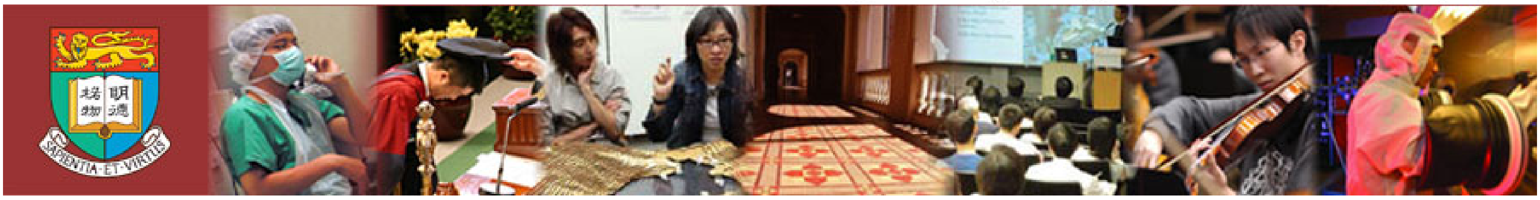

| | |
|---|---|
| **Title** | **Integration of System-Dynamics, Aspect-Programming, and Object-Orientation in System Information Modeling** |
| **Author(s)** | **Wang, J; Liu, D; Ip, W; Zhang, W; Deters, R** |
| **Citation** | **IEEE Transactions on Industrial Informatics, 2014, v. 10 n. 2, p. 847-853** |
| **Issued Date** | **2014** |
| **URL** | **http://hdl.handle.net/10722/202948** |
| **Rights** | **IEEE Transactions on Industrial Informatics. Copyright © IEEE.** |

# Integration of System-Dynamics, Aspect-Programming, and Object-Orientation in System Information Modeling

Junwei Wang, Dong Liu, Wai Hung Ip, Wenjun Zhang, *Senior Member*, *IEEE*, and Ralph Deters

***Abstract*—Contemporary information modeling of enterprise systems only focuses on the technical aspect of the systems, though it is known that they are social-technical (socio-tech) systems in essence. In fact, there are many lessons that can be learned from failures in the management of enterprise systems, which range from a small one (e.g., failure to install a printer driver) to a large one (e.g., nuclear power plant post-accident management). This paper, therefore, proposes that the enterprise system should be viewed as a socio-tech system. The paper presents a novel integrated approach to information modeling of socio-tech enterprise systems. In particular, the approach integrates object-orientation, systems-dynamics (as a means to represent high-level dynamics), and aspect-programming. The paper discusses an example to illustrate how the proposed approach works.**

***Index Terms*—Aspect-orientation, industrial information integration engineering (IIIE), information modeling, social-technical systems (socio-tech), systems-dynamics.**

## I. Introduction

ENTERPRISE systems are, in essence, social-technical (socio-tech) or human–machine systems [1]. Without this understanding, the life cycle engineering and management of socio-tech systems is less understood and thus will be poorly modeled. Contemporary literatures almost regard enterprise systems as technical systems, especially when information modeling of enterprise systems is concerned [20]–[27]. In sociology, human systems are addressed, leading to well-known theories about human, especially human group behaviors and subsequently, the behaviors management. Unfortunately, the two do not communicate with each other well, causing many failures in the management of enterprise systems, with the heading of human factors. Therefore, it is necessary to view enterprise systems as a socio-tech system and to understand their interactions and interdependencies (i.e., constraints) being brought to each other.

The socio-tech enterprise system is very complex because of additional complexity caused by constraints between the socio and technical system [28]–[32]. To manage such a socio-tech system successfully, the first step is to develop an information model of it [2], [3], [5], [14], [15]. An information model is a framework of concepts to be used to express the mini-world semantics [6]. An information model comprises two parts: 1) the structure of a universe of discourse (i.e., systems, subsystems, entities, and their properties and relationships); and 2) how the universe of discourse behaves when it receives some input and/or is subjected to disturbances from the outside of the universe under certain external environments.

Only a few researchers [14], [15] focus on the interdependencies between socio-tech systems and effects of the interdependencies on the systems' behaviors. Illustrative and formal methods were integrated for modeling socio-tech systems in the approaches described in [2], [3], and [5]. These approaches modeled the socio-tech systems from a pure technical point of view and failed to capture semantic elements on the social side. Rasmussen's framework [13] is widely applied for socio-tech systems' analysis and modeling. However, Rasmussen's framework emphasizes on the hierarchical structure in most socio-tech systems from top to down, e.g., government, regulators, company, management, staff, and work.

The challenges of information modeling of socio-tech systems result from the following.

1) Complex couplings between social and technical systems: Quite often, policies and regulations in the social system domain considerably affect the development and operation management in the technical system domain. On the other hand, the emerging of a new technical system could create new demands on adapting and adjusting the social system.

