

## A Survey on Application of Situation Calculus in Business Information Systems

Bing Li and Junichi Iijima

*Graduate School of Decision Science and Technology,  
Tokyo Institute of Technology, Japan  
{li.b.ab, iijima.j.aa}@m.titech.ac.jp*

### Abstract

*Business needs are expanding with evolution of the global economy. Customer-centric concepts are gradually becoming the main driver of businesses and will thus determine the development of business information systems. How to fully satisfy customer needs requires that these needs should be processed in a timely fashion and moreover intelligence should be enabled so as to solve some problems beyond the capabilities of human processing. The situation calculus – as a widely used formalism in AI fields – is just such an appropriate method that can bring intelligence to business systems. This paper attempts to give a survey on application of the situation calculus in business information systems.*

### 1. Introduction

With evolution of the global economy, business needs are greatly expanding by comparing with the traditional economies. Customer-centric or customer-oriented concepts gradually become the driving force of businesses and customers are more involved in the production phase of goods or services [1, 2, 3]. Meanwhile, how to satisfy various customer needs in this new economy is greatly challenging the current business information systems.

Similarly with the ICT trends explained in [4], the traditional ways to provide products or services to customers are mostly one to many, i.e., one product or service will satisfy a mass of customers. But in the coming era of information, e-services will occupy a remarkable portion of businesses. Many services will be integrated to serve a customer. The traditional ways of problem solving, usually manually processed by some experts, will be obstacles to provide services to customers in a timely fashion because of the general scarcity of experts and the shrinking time to market (TTM). Furthermore, the exponentially increasing

amount of information will go beyond the capabilities of human processing.

Generally speaking, intelligence should be enabled in business information systems so as to solve the problems in the face of human processing, such as delay time, error-prone and limited availability. But, as we know, it is hard work to build a comprehensive intelligent system that can completely implement human-level intelligence. Although such a system is still in desire, some degree of intelligence has been and will be implemented in business information systems.

This paper focuses on the situation calculus from the perspective of surveying its application in business information systems. The situation calculus is a dialect of logic language and has been widely used in AI related research. Recently this formalism has been applied to various business related systems

The rest of this paper is organized into three sections. Firstly, the situation calculus will be briefly introduced and reviewed in Section 2. In Section 3, applications of the situation calculus will be reviewed from three research fields, including web service composition, context awareness and business process management. Finally, the conclusion is given in Section 4.

### 2. The Situation Calculus

The situation calculus was proposed by John McCarthy in 1969 [5]. The basic concepts were further developed by Reiter in [6] and extended by subsequent research efforts [7, 8, 9, 10]. In this section only basic concepts and reasoning mechanism will be briefly reviewed so as to obtain effective understanding about the applications of this formalism in business systems. For comprehensive understanding about the situation calculus, readers can refer to the books [11, 12] and other related papers mentioned above.

## 2.1. Basic Concepts

The situation calculus is a dialect of logic language. Although the extended theories make it be second-order and many-sorted logic, most of its concepts can be reduced and represented in first-order logic [11].

The situation calculus is designed to specify and implement dynamical systems. The strength is its capability of representing and reasoning about actions. The key concepts include *action*, *situation* and *fluent*.

An *action* in the domain world is denoted by a function symbol, for example, *pickup(A)*, *walk(L)*, *check\_credit(CardNo)*.

A *situation* represents the world history, which can be interpreted as a sequence of actions. The constant  $s_0$  is used to denote the initial situation, namely, the empty sequence of actions, which can be interpreted as the initial state of the modeling world.

A predefined binary function symbol *do*: *Action*  $\times$  *Situation*  $\rightarrow$  *Situation*, maps an action *a* and a situation *s* to the successor situation resulting from performing *a* in the situation *s*. This successor situation is denoted as *do(a, s)*, which can be interpreted as the action sequence formed by adding action *a* to the action sequence *s*.

A predefined binary predicate symbol *Poss* is defined as *Poss*: *Action*  $\times$  *Situation*. *Poss(a, s)* means that the action *a* is possible to be executed in the situation *s*.

Besides, functions and relations will be introduced to model the dynamical world. Those whose (truth) values vary from situation to situation are called *fluents*, including relational fluents and functional fluents. For example, *colour(x, s)*, *height(y, s)*, *closeTo(r, x, s)*, *pay\_way(order, credit)*, *order\_status(order, filled, s)*. These fluents can reflect the changes of world states that are the result of named actions. For convenience, those whose (truth) values do not vary with situations can be called *non-fluents*, including relational non-fluents and functional non-fluents.

