

An Agent and Goal-Oriented Approach for Virtual Enterprise Modelling: A Case Study

Liu Zhi¹, Lin Liu²

¹Software Development Environment Key Laboratory,
Zhejiang University of Technology, Hangzhou, China, 310014
lzhi@cs.toronto.edu

²Department of Computer Science, University of Toronto,
Toronto, Ontario, Canada, M5S 3H5
Liu@cs.toronto.edu

Abstract. Virtual enterprise which is established to satisfy the requirements of market is a one-off, dynamic distributed organization. In a virtual enterprise, partners selected form an alliance to fulfill goals that otherwise can't be achieved for the limitation of individual's capabilities. Modelling virtual enterprise will assist in analyzing system construction issue and facilitate cooperation among the partners. This paper proposes to use strategic actor based on modelling framework i* to represent the construction and cooperation issue of virtual enterprise. The partners are treated as strategic agents and cooperation among them are depicted as dependency relationships. Through dependency relationships, the actors in the model can form a hierarchical and federated architecture and cooperate with each other. Example from an air-separator virtual enterprise is used to illustrate.

1 Introduction

A virtual enterprise is a dynamic alliance among companies. Through cooperating inside or outside companies, a virtual enterprise can develop some kind of products with shorter leadtime, better allover performance and lower cost to meet the rapidly changing market. Virtual enterprise is a temporary and virtual organization which is built to respond to the market more quickly and utilize various technologies and resources more efficiently in the entire virtual enterprise. It will be initiated by a core company which plays a role of leader. It will select proper cooperators to form a short-term relationship to collaborate in a one-off project. The initiator will select those partners who can provide resources needed. In fact, a company can take part in different dynamic alliances and act as a different virtual group in a virtual enterprise. There are several cooperation styles, such as supply chain, subcontract process, joint venture management and so on [18]. The cooperation eventually maximizes the virtual enterprise's utilization of resources, reduces high investment and risk and ultimately wins the competition. In order to form the short-term cooperative relationships with partners, the initial company must model the requirements of virtual enterprise and take on the high-level design task (product design task) as its goal. Then it can

decompose the task into some sub-tasks. These partners are called the virtual groups who can take on the coarse-grained sub-tasks in accordance with their goals. They also can decompose their sub-tasks into the fine-grained sub-tasks and then select corresponding partners and allocate these sub-tasks to them. So the whole high-level design task can be accomplished by these hierarchical partners. All the partners take on these sub-tasks are according to their goals and form a hierarchical alliance. It is very important to decompose the large and complex system into smaller, more manageable subsystems which can be dealt relatively.

Some architectures of virtual enterprise have been provided and many technologies have been applied in this field[2][12]. Many sophisticated methodologies, such as CIMOSA, GRAI-GIM and PERA, are used to describe the life-cycle an integrated enterprise[17]. But these existing enterprise modelling methodologies are process-oriented approaches and relevant solutions should be predefined. Based on these modelling methods, the enterprise is represented as various function models and information models. In the meantime, they don't provide any tools and methods to identify the requirements of virtual enterprise. Because virtual enterprise is a dynamic alliance and the partners who are selected to join the alliance according to their goals maybe have different solutions to accomplish the same tasks or sub-tasks. So how to analyze the requirement of virtual enterprise is the foundation stone for modelling it. The partners who join the alliance will take on the tasks or sub-tasks and can be treated as agents which are capable of flexible, autonomous actions in virtual enterprise[3][15][21]. It will be very suitable to adopt an agent-oriented approach to analyze and model the requirements of a virtual enterprise and accomplish it in an agent-oriented architecture[4][11][13][14][16]. Adopting an agent-oriented approach to model virtual enterprise aims at support elicitation, exploration, and analysis of the systems-in-the-world and it can help users and stakeholders articulate what they concern and what cooperation relationships between them[1][6].

In this paper, we adopt the *i** framework to model a virtual enterprise. The *i** framework [7] was developed for modelling intentional relationships among strategic actors, and helping to reason about changes in relationships among strategic actors. It adopts an agent-oriented perspective. In *i** framework, organizations are viewed as consisting of social actors who have freedom of action (similar to agents), but depend on each other for goals to be achieved, tasks to be performed, and resources to be furnished [8]. These dependencies are intentional and based on some important concepts such as goal, commitment, belief, capability, and so on. It is very useful to provide a higher level modelling method comparing to conventional modelling techniques such as data flow diagramming and object-oriented analysis. From this viewpoint, the *i** framework is similar to the Gaia methodology[11]. It also treats an organization as a collection of actors, but it more stresses on dependency relationship between the actors. It can support strategic actor to undertake the task according to its belief and capability in order to accomplish a goal. It also can reason about opportunities and vulnerabilities through all kinds of dependency relationships. This goal-oriented and agent-oriented analysis leads to the functional and non-functional requirements of the system-to-be. Based on the analysis, agent-oriented architectural design can be accomplished. This modelling approach will reduce a semantic gap between the agent-oriented software system and its elementary environment[9]. The framework has been applied in the context of requirements engineering, business process reengi-

neering, and other software processes. Now the framework is tending to form the basis of an agent-oriented system development paradigm.

