STUDY ON THE DESCRIPTION METHOD OF MANUFACTURING CAPABILITY BASED ON DESCRIPTION LOGICS IN CLOUD MANUFACTURING

Yongliang Luo[(a)], Lin Zhang[(a)], Fei Tao[(a)] ,Yongkui Liu[(a)], Lei Ren[(a)]

[(a)] School of Automation Science and Electrical Engineering, Beihang University,
Beijing 100191, P. R. China

yongliang2002@gmail.com, zhanglin@buaa.edu.cn,
taofei999@gmail.com, yongkuiliu@163.com, lei_ren@126.com

ABSTRACT

Manufacturing capability (MC) sharing is one of the most important innovations in cloud manufacturing (CMfg). In order to realize the on-demand usage of MC in the CMfg, the issue of how to describe MC need to be solved first. This paper proposes a new description method of MC based on description logics (DL) oriented to CMfg. Concepts and state of the art related to MC are summarized firstly, and then the characteristics and theoretical basis of current service description language are systematically analyzed. On this basis, the description model of MC is proposed. Finally, the static and dynamic information models based on DL of MC are mainly investigated combined with practical examples.

Keywords: cloud manufacturing, manufacturing capability, description language, dynamic description logics

1. INTRODUCTION

Service, environment protection and knowledge innovation are the main factors in manufacturing competition and they are the main issued (issues) to be faced and solved for manufacturing enterprises. At the meantime, Cloud manufacturing (CMfg) as a possible solution and new manufacturing paradigm is proposed by Bohu Li and Lin Zhang(2010) , it can provide theoretical and technical supports for the transformation from production-oriented manufacturing to service-oriented manufacturing. CMfg is a new service-oriented, knowledge-based, and intelligent networked manufacturing mode with high efficiency and lower consumption [Bohu Li and Lin Zhang *et al*. 2010]. In order to achieve the virtualization and servilization of manufacturing resources and manufacturing capability, CMfg is realized by combining with advanced manufacturing and information technologies organically, such as cloud computing, the internet of things, service-oriented technologies, high performance computing, information system integration, etc. CMfg can provide users with on-demand, safe and reliable application services in the whole life-cycle of products through network. It aims to achieve the agile, service-oriented, green and intelligent manufacturing, is a new phase of networked manufacturing, as well as the materialization of service-oriented manufacturing.

At presents, CMfg is in the early stage and many related key technologies are urgent to be solved, among which manufacturing capability (MC) sharing is one of the most important ones in CMfg. Because of its complexity and less relevant studies, there is no clear definition of MC currently. So far, although the concept and connotation of MC is proposed in voluminous literature, there are no conclusive answers yet. For example, Skinner proposed the concept of MC in 1969 firstly. In his opinion, MC is a concept comprising many elements, such as cost, delivery time, quality, and the relationship between them. Richard (1973) considered that the capability includes knowledge, skills, and experience of enterprises. Khalid(2002) concluded that MC is an effective integration of related resources in the process of achieving expected target task. MC reflects the completion of manufacturing objective, and it is a performance level of the standard which is pre-sat by working organization [Andreas Grobler, 2006]. Keen(2000) commented that MC is the integration of intangible and tangible resources, where the tangible resources include labor, capital, facilities and equipment, and the intangible resources include information, procedures, equipment and the organizational system. The relationship between MC and enterprise performance is primarily discussed from the perspective of achieving low operating costs and high product quality [Siri Terjesen, 2011].

Combined with above definitions about MC, according to task demands and resource characteristics of CMfg, in this paper, MC is considered as a kind of capability which can be formally represented in the manufacturing activities. MC, as an important kind of resources in CMfg system, is a manufacturing process, which reflects the capability of completing a manufacturing task or experiment supported by related manufacturing resources and knowledge. According to the concept and characteristics of MC in CMfg, MC can be divided into design capability (DC), simulation capability (SC), product capability (PC), and many other capabilities related to life-cycles of complex products. MC is closely related to manufacturing activities and manufacturing resources, and can only be reflected in concrete activity tasks and resource elements.