## 2.2. Basic Reasoning Mechanism

Actions in the domain are specified by precondition and successor state axioms. The precondition axiom of an action states that in what condition the action is possible to be performed. The condition can be represented by relational fluents or non-fluents.

The successor state axioms are also called effect axioms, which represent the effects of performing actions. The effects of performing an action are reflected by the changed truth values of related relational fluents. For example [11], to specify an

action, denoted as *drop(r, x)*. The precondition axiom can be represented as

$$Poss(drop(R, X), S) \equiv holding(R, X, S).$$

The successor state axiom can be

$$broken(X, do(A, S)) \equiv (A=drop(R, X) \wedge fragile(X, S)) \vee (broken(X, S) \wedge A \neq repair(R, X)).$$

From the above example, it can be seen that action specification has a routine format.

The underlying reasoning mechanism can be briefly explained as follows:

From the precondition axiom of an action  $a_1$ , we can know when it is possible to execute, e.g., denoting this situation as  $s_1$ ; then from the successor state axiom, we can infer the successor world situation, denoting as  $s_2$ .

Similarly, there may exist an action  $a_2$  whose precondition axiom is satisfied in the situation  $s_2$ . From the successor state axiom of  $a_2$ , the successor situation of performing  $a_2$  can also be inferred, denoted as  $s_3$ .

Accordingly, we can compose at least two scenarios. Given the initial situation  $s_1$  and the action sequence  $[a_1, a_2]$ , we can infer the successor situation  $s_3$  after performing  $a_1$  and  $a_2$  subsequently. On the other hand, given the initial situation  $s_1$  and the final situation  $s_3$ , we can infer that the action sequence  $[a_1, a_2]$  will be the possible reason or solution to change the situation from  $s_1$  to  $s_3$ . These two scenarios form basic reasoning mechanisms of the situation calculus and can be combined and used to many business applications.

## 2.3. Extended Languages

The basic concepts of the situation calculus are continuously extended to include more dynamic features by the subsequent research efforts [7, 10, 11].

Based on the situation calculus, a programming language – Golog, was developed in [7] to represent complex actions or procedures. Via macro-expansion, Golog provided a high-level programming of robots and software agents, process control, etc. A Golog interpreter in Prolog was provided in [11].

ConGolog extended the situation calculus to include concurrency. This language includes facilities for prioritizing the execution of concurrent processes, interrupting the execution when certain conditions become true, and dealing with exogenous actions [10].

Besides, many other research efforts extend the situation calculus from various aspects such as two recent ones in [13, 14], all of which make up of the whole theoretic set of the situation calculus.

Meanwhile, with extension of the situation calculus, as part of theoretical research, some problems in the face of this formalism are continuously challenged and approached [6, 15, 16], including the frame problem, qualification problem and ramification problems.

Despite of the existence of these problems, the situation calculus could be applied to business systems when some tradeoffs are made between expressiveness and reasoning. Based on the above brief explanation about the situation calculus, we will survey its current applications in business information systems.

### 3. Application of Situation Calculus

Applications of the situation calculus are driven by the business needs that require implementation of a higher degree of autonomy in business information systems. The exponentially expanding needs in the newly emerging economies will make the traditional business systems sluggish to respond to customers. Business intelligent systems are greatly desired so as to satisfy customers with quick responsiveness and high degree of customer orientation.

The situation calculus is widely used in AI related research fields and has been applied to various industrial systems. In this paper three research fields are surveyed in detail, including web service composition, context awareness and business process management. The underlying connections among the three fields are also explained later.

#### 3.1. Web Service Composition

From the perspective of service providers – companies, web service technologies potentially improve the flexibility of business processes and help to build up loosely coupled enterprise systems [17]. From the perspective of service consumers – customers, web service technologies make it possible to satisfy their needs with more ease. But huge numbers of available services will go beyond the manual control of humans, thus automatic composition of web services is needed.

In [18], web service composition approaches are classified into two realms: workflow composition and AI planning. The situation calculus based approaches fall in the latter one, which was proposed and explained in [19, 20, 21, 22, 23].

In [19], it was argued that computer-interpretable description of web services was critical to automating a diversity of tasks, including web service discovery, invocation, composition, monitoring, verification and simulation. The process perspective of web services was emphasized and furthermore such a process model should be described as an ontology of first-order logic. The situation calculus worked as the core theory of the promoted Process Specification Language (PSL) [24], which is used to describe the ontological process model.