In this paper, we explore the use of i^* framework for modelling virtual enterprise. We can use organizational view to capture goals of the initial company. Goals related to functional capabilities provide the basis requirements of virtual enterprise and goals related to business and system qualities provide the non-functional requirements (called softgoal “which is used to model quality attributes for which there are no a priori, clear-cut criteria for satisfaction, but are judged by actors as being sufficiently met”)[5]. Goals from the organizational view take a starting step to construct the virtual enterprise. This can be represented as a Strategic Dependency model in the i^* framework. Then the partners who take on those goals will decompose the pertinent goals or tasks into alternative solutions and the softgoals will be used to determine how each alternative solution relates to pertinent business and system qualities. This can be represented as a Strategic Rationale model in the i^* framework. Finally, the partners who take on the sub-goals and sub-tasks will refine the goal graph until the sub-goals and sub-tasks can be taken on by all hierarchical partners. The partners can be represented as actors in the i^* framework.

This paper outlines an idea paradigm to use strategic actor relationships in the i^* framework to represent the construction and cooperation issue in a virtual enterprise, and to illustrate how these models can support analysis and reason. Using the i^* framework can obtain a better understanding of the organizational relationships among those agents and can understand the rationale of the decisions.

Section 2 presents a brief overview of the i^* framework, including its basic concepts such as Strategic Dependency model and Strategic Rationale model used in the virtual enterprise. Section 3 describes the cooperation relationships modelling in the virtual enterprise and proposes a cooperative architecture supporting it. Section 4 discusses related work and future research work.

2 The i^* Framework for Modelling Virtual Enterprise

The i^* framework consists of two kinds of models. One is the Strategic Dependency model that describes the network of intentional relationships among actors. Actors depend on each other for goals to be achieved, tasks to be performed, and resources to be furnished. These dependencies are intentional based on important concepts such as goal, commitment, capability, belief, and so on[7]. Another is the Strategic Rationale model which describes and supports the reasoning that each actor has about its relationships with other actors, its alternative means to achieve its goals, and how the qualitative expectations of actors are satisfied by these alternatives. Actors in i^* framework are strategic because they evaluate their social relationships in terms of opportunities that they offer, and vulnerabilities that they may bring.

2.1 Strategic Dependency Model

A Strategic Dependency (SD) model consists of many nodes and links. Each node represents an actor, and each link between two actors indicates that one actor depends

on the other. The depending actor is called the depender and the actor who is depended on is called dependee. That means the depender needs some help to attain the goal from the dependee. The depender can achieve goals that are not able to be realized without the dependency, or not as easily or not as effectively. At the same time, the depender becomes vulnerable. If the dependee fails, the depender would be adversely affected in its ability to achieve its goals. There are four dependencies in Strategic Dependency model: goal dependency, task dependency, resource dependency, and softgoal dependency. In a goal dependency, an actor depends on another to make a condition come true; In a task dependency, an actor depends on another to perform an activity; In a resource dependency, an actor depends on another for the availability of an entity; The softgoal dependency is a variant of the goal dependency and it is not a priori, clear-cut criterion for what constitutes meeting the goal. It reflects different types of freedom that is allowed in the relationship between depender and dependee[7].

Figure 1 shows a Strategic Dependency model for the virtual enterprise. Here the virtual enterprise is constructed to produce a complex product, named as air-separator or oxygen-producer. It always needs a long time to produce an air-separator and it needs a lot of companies to cooperate with each other. The Product Designing and Quoting Group, Devices and Machines Manufacturing Group, Parts Supplying Group, and Product Assembling Group compose the virtual enterprise. These groups may consist of a set of related companies to finish respective goals. In fact, this forms co-operation in high level in the virtual enterprise. The virtual enterprise is triggered by the requirement of market. In other words, the Customer's order is the requirement of the market. When the customer has a desire to purchase an air-separator, it is very difficult to get a product in stock, because this kind of product is customized and can't be produced in batch. So the customer must hand over the order to the Product Designing and Quoting Group in order to customize the product. The Product Designing and Quoting Group will design the product in accordance with the performance of product and give the quoted price and the delivering date. For the customer, it can evaluate the design, price and delivering date and decide to accept or not. It can get the related information, such as price and producing time from the Devices and Machines Manufacturing Group, Parts Supplying Group, and Product Assembling Group. If not, it can negotiate with the Product Designing and Quoting Group. If the negotiation fails, the virtual enterprise will not be built. The Product Designing and Quoting Group will design the product according to the performance of product. In order to enhance competition of the product, the Product Designing and Quoting Group should give the Customer a good design, including good performance, good price, and short delivering time. Good performance means the product should sufficiently meet the need of the product. It demands the Product Designing and Quoting Group to have sufficient experiences in product design. In the meantime, the Group should consider the related constraints in the downstream of producing process in advance, including the constraints in the manufacturing process, assembling process, and so on. It can avoid doing over again in order to shorten the delivering time and it also can effectively reduce the designing surplus in order to cut down the product price. All these can make the designed product more competitive. These constraints can be represented as all kinds of interaction relationships. So the Product Designing and Quoting Group is a footstone to build the virtual enterprise.