In order to achieve on-demand use of MC in CMfg, the core technology about the formal description of MC needs to be solved firstly. Through the formal description of MC, it can realize not only the function sharing of resources, but also the sharing of the experience and knowledge in the manufacturing process, such as manufacturing flow, sensor data, experience of staff, and so on. The existing description languages of resources or services (such as OWL-S and WSMO) are difficult to meet the needs for the formal description of MC in CMfg because of the complexity and dynamic characteristics of MC. A new description language of MC (MCL) based on DL is presented here. The structure of the paper is structured as follows. The following section systematically analyzes and summarizes the characteristics and theoretical basis of current services description language. In section 4, description model of MCL is presented firstly, and then, the focus is shift to the static and dynamic information models based on DL of MC in combination with practical examples.

2. RELATED WORK

2.1. Description logics

Description logic (DL)[Frank Wolter *et al.* 2000, Monika, 2004] is a formal languages for knowledge representation particularly suitable for ontology. It is more expressive than propositional logic but has more efficient decision problems than first-order predicate logic. DL is used in artificial intelligence for formal reasoning on the concepts of an application domain. Importantly, it provides a logical formalism for ontology and the semantic web. A description logic (DL) model includes the following elements: concepts, roles, individuals and their relationships, where concepts are the integration of individuals, roles reflect the binary relation between different individuals.

Static field knowledge can be effectively represented and reasoned by DL, but it is difficult to satisfy the representation of the dynamic feature knowledge of service or action, while some elements such as "state" or "possible" not exist in DL. According to the above issues, some scholars extend the description logic based on action theory, dynamic logic and other related theories, and propose the dynamic description logic (DLL). DLL can not only describe static domain knowledge based on DL, but also achieve the representation and reasoning of action behavior based on the above knowledge. Basic symbols of DDL are as follows[SHEN Guohua, 2008]:

- Concept names: $C_1, C_2, \cdots$;
- Roles name: $R_1, R_2, \cdots$;
- Individual constant: a，b，c…;
- Individual variable: x，y，z…;
- Concept operation: $\neg, \cap, \cup$ and quantifier: $\exists$，$\forall$；
- Formula operation: $\neg, \wedge, \rightarrow$ and quantifier: $\forall$;
- Action name: $A_1, A_2, \cdots$;
- Action construction: such as ；(synthesis)，$\cup$（synchronization），*（repeat），?（switch）；
- Action variable: $\alpha, \beta, \cdots$;
- State variable: u, v, w,…。
- State change: $u \rightarrow v$

Definition 1 (Concept) [Frank Wolter *et al.* 2000]. In DDL, related concept are defined as follows：

(1) *Atomic* concept as P, *all* concept as $\top$ and *empty* concept as $\bot$;

(2) If C and D are concepts，then the following are concepts: $\neg C, C \cap D, C \cup D$;

(3) If R is a role and C is a concept , then $\exists R.C, \forall R.C$ are concepts;

(4) If C is a concept, α is a action, then the $[\alpha]C$ is also a concept.

Definition 2 (Action) [Frank Wolter *et al.* 2000]. A description of action like the form of expression: $A(x_1 ...... x_n) \equiv (P_A, E_A)$, where:

(1) A is a action name, it indicates the symbolic of action;

(2) $x_1 ..... x_n$ are individual variable, indicating the operation object of action. It is also called operation variable;

(3) P_A is a set of pre-conditions, it indicates that conditions must be satisfied before the action execution, form as:

$$P_A = \{con \mid con \in condition\};$$

(4) E_A is a set of post-conditions, it indicates the result set after the action execution, it can be divided into two subsets: *head* and *body*, where *head* = $\{con \mid con \in condition\}$, *body* are also conditions, if each condition of *head* can be satisfied in statement *u*, then each condition of *body* will be satisfied in statement *v*.