In [20, 21], the authors provided the situation calculus based semantics of DAML-S (now OWL-S). By mapping from DAML-S to the situation calculus, web services are precisely represented and the situation calculus axiomatization will enable more analysis by employing various analysis techniques such as Petri Nets.

Golog language [7, 11] was adapted and extended to enable programs that could be generic, customizable and usable in the context of the Web [22]. A new distinguished fluent in the situation calculus was introduced, which was called *Desirable(a, s)*, i.e., the action *a* is desirable in the situation *s*. Furthermore, action is executable when satisfying both *Poss(a, s)* and *Desirable(a, s)*. In this way user constraints can be introduced.

An intelligent infrastructure was proposed in [25] that could satisfy dynamic user constraints for web service composition. The situation calculus and generic exception patterns were employed. This infrastructure enabled translating user desires into computer processable information, decomposing complex user demands into primitive tasks and then deducing the actions required to achieve the user's demands. However, great research efforts were further needed in order to implement higher degree of intelligence.

In the realm of AI planning, besides the situation calculus, there are other research approaches. Hierarchical Task Network (HTN) planning was employed for web service composition and OWL-S service descriptions were transformed to a domain of SHOP2 (a HTN planning system) and correctness of the algorithm was also proved by showing the correspondence to the situation calculus semantics of OWL-S [26, 27, 28].

Software synthesis and composition methods were applied to web service composition in [29] and their work was based on similarities between web service composition and component-based development in software engineering.

#### 3.2. Context Awareness

Several terms are related to the concept of context awareness such as situation awareness or dependency. These concepts attempt to conceive and represent the environmental information, and accordingly enable making rapid and appropriate decisions, and taking effective actions. These concepts can be integrated into a more general framework of situation management [30], which can be applied to the management of complex and dynamical systems. Some theories can be used to specify and implement situation management, including situation awareness [31], situational control [32] and situation calculus [11]. With penetration of

ICT in the routine businesses, it is promising to apply these approaches to dynamically manage business services and the underpinning information systems that implement or at least support these services. These concepts have been applied to implement some intelligent systems in the ubiquitous environment.

An infrastructure of context awareness was proposed based on first order logic in [33]. Context is modeled by first order logic, which enables the precise expression of contextual information and automates the inductive and deductive reasoning about context. They envisioned and built up a smart space by using the devised infrastructure – Gaia [34], in which applications can be customized based on the space context. Their work showed the application of context awareness in the ubiquitous computing environment. The situation calculus theories were originally based on the first order logic [5, 6]. In this sense, the work in [33, 34] also showed the feasibility of applying the situation calculus in context awareness and the capability of action reasoning can also be effectively used in this field.

As a typical application of context awareness, how to provide personalized mobile services is an active research field [35, 36, 37, 38]. The concept of situation dependency was proposed in the field of mobile commerce in [35]. The services should be adapted to the customer's context, including spatial, temporal and personal information. Situation dependency was applied to mobile services and a prototype system was also implemented in [36, 37]. An architecture was proposed based on the situation calculus and mobile users could call on server to offer situation dependent services that would dynamically respond to their changing situations. The server embedded with a Prolog engine performed the inference work, i.e., reasoning about the user's contextual information, such as location and personal profile, and providing the individual-matched services to users.

In the current research on context awareness, context modeling or situation modeling seems to be the critical problem. How to efficiently and effectively model the contextual information will determine the success of industrial applications of these approaches. The ontological approaches attempted to tackle this problem in [39, 40].

### 3.3. Business Process Management

Business process management covers the whole life cycle of business processes, including process design, simulation, enactment, monitoring and control, diagnosis, etc [41]. Application of formal methods in business process management systems is critical to ensure correctness properties of business process

definition and furthermore enables the potential analysis [41, 42, 43]. The situation calculus can be applied to formally specify and analyze business processes by considering the intuitive mapping from an activity in a process to an action in the situation calculus domain.

Application of the situation calculus in business process simulation was firstly proposed in [44] and was used to enable formal analysis to test consistency in process models. This approach was later extended to a formal framework for enterprise and business process modeling in [45, 46]. The situation calculus was used to represent the operational enterprise knowledge. A formalized enterprise model consisted of some submodels, including organizational and process model. In this way, business processes and related concepts can be defined precisely and rigorously. Moreover, analysis and reasoning about these concepts can be enabled.