Manuscript received April 30, 2012; revised August 06, 2013; accepted January 02, 2014. Date of publication January 16, 2014; date of current version May 02, 2014. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) through a strategic project grant, and in part by the Innovation and Technology Commission of the Government of the Hong Kong Special Administrative Region (HKSAR) of the People's Republic of China through project ITS/043/11FX. The work of W. Zhang was supported by the East China University of Science and Technology (ECUST) through the Fundamental Research Funds for the Central Universities. The work of J. Wang was supported by the State Key Laboratory of Synthetical Automation for Process Industries, and by the National Natural Science Foundation of China (NSFC) under Grant 71001018. The work of W. H. Ip was supported by the Department of Industrial and Systems Engineering, Hong Kong Polytechnic University (H-ZD88). Paper no. TII-12-0323. *(Corresponding author: W. Zhang.)*

J. Wang is with the Complex Systems Research Center, East China University of Science and Technology, Shanghai 200237, China, and with the Department of Industrial and Manufacturing Systems Engineering, University of Hong Kong, Pokfulam, Hong Kong (e-mail: wangjw1979@gmail.com).

D. Liu is with the National Superconducting Cyclotron Laboratory, Michigan State University, East Lansing, MI 48824 USA (e-mail: liud@frib.msu.edu).

W. H. Ip is with the Department of Industrial and Systems Engineering, Hong Kong Polytechnic University, Hung Hom, Hong Kong (e-mail: mfwhip@inet.polyu.edu.hk).

W. Zhang is with the Complex Systems Research Center, East China University of Science and Technology, Shanghai 200237, China, and with the Advanced Engineering Design Laboratory, University of Saskatchewan, Saskatoon, SK S7N 5A9, Canada (e-mail: Chris.Zhang@Usask.ca).

R. Deters is with the Department of Computer Science, University of Saskatchewan, Saskatoon, SK S7N 5A9, Canada (e-mail: deters@cs.usask.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2014.2300703

2) There is an uncertain and vague relationship between any two social system entities, and between one social system entity and one technical system entity.
3) The unified modeling language (UML) notation of a triangle object and the evolutionary structure of the social system: Quite often, the structure is formed by evolution and interaction, and not by design.
4) The interaction between the social and technical systems involve complex dynamics.

This paper proposes an approach to information modeling of socio-tech systems. The approach is developed on the basis of object-orientation, aspect-orientation, and system-dynamics and integrates them. The remainder of this paper is structured as follows. Sections II and III present the proposed information modeling approach for socio-tech systems. Section IV demonstrates an application of the approach. Section V is a conclusion.

## II. Information Modeling for Socio-Tech Systems

An information model (or a conceptual model, roughly speaking) aims to represent system structures and static behaviors of systems. Such a model provides abstract and high-level views of the systems. The information model is a basis of developing more detailed behavior models of systems in design, construction, and operation management. The proposed approach to information modeling of socio-tech systems is developed based on the integration of object-orientation [6], aspect-programming [8], system-dynamics [9], and several new modeling constructs, where, in particular, it is the first time that system-dynamics modeling is brought to information modeling to our best knowledge. In the following, concepts and constructs for information modeling of socio-tech systems are presented. It should be noted that the approach developed in this paper may also be called general information model or framework for information modeling. Applying this approach to a particular application leads to an application specific information model.

### A. Object Abstraction

Object-oriented thinking with a root in ontology is widely endorsed with the development of object-oriented programming (OOP). The notions of objects in our approach are developed on the basis of those in software development. The programmers using object-oriented technologies view objects as the fundamental entities to represent a problem domain in programming languages. The following is a brief review of the semantics and notation of object [11]. An object is an instance of a class, i.e., a set of objects sharing common features. Each object has a unique identifier. Each object has lifetime: it is created at a time and destroyed at another time later. The states and behaviors of an object are represented in terms of attributes and operations. At any special point of its lifetime, an object has values for each attribute. An object links to other objects in a system and they interact with each other during their lifetime. The UML [11] supplies notations to describe the semantics of an object.

Fig. 1 shows a triangle object in UML. A triangle is a kind of polygon. The attributes of a triangle include its three vertices, the color of the border, and the fill color. The triangle can be moved to four directions. Moving is one of the behaviors of a triangle. In

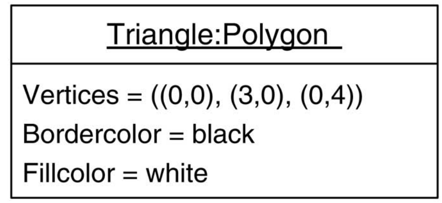


Fig. 1. UML notation of a triangle object.


Fig. 2. Dependency.

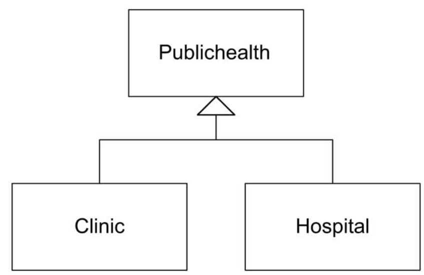


Fig. 3. Generalization.


Fig. 4. Association.

Fig. 1, the triangle's three vertices are (0,0), (3,0), and (0,4). Its border is black and it is filled in white. When the triangle is moved horizontally by three units, its vertices will change to (3,0), (6,0), and (3,4), while the colors of its border and filling are unchanged.