Definition 3 (Action concept) [Frank Wolter *et al.* 2000]. Action of DDL is defined as follows:

(1) Atomic action $A(\alpha_1, \cdots, \alpha_n)$ is a action;

(2) if α and β are action, then the following are concepts: $\alpha; \beta, \alpha \cup \beta, \alpha^*$;

(3) if ϕ is assertion formula, then ϕ ? is also a action.

2.2. Service description language

The mainstream description languages of service and resources in current are WSMO and OWL-S.

2.2.1. WSMO

WSMO [JIANG Zhixiong, 2008] is a conceptual model describing Semantic web services. Through its substance, it describes different aspects of web services, such as discovery, excitation, assemble and so on. Among the different aspects, WSMO focuses on solving the Web Services Interoperability problem. It mainly includes the following four elements:

(1) Ontology Ontology supplies the standardized definitions and descriptions of the message used in other elements, it concludes the conception, relation, function, axiom, the illustration of conception and relation, and some non-function attributes;

(2) Service Service describes the released Service semantic layer's functional attributes, and describe the way semantic web service communication and combination.

(3) Objective describes the services' objective type, describe the related users' request.

(4) Mediator describes the mapping relations among WSMO components' objective, and how to coordinate and interact among different WSMO elements.

WSMO is based on framework logic (F-logic), with the emphasis on describing the interoperability among the elements in service. WSMO, however, lacks the characterization of service's state and description of users' interaction. Moreover, the description of complex relationship between the elements in service is not enough either.

2.2.2. OWL-S

OWL-S [JIANG Zhixiong, 2008] is based on web ontological language (owl), which is recommended by W3C. Owl-s characterizes web service mainly in serviceprofile, processmodel and servicegrounding. Serviceprofile, which is similar to the yellow pages of service, describes the service's function and relative properties. Processmodel characters reflect service's process model, describe how a service works, and servicegrounding combines the process model, communication protocol and message format, and describes how to visit a service. The owl-s is based on description logic theory. owl-s cannot characterize the expression and ratiocination of services' dynamic natures' knowledge because description logic can only deal with the knowledge and express of service's static field, can not reflect the complex logic relationship of service and their dynamic characteristic.

In summary, ontology-based semantic web service description has been widely recognized, but there still exist the following deficiencies:

(1) It cannot describe service behavior's state transition and functional constraint information. The functional constraint information reflects only function's static information of service, ignores the description of service dynamic QoS attributes. Because of the above mentioned drawbacks, it is hard to guarantee the comprehensive description, accurate match as well as composition of service;

(2) The description of services' ongoing interaction is insufficient. The complex service cannot be completed only through once interaction since it may need a series of interactions with requester or other service.

(3) It is lack of characterization of the relation among composition elements and services since DL could only describe the binary relation for which simply relation is expressed by tree structure. Manufacturing capability service is multi-level and multi-dimensional with the emphasis on effective integration of resources of all kinds, so how to describe inside and outside complex logical relationships of service is one of the main problems of current service describe language.

In short, this paper uses the current research achievement related to resource and service description, combines the features and requirements of manufacturing capabilities in cloud manufacturing mode, uses dynamic description logical relevant theory, studies description method mainly according to MC's static information and dynamic behavior models.

3. THE DESCRIPTION MODEL OF MCL

According to the concept and classification of MC, the description model of MC is proposed as shown in fig 3. The top-level description model of MC as follows:

MC= (State/RelationSet, ObjectiveSet, QoSSet)

(1) State/relationSet It mainly describes various complex logic relationships among elements of MC in the task execution process, and different states of MC, such as available, busy idle state, etc.

(2) ObjectiveSet It is the core part of MC description model, in which the task object of MC and involving such matters as resources, execution process, performance and so on are expressed.

(3) QoSSet It is the integration of all level services quality of MC, such as business-level QoS, service-level QoS, and so on.

