Compositional Activity	Every simple activity in compositional activity is described by a **PortType** element includig many **Operation** elements
5	Relevant Data	Relevant data is described by **Message** element
6	Dynamic Mechanism	The Web Service description document should support the dynamic modification mechanism, in the workflow instance, the AO effector can operate on Web Service description document, including the adding, modifying and deleting of **Evaraible** and **Operation**

The Web Service description of Carriage Design is given in figure 8-11, and the Web Service description of Stress Simulation is given in figure 8-12. Only the PortType and Operation elements is given in figure 8-12 for simplicity.

```xml
<eVariables>
    <eVariable name="designinfo">
        <part name="design staff" type="xsd;integer"/>
    </eVariable>
</eVariables>
<portType>
    <operation name="carriagedrawing">
        <input message="designspecification"/>
        <output message="designdrawing"/>
    </operation>
    <operation name="carriagemodifying">
        <input message="designspecification"/>
        <output message="modificationresult"/>
    </operation>
</portType>
```

Figure 8-11 Web Service description of carriage design

```xml
<definitions targetNamespace="http://supplychain-flexibleworkflow/wsdl/subcontractor"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:plnk="http://schemas.xmlsoap.org/ws/2003/05/partner-link/"
xmlns:lns="http://supplychain-flexibleworkflow/wsdl/subcontractor">
<message name="materialExaminationMessage">
<part name="materialName" type="xsd:string"/>
<part name="materialType" type="xsd:string"/>
<part name="materialProducer" type="xsd:string"/>
<part name="materialAmount" type="xsd:integer"/></message>
<message name="resultMessage">
<part name="result" type="xsd:string"/>
<part name="detail" type="xsd:string"/></message>
<eVariables><eVariable name="examinationStaff">
<part name="staffName" type="xsd:string"/>
<part name="qualification" type="xsd:string"/>
</eVariable></eVariables>
<portType name="materialExaminationPT">
<operation name="ingredientExamination">
<input message="lns:materialExaminationMessage"/>
<output message="lns:resultMessage"/>
</operation>
<operation name="defectDetectionPT">.
<input message="lns:materialExaminationMessage"/>
<output message="lns:resultMessage"/>
</operation></portType>
<plnk:partnerLinkType name="meterialExaminationLinkType">
<plnk:role name="meterialExamination">
<plnk:portType name="lns:MeterialExaminationPT"/>
</plnk:role></plnk:partnerLinkType>
</definitions>
```

Figure 8-12Web Service description of Stress Simulation

The extensible property of Stress Simulation is composed of examiner name and his qualification, and the extensible function is composed of “stress computation” and “stress graphical simulation” while “stress computation” is default operation. The description of AO is relatively fixed, its description is not given. In order to better describe the Web Service flow of this EDP, the main PortType description of each activities of the EDP in figure 7-10 is given in table 8-3.

Table 8-3 Web Service simple list of custom truck EDP

ID	Activity	Web Service Description
1	Conceptual Design	<portType name="conceptual design PT"> <operation name="brainstorming"></operation> </portType>
2	Carriage Design	<portType name="Carriage Design PT"> <operation name="designspecificationanalysing"></operation> </portType>
3	Chassis Design	<portType name="Chasis Design PT"> <operation name="chasisdrawing"></operation> </portType>
4	Stress Simulation	<portType name="Stress Simulation PT"> <operation name="stresscomputating"></operation> <operation name="stresssimulating"></operation> </portType>
5	Assembly Design	<portType name="Assembly Design PT"> <operation name="Assemblydesigning"></operation> </portType>
6	Intensity Verification	<portType name="Intensity Verification PT"> <operation name="Intensitycomputing"></operation> </portType>

8.3.2Process Description of Flexible Workflow

The Web Service is independent function entity described by WSDL, the flexible workflow process specify a set of possible sequence of Web Service operations, the shared data, design partner in the design process and the role of the design partner. The flexible workflow process based on Web Service can be described by process description language. BPEL4WS（Business Process Execution Language for Web Service）is proposed in August, 2002, and is the combination of WSFL（Web Services Flow Language） in IBM and XLANG（XML LANGuage）in Microsoft. Different with the standards released by WfMC, BPEL4WS is not a standard on definition of business processes, but specifies business processes and their relations with Web Service. BPEL4WS represents the newest standard on Web Service business process description, but it mainly describes structured process with poor flexible process description. It can be known from literature review only few researches on flexible workflow based on BPEL4WS. The flexible workflow description based on BPEL4WS is given after analyzing the elements in BPEL4WS.

In BPEL4WS (Andrews, Curbera et al 2007), Partner Link Types defines the relation between process and service as well as their respective roles, Partner Links defines the link between process and service, while Partner defines the set of Partner Links. BPEL4WS defines rich data operation methods, among them, Variables is used to store the information in Web Service process, this information includes the related information in process and information exchanged between services. The type of Variables can be the type in WSDL, XML Schema simple type or XML Schema elements. In addition to the general expression, BPEL4WS defines the Boolean-valued expression, Deanline-value expression and Duration-value expression, these expressions are defined by XPath() 1.0 (Clark and DeRose 2006). The data relation between processes is expressed by correlation Sets.

The main functions of BPEL4WS is mainly realized by activity element, the activity element can be classified into basic activity and structured activity, as illustrated in table 8-4. The calling of Web Service is mainly realized by basic activity while the logic relation between Web Service callings is realized by structured activity. In flexible workflow model, the control relation is made up of static and dynamic arc, static arc can be described by structured activity in BPEL4WS. But for the expression of dynamic arc, there is no direct element in BPEL4WS. According to the static characteristics of dynamic control arc, the dynamic arc can be changed into static arc in the running of workflow instance, and the dynamic arc is determined jointly by subsequent rule and previous rule of activities.

Table 8-4 BPEL4WS activities

Basic Activity	Structured Activity
<invoke>:calling Web Service	**<sequence>**:sequence activity
<receive>:receiving Web Service	**<switch>**:condition activity
<assign>:data exchanging	**<while>**:cycle activity
<reply>:replying Web Service	**<flow>**:activity defining concurrency and synchronization
<throw>:exception handling	**<link>**:control link
<empty>:null activity	**<pick>**:waiting the event and activating activity
<wait>:waiting	**<scope>**:defining activities Series

In BPEL4WS, <link> defines the control relation of activities tacitly, the source Web Service node of <link> is defined by <source> while the target node of <link> is defined by <target>. In this view, the dynamic router can be defined as a set of several <link>, each group of <link>s corresponds with a routing rule, so the previous router rule is composed of a group of <source> and <link> elements, the subsequent router rule is compose of <target> and <link> elements. According to the definition of BPEl4WS, <source> includes the property of "transitionCondition", it can be expressed by Boolean-valued expression defined by XPath 1.0. <sourceGroup>和 <sourceGroups> is added in define previous router rule, and <targetGroup> and <targetGroups> is added to define subsequent router rule, the flexible workflow description based on BPEL4WS is given in table 8-5.

According to the method given in table8-5, part of the Web Service description documents of custom truck EDP is given in figure 8-13, description of flexible workflow is mainly described in the documents, the detail of Variables and exception handling is not given in the documents, the annotation part of the documents shows the flexible change in contrast to the static flow.

Table 8-5 Web Service description of flexible workflow process

ID	Relevant Element	Web Service Description Method
1	Dynamic Router	The dynamic router is described by structured activity, such as **<sequence>**, **<switch>**.
2	Previous Router	A**<sourceGroup>**elements including many <source> elements is a group to describe a rule,**<sourceGroup>** includes "defaultGroup" property, if the property value is "default", it means that this **<sourceGroup>** is the default rule,**<source>** element include "transitionCondition" property; the previous router can be described by **<sourceGroups> elements including many <sourceGroup>** elements,**<sourceGroups>**element, which describes ECA rules, includes "transitionCondition" property.
3	Subsequent Router	Similar with previous router, the **<targetGroup> and <targetGroups> elements have samilar description method with <sourceGroup> and <sourceGroups> elements**, the difference lies in that there are no "transitionCondition" property in **<target>**element, and the "transitionCondition" in **<targetGroups>** is used to determine if the default **<targetGroup>**can execute subsequent activity.
4	Compositional Flow	The compositional flow can be encapsulated into Web Service, the inner functions or activities can be described as Web Service at first, then the multi Web Service is described by BPEL4WS.
5	Dynamic Mechanism	Similar with the description of activity Web Service, the flow description of flexible workflow should be operated by AO in runtime, including the operation on **<sourceGroup>**and **<targetGroup>**, as well as on **<link>** and **<partnerLinkType>**.
6	**<link> and other related element**	According to the dynamic change of router rules, the relevant **<link>** elements and **<partnerLinkType>** should be added or deleted dynamically.

The flexibility of flow also is reflected in dynamically calling and binding of different operation, as well as the modification of eVariables elements. For example, in calling of the “chassis design” activity, the different drawing software can be used, and the design cycle time can also be modified.

```xml
<receive partnerLink="conceptual design"
portType="lns: conceptualDesignPT" operation="conceptualDesign"
inputVariable="designrequirements"
outputVariable="conceputalDesign" createInstance="yes">
<sourcegroups><sourcegroup>
<source linkName="conceptualDesign-to-carriageDesign1"/>
<source linkName="conceptualDesign-to-carriageDesign2"/>
<source linkName="conceptualDesign-to-chasisDesign"/>
</sourcegroup></sourcegroups>
</receive>
<invoke partnerLink="carriage design 1" portType="lns: carriageDesign1PT"
operation="carriageDesign" inputVariable="conceptualdesign"
outputVariable="carriagedesignspecification">
<targetgroups><targetgroup>
<target linkName="conceptualDesign-to-carriageDesign1"/>
</targetgroup></targetgroups>
<sourcegroups><sourcegroup>
<source linkName="carriageDesign1-to-stressSimulation"/>
</sourcegroup></sourcegroups>
</invoke>
<invoke partnerLink="carriage design 2" portType="carriageDesign2PT"
operation="carriageDesign" inputVariable="conceptualdesign"
outputVariable="carriagedesignspecification">
<targetgroups><targetgroup>
<target linkName="conceptualDesign-to-carriageDesign2"/>
</targetgroup></targetgroups>
<sourcegroups><sourcegroup>
<source linkName="carriageDesign2-to-stressSimulation"/>
</sourcegroup></sourcegroups>
</invoke>
<invoke partnerLink="chassis design" portType="chassisDesignPT"
```

Figure 8-13 Web Service description of flexible workflow process

8.4 Distributed Flexible Workflow Execution Algorithm

8.4.1 Execution Algorithm of Distributed Flexible Workflow Engine

The instance of flexible workflow is realized by workflow engine interpreting flexible workflow model. Based on the Web Service-based distributed workflow engine and flexible workflow description, the flexible workflow execution algorithm is given. The execution of flexible workflow is implemented by the collaboration of multi distributed workflow engines. The workflow engine interprets and activates the flexible workflow model, executes the flexible operation on flexible workflow and navigates the running of flexible workflow, and it finishes the creation, execution and management of flexible workflow instance by the interaction of users and application program. The execution of distributed flexible workflow is explained in figure 8-14 by the interaction of three distributed workflow engine, and the workflow sub-process execution algorithm is illustrated by the collaboration of Web Services in workflow engine. After the flexible workflow instance is started, the AO instance corresponding with the activity is started as well, and the AO instance can operate on the flexible elements in flexible workflow instance to realize the intelligence of flexible workflow by the autonomic computing and collaboration with other AOs. It is worth noting that the instance in the figure is destroyed after all instance is executed ensuring that all the AO instance in activity is always active to satisfy the requirement of AO collaboration.

The key of flexible workflow execution algorithm is the execution on flexible elements, an execution algorithm on extensible property, extensible activity and dynamic router is given in table 8-6.