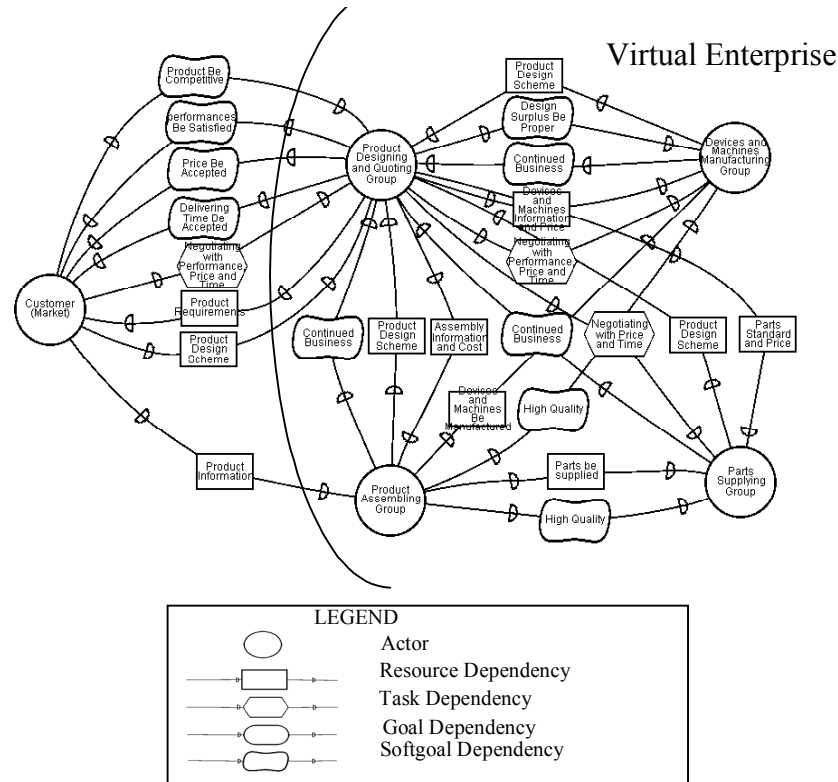


Fig. 1. The strategic dependency model of the virtual enterprise

In figure 1, the Customer depends on the Product Designing and Quoting Group to get the product design scheme and evaluate the scheme, including the performance, price, and delivering time. The Product Designing and Quoting Group will depend on the Customer to get the product requirements. It also depends on the Devices and Machines Manufacturing Group to get the proper manufacturing information in order to cut down the designing surplus. It also can get sufficient information about the manufacturing price and time in order to give quoted price for the Customer. When the Product Designing and Quoting Group has designed the devices and machines it would negotiate with the Devices and Machines Manufacturing Group. The performance, the price and the delivering date will be decided and cooperation among them will start. That will be a part of the virtual enterprise. In the meantime, the Devices and Machines Manufacturing Group will do its best to satisfy the Product Designing and Quoting Group in order to continue the business.

Besides the cooperation above, the Product Designing and Quoting Group will cooperate with the Parts Supplying Group and Product Assembling Group. The Parts Supplying Group will purchase and provide all kinds of standard parts or instruments for the product. The Product Assembling Group will get all kinds of design information, devices and machines manufactured, and standard parts and assemble them together. The product will be given a test run and then is checked and accepted by the

Product Designing and Quoting Group. It will be sent to the Customer by the Group and entire virtual enterprise will be dismissed.

2.2 Strategic Rationale Model

The Strategic Rationale (SR) model describes a more detailed level of an actor in order to model its internal intentional relationship and support to reason of each actor about its intentional relationships. The intentional elements, such as goals, tasks, resources and softgoals, appear in SR model not only as external dependencies, but also as internal elements arranged into a hierarchy of means-ends and task-decompositions relationships. A goal may be associated through means-ends with multiple, alternative ways to achieve it, which are usually represented as tasks. Task-decomposition links hierarchically decompose task into four types: sub-tasks, sub-goals, resources, and softgoals[6][7].