Where *ObjectiveSet* can be further subdivided and abstracted as follows:

ObjectiveSet = (InternalRelation, StaticAttribute, DynamicBehavior, ServiceQoS)

In the above description model, *Internal Relation* is the logical relationship among inside elements of MC, it is also the subset of *State/relationSet* refer to top-level description model of MC. *StaticAttribute* mainly describes static attribute of MC including basic function information, resources description information, historical cases information and so on. *DynamicBehavior* is mainly describes the dynamic information in the execution process of MC, such as operation process, completion situation, state transformation, timing interaction, etc. *ServiceQoS* is the service-level QoS description of MC; it is the subset of *QoSSet* in the above top-level description model of MC

.

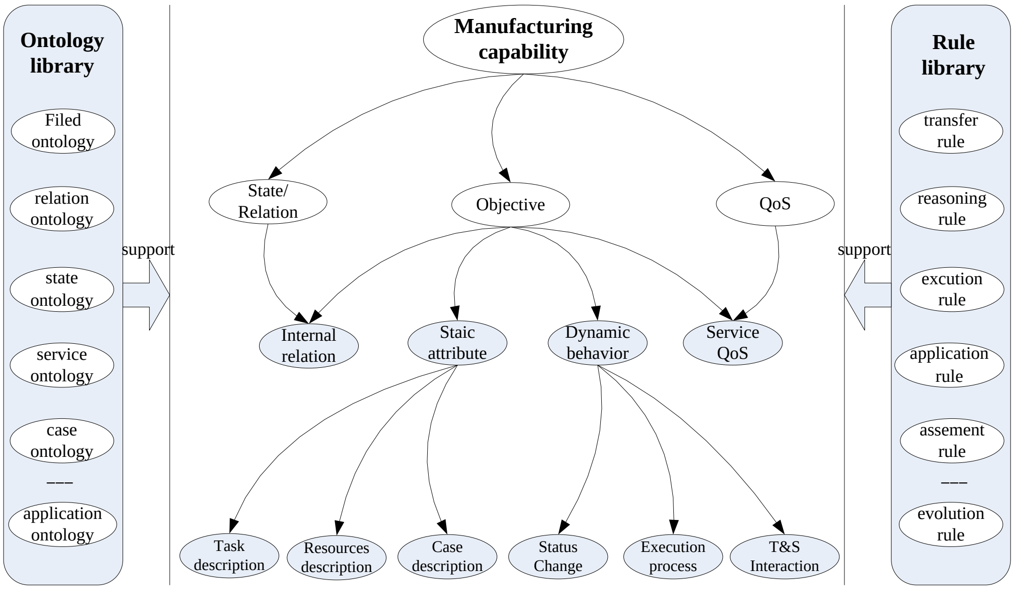

Fig.3. The description model of MC

In addition, ontology library and rule library are built in order to provide a support for the formal description and application reasoning of MC as shown in fig 3. In which ontology library includes filed ontology, state ontology, application ontology and basic terms of MC such as concepts, roles and relations, it is the precondition for achieving semantic match of MC. Rule library mainly includes various rules related to application process of MCS, such as transfer rule, reasoning rule, execution rule, assessment rule, evolution rule, etc. It provides a basis for realizing all kinds of intelligent reasoning.

This paper mainly studies on static attributes and dynamic behavior of MC as follows.

3.1. Static attribute modeling

In this section, static attribute framework of MC based on DL is proposed by combining "from top to bottom" level description method. The top level is static attribute description model of MC, which is defined as follows:

Definition 4 (MC static description model) the static attribute description model of MC (MC_SD) can be abstractly represented by a triple as follows:

$$MC\_SD= (Objective, Resources, MCase)$$

Where *Objective* is the manufacturing task, it is the basic information description set of expected target of MC. *Resources* generally are the various resources related to perform some tasks or activities. *MCase* includes relevant information related to cases of MC. All the above expression are concept of ontology, and expressed based on DL as follows:

$$MC\_SD ::=(\exists\, hasObjective.\ Objective) \cap$$
$$(\exists\, hasResources.Resources) \cap (\exists\, hasMcase.\ Mcase)$$

Above concepts in definition 4 could be further defined by following defines:

Definition 5 (Objective) Mission objective of MC could be expressed by a triple is defined as:

$$Objective= (Basicinfo, Requirement, Goalfeature)$$

Where *Basicinfo* mainly describes the basic information of one task, such as task name, function introduction, category, transaction mode, accounting policy, etc. *Requirement* reflects the basic conditions and relevant constraints involved in the process of one task. *Goalfeature* mainly describes the feature of the expected target oriented to a task, for example, in order to product a component by a manufacturing task, which *Goalfeature* means the various information about the component including quality, performance, cost, delivery time and so on.