Recent work of applying the situation calculus to business process management can be found in [47, 48, 49]. This work attempts to bridge the gap between the informal model used in the industry and the formal model in the academia society. The XML Process Definition Language (XPDL) [50] and the situation calculus were selected as the specification languages. A prototype system was designed and developed to perform the automatic transformation from an XPDL-defined process specification to the Prolog program that logically specifies the process in the situation calculus. The Prolog-defined process model worked as an inferable model that was available to potential analysis, including verification and planning.

The above approaches based on the situation calculus are still focusing on the business process modeling or designing phase. The situation calculus is used to perform formal representation and verification. More intelligent functions, such as process planning and diagnosis, could be extended in the future work. With expressiveness and reasoning capability of the situation calculus, these functions can potentially be implemented and thus business information systems with high-level autonomy can be obtained.

### 3.4. Something in Common

There is something in common for the above three fields that apply the situation calculus. Firstly, they are in face of modeling a dynamical domain that changes continuously. For the case of web service composition, it is obvious that the domain model should be modified when the web system introduces a new service available to customers. Similarly, a process model should be changed when a new activity is added into the process. For the case of context awareness,

contextual and situational information vary with time, location, person, etc.

Secondly, intelligence and autonomy are greatly desired so as to adapt to the ubiquitous computing environment. Huge numbers of web services need to be processed automatically and thus composite services could be provided to customers in a timely fashion. Contextual information should be reasoned and thus an appropriate and satisfactory service could be available to users. Business process management systems need to be robust and intelligent so as to rapidly design and implement effective and efficient processes and thus enable the flexibility of the whole enterprise systems.

Thirdly, service oriented concepts are underpinning these three fields. This is obvious for web service composition. Context awareness will be implemented by integrating some services from context or situation provider and generating a composite service after context reasoning. The recent evolution of business process management systems are technically enabled by the development of service oriented architecture technologies.

The situation calculus theories can provide the requirements mentioned above. Originally designed from first order logic and continuously extended to include dynamical features, these theories are capable of modeling dynamically changing systems. In some degree, the underlying reasoning mechanisms are sufficient to implement intelligent systems that could be put into practice. The concept of services can be represented by actions in the situation calculus with more precision. Application of the situation calculus will help to implement dynamic management of business information systems.

Besides the above three fields, many other research fields can also apply the situation calculus to build up practical business systems, including multi agent systems, autonomic systems, network management systems, etc. To some extent, these applications also have some features explained above.

## 4. Conclusion

In this paper, we survey the situation calculus from the perspective of applications in business information systems. The situation calculus theories are generally reviewed by explaining the basic concepts, reasoning mechanism and some extension. Then, three fields are surveyed that have employed the situation calculus to design and develop intelligent applications. Moreover, the common things that could link these three fields are explained.

It will take a long way to implement the ambitious vision of comprehensive intelligence in business

information systems. The theoretical and practical research on the situation calculus will continuously make contributions to this work. Some critical research efforts will include ontological representation of domain concepts, transformation between different models specified in different languages, and further development of expressive representation and effective reasoning.

## References

- [1] Tien, J.M., Berg, D., "A Case for Service Systems Engineering," *Journal of Systems Science and Systems Engineering*, Vol. 12, No.1, pp.13–38, March, 2003.
- [2] Maglio, P.P., Srinivasan, S., Kreulen, J.T., Spohrer, J., "Service Systems, Service Scientists, SSME, and Innovation," *Communications of the ACM* 49 (7), pp. 81-85, 2006.
- [3] Paloheimo, K., Miettinen, I., Brax, S., *Customer-Oriented Industrial Services*, Helsinki University of Technology, BIT Research Centre, 2004.
- [4] Weiser, M., Brown, J.S., "The coming age of calm technology," *Beyond Calculation: the Next Fifty Years*, Denning, P. J., Metcalfe, R. M. (Eds.), Copernicus, New York, NY, 75–85, 1997.
- [5] McCarthy, J., Hayes, P.J., "Some Philosophical Problems from the Standpoint of Artificial Intelligence," in: B. Meltzer, D. Michie (Eds.), *Machine Intelligence 4*, pp. 463–502, Edinburgh University Press, 1969.
- [6] Reiter, R., "The Frame Problem in the Situation Calculus: A Simple Solution (Sometimes) and a Completeness Result for Goal Regression," *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy*, pp. 359–380, Academic Press, San Diego, CA, 1991.
- [7] Levesque, H.J., Reiter, R., Lesperance, Y., Lin, F., Scherl, R.B., "GOLOG: A Logic Programming Language for Dynamic Domains," *Journal of Logic Programming* 31, pp. 59–84, 1997.
- [8] Levesque, H.J., Pirri, F., Reiter, R., "Foundations for the Situation Calculus," *Electronic Transactions on Artificial Intelligence* 2, pp. 159–178, 1998.
- [9] Pirri, F., Reiter, R., "Some Contributions to the Metatheory of the Situation Calculus," *ACM* 46 (3) pp. 325–361, 1999.
- [10] Giacomo, G.D., Lesperance, Y., Levesque, H.J., "ConGolog, A Concurrent Programming Language Based on the Situation Calculus," *Artificial Intelligence* 121, pp. 109–169, 2000.