In figure2, the goals of each group and internal intentional relationships are elaborated. The goal of the Customer is to get a customized product (air-separator) and is accomplished by “Requirements of Product Be Decided”. This task can be decomposed of three sub-tasks, such as “Performances Be Decided”, “Price Be Decided”, and “Deadline Be Decided”. For the Product Designing and Quoting Group, its goal is “Product Designing and Quoting”. There are three possible ways to accomplish it, such as “New_Product Designing and Quoting”, or “Transformed_Product Designing and Quoting”, or “Modified_Product Designing and Quoting”. These tasks respectively response for different products and each task is composed of two sub-tasks, related product designing and quoting. In any designing way, there are five main designing processes, including “General Designing”, “Device Designing”, “Machine Designing”, “Power Supply System Designing”, and “Detect System Designing”. In which, the Power Supply System includes many kinds of power transformers and the Detect System which will detect flow quantity, pressure on the pipes, resistance in the pipes and so on. There are a lot of valves, detecting instruments and electronic controlling instruments. So these sub-tasks have different complexities and they have different quoted prices. The Product Designing and Quoting Group will select related cooperative partners whom the tasks and sub-tasks can be allocated to and roughly design the product according to the requirements from the Customer and give its quoting price. The Customer evaluates the design and price and negotiate with the Product Designing and Quoting Group. When they come to an agreement, the partners will take the product order to do the detailed designs. The designing partners will consider the product manufacturing and cost information from the partners in the downstream and the Product Designing and Quoting Group can negotiate with them to ensure the competitive price, delivering time, and good performance. Because the product is rather complex, the partners can decompose the designing work and find partners to finish it. These partners can cooperate with each other inside or outside their own companies.

In figure 2 also shows the Strategic Rationale model to support reasoning about the product designing and quoting process. Each possible way has different implications for the qualities of goals or softgoals. A softgoal is usually not easy to quantified. The contributions to softgoals can be positive or negative, adequate or inadequate. The