This model can be represented by DL such that:

$$Objective ::=(\exists\, hasBasicinfo.Basicinfo) \cap$$
$$(\exists\, hasRequirement.Requirement) \cap$$
$$(\exists\, hasGoalfeature.Goalfeature)$$

Definition 6 (Resources) manufacturing resources could be defined by a binary, and then expressed by DL as follows:

$$Resources= (MajorRs, AuxiliaryRs, HumanRs)$$
$$Resources ::=(\exists\, hasMajorRs.\ MajorRs) \cap$$
$$(\exists\, hasAuxiliaryRs.\ AuxiliaryRs) \cap (\exists\, has\ HumanRs.\ HumanRs)$$
$$MajorRs::=(\exists\, MRsBasicinfo.\ Basicinfo) \cap$$
$$(\exists\, MRsFeature.Feature) \cap$$
$$(\exists\, MRsAwareinfo.Awareinfo)$$
$$AuxiliaryRs::=(\exists\, ARsBasicinfo.\ Basicinfo)$$
$$HumanRs::=(\exists\, HRsBasicinfo.Basicinfo) \cap$$
$$(\exists\, HRsSkill.\ K\&Skill) \cap$$
$$(\exists\, HRsAchievement.Achievement)$$

Where *MajorRs* is the major resources of MC, for example, the *MajorRs* of software MC is the manufacturing software, On the contrary, other resources relevant software's MC called auxiliary resources (*AuxiliaryRs*) such as computing resources. This paper mainly studies on *MajorRs*. It consists of *MRsBasicinfo*, *MRsFeature and MRsAwareinfo*, in which *MRsBasicinfo* describes the basic information of major resources, for instance, equipment model, production area, buying time, etc. *MRsFeature* describes the detail parameters of resources including equipment process, outline dimension and so on. *MRsAwareinfo* includes perceptive information of equipment in execution, such as idle state, running state and so on. In addition, human resources (*HumanRs*) is defined as a special resources as distinct from *MajorRs* and *AuxiliaryRs* in this model because of knowledge and experience sharing in CMfg. It describes the information of personal and organization related to the manufacturing process, including *HRsBasicinfo (*basic information of human), *HRsSkill* (personnel composition, personnel's seniority, duty division, knowledge skill) and *HRsAchievement*(education, training, awarded, and relevant achievement).

Definition 6 (*MCase*) Case of MC is represented by a triple such that:

MCase=(Mcasebasicinfo, Accomplishment)

Where *Mcasebasicinfo* is the basic information of case, it includes user requirement, completion time and others. *Accomplishment* includes more complete information of one task. It is formulated by DL as:

MCase::=(∃hasMcasebasicinfo. Mcasebasicinfo) ⊓ *(∃has Accomplishment. Accomplishment)*

3.2. Dynamic behavior modeling

Definition 7 (Constraints) In DDL various constraints like $\exists C, C(p), R(p.q), p=q$ and $p \neq q$, where $p,q \in N_i$

- As $p,q \in N_i$, each constraint has corresponding assertion formula, such that: $p=q$ and $p \neq q$ denote respectively, individual p equal q and individual p not equal q in knowledge base.
- To integer, real number, constraints can be expressed as mathematical equations and inequalities. For example, $p=q+n$, $p>q+n$, where $p,q \in N_i$ n is a constant.