Table 8-6 Execution algorithms on flexible elements

ID	Flexible Element	Execution Algorithm
1	Extensible Property	[1] AO dynamically modify extensible property according to the running of flexible workflow, including adding, deleting of extensible property and modification of extensible property value; [2] Workflow engine reads the default extensible property and its value; [3] Workflow engine calls the extensible property value;
2	Extensible Function	[1] AO dynamically modify extensible function according to the running of flexible workflow, including changing default extensible function, deleting or adding extensible function, and modify the content of

		extensible function; [2]Workflow engine reads the content of default extensible function, including the PortType, Operation and Operation binding; [3] Workflow engine adds the PortType information into workflow list manager; [4] Workflow list manager calls application according to the PortType information;
3	Dynamic Router	[1] AO operates on dynamic router according to the running of flexible workflow, including adding, deleting or modifying dynamic router as well as changing default dynamic router; [2] Workflow engine reads the subsequent router of activity, i.e., the **<sourceGroups>**information when executing a activity is finished; [3] Workflow reads the **<sourceGroup>** in **<sourceGroups>**; [4] Workflow engine computation the activation condition in **<link>**element according to the rule in **<sourceGroup>**, then send activation request to satisfied <link> element; [5] Workflow engine read the previous router**<targetGroups>** of activity which is pointed by above **<link>** element; [6] Workflow engine fetchs **<targetGroup>** in **<targetGroups>**; [7] Workflow engine judges if all of the activation request from **<link>**elements according to the router in **<targetGroup>**; [8] Workflow wait until all request is received; [9] If all request is received, then workflow engine judges the activation condition in**<targetGroups>**, if the condition can be satisfied, then execute the activity; [10] When the activity execution is finished, then workflow engine turn to procedure 1 until all activity in instance is executed;

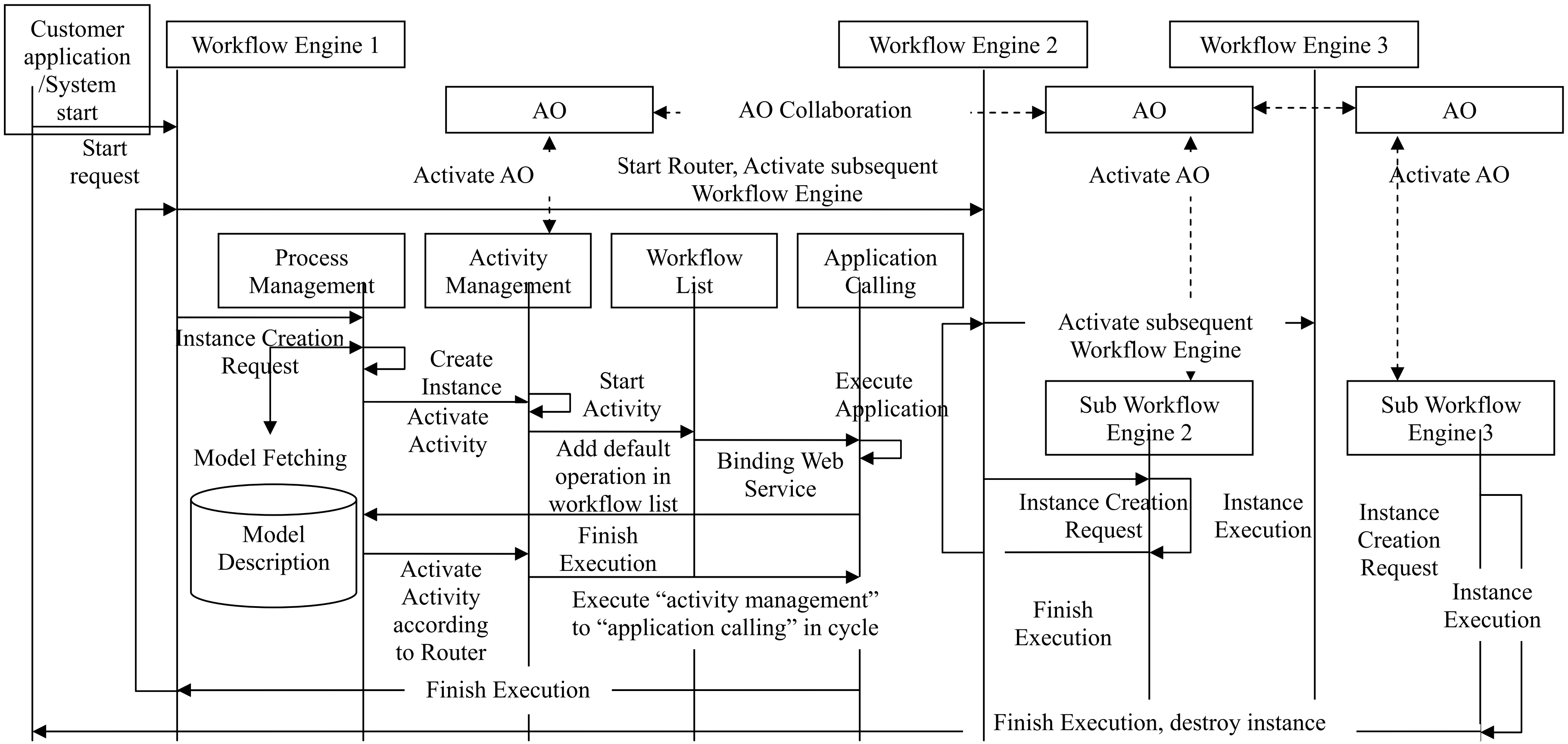


Figure 8-14 distributed flexible workflow execution algorithm

8.4.2 Dynamic Extension Algorithm of Flexible Workflow

In flexible workflow supporting dynamic engineering design process, some situation, such as adding new design activity, changing design process, and changing design process in design group according to the adjustment of general design process, are happened very often in the flexible workflow instance. These situations will cause three key problems, namely, adding nodes dynamically, deleting nodes dynamically and refining nodes dynamically. Based on the flexible workflow execution algorithm, the dynamic node adding algorithm, dynamic node deleting algorithm and dynamic node refining algorithm is studied.

For the dynamic node adding, it is actually discovery and dynamic calling on flexible workflow node described as Web Service. The discovery of flexible workflow node is realized by UDDI. After discovering needed calling Web Service, the related nodes should also be contacted in order the new node can be added. It is similar in the dynamic deleting node, the related node should be contacted and the new link relation should be re-established after the node deleting. The workflow logic relation should be figured out before the node is added or deleted. Some of the typical logic relation of node adding or deleting is illustrated in figure 8-15, (a) is sequence logic, (b) is and logic, (c) is or logic, (d) is loop logic. The grey part in figure is the workflow node may be added, the number in node is the serial number of workflow node, among them,

1) there only one possibility to add node for (a);

2) for (b), there are 4 possibility, situation 1 is equal to situation 2; for situation 3, the added node is in one branch of OR logic, the new node is only need to be added in this branch without considering another branch, so the situation 1, 2 and 3 all can be converted into sequence logic; for the situation 4, only one branch need to be added without disconnecting former link;

3) for (c), there are 4 possibility, their processing method is very similar with (b), only the difference of logic relation should be considered;

4) for (d), situation 1 and 2 all can be converted into sequence logic, only the adding sequence of node 1 and node 2 should be considered.

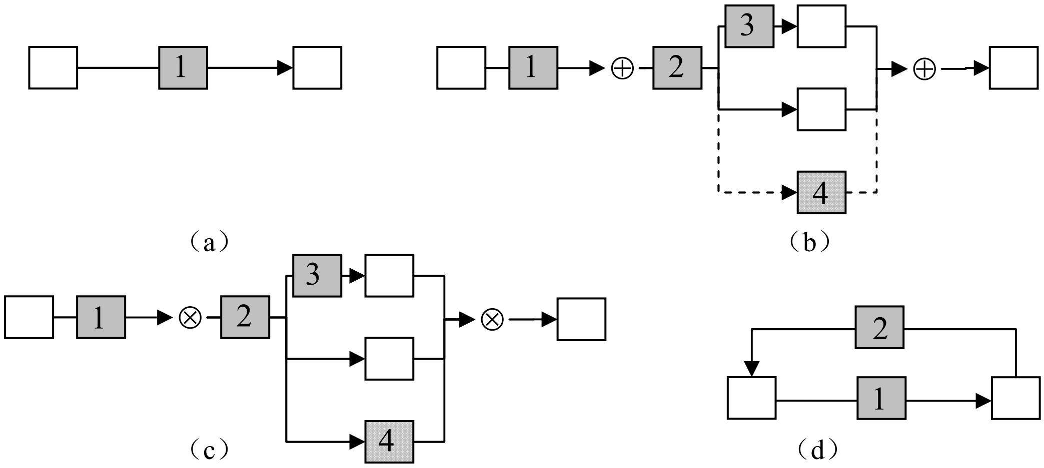


Figure 8-15some of the typical logic relation of node adding or deleting

Although figure 8-16 describes the node adding in typical logic, but if the grey node in the figure is the node to be deleted, the figure can also be the description of deleting node. So the node deleting algorithm is similar with node adding algorithm.

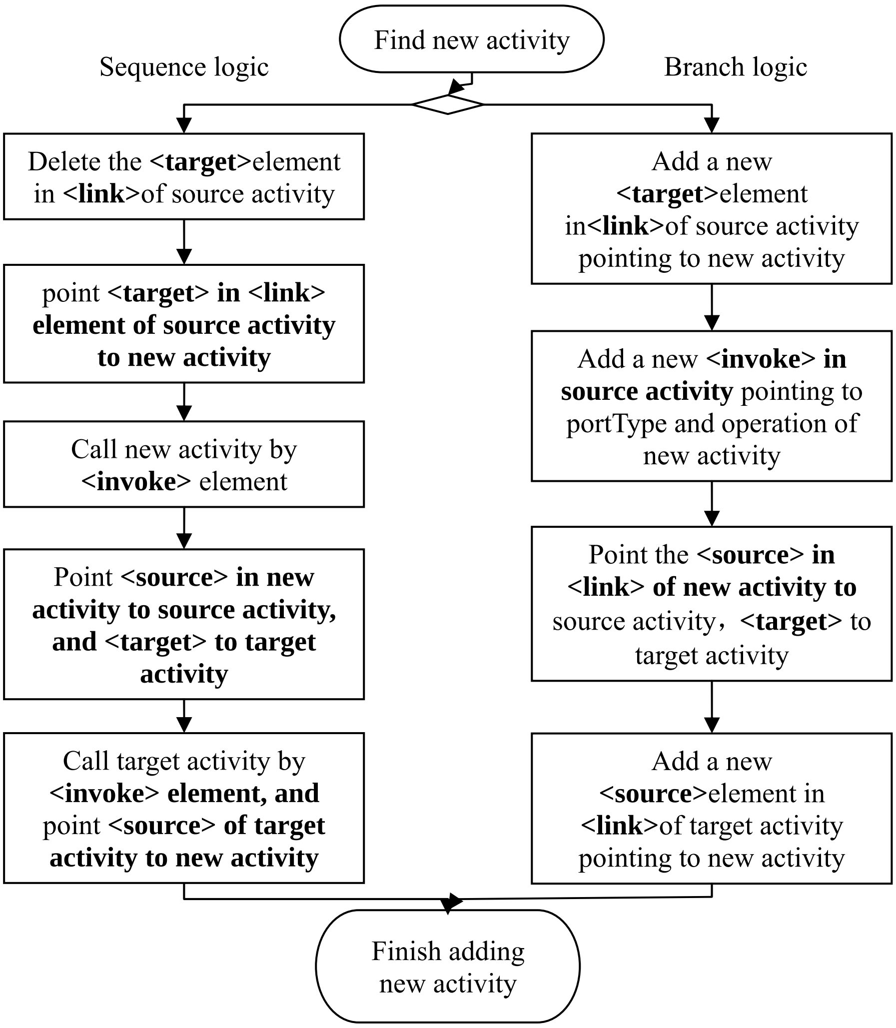


Figure 8-16 Flexible workflow dynamic node adding algorithm

Then it can be concluded that the node adding/deleting algorithm in different logic relation can be got from the node adding/deleting algorithm in sequence and branch logic. The dynamic node adding algorithm is given in figure 8-16.

For the dynamic node refining, the hierarchical activity model has been given in Chapter 6. According to this model, the dynamic node refining should be done in following way: the whole activity is described as Web Service by elements such as PortType and Operation, the function is described as Web Service flow of sub nodes, the algorithm is given in figure 8-17.

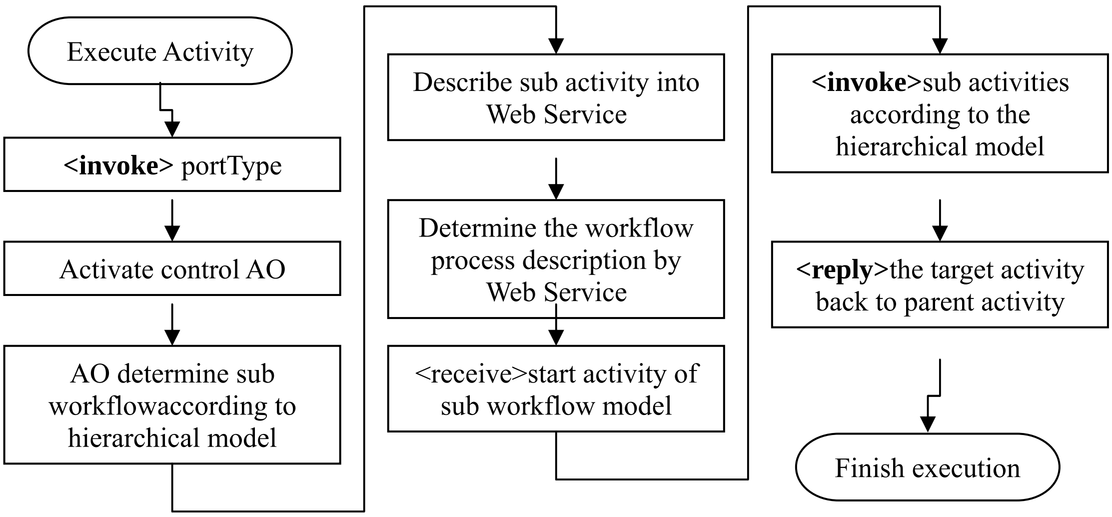


Figure 8-17 Flexible workflow node refining algorithm

8.5 Chapter Summary

Based on brief introduction of Web Service, a distributed flexible workflow engine framework is proposed in this chapter, the SOAP and WSDL in applied in distributed flexible workflow design. Aiming at the characteristics of flexible workflow and Web Service, the WSDL and BPEL4WS is extended, and a flexible workflow description method is proposed based on extended WSDL and BPEL4WS, and the description instance of flexible workflow is given to demonstrate the proposed method. Finally a distributed flexible workflow execution algorithm is given in flexible workflow execution and extending.

Chapter 9 Flexible workflow Management Prototype System supporting Engineering Design Process Management Implementation –a case study

Based on above theory research, a flexible workflow management prototype system supporting dynamic engineering design process is developed using Microsoft .Net; the proposed theory is demonstrated by the application of the prototype in a medium-sized customised truck company (The Customised Trucks). The Customised Trucks is a typical Make-to-Order (MTO) company, which manufactures products designed for a specific customer. The design process and the products are very complex while the batch size of the product is low. The design process is highly dynamic because it changes each time when there is a new order from customers.

