EmergencyGrid – Planning in Convergence Environments

Natasha C. Queiroz Lino, Clauirton de A. Siebra and Manoel Amaro
Center of Informatics, Federal University of Paraíba
[natasha,clauirton]@ci.ufpb.br, manoel.amaro@lavid.ufpb.br

Austin Tate
Artificial Intelligence Applications Institute, School of Informatics, University of Edinburgh
a.tate@ed.ac.uk

Abstract

Government agencies are often responsible for event handling, planning, coordination, and status reporting during emergency response in natural disaster events such as floods, tsunamis and earthquakes. Across such a range of emergency response scenarios, there is a common set of requirements that distributed intelligent computer systems generally address. To support the implementation of these requirements, some researchers are proposing the creation of grids, where final interface and processing nodes perform joint work supported by a network infrastructure. The aim of this project is to extend the concepts of emergency response grids, using a convergence scenario between web and other computational platforms. Our initial work focuses on the Interactive Digital TV platform, where we intend to transform individual TV devices into active final nodes, using a hierarchical planning structure. We describe the architecture of this approach and an initial prototype specification that is being developed to validate some concepts and illustrate the advantages of this convergence planning environment.

Introduction

We have seen, in recent decades, a steady increase in natural catastrophes resulting in loss of life and physical damage. The earthquake in Haiti (2010, over 300,000 victims) and tsunami in Japan (2011, over 20,000 victims) are examples of such events. In fact, weather related events are expected to increase in number and severity in the future, due to the impacts of climate changes.

Nowadays, modern technologies could effectively impact the ability to plan, coordinate and respond to such disasters. These technologies are related, for example, to emergency communications, earth observation and events monitoring. Interactive Digital TV (IDTV) is one of these technologies that are being used in the emergency domain as a way to warn people about emergency on time. The IDTV platform enables the configuration of an emergency warning broadcast system and the sending of alerts (earthquake, tsunami, etc.) to each device in the area covered. The alert signal uses some data space in one of the segments of the data stream, turns on all receivers, if turned off, and presents the alert information. An example of such alert is the *Earthquake Early Warning* (EEW), which was well-utilized with alert sound and emergency box superimposed on TV screen at time of the 2011 Tohoku earthquake and tsunami and many aftershocks in several days. In April 2011, the Chilean Subsecretary of Telecommunications also released a similar alert system.

With the planned coverage of 95% of the worldwide population with digital television, there are in fact opportunities for prompt deployment of public emergency warning systems via satellite or terrestrial TV network. This work proposes the extension of this IDTV use so that they can bring more advanced information rather than simple disaster warnings. In this new perspective, the idea is to consider each IDTV device as final nodes of a hierarchical planning and task support structure, so that all the components can be seen as an emergency grid. This grid should provide a convergence environment, integrating IDTV, Web and mobile phone platforms, so that they could change knowledge and services with each other.

To build such a grid, we have provided a semantic layer to the IDTV middleware, so that intelligent process support could be implemented on this layer, sharing knowledge and planning information via ontological descriptions. The central planning node is implemented using the *Knowledge as a Service* metaphor, so that planning resources can be accessed as a service.

The remainder of this work is organized as follows: the next section describes the main works about the use of intelligent systems in emergency response scenarios. Then, we discuss the general architecture of our approach and

---

technologies that we are using to create an emergency grid that involves the IDTV platform. After that, we illustrate the use of this architecture with the specification of an emergency response application. Finally, we comment on the main remarks and future research directions.

Intelligent Systems for Emergency Response

Recently, many projects and initiatives have been devoted to provide intelligent computational support for emergency management. The work of Wang *et al*. (2007), for example, proposes an algorithm for optimal emergency resource allocation scheme in order to solve collision problems among multiple disaster places and multiple resource suppliers. Also regarding resource manipulation, Liu (2004) proposes a *possibilistic* Petri net-based resource description language, and related matchmaking mechanism, to search for relevant resources over the Internet that can cooperate to prepare for and respond to environmental emergency situations. Specifications of multiagent architectures [Basak *et al*. 2011; Schoenharl and Madey 2006] and decision making support systems [Tufekci 1995; Hernandez and Serrano 2001] are also important contributions from the research community to disaster relief.