There are three kinds of relationships between objects: 1) dependencies; 2) generalizations; and 3) associations [11]. A dependency describes that an object uses the information or services of another object, which further means that the existence and/or functioning of the first object depends on the second. For example, a fire engine depends on fire hydrant at a certain urban location, which depends on the local water supply. The notation of dependency is shown in Fig. 2. A generalization is the relationship between a general kind of object and a specific kind of object. For example, both city hospital and a walk-in clinic are public health institutes, shown in Fig. 3. The opposite direction of generalization is called specification. An association describes the structural relationship between objects of two classes. The name of an association describes the nature of such relationship. The classes at both ends have their roles in the association. The multiplicity of the roles can also be specified in an association. The "whole/part" association relationships are called aggregations. Fig. 4 shows an example of association which describes many employees work for a plant.

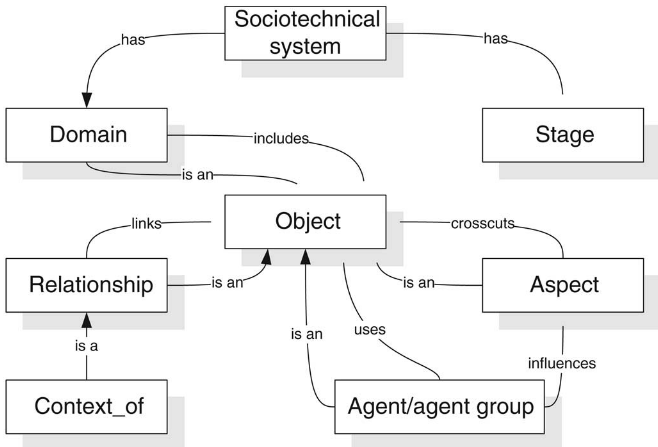


Fig. 5. Concept map of the special building blocks of socio-tech systems.

### B. Special Modeling Constructs for Socio-Tech Systems

We propose five modeling constructs especially useful for socio-tech systems: 1) domain; 2) aspect; 3) context-of; 4) agent/agent group; and 5) stage. Each of these constructs is existed in literature, but they have not been integrated to modeling socio-tech systems in a coherent way. The concept map of these blocks is shown in Fig. 5 and their semantics and notations are discussed in detail in the following sections.

*1) Domain:* A significant difference between a socio-tech system and a pure technology system is that the former often involves more diverse domains than the latter. A domain is a special object and it has unique identifier, lifetime, states, and behaviors. Each object exists in a certain domain that gives the context of all the objects inside. An entity can be modeled as an object or a domain based on the granularity of the model and information relativity principle [17]–[19], [33]. A domain has interfaces through which the objects inside the domain link to objects outside by relationships. In a socio-tech system, the domains can be classified into two categories: technical domains and social domains. We use rectangles with dashed bounders to denote domains.

*2) Aspect:* A common feature of many entities in the social domains like codes, regulations, policies, and guidelines is that they are enforced into the decision-making operations of some objects in technology domains in the cases of either normal operational states or emergent situations. The influences of some social entities are even pervasive in the whole technology domains. An option to model such entities is to specify every object influenced by such entities. However, merging social entities into technical domains will make the information model for technical objects too complicated, and it is hard to update all the technical objects once the social entities have been changed.

A similar problem has been observed in programming, which motivated the development of aspect-oriented programming (AOP) [7], [8]. In AOP terminology, the entities cross-cutting the system's functionalities are called aspects. In AspectJ [7], an implementation of AOP in Java, an aspect is defined by the joint points and advices. A joint point is the place where the system's function will be affected by this aspect. An advice is how the system's function is modified at the corresponding joint points. AspectJ has two kinds of advices: the advices being executed before the cutting point and the advices after the cutting point. We apply these concepts to information modeling of the cross-cutting social entities in socio-tech systems.

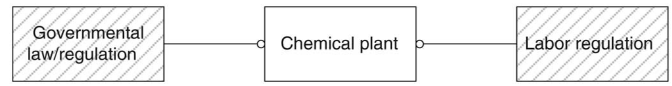


Fig. 6. Notations of aspects and context-of.

We generalize the notion of aspect in aspect programming and use it to model the scattering social entities such as codes, regulations, policies, guidelines, and their relations with technical entities. Besides the explicit social entities, there are other implicit social entities like public opinions and culture that work in the similar way in the socio-tech systems. These implicit entities are often ignored in traditional information analysis, because they are not proclaimed in writing and established by any person or organization. However, they do affect the behaviors of socio-tech systems and may contribute to a system's failures. In fact, this kind of information serves as a context that constrains a social-tech system's behaviors.

An aspect is a special object and it has unique identifier, lifetime, states, and behaviors. An aspect is linked to many objects influenced by it. An aspect specifies the operational decisions that should refer to this aspect and the advices enforced to such decisions by this aspect. To distinguish aspects (or aspect objects) from other objects, we use cross-hatched object shape to denote aspects shown in Fig. 6. The relationship between an aspect and its influenced object is called "context-of," which is to be discussed in Section II-B3.

*3) Context-of:* Context-of is a special kind of association between an aspect and another object. The behavior of an object is constrained by the aspects that are the context of the object. When the aspects change, the object has to change its behavior accordingly, although its structure may have never been changed. To distinguish context-of relationships from other associations diagrammatically, we use a plain association with an unfilled circle at the end to denote it, as shown in Fig. 6.