The main products for the company include the different kinds of customised trucks, such as the heavy-duty dump trucks and flat platform trailers. Nowadays, the market competition becomes fiercer, so an efficient EDP is very important for the company to reduce the design cycle time and improve the quality to win the market competition. But the design processes for different trucks are different, so the company typically takes long time to reconstruct and manage its design processes. The customized truck design process should be flexible enough to support the frequently changed EDP. Moreover, the new in-coming design orders of customized trucks often presents many new design requirements for the design team in the technical center of the customised trucks, new design methods, new design tools, as well as new design process emerge in the EDP of customised trucks, the intelligent changing of the customised engineering design process should aslo be supported.

9.1 Introduction to the Prototype System

Flexible Workflow Management Prototype System supporting Dynamic Engineering Design Process (FWMPS-DEDP) is a distributed flexible workflow management system supporting dynamic engineering design process management. It integrates flexible workflow modelling and execution with the core of multi-AOs flexible workflow model, and it can realize the intelligence by the multi AOs intelligence algorithm and their collaboration. FWMPS-DEDP supports the graphical modelling to define the model of dynamic engineering design process, and the design process can be dynamically created in FWMPS-DEDP. Furthermore the intelligence of a design process can be improved by the AO embedded in its design activity. The merits of this prototype is that by introducing the flexible workflow and multi AOs, not only the rapid re-constructing and dynamic adaptation of the design process can be realized, but also the traditional method to construct design process management system by programming is changed. FWMPS-DEDP can support the dynamic and cross-enterprise engineering design process management in a satisfied way.

The development environment of FWMPS-DEDP is windows XP/Microsoft VS .Net 2008 with C# language. The running environment is Windows/Microsoft .Net Framework 2.0, Microsoft IIS （Internet Information Server） is the web server. The database management system used is Oracle 9i, and the system management and user-interface is operated via IE browser.

9.2 Architecture and Modules of Prototype System

9.2.1 Architecture of Prototype System

Objectively speaking, workflow management system is a very large and complex system, and it is unrealistic to develop a complete workflow management system in limited time [208]. Based on many workflow management systems and open-source project, the modeling tool and light-weight workflow engine is developed, the system architecture of FWMPS-DEDP is given in figure 9-1.

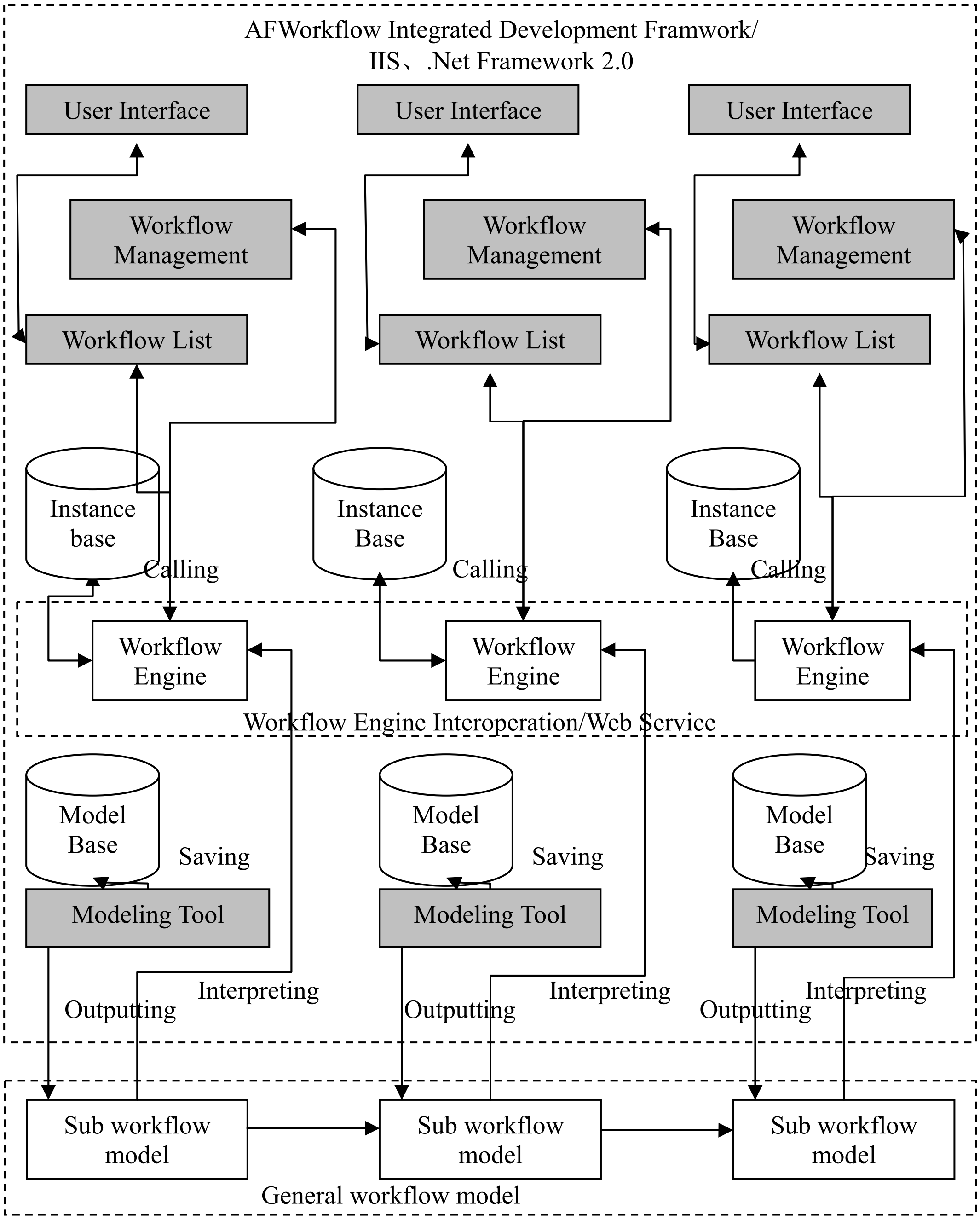


Figure 9-1 The system architecture of FWMPS-DEDP

The general flexible workflow mode is given in the dashed box of figure 8-1, and the general workflow can be decomposed into several sub workflows which is corresponding with the multi design activity or its sub design activity. The integrated development environment of FWMPS-DEDP is given in upper dashed box in the figure. Corresponding with the decomposed sub workflow, the integrated development environment is composed of FWMPS-DEDP modules deployed in number of IIS server. For each FWMPS-DEDP module, it is composed of modelling tool, workflow engine, workflow list management, workflow management tool and user interface, as well as workflow model and instance library. The flexible workflow model (description) is the link of different FWMPS-DEDP modules, different modelling tool define different part of workflow model jointly, while the execution link of FWMPS-DEDP modules is realized by the Web Services interoperation between the workflow engines. The design of flexible workflow engine based Web Service has been described in detail in chapter 8, it not illustrated in the figure separately.

When FWMPS-DEDP is running, one of its module (normally in leading design company/team) defines the general workflow model at first, then the general workflow model is decomposed into several sub model; after that the definition and interface detail of sub model is assigned into each design company/team, the inner detail of sub model is detail in each design company/team, then the sub model is assembled into final dynamic engineering design process, and the decomposed sub model is stored in the model library. The decomposed sub model is called and executed by workflow engine, the related workflow task is interacted with users by workflow list and user interface, the workflow management tool is in charge of the maintaining and monitoring of workflow engine and AO. When one sub model is executed, through interaction of workflow engines, the next sub model is executed by its workflow engine according to control logic of general workflow model.

9.2.2 System Modelling Tool

Flexible workflow mode is the foundation of realizing flexible workflow management system, and modelling tool is the main function module of FWMPS-DEDP. Modelling tool provides graphical modelling method, and user can defines the engineering design process according to the graphical elements in chapter 4. There are altogether two modelling interface provided, one is the interface for the general model, and another is for the sub model by giving definition of sub model and detail of interface. Besides the regular workflow node and control logic modelling, for each flexible workflow node, the extensible property, extensible function and workflow application data of node can be defined in modelling tool, and the AO can be defined and AO knowledge can be managed. By defining the dynamic router of flexible activity, the dynamic control logic of flexible workflow can be defined in workflow modelling tool.

The modelling tool of FWMPS-DEDP also includes the management and maintaining of workflow model, searching and browsing workflow model, the saving of workflow model by XML format. The interface of workflow modelling tool is given in figure 9-2, it is composed of title area, menu area and work area. In the left part of the work area is the control column, and user can drag the control to define the flexible workflow model. The sub module of modelling tool includes:

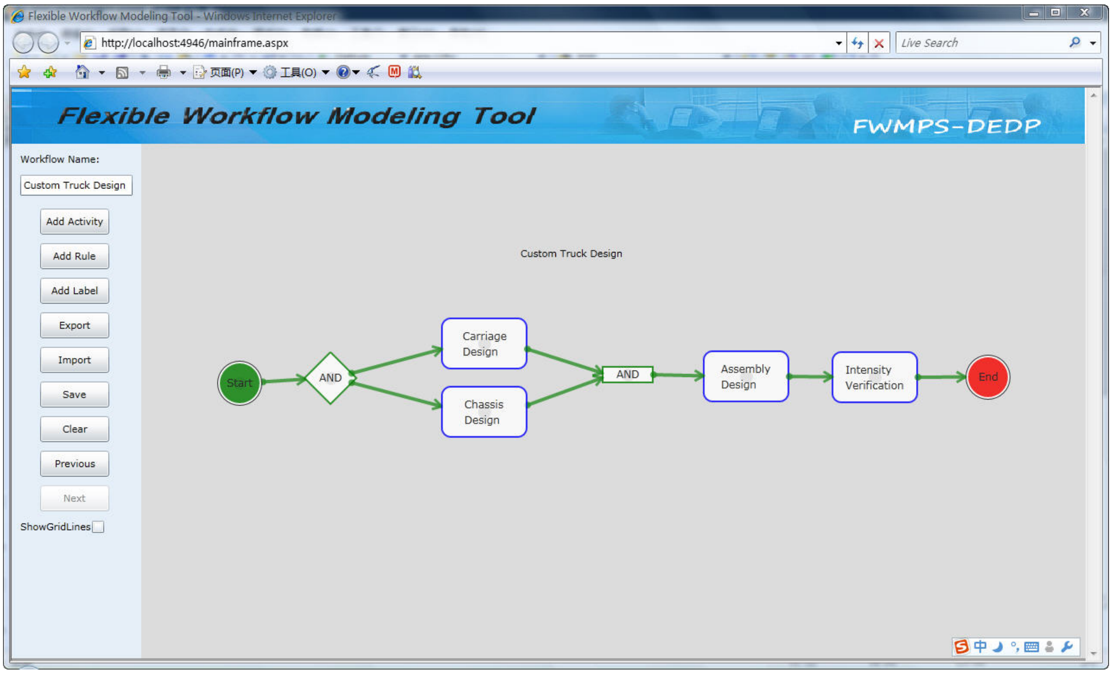


Figure 9-2 FWMPS-DEDP modelling tool

1) Property module: defining the extensible property of current node;

2) Operation module: defining the extensible function of current node;

3) Router module: defining the dynamic router of current node;

4) AO module: defining the set of intelligence algorithm and knowledge;

The sub-workflow model modelling tool is given in figure 9-3, the definition of current node in general workflow model is the basis of sub workflow model modelling.

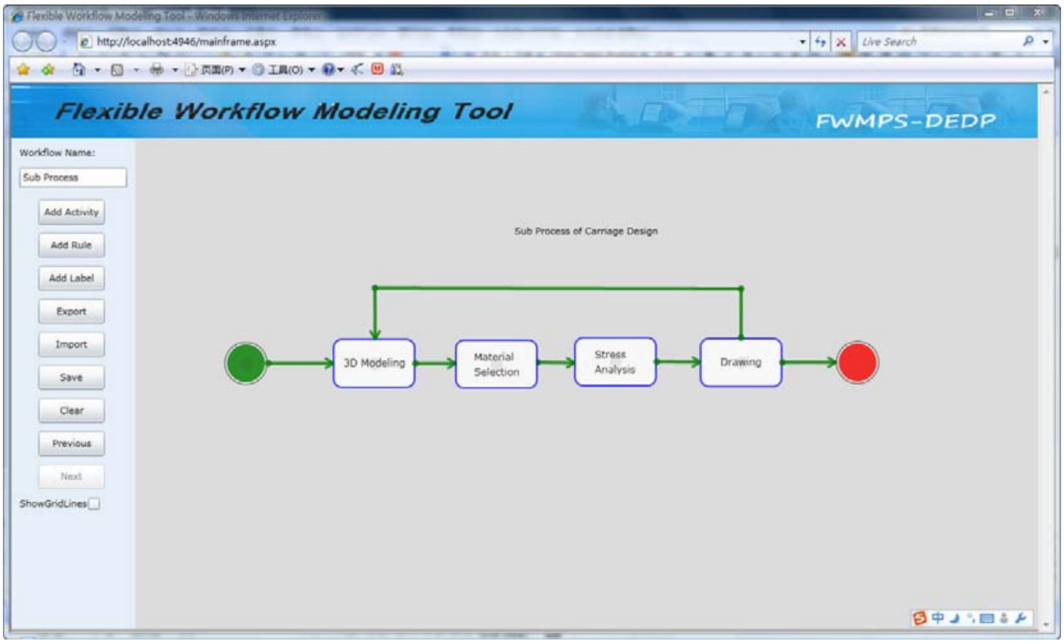


Figure 9-3 FWMPS-DEDP sub model modelling

The flexibility of the flexible workflow model defined by modelling tool is realized by the flexible elements. The extensible property defining interface is given figure 9-4.