These and other works highlight two research directions: low level approaches (*e.g*. resource search and allocation algorithms) and more general approaches (*e.g*. architectures and decision support systems). A different kind of approach aims to integrate previous solutions, or systems from different parts, to create more sophisticated disaster response solutions [Fortier and Volk 2006]. In this context, we see the *Grid* metaphor as one of the main research trends.

A *Grid* is a geographically distributed computation platform that can enable users to access various computing resources via a uniform computational interface [Foster and Kesselman 1999]. In grid computing, a single big task is split into multiple smaller tasks which are further distributed to different computing machines. Upon completion of these smaller tasks, they are sent back to the primary machine which in return offers a single output. Examples of Grid applications in the emergency response domain are the e-Response [Potter *et al*. 2004] and FireGrid [Upadhyay *et al*. 2008] research programmes.

e-Response is a simulated scenario in which a distributed team of specialist scientists use CoAKTinG (Collaborative Advanced Knowledge Technologies in the Grid) [Buckingham Shum *et al*. 2002] tools to coordinate emergency environmental protection activities. The domain used was an oil spill in the Solent, a strait separating the Isle of Wight from the mainland of England. FireGrid is an integrated emergency response system for fires in built environments. The broad objective is to provide fire fighters with as much useful information as possible that enables them to make sound and informed judgments while tackling the fire. To achieve this goal, the system provides the continuous assessment of the state of the building, forecasting the likelihood of future events and conveying this information to the responders at the scene.

Setting a Convergence Planning Environment

While all works discussed in the previous section are targeted at providing support for emergency response teams, we take a different approach, whose aim is to support civilians in processes such as evacuations of unsafe areas. In a similar way that FireGrid intends to provide fire fighters with useful information to support their decisions, our approach intends to also provide useful information to civilians, so that they can save themselves. For that end, common domestic devices, such as TVs and mobiles phones, should be used. This paper, in particular, focuses on the IDTV platform. The next sections discuss the technologies that we are using to extend the use of intelligent resources to this platform, creating a convergence environment where planning activities and their outcomes can be better delivered to normal civilians.

General Architecture

Figure 1 shows a conceptual view of the system. The *Planning and control center* composes the main node of the grid and it accounts for providing the planning services. To that end, it is being implemented in accordance with the *Knowledge as a Service* (KaaS) [Beijun 2010] metaphor. When TV devices receive a broadcast that contains warnings about a disaster, they inform their users about this disaster using messages and sounds in the display. This is the normal procedure in current emergency warning systems. However, this message also asks users to press a button on their remote control to get instructions about disaster procedures and actions to be carried out. An example is discussed latter on in this paper.

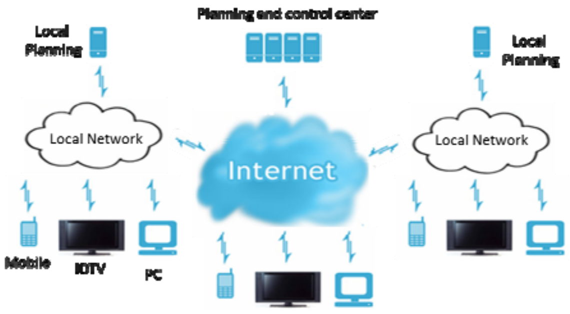

Figure1. Conceptual view of a convergence environment

Note that we may have local planning nodes to provide scalability to the system. In this case we can have three or more levels in the planning hierarchy. Several works present proposals about how to control and coordinate components in a hierarchical planning structure [Durfee and Montgomery 1991; Cox *et al*. 2005; Clement and Durfee 2001]. In our case, we are using extensions based on the I-X architecture [Tate 2000], which can be seen in [Siebra and Lino 2006]. However this discussion is out of the scope of this paper, so that we focus on the creation of the convergence environment and its extension to other platforms.

IDTV Architecture

To provide support to more advanced applications, we have created a *semantic* layer as part of the IDTV middleware. In fact, without this layer, the IDTV platform suffers from the same limitations as the World Wide Web. Current computational processes that run on the Web only account for leading the information transport, so that they do not have access to the meaning of the page content. The main reason is the form in which the information is structured, which is appropriate to the human user manipulation rather than computational processes. Thus, today we have a Web of documents rather than a Web of information, where computers can only provide limited assistance during the access and processing of information

The Semantic Web [Shadbolt *et al*. 2006] is the main W3C resultant technology for the problem discussed above. Its aim is to enable machines to understand the meaning of information on the Web. Some of its advantages are: sharing and reuse of data in different applications, automatic processing of data by computers, and semantic connections between data and the real world.