- [11] Reiter, R., *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*, MIT Press, 2001.
- [12] Brachman, R., Levesque, H., *Knowledge Representation and Reasoning*, Morgan Kaufmann, 2004.
- [13] Demolombe, R., Fernandez, A.M.O., "Intention Recognition in the Situation Calculus and Probability Theory Frameworks," *CLIMA VI, LNAI 3900*, pp. 358–372, 2006.
- [14] Denecker, M., Ternovska, E., "Inductive Situation Calculus," *Artificial Intelligence 171*, pp. 332–360, 2007.
- [15] Thielscher, M., "From Situation Calculus to Fluent Calculus: State Update Axioms as a Solution to the Inferential Frame Problem," *Artificial Intelligence 111*, pp. 277–299, 1999.
- [16] McIlraith, S.A., "Integrating Actions and State Constraints: A Closed-Form Solution to the Ramification Problem (Sometimes)," *Artificial Intelligence 116*, pp. 87–121, 2000.
- [17] Moitraa, M., Ganesh, J., "Web Services and Flexible Business Processes: Towards the Adaptive Enterprise," *Information & Management 42*, pp. 921–933, 2005.
- [18] Rao, J., Su, X., "A Survey of Automated Web Service Composition Methods," *Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC)*, San Diego, California, USA, 2004.
- [19] Gruniger, M., Hull, R., McIlraith, S., "A First-Order Ontology for Semantic Web Services," *Proceedings of W3C Workshop on Frameworks for Semantics in Web Services*, 2005.
- [20] Narayanan, S., McIlraith, S., "Analysis and Simulation of Web Services," *Computer Networks 42*, pp. 675–693, 2003.
- [21] Narayanan, S., McIlraith, S., "Simulation, Verification and Automated Composition of Web Services," *Proceedings of the Eleventh International World Wide Web Conference*, Honolulu, Hawaii, 2002.
- [22] McIlraith, S., Son, T., "Adapting Golog for Composition of Semantic Web Services," *Proceedings of the Eighth International Conference on Knowledge Representation and Reasoning*, Toulouse, France, 2002.
- [23] McIlraith, S., Son, T., Zeng, H., "Semantic Web Services," *IEEE Intelligent Systems 16(2)*, pp. 46–53, 2001.
- [24] Gruninger, M., Menzel, C., "Process Specification Language: Theory and Applications," *AI Magazine 24*, pp. 63–74, 2003.
- [25] Nariai, K., Paik, I., Shinozawa, M., "Planning and Composition of Web Services with Dynamic Constraints Using Situation Calculus," *Proceedings of The Fifth International Conference on Computer and Information Technology (CIT)*, 2005.
- [26] Sirin, E., Parsia, B., Wu, D., Hendler, J., Nau, D., "HTN Planning for Web Service Composition Using SHOP2," *Journal of Web Semantics*, Vol.1, No. 4, pp. 377–396, 2004.
- [27] Wu, D., Parsia, B., Sirin, E., Hendler, J., Nau, D., "Automating DAML-S Web Services Composition Using SHOP2," D. Fensel et al. (Eds.): *ISWC 2003, LNCS 2870*, pp. 195–210, 2003.
- [28] Nau, D., Au, T., Ilghami, O., Kuter, U., Murdock, J., Wu, D., Yaman, F., "SHOP2: an HTN Planning System," *Journal of Artificial Intelligence Research 20*, pp. 379–404, 2003.
- [29] Matskin, M., Rao, J., "Value-Added Web Services Composition Using Automatic Program Synthesis," *Proceedings of the International Workshop on Web Services, E-Business, and the Semantic Web*, CAiSE, Canada, 2002.
- [30] Jakobson, G., Buford, J., Lewis, L., "A Framework of Cognitive Situation Modeling and Recognition," *Proceedings of Military Communications Conference (MILCOM)*, 2006.
- [31] Endsley, M.R., "Toward a Theory of Situation Awareness in Dynamic Systems," *Human Factors 37(1)*, pp. 32–64, 1995.
- [32] Pospelov, D.A., *Situation-Driven Control: Theory and Practice*, Moscow, Nauka, 1986.
- [33] Ranganathan A., Campbell R.H., "An Infrastructure for Context-Awareness Based on First Order Logic," *Personal Ubiquitous Computing*, Vol. 7, No. 6, pp. 353–364, 2003.
- [34] Román, M., Hess, C.K., Cerqueira, R., Ranganathan, A., Campbell, R.H., Nahrstedt, K., "Gaia: A Middleware Infrastructure to Enable Active Spaces," *IEEE Pervasive Computing 10-12*, pp. 74–83, 2002.
- [35] Figge, S., "Situation-Dependent Services – A Challenge for Mobile Network Operators," *Journal of Business Research, Mobility and Markets: Emerging Outlines of M-Commerce*, Vol.57, No.12, pp.1416–1422, 2004.
- [36] Ding, X.J., Iijima, J., "A Study on Implementing Situation Dependent Services Based on Logical Approach," *the Journal of Japan Society for Management Information (JASMIN)*, Vol. 16 No.1, pp. 49–62, June 2007.
- [37] Ding, X.J., Iijima, J., "Logical Approach for Dynamic System in Situation Dependent Services," *Proceedings of the 3rd IEE Mobility Conference*, 2006.