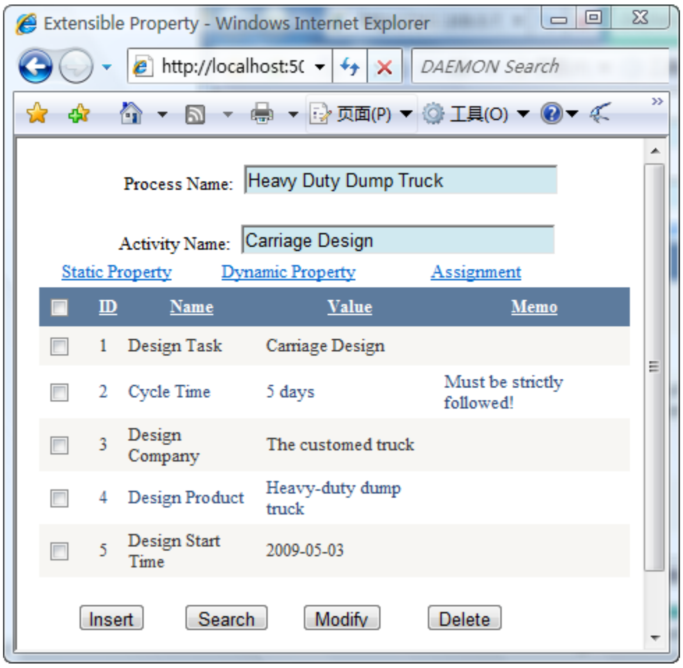


Figure 9-4 extensible property defining interface

In figure 9-4, the static and dynamic extensible property can be defined, and the operator of activity is assigned according to the department, role and operator mode.

The extensible function defining interface is given in figure 9-5, the detail of extensible function can be configured.

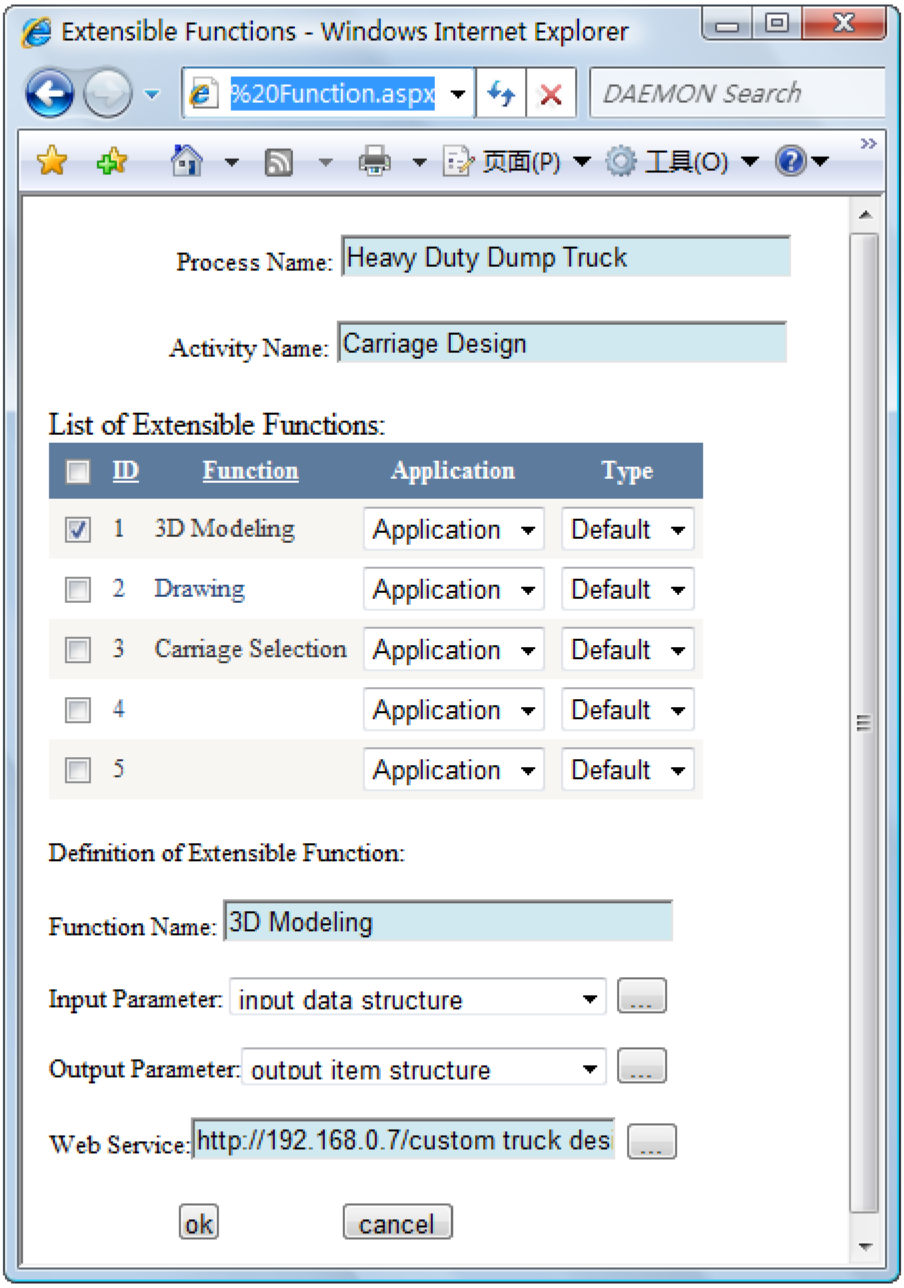


Figure 9-5 Extensible function defining interface

The intelligence of flexible workflow is realized by AO, according to the AO knowledge expression in 7.2.2, the AO reasoning knowledge edit interface is given in figure 9-6. In the prototype system, the AO reasoning program is not loaded dynamically, instead it is load manually in practical application.

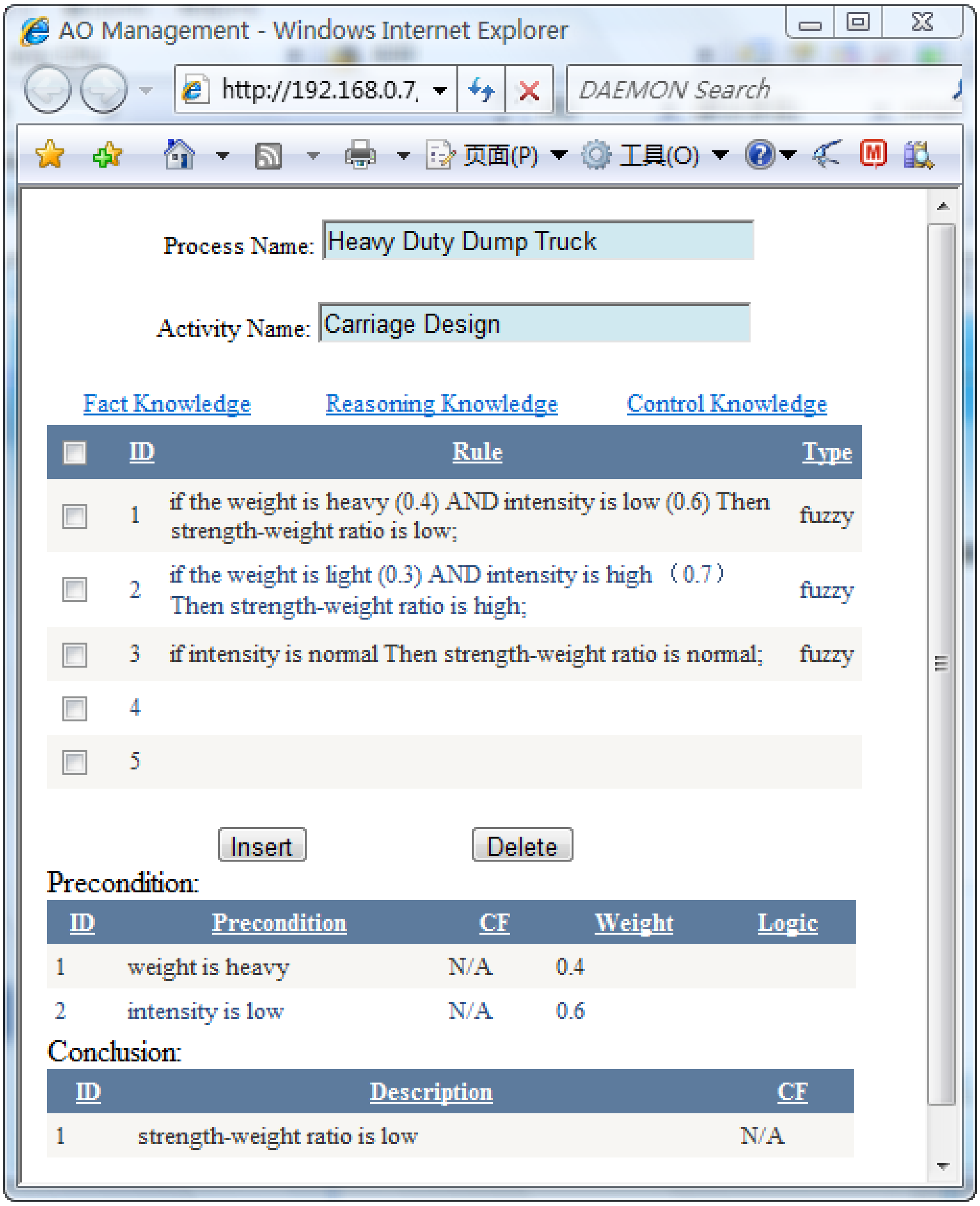


Figure 9-6 AO knowledge edit interface

9.2.3 Workflow List Management and User Interface

Workflow list is the interface of workflow engine and user, the task of flexible workflow instance is loaded into the workflow list, and user acquires task via workflow list to execute the task. The workflow list in FWMPS-DEDP is given in figure 9-7. The left part is the workflow model and its instance while the right part is the workflow list of the instance. User of FWMPS-DEDP can get into the task process interface by pressing the task processing button in the workflow list interface.

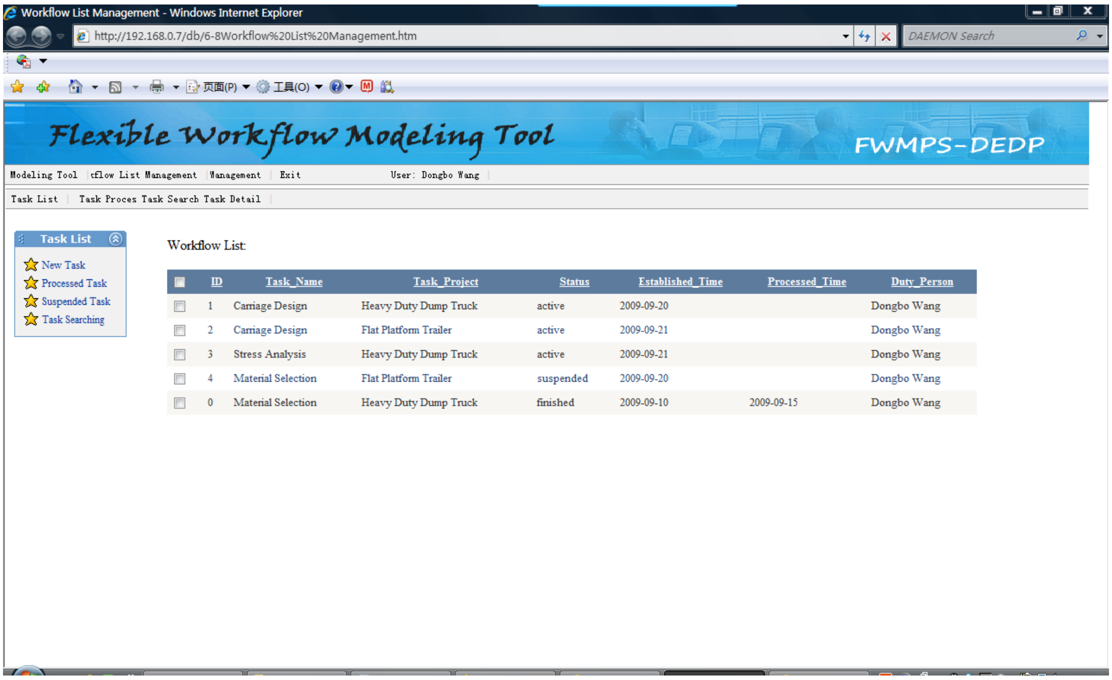


Figure 9-7 The workflow list interface

One task processing interface in FWMPS-DEDP is given in figure 9-8, the core of the interface includes the description of task, input data, output data and operation button. Because the task operation, input and output data is different in different tasks, the task processing interface is normally created according to the definition of task and its related data. Specifically the workflow instance variables and operation defined by user is written into ASPX document by Write method in Response object built-in ASP.Net together with the JavaScript. The actually practice at present is that the task processing interface template is design before hand by commonly used webpage development tool, such as VS.Net and FrontPage, the template is mainly design in its layout and description information; based on the template, the instance variables is written into the template dynamically; for some complex task process interface, the task process interface is even needed to be customized.