As we see, semantic representations are mainly important for systems integration and information sharing. Such features are the fundamental basis for a convergence environment. The Coalition Search and Rescue Task Support (CoSAR-TS) [Tate *et al*. 2006] is a good example of planning integration to other web services, supported by a semantic web environment. Emergency response operations by nature require the kind of rapid dynamic composition of available services making it a good use case for Semantic Web technologies.

IDTV Semantic Data Format

In the current IDTV standards, transmission of information, in a broadcast stream, is purely based on metadata definitions of tables and information services. The SI (Service Information) tables extend the PSI (Program Specific Information) tables, of the MPEG-2 standard, defining a set of structures that have descriptive data that transport specific IDTV information. Table 1 transcribes part of the MPEG-2 PSI/SI metadata table, which shows the fields 41, 42 and 43 related to the definition of an emergency alert.

The use of such tables facilitates the creation, processing, and rapid extraction of information. However, the SI tables are considered rigid metadata. Many services need more detailed information that cannot be satisfactorily defined within the SI tables. To that end, we have provided an ontological description to the IDTV operational data, so that external processes can understand the semantic meaning of their elements.

Table 1 - Part of the MPEG-2 PSI/SI metadata table

#	Metadata	Source	Description
…	…	…	…
41	state_area_code	NIT/PMT	Target state to emergence information transmission
42	microregion_area_cod	NIT/PMT	Target micro-region to emergence information transmission
43	signal_level	NIT/PMT	Specific emergency alert, which is defined by government organizations

In the proposed ontology, for example, we have the *EmergencyAlert* class. This class represents a signaling element that is transmitted by content providers to inform the population of a specific region about an imminent emergency situation. Another important element of this ontology is the *MMContent* class that represents a generic multimedia content entity and is the basis for all content construction that is used in the IDTV platform. The *EmergencyAlert* and *MMContent* are related by the *isEmergencyAlertTransmittedInto* property. This property indicates that a specific emergency alert is contained into a specific multimedia content during the IDTV transmission. Similarly the *hasLocationAlertFor* property relates the *EmergencyAlert* and *GeographicArea* classes. It indicates the scope of an emergency alert in terms of a geographic area.

Planning as a Service

In the proposed architecture, planning activities are mainly carried out in a server, rather than middleware. This approach is justified because such planning activities require a high processing power and data manipulation. This is a constraining factor, since current set-top-boxes do not have high processing power. In addition, another reason is that the middleware native operations have priority over computing resources usage. As a consequence, for instance, if the middleware needs more memory or processing power, it can demand computational resources that are being used by an upper level application and all data can be lost. Thus, the demanding part of processes is being developed in accordance with the KaaS [Beijun 2010] paradigm, so that set-top-boxes only need to send and receive information from/to such services, carrying out simple parts of the whole planning process. Another motivation to allocate the whole demanding process in a server is the easier access from/to any other computational process and available data. For example, we can integrate services from other computational platforms, such as mobile and personal computers, and also compose new services using other available web services.

Two main advantages of the KaaS paradigm can be stressed. First, the models used by this paradigm are based on formal semantic representations, so that we do not have the same problems that are found in other web services. Second, the knowledge servers have the capacity of accessing data from different sources, instantiating their representations and generating knowledge to be delivered via intelligent process such as a distributed planning algorithm. Figure 2 shows a conceptual view of a service, according to the KaaS approach.

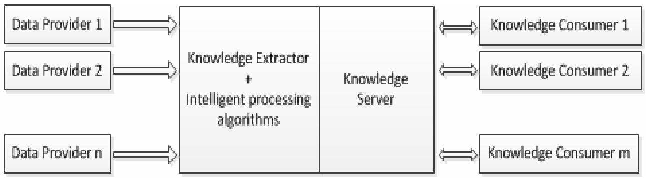

Figure 2. KaaS conceptual view [Xu and Zhang 2005].

According to this figure, the KaaS framework defines three logic components: (1) Data Providers, (2) Knowledge Server (Knowledge Extractor and Intelligent Processing algorithms) and (3) Knowledge Consumers. Considering our approach, Data Providers are sources of useful information that can assist the plan creation. For example, if the planning aim is to allocate tasks for emergency response teams, data providers could be represented by police stations, fire brigade centers and hospitals. The *Knowledge Server* runs a hierarchical multiagent planning algorithm, which is discussed in the next section. Finally, the *Knowledge Consumers* are represented by civilians, which can access emergency procedures via domestic devices, such as TVs and mobile phones.