- [38] Anagnostopoulos, C.B., Tsounis, A., Hadjiefthymiades, S., "Context Awareness in Mobile Computing Environments," *Wireless Personal Communications* 42, pp. 445–464, 2007.
- [39] Ranganathan A., McGrath, R.E., Campbell R.H., Mickuna, D.M., "Ontologies in a Pervasive Computing Environment," *Proceedings of Workshop on Ontologies and Distributed Systems*, IJCAI, 2003.
- [40] Chen, H., Finin, T., Joshi, A., "An Ontology for Context-Aware Pervasive Computing Environments," *Special Issue on Ontologies for Distributed Systems, Knowledge Engineering Review*, 2003.
- [41] Van Der Aalst, W.M.P., Ter Hofstede, A.H.M., Weske, M., "Business Process Management: A Survey," *Proceedings of BPM 2003, LNCS 2678*, pp. 1–12, Springer, 2003.
- [42] Van Der Aalst, W.M.P., Ter Hofstede, A.H.M., "Verification of Workflow Task Structures: a Petri-Net-Based Approach," *Information Systems*, Vol. 25, No.1, pp. 43–69, 2000.
- [43] Van Der Aalst, W.M.P., Van Hee, K.M., *Workflow Management: Models, Methods, and Systems*, MIT Press, 2002.
- [44] Plexousakis, D., "Simulation and Analysis of Business Processes Using GOLOG," *Proceedings of Conference on Organizational Computing Systems*, pp. 311–322, 1995.
- [45] Koubarakis, M., Plexousakis, D., "A Formal Model for Business Process Modeling and Design," *Proceedings of CAiSE 2000*, pp. 142–156, 2000.
- [46] Koubarakis, M., Plexousakis, D., "A Formal Framework for Business Process Modelling And Design," *Information Systems* 27, pp. 299–319, 2002.
- [47] Li, B., Iijima, J., "Bridging the Gap Between XPD L and Situation Calculus: A Hybrid Approach for Business Process Verification," *Proceedings of the 5th International Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems (MSVVEIS)*, ICEIS, Funchal, Portugal, 2007.
- [48] Li, B., Iijima, J., "Architecture on a Hybrid Business Process Design and Verification System," *Proceedings of International Symposium on Information System & Management (ISM)*, Shanghai, China, 2007.
- [49] Li, B., Iijima, J., "A Hybrid Approach for Business Process Verification," *Proceedings of The IFIP International Conference on Research and Practical Issues of Enterprise Information Systems (CONFENIS)*, Beijing, China, 2007 (forthcomming).
- [50] WfMC, Process Definition Interface – XML Process Definition Language, Document Number WfMC-TC-1025, Version 2.00, October 3, 2005.