*4) Agent and Agent Group:* In technical system modeling, the users are considered as actors outside of the systems. However, we have to include the persons and the societies into the models for socio-tech systems, because they influence the behavior of the technical system in many aspects, and they are of the most interest in some critical situations. For instance, the society opinions change the aspects like policies and regulations in socio-tech systems. The cooperation of individuals affects the rescue speed in an emergency. In our methodology, individuals are modeled as agents and the society is modeled as an agent group. Both agent and agent group have complicated behavior patterns, and they are essential parts of socio-tech systems. We use the notation of actor in UML to denote the agent and the agent group, shown in Fig. 7.

*5) Stage:* A socio-tech system has different operational statuses, e.g., normal status and exceptional status. The structure and behavior of a socio-tech system will change due to the outside disturbances or designed mechanisms when it transfers from one operational status to another. We use the

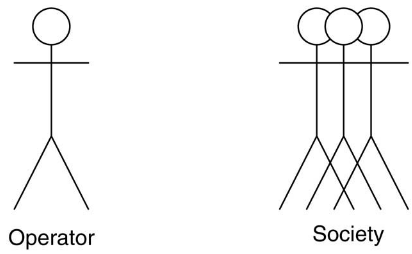


Fig. 7. Notations of agent and agent group.

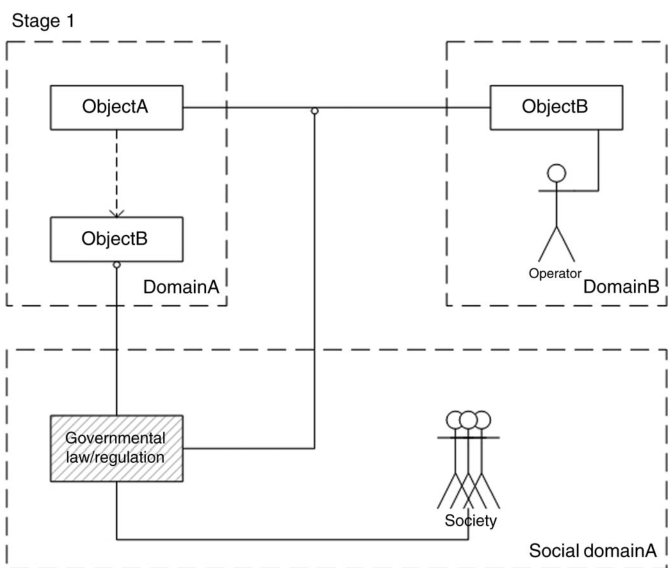


Fig. 8. OARD model.

concept of stage to model the evolution of a socio-tech system. A stage is in fact the context of all the information described in the model of a socio-tech system. The same socio-tech system in different stages will have different objects, aspects, and relationships between objects and aspects. How many stages a socio-tech system should have depends on design of technical domains and the social domains. Therefore, it should be clarified that what stage the system is in for all the information models of a socio-tech system. As a default, the stage of a socio-tech system is normal operational status.

## III. Object-Aspect-Relationship-Domain Model

Thus, by far, the building blocks to model socio-tech systems in our proposed approach include the basic object-oriented (OO) modeling methods and the special modeling blocks for socio-tech systems. The model developed in this way is called an Object-Aspect-Relationship-Domain modal (OARD). Fig. 8 shows an example of OARD model. Note that each relationship in the model is by nature an object (see the concept map in Fig. 5), and therefore, it has lifetime and can be associated to an aspect, which means aspects can change the dependencies and associations between objects. In this way, this model has the ability to model the structural change of a socio-tech system.

The behaviors of a socio-tech system are determined by the behaviors of its subsystems (objects, aspects, and agents in the OARD model) and the structure of the system [9], [10]. The structure is how the relationships link the subsystems together, which is described in the OARD models. However, it is difficult to get a particular view about the logics of system behavior in a scenario of interest in the whole picture of an OARD model. We further propose to apply the casual loop diagram [9] to describe the cause–effect relationships between entities that are important for a scenario in interest. The entities are in fact the attributes of objects in the system.

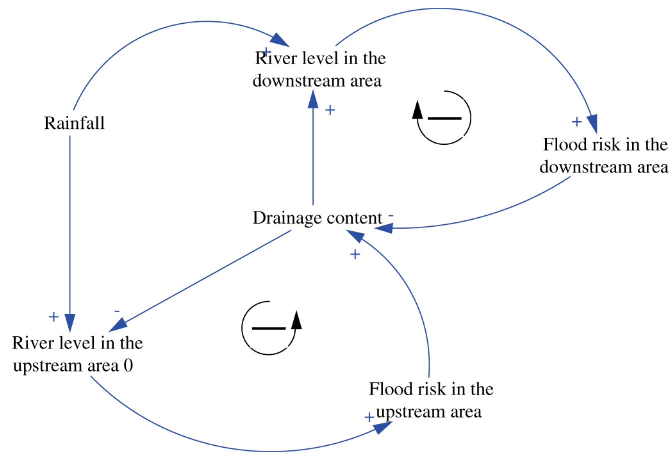


Fig. 9. Casual loop diagram for a river–dam system.