The work of Paik *et al.* (2006) discusses some issues about the configuration of planning as a service and describes a framework for intelligent semantic web services that supports planning and scheduling aspects by a combined HTN planner and CSP (Constraint Satisfaction Problems) techniques. Note that the planning as a service approach is different from other approaches, which use planning mechanisms for the Web services composition problem [Traverso and Pistore 2004; Bo and Zheng 2009]. In the former case, planning is in fact the service, while this latter approach uses planning to compose the most diverse kinds of services.

Planning Aspects

The planning server is being specified in accordance with the I-X technology [Tate *et al*. 2006], which intends to provide a well-founded approach to allow humans and computer systems to cooperate in the creation or modification of some product, such as a plan. The use of I-X is justified because its planning representation is based on a formal ontology, called <I-N-C-A> (<Issues – Nodes – Constraints – Annotations>) [Tate 2003]. Thus, this ontology can be represented in the IDTV semantic layer as a domain ontology.

The main role of I-X planning agents is to provide actions to decompose higher level; more abstract activities until there are only executable activities. The important point in this discussion is to know that each planning step is implemented by an *activity handler*, which propagates the components through *constraint managers* to validate their constraints. Thus, all agents have a set of activity handlers that they use to refine or perform their activities. In a general way, the process follows these steps:

1. When an activity a is received, the agent's controller component selects a set H of activity handlers, which matches the description of a;
2. Each handler $h \in H$ uses one or more constraint managers to return its status (possible, impossible or not ready);
3. An optimal strategy, or an user, chooses one of the proposed handlers, committing to the performance of a;
4. During the execution, constraint managers are still monitoring the constraints of a, warning in case of problems, and maybe proposing continuations.

The role of constraint managers in this process is to maintain information about a plan while it is being generated and executed. The information can then be used to prune search where plans are found to be invalid as a result of propagating the constraints managed by these managers. The principal advantage of using constraint managers is their modularity. We can design managers to deal with specific types of constraints, such as the types discussed here (e.g., temporal, resource, commitment, etc.)

Together, the constraint managers form the model manager of the agent. Each constraint manager considers a set of specific constraints in a well-defined syntax, based on the support provided to a higher level of the planner where decisions are taken. However, they do not take any decision themselves. Rather, they are intended to maintain all the information about the constraints they are managing and to respond to questions being asked of them by the decision making level [Tate *et al*. 2006].

IDTV Emergence Response Application

This section details how this approach will be evaluated via a practical prototype that is in ongoing development. The prototype scenario represents part of Joao Pessoa (JP), the eastern-most city in Brazil. According to some scientists, there is a small chance that a mega-tsunami, originated from an earthquake close to Canary Islands, can reach the coast of JP (Figure 3). This region has a high population density, so that a simple emergency alert can create serious problems. For instance, the disordered use of the five coast evacuation routes may create big traffic jams.

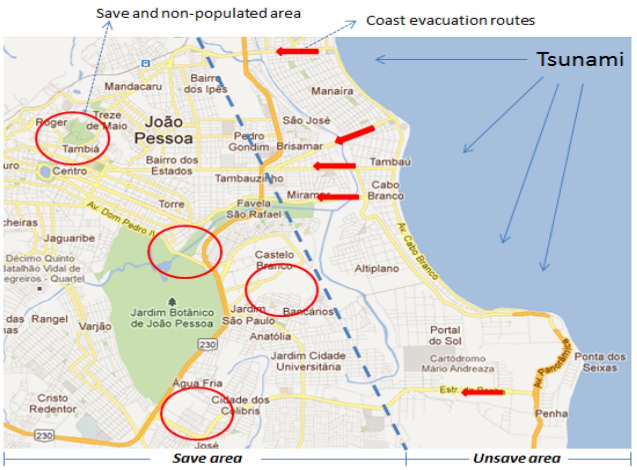

Figure 3. Map of Joao Pessoa city coast.