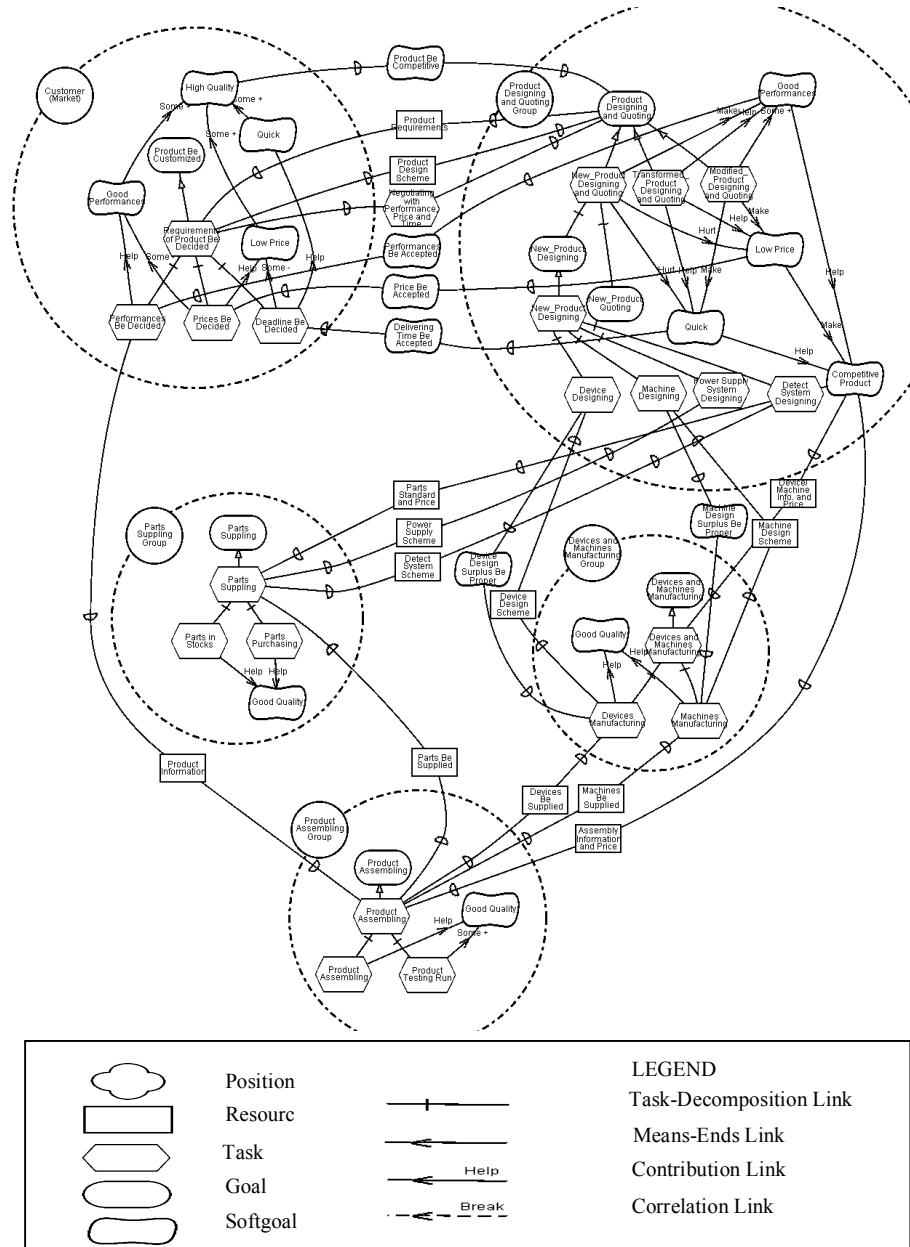


Fig. 2. The strategic rationale model of the virtual enterprise

treatment of softgoals is based on a framework developed by Lawrence Chung for dealing with non-functional requirements in software engineering [10]. The positive contribution types for softgoal are Help (positive but not sufficient), Make (positive and sufficient) and Some+(positive in unknown degree). The corresponding negative

types are Hurt (negative but partial), Break (negative and sufficient) and Some- (negative in unknown degree). In figure 2, Modified_Product Designing is based on the existed designs and just need to do some little modifying work. So the designing process will be quick, low price, and have some positive effect to good performance; The Transformed_Product Designing is also based on the exist designs, but need to do a rather big transforming design. It will be some quick, some low price, and have some positive to good performance. These two processes are case-based designs and rather steady because there are already much experiences in the practices. But in the meantime they also have a difficult to break through the constraints of those models. The New_Product Designing has an innovative designing method and may make sufficient positive effect to good performance, but it needs more time and high price. So the Strategic Rationale model provides a systematic way to explore the possible alternative in designing process. This would be help for the cooperators to decide which way they should take and sufficiently meet the requirements of the Customer.

In figure 2, there are still other Groups in the virtual enterprise. These groups supply related constraint information for the Product Designing and Quoting Group and cooperate with the Group to accomplish devices and machines manufacturing, parts supplying, and product assembling. In Devices and Machines Manufacturing Group, there are a lot of manufacturing experiences. The redundant surplus in design is the main reason for cost inflation, and sometime it is a factor to prolong the delivering time. So the control of surplus is very important in product design. But it must be large enough to ensure the product quality in order to satisfy the Product Designing and Quoting Group to continue their market in the future. The Parts Supplying Group is similar to the Devices and Machines Manufacturing Group. It has two ways to supply the parts, in stocks and purchasing. It also needs to provide good quality work to continue their cooperation. The Product Assembling Group will assemble the devices, machines and parts supplied by the groups mentioned above in accordance with the design of the product and will give the product a test run. Then the product will be checked and accepted by the Product Designing and Quoting Group and delivered to the Customer.

2.3 Roles, Positions and Agents

In i^* , an agent is an actor used to refer to the concrete, physical manifestation; A role is an abstract and serves as holder of intention. It can be reasoned about before it is allocated to an agent. Position is typically assigned to an agent as a unit, serves as an intermediate mapping between agents and roles and eventually is occupied by some agents. Positions can cover roles, agents can occupy positions, and agent can also play roles directly [7]. The term actor is used to refer to the generic concept. agent, role, and position are specializations of the actor concept.

In the virtual enterprise, the partners cooperate with each other as virtual groups according to their goals. In i^* , goals always belong to agents and the global goal is distributed and propagated over the network of dependency relationships to form hierarchical intentions (goals). So an actor can represent an organization in which there are some other actors to take on the sub-tasks. Here the functional requirements are

3.1 Cooperation in the Virtual Enterprise

When the virtual enterprise is built, the partners will cooperate with each other according to their dependency relationships. The dependencies propagating over the network of dependency relationships make the dependers and dependees cooperate effectively.