Fig. 9 shows a casual loop diagram for a river–dam system. The river levels in the upstream area and the downstream area of a dam on a river can be adjusted by increasing or decreasing the drainage content. In the case of continuous heavy rain, the drainage content should be adjusted in order to maintain the flood risk in the area at the safe level. The positive or negative sign on an arc indicates that the levels of the factor at two side of the arc will change in the same or opposite direction. The positive or negative sign of a loop indicates that the loop is a positive or negative feedback loop. In Fig. 9, both loops are negative feedback loop, which means we can control the flood risk in both upstream and downstream areas by adjusting the drainage content, and yet decreasing the risk in one area will increase the risk in the other area.

## IV. Case Study

In this section, we present a case study to show the effectiveness of the proposed information modeling approach. The case system is the water infrastructure of Walkerton which experienced the 2000 *E. coli* outbreak [12]. Section IV-A briefly reviews the drinking water infrastructure experiencing the Walkerton *E. coli* outbreak in 2000. Section IV-B presents the information model of the drinking water infrastructure of Walkerton at that time using the proposed approach. Section IV-C presents a further discussion of the superiority of the proposed approach to Rasmussen's approach.

### A. Walkerton Drinking Water System in May 2000

Walkerton is a small town in southern Ontario with 4800 residents. In May 2000, 7 people died and more than 2300 became ill because of the drinking water contaminated by *E. coli*. The inquiry report [12] tells the details of the tragedy. We

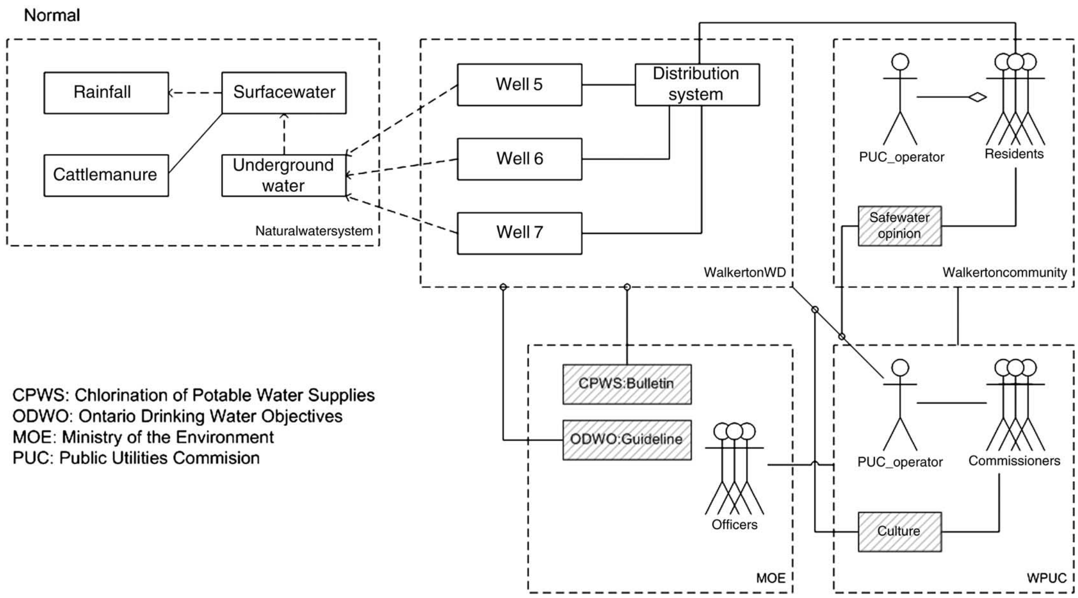


Fig. 10. OARD model of Walkerton drinking water infrastructure.

consider the drinking water infrastructure as a socio-tech system and develop an information model for it.

The Walkerton drinking water facility was managed by the local Public Utilities Commission (PUC). The PUC was overseen by the Ministry of Environment (MOE) of Ontario. A general manager of PUC is responsible for the decision-making and operation of the water system. There are three wells supplying raw water to the water distribution system in Walkerton, namely Well 5, Well 6, and Well 7. Each well has a chlorinator that should add sufficient chlorine to reduce contaminants in the raw water according to the "Chlorination of Potable Water Supplies (CPWS)", a bulletin issued by the MOE. The MOE had published a guideline entitled the Ontario Drinking Water Objectives (ODWO) that requires continuous turbidity monitoring. Unfortunately, the guideline was not implemented by the PUC in 2000. However, the PUC operators should manually measure the chlorine residual daily, which was a common practice at the PUC. The investigation of the outbreak showed that the contamination came into the water system through Well 5. Well 5 was a swallow well, and its water is affected by the surface water around it due to the geological conditions. The heavy rainfall between May 8 and May 12, 2000, especially 134 mm of rain on May 12, made the surface water contaminated by the cattle mature on a farm closed to Well 5. It is the allowance culture for incompetent operation in PUC that contributed to the failure to detect the contamination in Well 5 and effective response to the outbreak. The public opinions that the untreated water was safe and chlorinated water tasted badly also contributed to the mistake of the PUC operators.