In the proximity of a tsunami event, the broadcasters send warning messages (Figure 4, left hand side), which are described via metadata, to be displayed by IDTV devices. We intend that when users press the green remote control button, an instance of the *EmergencyAlert* class is created and sent to the planning server in the form of a request, together with parameters that describe the users of this device and support the planning process. At the moment, we are considering only two parameters: user´s address and locomotion type.

Emergency Alert System

ALERT!

TSUNAMI WARNING. Press the GREEN button for furher informatons and evacuation instructions.

At the end of count down, press the green button and follow the map route

00:05:37

Figure 4. Examples of interfaces in IDTV platform.

When the server receives a request, it tries to allocate the best route from the user´s address to one of the safe areas, considering the traffic already allocated. The planner also returns the time that users must evacuate their homes. The clock carries out a count down until zero. At this moment, users must press the green button and evacuate their homes (Figure 4, right hand side). Obviously, this process is only valid to users whose locomotion way is defined as “car”. Otherwise (walk, bus, taxi, bike, etc.), a simple message is returned, asking an “as soon as possible” evacuation.

The first activity of the planning server is to acquire information, from *Data Providers* (Figure 2), about the event. In this application, important data is related to locations of safe areas and likely remainder time to disaster. After that, the allocation is carried out on demand. Sometimes we may have a route allocation that seems longer and non optimal. This is an effect of the on demand feature of this system. In order, we cannot have a pre-defined plan in advance because the planning system does not know how many civilians will be in the area at the moment of the alert broadcast.

Replanning activities are also limited in this scenario, since civilians are not monitored and they lose the communication channel after leaving their homes. This can create serious problems. For example, consider that one of the routes is blocked due to an accident. Consequently, other routes should be generated for the vehicles that are using the blocked route. This problem will only be considered after the integration of the mobile phone platform into this convergence scenario.

While the IDTV semantic representation and communication protocol between middleware and server is complete, we are still working on the planning service, mainly in the implementation of activity handlers. Three main concepts of the I-X architecture are appropriate for our implementation:

- Support for activity monitoring: we intend initially to only use the green button feedback (Figure 4, right hand side) as an indication that the plan is being followed. Future versions, using the mobile phone platform, will tend to use more advanced monitoring approaches;
- Support for *Standard Operating Procedures*: pre-planned set of activities, which can be used in specific situations, can be implemented as activity handlers;
- Modular implementation of activity handlers: at this moment we have only one type of handler that is *AlllocateRouteAndStartTime*. However we can have several versions (algorithms) of this implementation, each of them as a different activity handler.

We intend to use a simulator, such as Hermes [Xithalis 2008] to evaluate different versions of this handler. This application is a simple network simulator that allows us to design a network for a city and observe the level of service it can provide, i.e. number of vehicles and total trip time.

Conclusions and Research Directions

This work discusses a planning architecture where emergency response activities are provided via a server, according to the KaaS paradigm. This paradigm enables, among other features, an appropriate semantic description to data that comes from different platforms. Our main aim is to use the KaaS metaphor as a form to enable convergence among different computational platforms, such as the IDTV, mobile phone and Web. Our initial focus was on IDTV platform, where a complete semantic model was defined for its data. However, future versions intend to consider the mobile phone platform, mainly as a way to extend re-planning strategies and monitoring abilities.

We are still implementing the planning server; however some important requirements have already identified. The principal question is how to implement an optimization planning mechanism that can use the evacuation waiting time to re-plan routes. This re-planning must be carried out in real time and have low impact on unaffected users.

References

Basak, S., Modanwal, N., and Mazumdar, B. 2011. Multi-Agent Based Disaster Management System: A Review. *International Journal of Computer Science & Technology*, 2(2): 343-348.

Beijun, S. 2010. A Design Framework for Public Knowledge Service Platform. *Proceedings of the International Workshop on Knowledge as a Service*, Xiamen, China.

Bo, Y., and Zheng, Q. 2009. Semantic Web service composition using Graphplan. *Proceedings of the 4th IEEE Conference on Industrial Electronics and Applications*, Xian, China, pp. 459-463.

Clement, B., and Durfee, E. 2001. Performance of Coordinating Concurrent Hierarchical Planning Agents Using Summary Information. *Lecture Notes in Computer Science*, Vol. 1986/2001, pp. 214-230.