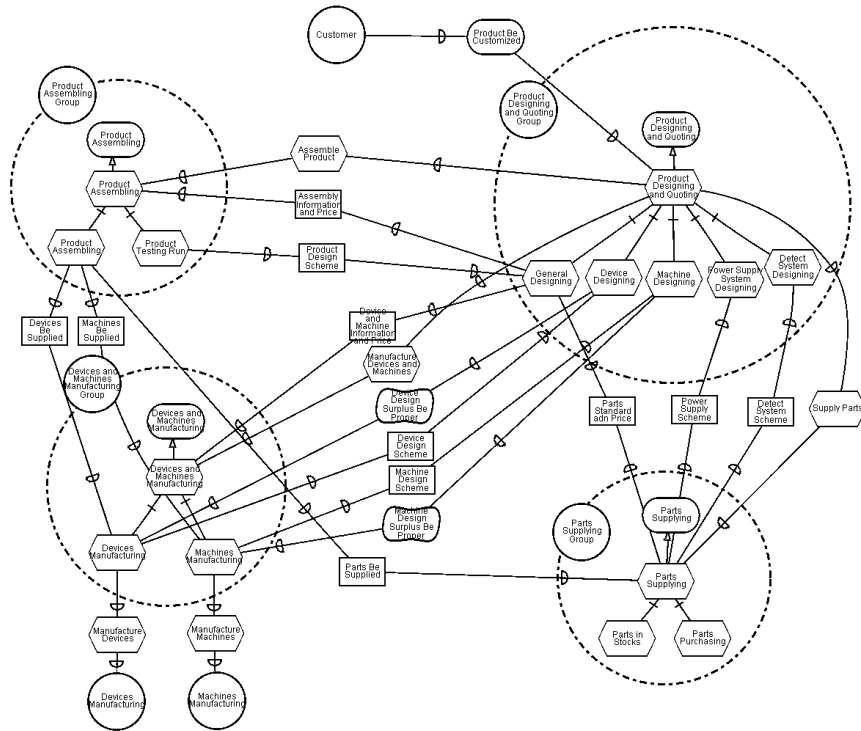


Fig. 4. The cooperation relationships in the virtual enterprise

In figure 4, the Customer depends on the Product Designing and Quoting Group (goal-dependency) to customize the product and doesn't care how the Group achieves the goal. In the virtual enterprise, the Product Designing and Quoting Group takes on a leader role to establish and organize the virtual enterprise in order to accomplish the tasks. The sub-tasks in the Product Designing and Quoting Group, such as General Designing, Device Designing, Machine Designing, will finish the general product design and all kinds of designs of devices and machines in this product. They respectively depend on Devices and Machines Manufacturing (task-dependency) to manufacture the devices and machines, depends on Parts Supplying (task-dependency) to supply the parts, depends on Product Assembling (task-dependency) to assemble the product. There are some resource-dependencies between tasks for the availabilities of entities (physical or informational). The Device Designing depends on the Devices

Manufacturing (softgoal-dependency) to achieve proper surplus design for devices. The depender will make the final decision based on information that it get from these partners in the downstream in order to cut down the surplus properly. Beside the dependencies among the actors, there are many task decomposition links in the model which can describe the decomposition of tasks.

In figure 4, the sub-tasks of Devices Manufacturing and Machine Manufacturing are respectively allocated to two manufacturing partners. These two manufacturing partners can be selected by the Devices and Machines Manufacturing Group and can get those dependency relationships and related information from the their upper actor. From this view, the whole dependency relationships among the actors are formed a hierarchical architecture and it can constructed as a multi-agent system based on the federation architecture to support the virtual enterprise[15][21].

3.2 Cooperative Architecture for the Virtual Enterprise

Based on the analysis with the i^* framework, an agent and goal-oriented model for the virtual enterprise can be built. Because the virtual enterprise is a dynamic alliance, when the goals and tasks have been allocated to partners, the allying process is finished. In order to support the allying process and cooperation in virtual enterprise, we propose a hierarchical and federated architecture in figure 5. In this architecture, the partners can communicate with each other with the well-known Contract Net Protocol[20] when they ally with each other. Those agents(not at the lowest level) can dynamically take two roles: manager or contractor[14]. As a manager, the agent will decompose and allocate the task or sub-tasks to its partners. It can announce the task or sub-tasks to relevant partners and collect the bids from partners and awards the task or sub-tasks to the best bidders. When the task or sub-tasks have been allocated to the partners, the manager and the partners form a federation. The manager will cooperate and coordinate with the partners and support communication among the agents in its federation. Agents in a federation can communicate with others in different federations through their respective upper manager. This makes the realization of communication more convenient.

In figure 5, the hierarchical and federated architecture not only supports the task to be decomposed and allocated, but also to facilitate the tasks to be managed. In a federation, these partners cooperate with each other in order to accomplish the tasks or sub-tasks. These partners may come from different companies or come from different work units in the same company. They can overlap to implement their respective tasks or sub-tasks allocated to them based on various dependency relationships. That means those companies cooperating with each other in the virtual enterprise as different virtual groups, such as G_1 , G_2 , or G_3 , respond different tasks or sub-tasks in the high-level cooperation. In the low-level cooperation, the developers and work units in same company or in different companies can form all kinds of work teams, such as T_1 , to take on the decomposed tasks or sub-tasks. This makes the virtual enterprise more flexible and more competitive. In this paper, a partner can be a company, or a work unit, or a developer in a company.