### B. The Information Model of Drinking Water Infrastructure

Fig. 10 shows the OARD model developed for the Walkerton drinking water infrastructure. There are five domains in the normal stage of the system: 1) natural water system; 2) water distribution system; 3) Walkerton PUC; 4) MOE; and 5) Walkerton community. In the natural water system, the underground

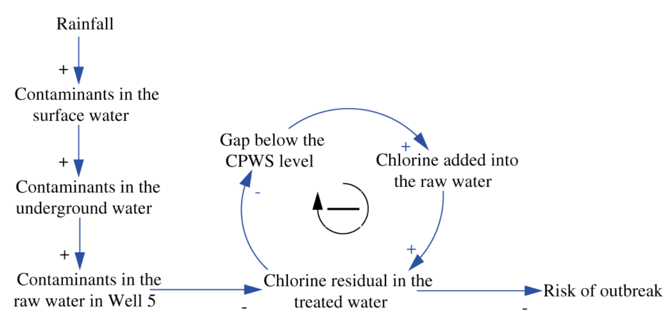


Fig. 11. Casual loop diagram for Walkerton drinking water contamination.

water depends on the surface water that further depends on the rainfall. The surface water can be contaminated by the cattle manure. Well 5, Well 6, and Well 7 all depend on the underground water. They supply water to the water distribution system. The Walkerton residents get water from the water distribution system. The PUC is set up by the Walkerton community and overseen by the MOE. The CPWS and ODWO are context of the water distribution system. The PUC operator is in charge of the operation of the water distribution system. The culture of PUC and the opinions of safe water in Walkerton community are context of the relationship between the PUC operator and the water distribution system. Fig. 10 shows the "normal" stage of the system. The system structure changed during the outbreak when A&L Canada Laboratories, the local health unit, MOE's Spills Action Center (SAC), the local municipality, and the local public media joined the system.

Now, we use the casual loop diagram to show the reason of the outbreak in Fig. 11. Rainfall and the contaminants in the water system have positive effect on the risk of outbreak, and the chlorine residual in the treated water has negative effect on the risk of outbreak. The chlorine residual in the treated water is controlled by monitoring and adding enough chlorine into the raw water. In Walkerton outbreak, failing to monitor and maintain the chlorine residual in the treated water is the most dominating reason for the tragedy.

### C. Discussion

As stated in Section I, Rasmussen's framework [13] is a popular tool for socio-tech system analysis and modeling. The differences between our approach and Rasmussen's framework are as follows.

1) Rasmussen's framework emphasizes more on the social side of systems and defines a relatively strict model structure in terms of social hierarchies, while our approach does not have this limitation. The OARD model can be easily extended and scaled using the notion of domains and objects.
2) Rasmussen's framework does not provide the notations for capturing different types of relationships, while our approach provides types of dependency, association, generalization, and context-of.
3) Rasmussen's framework does not distinguish the physical entities, social entities, and human agents by notations, while object, aspects, and agents are basic elements of the OARD model in our approach.
4) It is a common practice to confuse the objects and their behaviors in the map by applying Rasmussen's framework, while our approach can help prevent such practices.

The above-mentioned differences can be observed by comparing the case study of the Walton outbreak presented in this paper with that using Rasmussen's framework [16].

## V. Conclusion

This paper presented an integrated information modeling approach for socio-tech systems with consideration of all aspects of information covering technical system and social system and their interaction and dynamics. The completeness of information modeling with one coherent framework or approach may be highest among other approaches in the current literature. Such an achievement is due to the proposed integration of three powerful paradigms in information and dynamics modeling: object-orientation, aspect-programming, and system-dynamics.

It is noted that nobody in the literature brings system-dynamics to modeling of information dynamics. However, in a complex socio-tech system, information certainly changes with respect to time and event. Further, information changes come in three aspects: change of a technical system (e.g., vehicle), change of information that is carried with or by the technical system, and interaction between the technical system and social system. With the proposed approach, all the three aspects of information can be modeled. The proposed approach essentially focused on modeling of interactions between social systems and technical systems. When our approach is used to a particular application, the existing approaches (in literature) to both technical system and to social system should be employed together with the modeling constructs proposed in our approach. Nevertheless, the proposed constructs are also applicable to a "pure" technical system or to a "pure" social system.

Besides the benefit of completeness in information modeling for socio-tech systems, as elaborated in the above, another benefit is that the proposed approach facilitates implementation and utilization, as all the new constructs can be "fabricated" with UML constructs in the UML framework.

In conclusion, the proposed approach can well model the interaction and its dynamics of any socio-tech system. The proposed approach can be used for fault diagnosis and prediction for complex socio-tech systems. (Note: Any technical system can be viewed as a socio-tech system with the difference being the degree of interaction between the two.)

## References


[1] L. Xu, "Enterprise systems: State-of-the-art and future trend," *IEEE Trans. Ind. Informat.*, vol. 7, no. 4, pp. 630–640, Nov. 2011.

[2] Y. Ashparie, O. Bashorun, G. Koch, G. Siegel, and P. Stoyanov, "Infrastructure analysis system (IAS)," in *Proc. IEEE Syst. Inf. Eng. Des. Symp.*, Charlottesville, VA, USA, 2005, pp. 40–44.

[3] M. Bassiouni and R. Guha, "Modeling and analysis of temporal failure and degradation behavior of critical infrastructure systems," in *Proc. 35th Annu. Hawaii Int. Conf. Syst. Sci*., Hawaii, USA, 2002, pp. 689–694.

[4] G. Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language User Guide*, 2nd ed. Reading, MA, USA: Addison-Wesley, 2005.