Definition 8 (Action behavior model of MCS) Action behavior description models of MCS based on DDL is defined as:

$$MCS\_AB(\eta_1, \eta_2, \ldots, \eta_n) = (P_S, E_S)$$

Where

- *MCS_AB* is the action name of MCS, it is the action's unique identifier;
- $\eta_1, \eta_2, \ldots, \eta_n$ are operation variable, are indicate the object of operation;
- P_S is a set of requirement conditions, it indicates that equirement conditions must be satisfied before the action execution, form as: P_S={Res|Res∈Requirement }
- E_S is a set of result conditions, it indicates the result set is generation after the action execution, it is the set of constraint subset: *head* and *body*, where *head={h|h ∈ Requirement }，body={b|b ∈Requirement }*

Definition 9 (Execution process description model of MCS) If there are two processes A and B, includes four types : Sequence(；), Synchronization(⋃), Iterate(*), Switch(?), iff the follow constraints are satisfied:

- Sequence(A,B)|=A;B, satisfy
  $\{\langle u,w\rangle \mid u,v,w \in W, S_A[u \to v] \wedge S_B[v \to w]\}$
- Synchronization(A,B)|=A⋃B, satisfy
  $\{\langle u,v\rangle \mid u,v \in W, S_A[u \to v] \vee S_B[u \to v]\}$
- Iterate(A)|=A*, satisfy
  $\{\langle u,v\rangle \mid u,v \in W, S_A[u \to v] \vee S_A[u \to v] \vee \cdots\}$
- Switch(B)|=B?, satisfy
  $\{\langle u,u\rangle \mid u \in W, u \models B\}$

Definition 10 (State description model of MCS) State description of MCS is composition of all assertion such as individuals, attributes and relations of each state. Let *w* is the set of assertion formula, define $w=\{\varphi_1, \ldots, \varphi_n\}$, and $I(w)=\{\Delta^I, C_i^{I(w)}, \ldots, A_j^{I(w)}, \ldots, E_k^{I(w)}\}$ is the interpretation of *w*, is also called model of *w*, abbreviated as: $I(w)=\{\Delta^I, \bullet^{I(w)}\}$ .Where Δ^I is the domain of interpretation, $\bullet^{I(w)}$ is the function of interpretation. Atomic concept C_i maps a set $C_i^I \subseteq \Delta^I$, relation A_j maps a set $A_j^I \subseteq \Delta^I \times \Delta^I$, every individual maps a set $E_j^I \subseteq \Delta^I$

Definition 11 (State transformation model of MCS) Let two interpretation models of state *u* and state *v* in MCS: $I(u)=\{\Delta^I, \bullet^{I(u)}\}$, $I(v)=\{\Delta^I, \bullet^{I(v)}\}$,then let MCS $S=(P_S, E_S)$,there exists a state transformation: $u \to v$,and dynamic attribute γ_i will be changed accordingly, iff it satisfy the following conditions:

- $I(u) \models \alpha_1 \wedge \alpha_2 \wedge \ldots \wedge \alpha_i \wedge \ldots$, where $\alpha_i \in P_S$;
- Every ordered pair *head/body* in set E_S, satisfy: $I(u) \models \varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_i \wedge \ldots \Rightarrow I(v) \models \phi_1 \wedge \phi_2 \wedge \ldots \wedge \phi_i \wedge \ldots$, where $\varphi_i \in head$, $\phi_i \in body$;
- There exists MCS attribute set: $Attribute=(Satt, Datt)$ of *S*, where *Satt* is the set of static attribute, *Datt* is the set of dynamic attribute, and $\gamma_i \in Datt$, when state of MSC is changed, the γ_i will be changed accordingly.

So it expresses the state *v* generated from *u* in *S*, and defined as: $S[u \to v \mid \gamma_i]$.

4. CONCLUSIONS

Manufacturing capability sharing is a core of CMfg philosophy. Meanwhile, formal description of MC is the key technologies to achieve on-demand use. This paper studies the description method of MC based on description logics by combining the concept and characteristics of MC, and then the static information and dynamic information models based on DL of MC are mainly investigated. In the future, a prototype of MC formal description will be developed according to above proposed methods and related technologies.