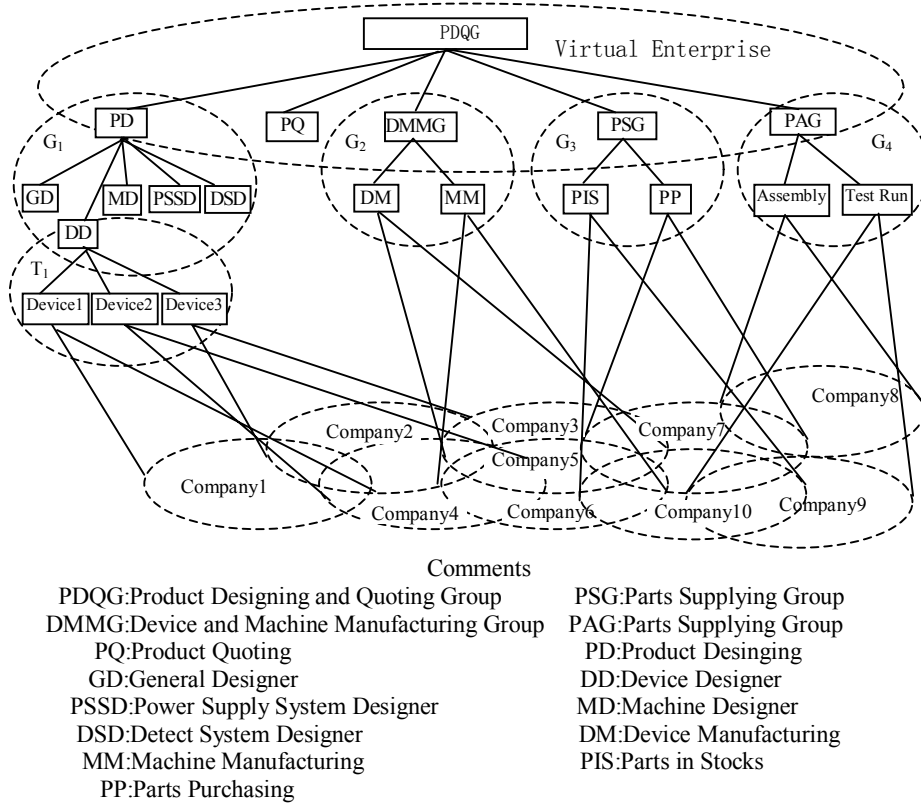


Fig. 5. A hierarchical and federated architecture for the virtual enterprise

4 Discussion

In this paper, we have illustrated the use of i^* framework to model an air-separator virtual enterprise. The i^* framework has been applied in many fields, but what it has modelled are always very concrete applications, such as trust-related issues, intellectual property management issues. The virtual enterprise is a very complex and distributed system and it is a dynamic alliance. The initial company establish and organize the virtual enterprise in order to sufficiently meet the requirements of the market. It will plan the product design and manufacturing process and select related partners according to their goals, intentions, and capabilities to join the dynamic alliance and share the task. It also needs to model the various dependency relationships among the partners in order to support their cooperation. So it is very important to model the virtual enterprise in order to reveal what parts are in it and how they interact and cooperate.

The i* framework was developed for modelling intentional relationships among strategic actors. The cooperators in the virtual enterprise are treated as actors and cooperate with each other in a network of dependency relationships. Some actors depend on goals and others depend on tasks or softgoals. Because some task are so complicated that they are decomposed into several sub-tasks and allocate these sub-tasks to actors (partners) which have been selected to join the alliance. Here we specialize the actors who join the virtual enterprise as positions and assign the positions to agents who represent the partners. This realizes the task decomposition and task allocation. The i* model with the dependency relationships encourages and facilitates the analysis of requirements and cooperation in the virtual enterprise. We also propose a cooperative architecture based on this model to support virtual enterprise to be constructed and to be managed.

This paper has taken a rather simplistic view of the virtual enterprise even though we provide a complex alliance to produce a kind of product, air-separator. We just give the coarse-grained analysis about it. It still needs rather deep study in the fine-grained level. In this paper, we model the virtual enterprise using the i* framework aiming at analyzing the requirements of virtual enterprise and cooperation among the partners. There are still lots of works to do in this field. In future work, we plan to apply the i* framework to analyze and reason opportunities and vulnerabilities among the partners in cooperation through all kinds of dependency relationships using the softgoals. We also plan to combine other technologies, such as scenarios[19], to achieve the development from early requirement analysis to later detailed architecture design.