[5] M. Chakrabarty and D. Mendonca, "Integrating visual and mathematical models for the management of interdependent critical infrastructures systems," in *Proc. IEEE Int. Conf. Man Cybern.*, The Hague, The Netherlands, 2004, vol. 2, pp. 1179–1184.

[6] J. R. Dittrich, "Object-oriented database system: The notion and issue," in *Proc. Int. Workshop Object-Oriented Database Syst*., Hoboken, NJ, USA: Wiley-IEEE Computer Society Press, 1986, pp. 2–6.

[7] J. D. Gradecki and N. Lesiecki, *Mastering AspectJ: Aspect-Oriented Programming in Java*, Hoboken, NJ, USA: Wiley, 2003.

[8] G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C. Lopes, J.-M. Loingtier, and J. Irwin, "Aspect-oriented programming," in *Proc. ECOOP'97,* Jyväskylä, Finland, 1997, pp. 220–242.

[9] C. W. Kirkwood. (1998). *System Dynamics Methods: A Quick Introduction* [Online]. Available: http://www.public.asu.edu/~kirkwood/sysdyn/SDIntro/preface.pdf, accessed on Jan. 31, 2014.

[10] Y. Lin and W. J. Zhang, "Towards a novel interface design framework: Function-behavior-state paradigm," *Int. J. Hum. Comput. Stud.*, vol. 61, no. 3, pp. 259–297, Sep. 2004.

[11] OMG. (2005). *Unified Modeling Language* [Online]. Available: http://www.uml.org/, accessed on Jan. 31, 2014.

[12] J. D. O'Connor. (2002). *Report of the Walkerton Inquiry*, Publications Ontario [Online]. Available: http://www.attorneygeneral.jus.gov.on.ca/english/about/pubs/walkerton, accessed on Jan. 31, 2014.

[13] J. Rasmussen, "Risk management in a dynamic society: A modeling problem," *Saf. Sci.*, vol. 27, pp. 183–213, 1997.

[14] S. Rinaldi, J. Peerenboom, and T. Kelly, "Identifying, understanding, and analyzing critical infrastructure interdependencies," *IEEE Control Syst. Mag.*, vol. 21, no. 6, pp. 11–25, Dec. 2001.

[15] S. Rinaldi, "Modeling and simulating critical infrastructures and their interdependencies," in *Proc. 37th Annu. Hawaii Int. Conf. Syst. Sci.*, Hawaii, USA, 2004, pp. 873–880.

[16] D. M. Woo and K. J. Vicente, "Socio-technical systems, risk management, and public health: Comparing the North Battleford and Walkerton outbreaks," *Rel. Eng. Syst. Saf.*, vol. 80, pp. 253–269, 2003.

[17] W. J. Zhang, "Integrated CAD and CAM of mechanisms," Ph.D. dissertation, Dept. Mech. Marine Eng., Delft Univ. Technol., Delft, The Netherlands, 1994.

[18] W. J. Zhang and K. van der Werff, "A critique of conceptual data modeling notions relative to the machine design domain," in *Proc. 1994 ASME Eng. Database Symp. '94,* Minneapolis, MN, USA, Sep. 11–14, 1994, pp. 59–66.

[19] W. J. Zhang and Q. Li, "Information modelling for made-to-order virtual enterprise manufacturing systems," *Int. J. Comput. Aided Design*, vol. 31, pp. 611–619, Oct. 1999.

[20] S. Fang, L. Xu, H. Pei, Y. Liu, Z. Liu, Y. Zhu *et al.*, "An integrated approach to snowmelt flood forecasting in water resource management," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 548–558, Feb. 2014.

[21] L. Xu, W. Viriyasitavat, P. Ruchikachorn, and A. Martin, "Using propositional logic for requirements verification of service workflow," *IEEE Trans. Ind. Informat.*, vol. 8, no. 3, pp. 639–646, Aug. 2012.

[22] W. He and L. Xu, "Integration of distributed enterprise applications: A survey," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 35–42, Feb. 2014.

[23] Y. Li, B. Cao, L. Xu, J. Lin, S. Deng, Y. Yin *et al.*, "An efficient recommendation method for improving business process modeling," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 502–513, Feb. 2014.

[24] L. Xu, C. Wang, Z. Bi, and J. Yu, "AutoAssem: An automated assembly planning system for complex products," *IEEE Trans. Ind. Informat.*, vol. 8, no. 3, pp. 669–678, Aug. 2012.

[25] L. Duan and L. Xu, "Business intelligence for enterprise systems: A survey," *IEEE Trans. Ind. Informat.*, vol. 8, no. 3, pp. 679–687, Aug. 2012.

[26] J. W. Wang, F. Gao, and W. H. Ip, "Measurement of resilience and its application to enterprise information systems," *Enterprise Inf. Syst.*, vol. 4, no. 2, pp. 215–223, May 2010.

[27] V. N. Kasyanov and E. V. Kasyanova, "Information visualisation based on graph models," *Enterprise Inf. Syst.*, vol. 7, no. 2, pp. 187–197, Feb. 2013.

[28] J. W. Wang, H. F. Wang, W. J. Zhang, W. H. Ip, and K. Furuta, "On a unified definition of the service system: What is its identity?" *IEEE Syst. J.* [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6544571, accessed on Jan. 31, 2014.