ACKNOWLEDGMENTS

This paper is partly supported by the Doctoral Fund of Ministry of Education of in China (20101102110009), the National High-Tech. R&D Program (2011AA040501),China.

REFERENCES

Skinner W, 1969. Manufacturing –missing link in corporate strategy. *Harvard business review*. 6-9, 136-145

Richard G B, 1973. The organization of industry. *Economic Journal*. 82, 883-896

Keen P, Mcdonald M, 2000. *The process edge: creating customer value and business wealth in the internet era*. McGraw –Hill. New York

LI Bohu, ZHANG Lin, WANG Shilong, TAO Fei, et al. 2010. Cloud manufacturing: a new service-oriented manufacturing model. *Computer Integrated Manufacturing Systems*, 6(1), 1-7.

ZHANG Lin, LUO Yongliang, TAO Fei, REN Lei, et al. 2010. Study on the key technologies for the construction of manufacturing cloud. *Computer Integrated Manufacturing Systems*, 16(11), 2510-2520

ZHANG Lin, LUO Yongliang, FAN Wenhui, TAO Fei, REN Lei, 2011. Analyses of cloud manufacturing and related advanced manufacturing models. *Computer Integrated Manufacturing Systems*, 17(3), 0458-0468.

Siri Terjesen, Pankanj C. Patel, Jeffrey G. Covin. 2011. Alliance diversity, environmental context and the value of manufacturing capabilities among new high technology ventures. *Journal of operations management*, 29,105-115

Khalid Hafeez, YanBing Zhang, Naila Malak. 2002. Determining key capabilities of a firm using analytic hierarchy process. *International journal of production economics*, 76, 39-51

Andreas Grobler. 2006. An empirical model of the relationships between manufacturing capabilities. *International journal of operations & production management*, 26(5),458-485

JIANG Zhixiong. 2008. Description of Web services based on dynamic semantic. *FUDAN university doctoral dissertation*.

Monika Solanki, Antonio Cau, Hussein Zedan. 2004. Augmenting semantic web service descriptions with compositional specification. I*n proceedings of the 13th international conference on world wide web*, ACM press,544-552

Frank Wolter, Michael Zakharyaschev. 2000. Dynamic description logics. *Advances in model logic*.

SHEN Guohua. 2009. Research on modeling and reasoning about web services based on description logics. *Nanjing university of aeronautics and astronautics doctoral dissertation.*

AUTHORS BIOGRAPHY

Yongliang Luo received the B.S. degree and the M.S. degree in 2006 and 2009 from the Department of Computer Science at Shandong University of Science and Technology, China. He is currently working for the Ph.D. degree in Modeling simulation theory and technology at Beihang University. His research interests include service-oriented manufacturing and integrated manufacturing systems.

Lin Zhang received the B.S. degree in 1986 from the Department of Computer and System Science at Nankai University, China. He received the M.S. degree and the Ph.D. degree in 1989 and 1992 from the Department of Automation at Tsinghua University, China, where he worked as an associate professor from 1994. He served as the director of CIMS Office, National 863 Program, China Ministry of Science and Technology, from December 1997 to August 2001. From 2002 to 2005 he worked at the US Naval Postgraduate School as a senior research associate of the US National Research Council. Now he is a full professor in Beijing University of Aeronautics and Astronautics. He is an Editor of "*International Journal of Modeling, Simulation, and Scientific Computing*", and "*Simulation in Research and Development*". His research interests include integrated manufacturing systems, system modeling and simulation, and software engineering.

Fei Tao is currently an associate professor at Beihang Universigy since April 2009. His research interests include service-oriented manufacturing, intelligent optimizaiton theroy and algorithm, resource servcie management. He is author of one monograph and over 20 journal articles of these subjects. Dr. Tao was awarded the Excellent Doctoral Dissertation of Hubei Province, China and was elected to be a research affiliate of CIRP in 2009. He is the founder and editore-in-chief of *Internatioanl Journal of Servcie and Computing-Oriented Manufacturing* (IJSCOM).