Buckingham Shum, S., De Roure, D., Eisenstadt, M., Shadbolt, N. and Tate, A. 2002. CoAKTinG: Collaborative Advanced Knowledge Technologies in the Grid. *Proceedings of the Second Workshop on Advanced Collaborative Environments*, Eleventh IEEE Int. Symposium on High Performance Distributed Computing, Edinburgh, Scotland.

Cox, J., Durfee, E, and Bartold T. 2005. A distributed framework for solving the Multiagent Plan Coordination Problem. *Proceedings of the Fourth International Joint Conference on Autonomous agents and Multiagent Systems*, Utrecht, The Netherlands.

Durfee, E., and Montgomery, T. 1991. Coordination as distributed search in a hierarchical behavior space. *IEEE Transactions on Systems, Man and Cybernetics*, 21(6): 1363-1378.

Fortier, S., and Volk, J. 2006. Defining Requirements for ad hoc Coalition Systems during Disasters. *Proceedings of the IEEE International Conference on Computational Cybernetics*. Budapest, Hungary, pp. 1 – 6.

Foster, I., and Kesselman, C. 1999. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann.

Hernandez, J., and Serrano, J. 2001. Knowledge-based models for emergency management systems. *Expert Systems with Applications*, 20(2):173–186.

Liu, K. 2004. Agent-based resource discovery architecture for environmental emergency management. *Journal Expert Systems with Applications*, 27(1):77-95.

Paik, I., Maruyama, D., and Huhns, M. 2006. A Framework for Intelligent Web Services: Combined HTN and CSP Approach. *Proceedings of the IEEE International Conference on Web Services*, pp. 959-962.

Potter, S., Tate, A., and Dalton, J. 2004. Collaborative Task Support and e-Response, *AISB Quarterly*, No. 115, Winter 2004.

Schoenharl, T., and Madey, G. 2006. WIPER: A Multi-Agent System for Emergency Response, *Proceedings of the 3rd International Conference on Information Systems for Crisis Response and Management*, Newark, NJ, USA.

Shadbolt, N., Berners-Lee, T., and Hall, W. 2006. The Semantic Web Revisited. *IEEE Intelligent Systems*, 21(3): 96-101.

Siebra, C., and Lino, N. 2009. Aspects of Planning Support for Human-Agent Coalitions. *Journal of the Brazilian Computer Society*, 15(4):41-55, Rio de Janeiro, Brazil.

Tate, A. 2000. Intelligible AI Planning, in Research and Development in Intelligent Systems. *Proceedings of the Twentieth International Conference on Knowledge Based Systems and Applied Artificial Intelligence*, Cambridge, UK, pp. 3-16.

Tate, A. 2003. <I-N-C-A>: a Shared Model for Mixed-initiative Synthesis Tasks. *Proceedings of the Workshop on Mixed-Initiative Intelligent Systems* at the International Joint Conference on Artificial Intelligence, Acapulco, Mexico, pp. 125-130.

Tate, A., Dalton, J., Bradshaw, J., and Uszok, A. 2006. Task Support: Intelligent Task Achieving Agents on the Semantic Web. *Air Force Research Laboratory Technical Report AFRL-IF-RS-TR-2006-91*.

Traverso, P., and Pistore, 2004. Automated Composition of Semantic Web Services into Executable Processes. *Lecture Notes in Computer Science*, Vol. 3298/2004, pp. 380-394.

Tufekci, S. 1995. An integrated emergency management decision support system for hurricane emergencies. *Safety Science*, 20(1):39–48.

Upadhyay, R., Pringle, G., Beckett, G., Potter, S., Han, L., Welch, S., Usmani, A., and Torero, J. 2008. An Architecture for an Integrated Fire Emergency Response System for the Built Environment. *Proceedings of the. 9th IAFSS International Symposium on Fire Safety Science*, Karlsruhe, Germany.

Wang, S., Wang, Y., and Sun, J. 2007. An Optimized Emergency Resources Allocation Algorithm for Large-Scale Public Emergency. *Proceedings of the International Conference on Machine Learning and Cybernetics*, Hong Kong, pp. 119-123.

Xithalis, C. 2008. Synchronous Control Method for Persona Rapid Transit Systems. *Proceedings of the 10th International Conference on Application of Advanced Technologies in Transportation*, Athens Greece.

Xu, S., and Zhang, W. 2005. Knowledge as a Service and Knowledge Breaching. *Proceedings of 2005 IEEE International Conference on Services Computing*, 1:87-94, Orlando, FL, USA.