[29] M. Salge and P. M. Milling, "Who is to blame, the operator or the designer? Two stages of human failure in the Chernobyl accident," *Syst. Dyn. Rev.*, vol. 22, pp. 89–112, 2006.

[30] J. W. Rudolph, J. B. Morrison, and J. S. Carroll, "The dynamics of action-oriented problem solving: Linking interpretation and choice," *Acad. Manag. Rev.*, vol. 34, pp. 733–756, 2009.

[31] Z. Mohaghegh, R. Kazemi, and A. Mosleh, "Incorporating organizational factors into probabilistic risk assessment (PRA) of complex socio-technical systems: a hybrid technique formalization," *Rel. Eng. Syst. Saf.*, vol. 94, pp. 1000–1018, 2009.

[32] N. G. Leveson, "Technical, and managerial factors in the NASA Challenger, and Columbia losses: Looking forward to the future," in *Controversies in Science and Technology*, K. A. Cloud-Hansen, J. Handelsman, and D. L. Kleinman, Eds. Madison, WI, USA: Univ. Wisconsin, 2007, pp. 237–261.

[33] J. W. Wang, H. F. Wang, J. L. Ding, K. Furuta, W. H. Ip, and W. J. Zhang, "On domain modelling of the service system with its application to enterprise information systems," *Enterprise Inf. Syst.* [Online]. Available: http://www.tandfonline.com/doi/pdf/10.1080/17517575.2013.810784, accessed on Jan. 31, 2014.

**Junwei Wang** received the Ph.D. degree on the topic of optimization in systems engineering from Northeastern University, Shenyang, China, in 2006, and the Ph.D. degree on the topic of resilience engineering in mechanical engineering from the University of Saskatchewan, Saskatoon, SK, Canada, in 2013.

From 2011 to 2013, he was a Japan Society for the Promotion of Science (JSPS) Post-Doctoral Fellow with the Department of Systems Innovation, University of Tokyo, Tokyo, Japan. He is currently an Assistant Professor with the Department of Industrial and Manufacturing Systems Engineering, University of Hong Kong, Pokfulam, Hong Kong, since March 2014. His current research interests include modeling and optimization for resilience engineering of complex networked systems. He has published 18 refereed technical papers and conference papers, and has co-authored a book on optimization algorithms.

**Dong Liu** received the Ph.D. degree on the topic of architectural style of RESTful Service Composition and its performance, scalability, and reliability from the Department of Computer Science, University of Saskatchewan, Saskatoon, SK, Canada, in 2013.

From 2009 to 2012, he worked as a System Analyst at Canadian Light Source, Saskatoon, where he took part in the design and development of the Science Studio. He is currently a Software Engineer at the Facility for Rare Isotope Beams, Michigan State University, East Lansing, MI, USA. His research and development interests include distributed systems, system performance and scalability, enterprise application integration, RESTful services, big data, and event-driven programming.

**Wai Hung Ip** received the M.Sc. degree from Cranfield University, Bedford, U.K., in 1983, the MBA degree from Brunel University, London, U.K., in 1989, and the Ph.D. degree in manufacturing engineering from Loughborough University, Loughborough, U.K., in 1993.

He is an Associate Professor with the Department of Industrial and Systems Engineering, Hong Kong Polytechnic University, Hung Hom, Hong Kong. Over the span of more than 20 years of work in industry and education, he has published more than 100 international journals and conference articles in the areas of production management, enterprise resource planning, and supply chain and logistics systems. His research interests include the modeling and application of operations research and artificial intelligence techniques.

Dr. Ip is a Member of the Institution of Mechanical Engineers, the Institution of Electrical Engineers (IEE, U.K.), and the Institution of Hong Kong Engineers.

**Wenjun Zhang** (M'96–SM'09) received the Ph.D. degree from Delft University of Technology, Delft, The Netherlands, in 1994.

He is currently a Professor of Mechanical Engineering at the University of Saskatchewan, Saskatoon, SK, Canada. He has published over 320 refereed technical publications, 193 papers in refereed journals and over 130 papers in refereed conference proceedings, in a broad scope of fields including design and mechatronics, manufacturing, informatics, and human–machine systems.

Dr. Zhang is a Member of ASME, Senior Member of SME, and a member of SPIE. He received the Outstanding Researcher Award by the College of Engineering at the University of Saskatchewan, in 2011, and the Distinguished Graduate Supervisor Award in 2012 by the University of Saskatchewan. He has served as a Technical Editor for IEEE/ASME TRANSACTIONS ON MECHATRONICS and an Associate Editor for the *International Journal of Systems Science and Enterprise Information Systems*.

**Ralph Deters** received the Ph.D. degree in computer science from the Federal Armed Forces University, Munich, Germany, in 1998, and the Diplom. in computer science from the Technical University of Berlin, Berlin, Germany, in 1992.

He has been a Professor with the Department of Computer Science, University of Saskatchewan, Saskatoon, SK, Canada, since 1999. His research focuses on mobile and ubiquitous computing. Within mobile and ubiquitous computing, he is currently working on the issues of linking EIS with mobile devices, cross-platform development of mobile apps, the personal cloud (ubiquitous cloud computing), dependability and latency issues in mobile cloud computing, RESTful web services, and cloud middleware.