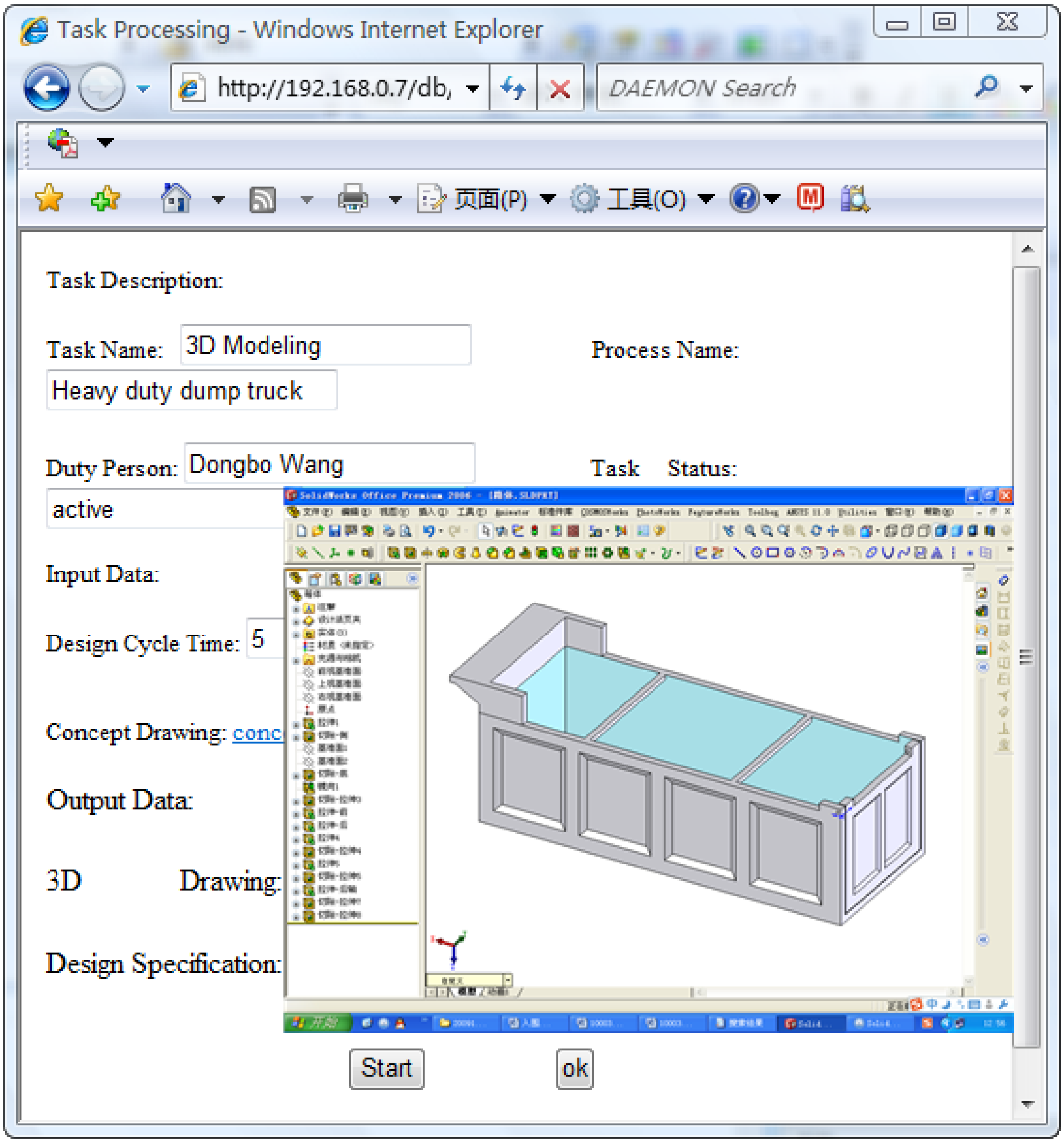


Figure 9-8 Task processing process

9.2.4 Workflow Management Tool

The workflow management tool mainly is responsible for the management of workflow system and the maintaining of workflow instance running. The workflow management tool of FWMPS-DEDP mainly have following functions, the workflow management tool is given in figure 9-9.

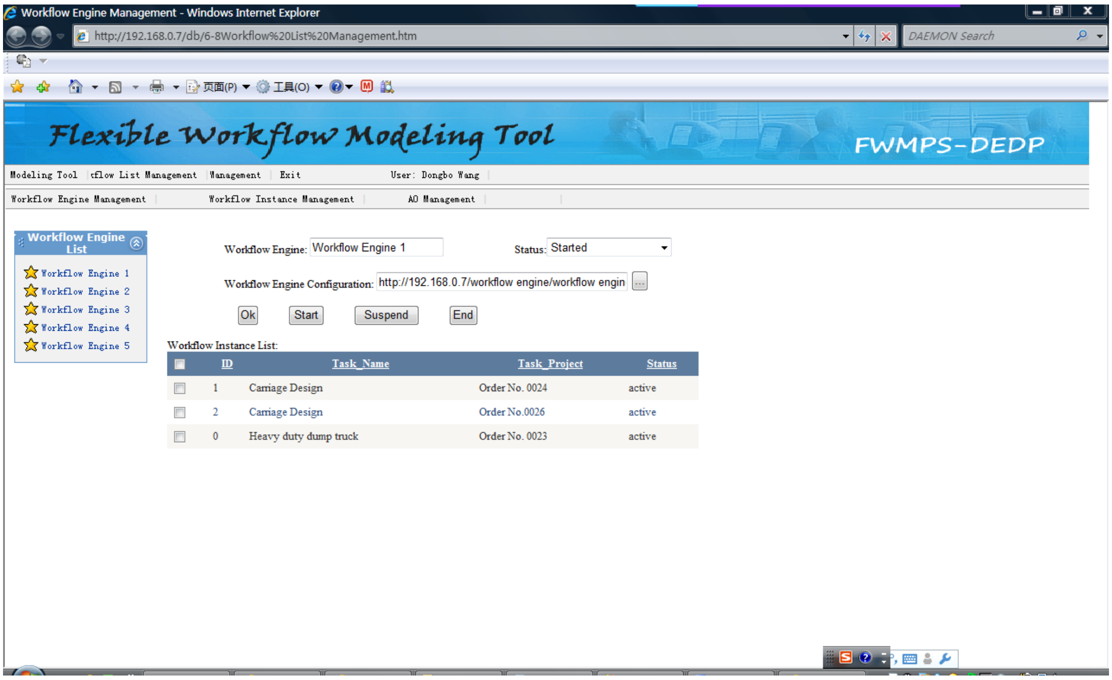


Figure 9-9 workflow engine management interface

1) Workflow engine management: including the starting and suspending of workflow engine, workflow engine running status querying;

2) Workflow instance management: including the running monitoring of workflow instance, workflow activity running monitoring;

3) AO management: including the starting and suspending of AO, and AO operation querying;

The left part of figure 9-9, gives he workflow engine list of FWMPS-DEDP, the detail of current workflow engine, including the name of workflow engine, configuration file of workflow engine, is given in right part. User can modify the status of workflow engine. The left part of interface givens the workflow instances list running on the current workflow engine.

The workflow instance management interface is given in figure9-10, the classified workflow instance is given in left part, the actually running process of current instance is given in right part, and the highlighted node means the node is being executed. The flexible change can be seen clearly in the difference between figure 9-2 and 9-10, the concrete change of flexible elements in the current node can be shown by pressing the detail browsing button.

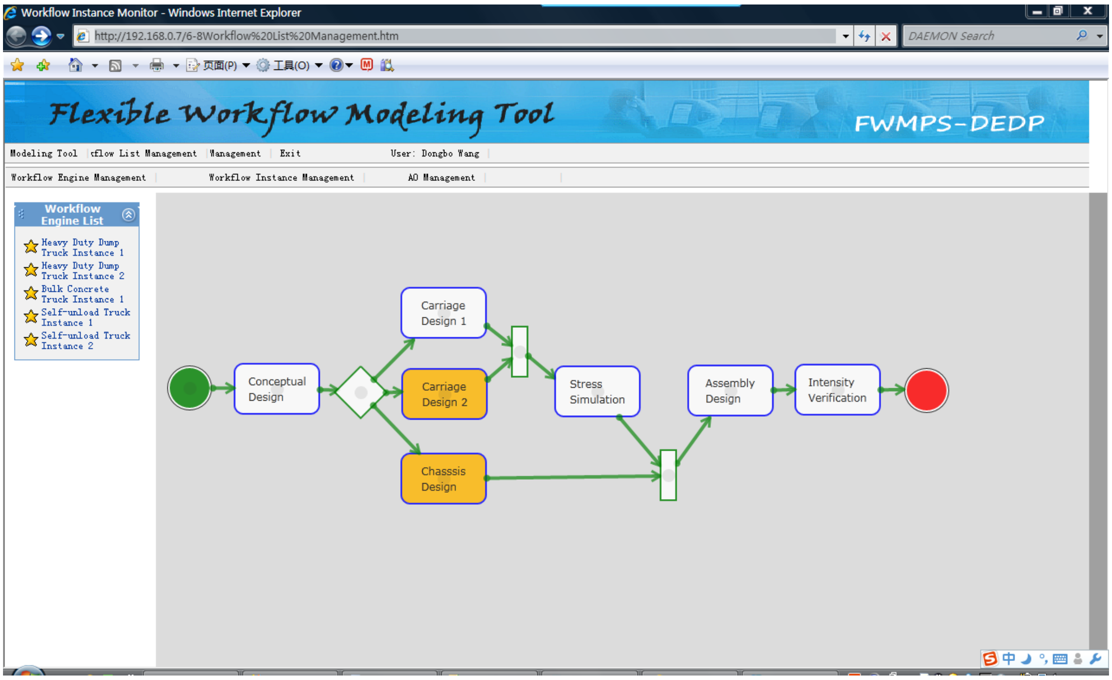


Figure 9-10Workflow instance management interface

The flexible change can be shown in the flexible workflow instance management interface while the change is realized by the operation on flexible elements; the AO management interface is given in figure 9-11.

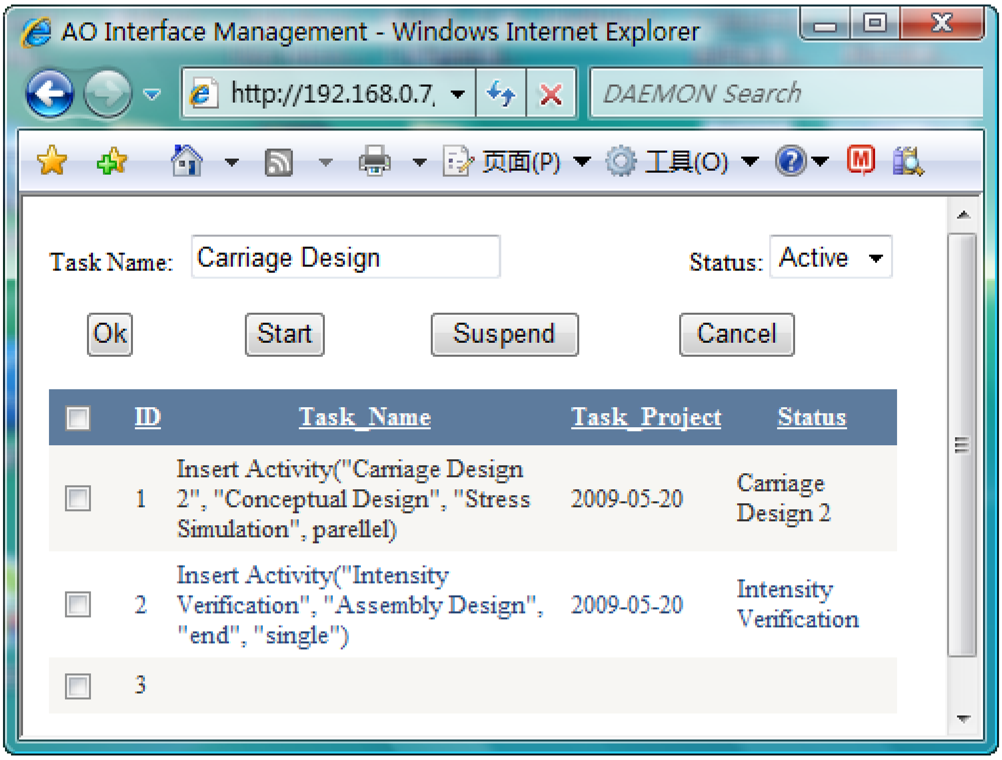


Figure 9-11 AO management interface

9.2.5 Database Design of Prototype System

In the architecture of Prototype system, the model library and instance library is the basis of modelling tool and workflow engine. A fine database design is the grantee for flexible workflow modelling. The multi AOs flexible workflow includes flexible elements, such as extensible property, as well as many interrelated entities such as AO. In dynamic instance-based flexible workflow modelling mode, the instance and workflow elements and their relation in instance is changed dynamically, a UML (Unified Modeling Language) class diagram of flexible workflow model and instance in dynamic engineering design process is given in figure 8-14. Each set of tables same with tables in the diagram should be created in each design node in the dynamic design process.

As shown in figure 9-12, the model table is the core of the whole database, the general description information of the model is stored in the model table. There is one to many relations between model table and node table, the description of node information is stored in node table.The information in node table can be described further by sub tables which have many to one relation with node table, such as extensible property table, extensible function table, previous dynamic router table, subsequent dynamic router table, AO table, AO knowledge table, input and output table. The knowledge rules is stored in AO knowledge table which has many to one AO table.

The general description of instance created in running of flexible workflow model is stored in instance table which has many to one relation with model table. Because one model can be instanced into many instances, each instance is composed of many sub nodes and their sub elements; meanwhile each instance is created dynamically, the elements in each node is often changed, if the information of all instance is stored in a single table, the data amount of the table will be very large, and the data in different instance will be accessed in this single at the same time, there must be data access problem. As explained in chapter one, each instance can be described by many instance nodes and their relations which are instanced from node table and its sub tables. It is realized in following way; when an instance is created in FWMPS-DEDP, a set of instance node table and its sub tables is created by PL/SQL program in Oracle with node table and its sub tables as template; when the table is created successfully, the name of table and columns are stored in a table titled instance data dictionary. Instance data dictionary is core of management for workflow instance in FWMPS-DEDP, the table and column names of current instance can be acquired by the instance name from the instance data dictionary, then the dynamic SQL procedure can be created according to the table and columns, the instance can be created by executing the SQL procedure.