References

1. Bashar Nuseibeh, Steve Easterbrook. Requirements Engineering: A Roadmap. *Proceedings of International Conference on Software Engineering (ICSE-2000)*, 4-11 June 2000, Limerick, Ireland, ACM Press
2. Bloch, M. and Pigneur, Y. The Extended Enterprise: a Descriptive Framework, Some Enabling Technologies and Case Studies, *Proceedings of the 2nd Int. Conf. Network organization Management*, June 95
3. Carlos A. Iglesias, Mercedes Garijo and Jose C. Gonzalez. A Survey of Agent-Oriented Methodologies. *Proceedings of the 5th International Workshop on Intelligent Agents V: Agent Theories, Architectures, and Languages (ATAL-98)*
4. Cutkosky, M.R., Englemore, R.S., Fikes, R.E., Genesereth, M.R., Gruber, T.R., Mark, W.S., Tenenbaum, J.M. and Weber, J.C. (1993) PACT: An Experiment in Integrating Concurrent Engineering Systems, *IEEE Computer*, 26(1):28-37
5. Daniel Gross, Eric Yu. Evolving System Architecture to Meet Changing Business Goal: an Agent and Goal-Oriented Approach. *ICSE-2001 Workshop: From Software Requirements to Architectures*. May 2001, Toronto, Canada. pp. 13-21
6. Eric Yu. Agent-Oriented Modelling: Software Versus the World. *Proceedings of Agent-Oriented Software Engineering AOSE-2001 Workshop*, Montreal, Canada, May 29th 2001

7. Eric Yu. Modelling Strategic Relationships for Process Reengineering, *Ph.D. thesis*. Dept. of Computer Science, University of Toronto, 1995
8. Eric Yu and John Mylopoulos. Modelling Organization Issues for Enterprise Integration. *Proceedings of International Conference on Enterprise Integration and Modelling Technology*, October 28-30, 1997, Turin, Italy
9. Jaelson Castro, Manuel Kolp, and John Mylopoulos. Towards Requirements-Driven Information Systems Engineering: The Tropos Project. *Information Systems*, 27(6), 365-389, 2002
10. Lawrence Chung, Brian A. Nixon, Eric Yu, and John Mylopoulos. *Non-Functional Requirements in Software Engineering*. Kluwer Academic Publishers, 2000
11. M. Wooldridge, N. Jennings, and D. Kinny. The Gaia Methodology for Agent-Oriented Analysis and Design. *Journal of Autonomous Agents and Multi-Agent Systems*, 3(3):285-312, 2000
12. Mingwei Zhou, Laszlo Nemes and et al. A Framework for Design a Virtual Manufacturing Enterprise and Its Implementation as a Workbench. DIISM '98. IFIP TC5 WG.5.3/5.7 3rd International Working Conference on the Design of Information Infrastructure Systems for Manufacturing II (1998: Fort Worth, TX) Boston: Dordrecht: London: Kluwer Academic, 1999
13. Nicholas R. Jennings. Controlling Cooperative Problem Solving in Industrial Multi-Agent Systems using Joint Intentions. *Artificial Intelligence*, 75(2) 195-240, 1995
14. Nicholas R. Jennings. A Roadmap of Agent Research and Development. *Int Journal of Autonomous Agents and Multi-Agent Systems*, 1, 7-38(1998)
15. Nicholas R. Jennings. On Agent-based Software Engineering. *Artificial Intelligence*, 117 (200)277-296
16. Nicholas R. Jennings. An Agent-Based Approach for Building Complex Software Systems. *Communications of The ACM*, April 2001/Vol. 44, No. 4
17. Peter Bernus, Laszlo Nemes and Theodore J. Williams. (ed.) *Architectures for Enterprise Integration*. Chapman & Hall. 1996.
18. Rick Dove. Agile Enterprise and Agile Production. *Production Magazine*, 1995.11
19. R.J.A. Buhr, D. Amyot, M. Elammari, D. Quesnel, T. Gray, and S. Mankovski. High Level, Multi-Agent Prototypes from a Scenario-Path Notation: A Feature-Interaction Example. In: H.S. Nwana and D.T. Ndumu (Eds), *Third Conference on Practical Application of Intelligent Agents and Multi-Agent Technology*, London, UK, 1998, pp. 255-276
20. Smith R. G. The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. *IEEE Transactions on Computers*, 1980, c-29(12):1104-1113
21. Weiming Shen, Douglas H. Norrie and Jean-Paul A. Barthes. (2001) *Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing*. Taylor & Francis