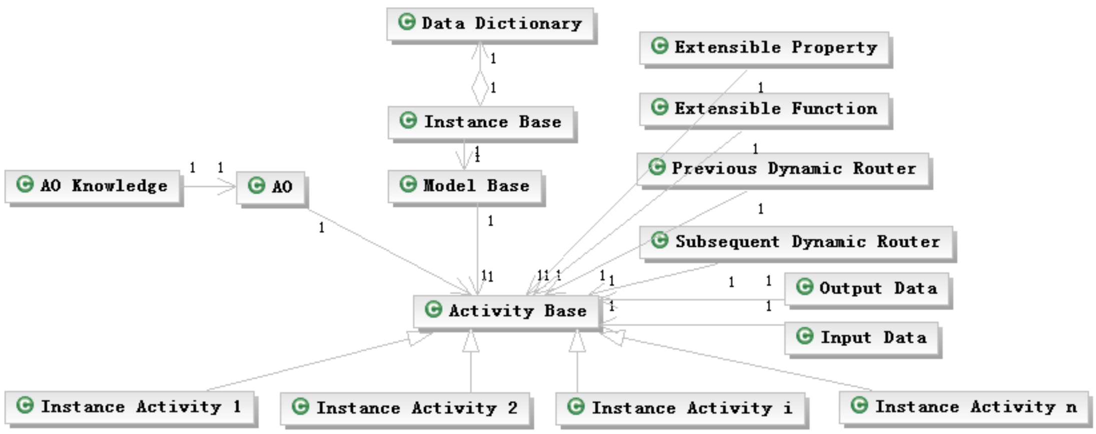


Figure 9-12The database UML class diagram of FWMPS-DEDP

9.3 Application of Engineering Design Process Management Prototype System

9.3.1 Introduction of the Engineering Design Process Management Prototype System

9.3.2 Application of Prototype System

FWMPS-DEDP has been applied in the Customised Trucks for nearly 5 months and6 different kinds of customised trucks including flat platform trailer, heavy-duty dump truck, bulk concrete truck, self-unload truck, concrete mixer and container truck, totally32 trucks have been designed using FWMPS-DEDP. Almost all of the trucks' EDPs are better managed and supported; nearly 69% (22 of 32) of the EDPs can determine dynamically its design process by applying the knowledge of AOs. The extent and scope of intelligence will be further increased with the improvement of the reasoning algorithm and AO knowledge.

To evaluate the effectiveness of the application of FWMPS-DEDP system, a questionnaire was designed to gather feedback from various users. The questionnaire covers issues on Reducing Design Cycle Time, Design Information Transformation, Design Information Searching

Performance, Supporting Performance of EDP, Dynamic EDP Management System Reconstruction, Reducing System Redevelopment Time and Cost, EDP Management Performance and Intelligence Accuracy. Each of these questions was scored on one to five scales: one being the worst and five begin the best.

The questionnaires were sent to 60 selected users including 40 designers and 20management staff in the Customised Trucks Company. About 51 questionnaires were returned and the results in terms of the total number of scores against each question are summarised in Table 9-1.

Table 9-1 Summary of FWMPS-DEDP questionnaires

Criteria /Score	1	2	3	4	5
Reducing Design Cycle	0	3	12	20	16
Design Information Transformation	0	2	7	16	26
Design Information Searching Performance	1	5	15	21	9
Supporting Performance of EDP	0	1	12	26	12
Rapid EDP Management System Reconstruction	0	0	5	12	34
Reducing EDP Management System Redevelopment Time and Cost	1	0	10	6	34
EDP Management Performance (For Management Staff)	0	3	4	6	7
Intelligence Scope	2	3	19	16	11

From Table 9-1 it is clear that FWMPS-DEDP has excellent performance in Design information Transformation, Rapid EDP Management System Reconstruction and Reducing EDP Management System Redevelopment Time and Cost; has good performance in Reducing Design Cycle, Design Information Searching Performance, Supporting Performance of EDP and EDP management Performance. These results also indicate that FWMPS-DEDP can support the dynamic and creative EDP management satisfactorily. Although 11 users believe the Intelligence Scope of FWMPS-DEDP is excellent and 16 users believe the Intelligence Scope is good, there are still 19 users selecting medium, 3 users selecting bad and 2 selecting very bad. From the above, nearly 69 % of the EDP can dynamically schedule its design process using the knowledge of AOs. The Intelligence Performance of FWMPS-DEDP is satisfied with these initial results. The Intelligence

Scope will be further improved by increasing the amount of AO knowledge and improving the reasoning algorithms of the system.

9.3Chapter Summary

A flexible workflow management prototype system supporting engineering design process management implementation developed by Microsoft.Net technology is introduced at first to demonstrate the proposed theory in this thesis. The architecture and modules of prototype system is introduced in detail including its system modelling tool, workflow list management and user interface, workflow management tool, and database design of prototype system. Finally an engineering design process a MTO company is introduced, the prototype system is applied in the engineering design process in the MTO company, these results also indicate that FWMPS-DEDP can support the dynamic and creative EDP management satisfactorily.

Chapter 10 Discussion and Future Work

This thesis details the research work carried out in order to develop a framework to support the dynamic engineering design process management, which will enable the control, decision making and management of dynamic changing engineering design process. This chapter discusses the overall research conducted as well as the contribution to the current research work on engineering design process and its supporting flexible workflow technology. The first section discusses about the research results. Second section proposes and discusses the future work that needs to be carried out to extend this research in different directions.

10.1 Research Results

This research makes the following contributions to the existing engineering design knowledge:

1. Critical review of the existing methods and frameworks in support dynamic engineering design process and its supporting workflow technology (Chapter 3).

2. Proposing an innovative flexible workflow based on autonomic objects and an engineering design process management framework based multi-autonomic objects flexible workflow (Chapter 5).

3. Proposing a dynamic instance-based flexible workflow modelling method, and based on above method, proposing a flexible workflow model based on dynamic instance (Chapter 6).

4. Presenting formalized expression of flexible workflow change and proposing an autonomic object intelligence algorithm based on weighted fuzzy reasoning (Chapter 7).

5. Presenting a distributed flexible workflow engine realized by web service, give a description of flexible workflow based on web service and proposing a distributed flexible workflow execution algorithm (Chapter8).

6. Implementation of flexible workflow management prototype system supporting engineering design process (Chapter 9).

The main contribution is propose an innovative flexible workflow multi-autonomic objects, and propose an engineering design process supporting framework based on multi-autonomic objects flexible workflow. The following subsections highlight these contributions in detail.

10.1.1 Review of the existing frameworks supporting technology

The review of existing frameworks and its supporting technology shows that there is no suitable framework and supporting technology to support the dynamic engineering design process management efficiently. The current frameworks mainly are established based on static workflow technology. Static workflow cannot respond to the change of engineering design process actively, the framework should be reconstructed to changed process with modified model. As to the supporting technology, the current supporting flexible workflow technology cannot support the cross-enterprise engineering process and inter-operation of different flexible workflow; the framework is based on the a proposed multi-autonomics object flexible workflow which is based on autonomic computing technology. The flexibility of workflow model should also be improved, and the traditional two stage workflow modelling method hinders the workflow flexibility. A new flexible workflow modelling method supporting dynamic engineering process is proposed, the proposed flexible workflow modelling method is called dynamic-instanced based workflow modelling method which can change in the workflow instance. Besides, most importantly, the current flexible workflow is more or less passive response to the change, but the fine adaptation is strongly needed for the change and uncertainness, especially the knowledge in the engineering design process is uncertain, uncertain intelligence processing capability is key to flexible workflow intelligence.

10.1.2 Proposing Multi Autonomic Objects flexible workflow and engineering design process management framework

An engineering design process management framework is proposed based on flexible workflow technology. Aiming at complex workflow inter-operation in dynamic engineering design process, a virtual workflow is proposed, by which the workflow inter-operation can be changed into the transfer of control and data information between virtual workflow nodes; then the flexible workflow technology is applied in all level of dynamic engineering process; afterwards, a dynamic engineering process management framework based on multi autonomic flexible workflow is proposed, the autonomic object embedded in flexible node is given, and then the flexible workflow architecture based on multi-autonomic object is designed, the intelligence of flexible workflow is realized by the autonomic object embedded in every workflow nodes and their cooperation.

10.1.3 Key technologies of multi autonomic objects flexible workflow

For the multi autonomic objects flexible workflow, mainly the three key technologies is researched as shown in follows:

In order to improve the flexible description of workflow model in dynamic engineering design process, the traditional “defining and execution” two stage is broken through. At first, the evolution-based workflow modelling method is given at first, a dynamic instance-based workflow modelling method is given further, the workflow model can be created into different dynamic instance in different running environment; second, by analysing the characteristics of flexible workflow, a meta-model in dynamic instance modelling is given and a multi-autonomic objects flexible workflow modelling method is given, and flexible workflow model definition and formal expression, including flexible workflow activity and its autonomic object, workflow control relation, workflow data relation, and workflow activity hierarchical model, is given in detail; finally, the detail structure and graphical expression of flexible workflow activity is given.

Based on multi-autonomic objects, the flexible workflow changing expression is described, and the realization of flexible workflow intelligence is introduced in autonomic object detecting and execution, autonomic object intelligent reasoning; combining with the uncertain flexible workflow autonomic object intelligence, represented by fuzziness, a flexible workflow intelligence algorithm based on extended Mamdani fuzzy reasoning system is proposed; in the proposed flexible workflow intelligence algorithm, the autonomic object knowledge composing and expression is given, and the DEMATEL (Decision Making Trail and Evaluation Laboratory) is introduced in computing autonomic object knowledge independence weight, which can modify the knowledge importance weight got by AHP; then the weighted fuzzy reasoning, as well as precise and fuzzy hybrid knowledge reasoning algorithm is researched and designed with emphasis; the proposed flexible workflow autonomic object intelligence algorithm is demonstrated by a sub-contract enterprise application by a Matlab computing instance; finally a bionic flexible workflow model and its adaptation algorithm is proposed, the algorithm is testified by an example of formulating fixed working hour in a production workflow.

In order to research on and realize the application of proposed multi-autonomic objects flexible workflow, a distributed flexible workflow engine structure based on web service is proposed, and distributed flexible workflow realization is given by applying SOAP and WSDL; according to the characteristics of flexible workflow and web service, a flexibleworkflow describing method is given with example by extending both WSDL and BPEL4WS; a flexible workflow engine execution algorithm and dynamic extending algorithm is given finally.

10.1.4 Development and application of flexible workflow management prototype system supporting engineering design

Finally, a Flexible workflow management prototype system supporting dynamic engineering design process is developed in Microsoft VS. Net 2005 environment, the prototype system is a distributed flexible workflow management system supporting dynamic engineering design process management, it integrates workflow modelling and executing with the help of multi-AOs flexible workflow model, the decision making in engineering design process can be realized by the multi AOs intelligence algorithm and their collaboration. The prototype system supports the graphical modelling to define the model of dynamic engineering design process, by which the engineering design process can be created in the prototype system.

To demonstrate the proposed prototype system and related theory, the prototype system has been applied in a customised trucks company for nearly 5 months and 6 different kinds of customised trucks including flat platform trailer, heavy-duty dump truck, bulk concrete truck, self-unload truck, concrete mixer and container truck, totally 32 trucks has been designed with prototype system. The dynamic engineering design process and its decision making can be supported and managed in a satisfactory way.

10.2 Future Research Directions

Because dynamic engineering design process management and its supporting flexible workflow technology are very complicated, there are certain future research directions should be done, including improvement of engineering design process framework based on multi autonomic objects flexible workflow and improvement of flexible workflow intelligence.

10.2.1 Improvement of engineering design process framework based on multi autonomic objects flexible workflow

The engineering design process framework can support and manage the dynamic engineering design process by multi autonomic objects flexible workflow. But the dynamic engineering design process is a very complex process, it relates with design process arrangement, complicated design activities, design cooperation, design decision making, and so on. In this thesis, the engineering design process framework is mainly research in information supporting aspects to support the management and control of dynamic engineering process, how to improve the dynamic management of the concrete design activities its self in practical engineering design process should be further researched.

Dynamic instance-based flexible workflow modelling method and related workflow model improve the flexibility greatly, but the improvement flexibility unavoidably causes the increases of workflow model structure and semantic mistakes as well as the increase of workflow runtime mistakes. In this thesis, how to improve the flexibility and intelligence of flexible workflow is paid much attention, the verification of flexible workflow model is not considered. Currently, there many researches on how to improve the workflow correctness b workflow model verification and workflow simulation, how to improve the correctness of multi autonomic objects flexible workflow should be further researched.

10.2.2 Improvement of flexible workflow Intelligence

The autonomic computing technology is introduced into the flexible workflow intelligence, the multi-autonomic objects flexible workflow intelligence algorithm with the core of extended Mamdani system, but in the engineering design process application, most of the intelligence reasoning is realized by case-based reasoning instead of fuzzy reasoning. How to improve the learning capability, how to use the rough set to dig the reasoning results further, how to enrich the reasoning case base, how to introduce the artificial neural network into the autonomic computing, how to integrate the rule-based reasoning and the case-based reasoning should all be further researched to improve the flexible workflow intelligence.

10.3Chapter Summary

This chapter reviews the overall research work carried out during this PhD and discuss the research results, including:

1. Review of the existing frameworks supporting technology;

2. Proposing multi autonomic objects flexible workflow and engineering design process management framework;

3. Key technologies of multi autonomic objects flexible workflow;

4. Development and application of flexible workflow management prototype system supporting engineering design;

Then the future research directions are discussed in following aspects: improvement of engineering design process framework based on multi autonomic objects flexible workflow and improvement of flexible workflow intelligence.

Chapter 11 Conclusions

11.1 Overall conclusion

This thesis has demonstrated that the proposed multi-autonomic objects flexible workflow is an important means to support and manage the dynamic engineering design process. The engineering design process is a highly dynamic and uncertain process, and the conventional workflow technology cannot satisfy the requirements of the creative and complex EDP very well for its fixed logic. The multi autonomic objects flexible workflow based on autonomic computing technology is applied to establish the engineering design process management framework. In the proposed engineering design process management framework, in order to improve the flexibility of flexible workflow, an instance-based flexible workflow modelling method and its formal modelling expression are given, and the autonomic object is embedded in the activity of the flexible workflow model to improve the intelligence of flexible workflow. An flexible workflow autonomic object intelligence algorithm is proposed to realize the intelligence of flexible workflow based on extended Mamdani reasoning, the reasoning instance in a Make-to-Order company is given to demonstrate the algorithm, and based on above algorithm, an multi autonomic objects collaborative reasoning algorithm is given to realize the collaborative intelligence of flexible workflow. Considering the inherent characteristics of cross-enterprise and distribution of flexible workflow, a framework of distributed flexible workflow based on web service is proposed, and the description of flexible workflow based on web service and distributed flexible workflow execution algorithm is given to introduce a mechanism for realizing the multi autonomic objects flexible workflow based on the web service. Finally, in order to demonstrate the research work proposed in the thesis, a flexible workflow management prototype system supporting dynamic engineering design process is developed using Microsoft .Net 2005, the proposed theory is demonstrated by the application of the prototype in a Chinese design company satisfactorily.

11.2 Further research work

The key techniques of flexible workflow supporting dynamic engineering design process are researched in this thesis. Based on the future research directions proposed in Chapter 10, there are still further research work can be done to improve the dynamic engineering design process management and its supporting flexible workflow technology.

These directions are:

- Integrating the proposed dynamic engineering design process supporting framework with the existing design activities, especially with the application of computer-based design applications.
- Improving the correctness of flexible workflow with the changing flexible workflow model and embedded intelligent autonomic objects.
- Further improving the intelligent response speed of multi autonomic objects flexible workflow in complex engineering design process.
- Mapping the flexible workflow model based on the web service to the engineering design process, and realizing the run-time change of the engineering design process.
- Improving the flexible workflow management prototype system supporting dynamic engineering design process, especially on the practicality and applicability of the prototype system.

Reference

Aalst, W., Hofstede, A. (2005) “YAWL: yet another workflow language,Information Systems”, Vol. 30, No. 4, pp. 245–275.

Andrews, T., Curbera, F. et al. (2006) “Business Process Execution Language for Web Services Version 1.1”, http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/. 2006-07-26.

Barrett, R., Maglio, P. P., Kandogan, E, Bailey, J. (2005) “Usable autonomic computing systems: The system administrators’ perspective”, Advanced Engineering Informatics, Vol. 19, No. 3, pp. 213-221.

Bento, J., Feijó, B., Smith, D.L. (1997) “Engineering design knowledge representation based on logic and objects”, Computers & Structures, Vol. 63, No. 5, pp. 1015–1032.

Biegus, L., Branki, C. (2004) “InDiA: a framework for workflow interoperability support by means of multi-agent systems”, Engineering Applications of Artificial Intelligence, Vol. 17, No. 7, pp. 825-839.

Blake, M. B., Gomaa, H. (2005) “Agent-oriented compositional approaches to services-based cross-organizational workflow”, Decision Support Systems, Vol. 40, No. 1, pp. 31-50.

Booth, D., Haas, H. et al. (2005) “Web Services Architecture”, http://www.w3.org/TR/ws-arch/. 2005-10-20.

Borg, J. C., Yan, X. T., Juster, N. (2000) “Exploring decisions’ influence on life-cycle performance to aid Design for Multi-X”, Artificial Intelligence for Engineering Design, Analysis and Manufacturing, Vol. 14, pp. 91-113.

Bray, T., Paoli, J. et al. (2006) “Extensible Markup Language (XML) 1.0 (Fourth Edition)”, http://www.w3.org/TR/2006/REC-xml-20060816/. 2006-08-16.

Bruning, J. (2011) “UML Metamodel-based Workflow Modeling and Execution”, 15th IEEE International Enterprise Distributed Object Computing Conference, pp. 97-106.

Cao, J., Wang, J., Zhang, S., Li, M. (2004) “A dynamically reconfigurable system based on workflow and service agents”, Engineering Applications of Artificial Intelligence, Vol. 17, No. 7, pp. 771-782.

Chao, K. M., Norman, P., Anane, R., James, A. (2002) “An agent-based approach to engineering design”, Computers in industry, No. 48, pp. 12-27.

Christensen, E., Curbera, F. et al. (2006) “Web Services Description Language (WSDL) 1.1”, http://www.w3.org/TR/wsdl. 2006-06-12.

Chung, P.W.H., Cheung, L., Stader, J., Jarvis, P., Moore, J., Macintosh, A. (2003) “Knowledge-based process management—an approach to handlingadaptive workflow”, Knowledge-Based Systems, Vol.16, No. 3, pp.149-160.

Clark, J. and DeRose, S. (2006) “XML Path Language (XPath) Version 1.0”, http://www.w3.org/TR/xpath. 2006-07-26.

Clarkson, J., Mélo, A. and Eckert, C. (2001) “Visualization techniques to assist design process planning”, International conference on engineering design, pp.1-23.

Coates, G. (2006) “Agent co-ordination aided distributed computational engineering design”, Expert Systems with Applications, No. 31. Pp. 776-786.

Conroy, G., Soltan, H. (1997) “ConSERV, a methodology for managing multi-disciplinary engineering design projects”, International Journal of Project Management, Vol. 15, No. 2, pp. 121-132.

Crowe, M., Kydd, S. (2001) “Agents and suggestions in a Web-based dynamic workflow model”, Automation in Construction, Vol.10, No. 5, pp. 639-643.

Damien, M., Martin, E., Håkan, P. and Robert, B. (2014) “Integration of the computer-based design analysis activity in the engineering design process - A literature survey”, 10th International Symposium on Tools and Methods of Competitive Engineering - TMCE'14 2, p.1181-1194.

Deng, S., Yu, Z., Wu, C. (2004) “Research and Design of Dynamic Workflow Modelling Method”, Computer Integrated Manufacturing Systems, No. 6, pp. 601-608. (In Chinese)

Eckert, C. M., Clarkson, P. J. and Stacery, M. K. (2001) “Information flow in engineering companies: Problems and their causes”, Proceedings of the 13th International Conference on Engineering Design: Design Management – Process and Information Issues, pp. 21-23.

Eynard, B., Gallet, T., Nowak, P. and Roucoules, L. (2004) “UML based specifications of PDM product structure and workflow”, Computers in Industry, Vol. 55, No. 3, pp. 301-316.

Fung, R. Y. K., Au, A. Y. M., Ip, A. W. H. (2006) “Petri Net-based workflow management systems for in-process control in a plastic processing plant”, Journal of Materials Processing Technology, Vol. 139, No. 1-3, pp. 302–309.

Germani, M., Mandolini, M., Mengoni, M., Peruzzini, M. (2010) “Collaborative Design System for Supporting Dynamic Virtual Enterprises”, IFIP Advances in Information and Communication Technology, Vol. 336, pp. 577-584.

Girard, P. (2001) “An agent environment to support co-ordination between design actors”, International Conference on Engineering Design, pp. 67-87.

Girard, P., Doumeingts, G. (2004) Modelling the engineering design system to improve performance, Computers & Industrial Engineering, Vol. 46, No. 1, pp. 43-67.

Gong, X. (2004) “Research on Web Services Based Distributed Workflow Management System”, PhD Thesis, Northwest University, China. (In Chinese)

Gonnet, S., Henning, G., Leone, H. (2007) “A model for capturing and representing the engineering design process”, Expert Systems with Applications, Vol. 33, No. 4, pp. 881-902.

Habhouba, D., Cherkaoui, S., Desrochers, A. (2011) “Decision-Making Assistance in Engineering-Change Management Process”, Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, Vol. 41, No. 3, pp.344 - 349.

Hamani, N., Dangoumau, N., Craye, E. (2006) “An iterative method for the design process of mode handling models", Computational Engineering in Systems Applications, IMACS Multiconference on, pp. 1431 - 1436.

Hao, Q., Shen, W, Zhang, Z., Park, S. W., Lee, J. K. (2006) "Agent-based collaborative product design engineering: an industrial case study", Computers in Industry, Vol. 57, No.1, pp. 26-38.

Harrington, B.W. (1998) "Development of software tools for automation and acceleration of the engineering design process", 1998 IEEE Aerospace Conference, pp. 265-275.

Ha, S., Suh, H. (2008) "A timed colored Petri nets modeling for dynamic workflow in product development process", Computers in Industry, Volume 59, Issues 2–3, March 2008, Pages 193-209.

Heer, T. and Wörzberger, R. (2011) "Support for modeling and monitoring of engineering design processes", Computers & Chemical Engineering, Vol. 35, No. 4, pp. 709–723.

Heinl, P., Horn, S., Jablonski, S. et al. (1999) "A comprehensive approach to flexibility in workflow management systems", In Proc. WACC '99.

Heller, M., Jager, D., Schluter, M., Schneider, R., Westfechtel, B. (2004) "A management system for dynamic and interorganizational design processes in chemical engineering", Computers & Chemical Engineering, Vol. 29, No. 1, pp. 93-111.

Hicks, B. J., Culley, S.J., Allen, R.D. and Mullineux, G. (2002) "A framework for the requirements of capturing, storing and reusing information and knowledge in engineering design", International Journal of Information Management, Vol. 22, No. 4, pp. 263–280.

Hollingsworth, D. (2005) "Workflow Management Coalition The Workflow Reference Model", http://www.wfmc.org/standards/docs/tc003v11.pdf. 2005-09-20.

Horváth, I. (2000) "Conceptual design: inside and outside Engineering design in integrated product development", proceedings of the 2nd international seminar and workshop, pp. 63-72.

Hu, J., Grefen, P. (2003) "Conceptual framework and architecture for service mediating workflow management", Information and Software Technology, Vol.45, No. 13, pp. 929–939.

Hung, K. C., Julian, P., Chien, T., Jin, W. T. (2010) "A decision support system for engineering design based on an enhanced fuzzy MCDM approach", Expert Systems with Applications, Vol. 37, No. 1, pp. 202-213.

Hurst, K. (1999) "Engineering Design Principles", London, Oxford: Elsevier Science & Technology Books.

Hwang, S. Y., Tang, J. (2004) "Consulting past exceptions to facilitate workflow exception handling", Decision Support Systems, Vol. 37, No. 1, pp. 49-69.

i2 Corporation. (2005). "The Next Generation: Synchronizing Demand and Supply", http://www.i2.com/solutionareas/sixone/synchronize.cfm, 2005-07-30.

IBM. (2005) "An architectural blueprint for autonomic computing", http://www-03.ibm.com/autonomic/pdfs/AC_Blueprint_White_Paper_4th.pdf, 2005-10-02.

IBM. (2006) "Developer Works". http://www-900.ibm.com/developerWorks/xml/index.shtml. 2006-06-01

IBM. (2006) “Web Services Conceptual Architecture (WSCA 1.0)”,
http://www-306.ibm.com/software/solutions/webservices/pdf/WSCA.pdf. 2006-06-01.

Jacobsen, K., Eastman, C., Jeng, T. S. (1997) “Information management in creative engineering design and capabilities of database transactions”, Automation in Construction, Vol. 7, No. 1, pp. 55-69.

Jean-Francois, B. and Pascal, L. (2002) “A co-operation framework for product-process integrationin engineering design”, Design Studies, Vol. 23, No. 6, pp.497–513.

Jegou, D., Kim, D., Baptiste, P., Lee, K. (2006) “A Contract net based intelligent agent system for solving the reactive hoist scheduling problem”, Expert Systems with Applications, Vol. 30, No. 2, pp. 156-167.

Kumar, A., Wainer, J. (2005) “Meta workflows as a control and coordination mechanism for exception handling in workflow systems”, Decision Support Systems, Vol. 40, No.1, pp. 89-105.

Lau, H.C.W., Lee, W.B. and Lau, P.K.H. (2003) “Flexible workflow integration: an object technology approach”, Industrial Management &Data Systems, Vol. 103, No. 3, pp. 167-176.

Lees, B., Branki, C., Aird, I. (2001) “A framework for distributed agent-based engineering design support”, Automation in Construction, Vol. 10, No. 5, pp. 631 – 637.

Lee, K., Lee, K. (2001) “Framework of an evolutionary design system incorporating design information and history”, Computers in Industry, Vol. 44, No. 3, pp. 205-227.

Li, H., Yang, Y., Chen, T.Y. (2004) “Resource constraints analysis of workflow specifications”, Journal of Systems and Software, Vol. 73, No. 2, pp. 271-285.

Lindow, K., Woll, R., Stark, R. (2012) “A Conceptual Framework for Sustainable Engineering Design”, Proceedings of the 19th CIRP Conference on Life Cycle Engineering, pp. 197-202.

Lin, H. (2014), “The Design and Implementation of PDM System for Small and Medium Enterprises”, Applied Mechanics and Materials, Vol. 607, pp. 826-829.

Li, J. (2003) A Multi-Agent Negotiation-based Decision Framework for Extensible Product Life Cycle, Thesis, Texas Tech University, 2003.12.

Liu, D., Wang, J., Chan, S., Sun, J., Zhang, L. (2002) “Modelling workflow processes with colored Petri nets”, Computers in Industry, Vol. 49, No. 3, pp. 267-281.

Liu, D., Wu, M., Lee, S. (2004) “Role-based authorizations for workflow systems in support of task-based separation of duty”, Journal of Systems and Software, Vol. 73, No. 3, 375–387.

Liu, J., Zhang, S., Hu, J. (2005) “A Case study of an inter-enterprise workflow-supported supply chain management system”, Information & Management. Vol. 42, No. 3, pp. 441-454.

Li, W. D., Fuh, J.Y.H., Wong, Y.S. (2004) “An Internet-enabled integrated system for co-design and concurrent engineering”, Computers in Industry, Vol. 55, No. 1, pp. 87–103.

MacGillivray, M., Domina, T., Lerch, T., Kinnicutt, P. (2007) “Engineering design process –An Interdisciplinary Approach”, International conference on engineering education, pp. 1-5.

Madhusudan, T. (2005) “An agent-based approach for coordinating product design workflows”, Computers in Industry, Vol. 56, No. 3, pp. 235–259.

Mantripragada, R., Kinzel, G., Altan, T. (1996) “A computer-aided engineering system for feature-based design of box-type sheet metal parts”, Journal of Materials Processing Technology, Vol. 57, No. 3-4, pp. 241-248.

Manwaring, M.L., Jones, K.L., Glagowski, T.G. (1996) “An engineering design process supported by knowledge retrieval from a spatial database”, Proceedings of Second IEEE International Conference on Engineering of Complex Computer Systems, pp. 395 – 398.

Marquardt, W. and Nagl, M. (2004) “Workflow and information centered support of design processes-the IMPROVE perspective”, Computer Aided Chemical Engineering, Vol. 15, pp. 101-124.

Matzke, W. E., Strube, G. (2006) “A Management tool for the performance management of distributed (global) dynamic engineering design processes”, 2006 IEEE International Engineering Management Conference, pp. 146 – 151.

Mavrikios, D., Alexopoulos, K., Xanthakis, V. et al. (2011) “A Web-Based Platform. For Collaborative Product Design, Review and Evaluation”, Digital Factory for Human-oriented Production Systems, London, Springer.

McGreavy, C., Wang, X. Z., Lu, M. L., Zhang, M., and Yang, S. H. (1996), “Objects, agents and work flow modelling for concurrent engineering process design”, Computers & Chemical Engineering, No. 20, pp. 1167-1172.

Merlo, C., Girard., P. (2004) “Information system modelling for engineering design co-ordination”, Computers in Industry, Vol.55, No. 3, pp. 317-334.

Microsoft. (2006) “Enterprise UDDI Services: A Synopsis”. http://www.microsoft.com/windowsserver2003/techinfo/overview/uddisynop.mspx. 2006-06-12.

Mitra, N. (2005) “SOAP Version 1.2 Part 0: Primer”, http://www.w3.org/TR/soap12-part0/. 2005-10-20.

Moreau, K. A., Back, W. E. (2000) “Improving the design process with information management”, Automation in Construction, No. 10, pp. 127-140.

Muller, R., Greiner, U., Rahm, E. (2004) “AGENT GENTWORK: a workflow system supporting rule-based workflow adaptation”, Data & Knowledge Engineering, Vol. 51, No. 2, pp. 223-256.

Nagl, M., Westfechtel, B., Schneider, R. (2003) “Tool support for the management of design processes in chemical engineering”, Computers & Chemical Engineering, Vol. 27, No.2, pp. 175–197.

Ong, N. S., Foo, W. C. (2004) “A real-time workflow tracking system for a manufacturing environment”, International Journal of Productivity and Performance Management, Vol. 53, No. 1, pp. 33-43.

Orlowska, M. E., Sadiq, S. W., Sadiq, W. (2005) “Specification and validation of process constraints for flexible workflows”, Information Systems, Vol. 30, No. 5, pp. 349-378.

Pahl, G. and Beitz, W. (2003) “Engineering Design a Systematic Approach”, London, Springer-Verlag.

Rehman, F. and Yan, X. T. (2011) “Application of Context Knowledge in Supporting Conceptual Design Decision Making”, International Journal of Product Development, Vol.13, No. 1, pp. 47-66.

Rehman, F., Duffy, A., Yan, X. T. (2008) “Design Management and Prediction”, Xi’an, China, Northwestern Polytechnical University Press.

Roozenburg, N.F.M., Eekels, J. (1995) “Product Design, Fundamentals and Methods”, Chichester, UK, Wiley.

Sterritt, R., Parashar, M., Tianfield, H., Unland, R. (2005) “A concise introduction to autonomic computing”, Advanced Engineering Informatics, Vol. 19, No. 3, pp. 181-187.

Schulz, K. A., Orlowska, M. E. (2004) “Facilitating cross-organisational workflows with a workflow view approach”, Data & Knowledge Engineering, Vol. 51, No. 1, pp. 109-147.

Smith, R. G. and Davis, R. (1981) “Framework for cooperation in Distributed Problem Solving”, IEEE Trans. On systems, Man and Cybernetic, Vol. 11, No. 1, pp. 61-70.

Son, J. H, Oh, S. K., Choi, K. H., Lee, Y. J., Kim, M. H. (2003) “GM-WTA: An efficient workflow task allocation method in a distributed execution environment”, Journal of Systems and Software, Vol. 67, No. 3, pp. 165-179.

Tang, D., Zheng, L., Li, Z., Li, D., Zhang, S. (2000) “Re-engineering of the design process for concurrent engineering”, Computers and Industrial Engineering, Vol. 38, No. 4, pp. 479 -491.

Tsai, A., Wang, J., Tepfenhart, W., Rosca, D. (2006) “EPC Workflow Model to WIFA Model Conversion”, IEEE International Conference on Systems,Man and Cybernetics, pp. 2758-2763.

Wagner, T., Guralnik, V., Phelps, J. (2003) “T ÆMS agents: enabling dynamic distributed supply chain management”, Electronic Commerce Research and Applications, Vol. 2, No. 2, pp. 114-132.

Wang, D., Yan, X. T., Ion, W. J., Wang, R. (2007) “A Framework for supporting intelligent and flexible workflow-based engineering design process”, International Journal of Design Engineering, Vol. 1, No.1, pp. 1-20.

Wang, M., Wang, H. and Xu, D. (2005) “The design of intelligent workflow monitoring with agent technology”, Knowledge-Based Systems, Vol. 18, No. 6, pp. 257-266.

Wang, S., Shen, W., Hao, Q. (2006) “An agent-based Web service workflow model for inter-enterprise collaboration”, Expert Systems with Applications, Vol. 34, No. 4, pp. 787-799.

Wang. Y., Wang. D., Wang. Zeng. And Liu. Z. (2011) “Study on bionic flexible workflow modeling and adaptation algorithm”, Application Research of Computers, Vol. 28, No. 5, pp. 1692-1695. (In Chinese)

Wetzela, I. and Klischewski, Ralf. (2004) “Serviceflow beyond workflow? IT support for managing inter-organizational service processes”, Information Systems, Vol. 29, No. 2, pp. 127-145.

WfMC. (2006) “Process Definition Interface – XML Process Definition Language”.

http://www.wfmc.org/standards/docs/TC-1025_xpdl_2_2005-10-03.pdf. 2006-9-5.

WfMC. (2005) “Workflow Management Application Programming Interface (Interface 2&3) Specification”, http://www.wfmc.org/standards/docs/if2v20.pdf. 2005-11-06.

Wright, I. C. (1997) “A review of research change management: product design into engineering implications for product design", Design Studies, Vol. 18, No.1, pp. 33-42.

Xu, D., Wang, H. (2002) "Multi-agent collaboration for B2B workflow monitoring", Knowledge-Based System, Vol. 15, No. 8, pp. 485-491.

Xue, D., Yang, H. (2004) "A concurrent engineering-oriented design database representation model", Computer Aided Design, Vol. 36, No. 10, pp.947-965.

Wallace, K., Burgess, S. (1995) "Methods and tools for decision making in engineering design", Design Studies, Vol. 16, No. 4, pp. 429-446.

Westerberg, A. W., Subrahmanian, E. et al. (1997) "Designing the Process Design Process", Computers & Chemical Engineering, Vol.21, No. 10, pp. S1–S9.

Xu, L., Li, Z., Li, S., Tang, F. (2007) "A decision support system for product design in concurrent engineering", Decision Support Systems, Vol. 42, No. 4, pp. 2029-2042.

Zhu. L. (2003) "A study on the Development of Man-hour Ration Based on Artificial Neural Networks", Thesis, Northwestern Polytechnical University, China. (In Chinese)

Zhang, L. (2010) "Role-based workflow model in SRS", the 2nd International Conference on Computer and Automation Engineering, 2010.

Appendix-1: Glossary of Terms

Term	Definition
AHP	Analytic Hierarchy Process
AM	Autonomic Manager
AO	Autonomic Object
ASP	Active Server Page
BP	back propagation
BPEL4WS	Business Process Execution Languagefor Web Services
BPML	Business Process Modeling Language
C#	C Sharp
CAD	Computer Aided Design
CAM	Computer-Aided Manufacturing
CAPP	Computer-Aided ProcessPlanning
CAS	Complex Adaptive System
CE	Concurrent Engineering
CF	Credibility Factor
CNC	Computer numerical control
COA	Center of Area
CORBA	Common Object Request Broker Architecture
DCOM	Distributed Component Object Model
DEMATEL	Decision Making Trial and Evaluation Laboratory
DG	Directed Graph
ECA	Event Condition Action
ECOA	Extended Center of Area
EDP	Engineering Design Process
EM	Extensible Madmani

EPC	Event-driven Process Chain
FEA	Finite Element Analysis
Gantt	Gantt Chart
HTTP	HyperText Transfer Protocol
IBM	International Business Machine
IIS	Internet Information Server
IPO	Input Process Output
MATLAB	Matrix Laboratory
MOM	Middle of Maximum
MTO	Make To Order
OA	Office Automation
PDM	Product Data Management
PDS	Product Design Specification
PERT	Program/Project Evaluation and Review Technique
RMI	Remote Method Invocation
RPC	Remote Procedure Call
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
UDDI	Universal Description Discovery and Integration
UML	Unified Modelling Language
WAPI	Workflow Application Programming Interface
WFCP-net	Workflow-net based on Coloured Petri net
WfMC	Workflow Management Coalition
WfMS	Workflow Management System
WSDL	Web Services Description Language
WSFL	Web Services Flow Language
XLANG	XML LANGuage

XML	Extensible Markup Language
XPath	XML Path Language
XPDL	XML Process Definition Language
XSLT	Extensible Style sheet LanguageTransformations

Appendix-2: List of Figures

Appendix-3: List of Tables

Appendix-4: Ph.D. Research Publications

[1] Dongbo Wang, Xiu-Tian Yan, William J. Ion, Runxiao Wang. A Framework for supporting intelligent and flexible workflow-based engineering design process. International Journal of Design Engineering, Vol. 1, No.1, 2007: 1-20.

[2] Dongbo Wang, Xiu-Tian Yan, Ningsheng Guo, Tao Li. A multi Autonomic Objects Flexible Workflow and its knowledge acquisition supporting Engineering Design Process, Advanced Materials Research, Vol. 97-101, 2010: 3341-3344. (EI: 20101612863579)

[3] Dongbo Wang, Xiu-tian Yan, Bing Chen, Yangxi Fan. A Knowledge Acquisition approach for multi-autonomic objects flexible workflow. Proceedings of The 39th International Conference on Computers & Industrial Engineering (CIE39). Troyes, France. July, 2009: 1397-1401. (EI: 20094712467521)

[4] Dongbo Wang, Bing Chen, Yangxi Fan, Leijin Fu. Research on Multi Autonomic Objects Flexible Workflow based on Complex Adaptive System. The 1st International Conference on Information Science and Engineering. Nanjing, China. Dec, 2009: 3024 - 3027. (EI: 20102212964787)

[5] Dongbo Wang, Runxiao Wang, Xiu-Tian Yan, Yijun Sheng. Research on the Flexible Workflow Based on Multi Autonomic Objects. Computer Integrated Manufacture System. Vol. 13, No.5, 2007: 955-960. (EI: 072710691870, In Chinese)

[6] Wang Yinghui, Wang Dongbo, Wang Zenglei, Liu Zhizhong. Study on bionic flexible workflow modeling and adaptation algorithm, Application Research of Computers, No.5, 2011. (In Chinese)

[7] Wang Dong-bo, Yan Xiu-tian, Li Zhi-yi, Sun Yi-xiao. Research on agile supply chain based on multi-autonomic objects flexible workflow. Application Research of Computers, Vol. 27, No.3, 2010. (In Chinese)

[8] Dongbo Wang, Xiu-Tian Yan, William J. Ion, Runxiao Wang. An approach to intelligent and flexible workflow model based engineering design management, Harbin, china. Jan, 2006: 73-79.

[9] Dongbo Wang, Xiu-Tian Yan, Runxiao Wang, Weigang Zhao. Research on the Flexible

Execution of Distributed Workflow Engine based on Web Service. Proceedings of the 2nd International Conference on Software, Knowledge, Information Management and Applications. Kathmandu, Nepal. March, 2008.

[10] Wang Dongbo, Wang Runxiao, Huang Wei, Hu Zhiqiang. Research on flexible workflow model based on dynamic instance modelling. Journal of Mechanical Science and Technology. Vol. 26, No.11, 2007: 1413-1417. (In Chinese)

[11] Dongbo Wang, Xiu-Tian Yan, William J. Ion, Runxiao Wang, et al. An Approach to a Web-based Flexible Workflow Modeling. Transactions of the VSB-Technical University of Ostrava Mechanical Series. No. 2, 2005: 179-184.

[12] Wang Dongbo, Wang Runxiao, Tong Xufeng, Qiu Fangliang. Research on flexible workflow model based on dynamic structure. Computer Application Research. Vol. 21, No. 6, 2004: 67-69, 81.

Appendix-5: Project related

[1] Doctoral New Teacher Fund of Ministry of Education of China (Grant No. 200806991047), 2008.

[2] Science Innovation Fund of Northwestern Polytechnical University of China (Grant No. 07XE0126), 2007.

[3] Research on integration of digital manufacturing resources management system and PDM system, Xi'an Science Project, 2010.