

PREFACE

AI has long lead the march towards intelligent behavior in computer systems. This has lead to emergence of techniques in perceiving the world (e.g., web-based systems, vision, speech, NLP), manipulating internal representations (e.g., knowledge representation, logic, uncertainty, utility, constraints), analyzing information (e.g., search, learning, planning, scheduling) and altering the world (e.g., robotics, multi-agent systems). They have resulted in many applications with profound impact to our daily lives and remarkable commercial successes. The time has come for AI to address even more challenging problems.

One such problem is sustenance and improvement of quality of life for global citizens despite over-population and limited resources. As the world's population increases and it puts increasing demands on the planet's limited resources due to shifting life-styles, we not only need to monitor how we consume resources but also optimize resource usage. Some examples of planet's resources citizens consume are water; energy sources – hydrocarbons, electricity; air – conversely, green-houses gases emitted; and food sources – plants and animals. Everyone (individuals, enterprises, governments) can do something in their realm for reducing resource footprint to alleviate the worsening situation. This is both an ethically good thing to do as well as makes a lot of economic sense.

Championing this cause, industry has started using the prefix “smarter” to cyber-physical systems, which involve sensing the data through physical instruments, interconnecting and integrating them from multiple sources and analyzing them for intelligent patterns. There are multiple domains, all converging on a broader “Smarter Planet” theme. Consider a city. The city management, citizens and businesses may be interested in optimizing their usage of energy and water, while providing efficient infrastructure, transportation and public safety to all participants.

The trend towards optimized cyber-physical systems, however, is not new to AI and in fact a subset of what has been pursued as intelligent behavior, but on a much larger scale of people and resources involved. We believe that AI has a big role to play in making all the stakeholders in the scope under consideration (individuals, building, campus, city, leading to the planet), and hence the planet, intelligent.

The aims of the workshop are to:

1. Draw attention of AI community to the domain of cyber-physical systems
2. Identify unique issues of this domain and what new techniques may be needed
3. Provide a platform for sharing best-practices and discussion

We have identified 6 papers to be presented at this workshop, 4 as full papers and 2 as short papers. These papers focus on a variety of intelligent planet problems including data acquisition, data modeling, analytics and case studies.

We thank the program committee and other reviewers for their efforts in providing constructive reviews to the submissions.

We hope that this workshop is as enjoyable to the participants as it was for us to organize it.

Co-Chairs:

Biplav Srivastava

IBM Research – India, New Delhi, India

Email: sbiplav@in.ibm.com, Ph: +91-9810 404142

Carla Gomes

Cornell University, USA

Email: gomes@cs.cornell.edu

Anand Ranganathan

IBM T. J. Watson Research Center, Hawthorne, USA.

Email: anand@us.ibm.com, Ph: +1-914-784-6257

We were guided at every step by a very active steering committee.

Steering Committee:

- Kanna Rajan, MBARI, USA
- Jussi Rintanen, NICTA, Australia
- Ullas Nambiar, IBM Research – India
- Raghuram Krishanpuram, IBM Research - India

Program Committee

Rajarshi Das	IBM TJ Watson Research Center
Hasan Davulcu	Arizona State University
Dipti Deodhare	
Aditya Ghose	University of Wollongong
Carla Gomes	Cornell University
Richard Goodwin	IBM
Anupam Joshi	University of Maryland Baltimore County
Subbarao Kambhampati	Arizona State University
Craig Knoblock	Information Sciences Institute, University of Southern California
Raghuram Krishnapuram	IBM India Research Lab
Ullas Nambiar	IBM India Research Lab
Kanna Rajan	Monterey Bay Aquarium Research Institute
Anand Ranganathan	IBM T.J. Watson Research Center
Jussi Rintanen	NICTA
Prasan Roy	IBM Research
Abdul Sattar	Griffith Univeristy
Biplav Srivastava	IBM India Research Labs
L. Venkata Subramaniam	IBM Research - India
Rosario Usceda-Sosa	

A Negotiation Protocol for Multiple Interdependent Issues Negotiation Over Energy Exchange

Muddasser Alam, Alex Rogers and Sarvapali D. Ramchurn

Intelligence, Agents, Multimedia Group,
School of Electronics and Computer Science
University of Southampton, UK.
{ma606, acr, sdr}@ecs.soton.ac.uk

Abstract

We present a novel negotiation protocol to facilitate energy exchange between off-grid homes that are equipped with renewable energy generation and electricity storage. Our solution imposes additional constraints on negotiation such that it reduces a complex interdependent multi-issue problem to one that is tractable. We prove that using our protocol, agents can reach a Pareto-optimal, dominant strategy equilibrium in a decentralized and timely fashion. We empirically evaluate our approach in a realistic setting. In this case, we show that energy exchange can be useful in reducing the capacity of the energy storage devices in homes by close to 40%.

1 Introduction

It is estimated that 1.6 billion people live without access to electricity; mostly in the Indian subcontinent and Sub-Saharan Africa [IEA, 2008]. This population is divided in very small communities scattered over vast areas of land and hence, providing them with centralized power distribution has not been justified to date due to the huge capital cost of such infrastructure and the lack of demand. However, the opportunity for demand cannot become established if there is no supply. To address this dilemma, a number of initiatives have begun to provide these communities with off-grid renewable microgeneration. infrastructure such as solar panels, and electric batteries for storage¹. However, these resources are operated in isolation. Our aim is to investigate whether the interconnection and decentralized coordination of such resources could result in their more efficient use and be beneficial to the connected homes in the community. We envision that such interconnection and coordination could lead to the creation of a grid from the ground up and allow these communities to take an evolutionary jump straight to the smart grid, in the same way that many of them have bypassed landlines and leapt straight to cell phone technology.

As a first step towards this vision, we investigate the possibility of exchanging energy between two interconnected homes. The core challenge is to enable the autonomous coordination of energy infrastructure in these individual households to ensure that energy is used efficiently. Given the lack of a centralised system, absence of banking/payment systems and low-processing power at hand, we need a tractable and decentralized energy exchange solution, that can operate

without financial payments between houses, and provides a benefit to all participants, compared to being disconnected.

Now, the idea of energy exchange is not new. There are several real world examples where energy exchange is used to improve energy management between countries (e.g. Finland and Sweden) and between cities (e.g. Delhi and Madhya Pradesh, India). Indeed, exchange of energy has already been shown to result in efficient use of energy and cost savings in utility companies [Ehtamo et al., 1988]. In this context, Ruusunen et al. [1991] considered a group of utility companies, each owning a generator, connected together to form a *power pool*. The cost of energy generation was different for each company and varied over the course of a day, allowing energy exchange to be beneficial to all. In their solution, energy generation and consumption is monitored by the *pool operator* who audits the cost and utility functions of the participants and distributes the cost savings among them. The energy exchange takes place according to a *static* exchange agreement which is established manually, prior to the exchange.

In contrast, Rosenschein and Zlotkin [1994] argued that in general, such mediated solutions are unnecessary, and automatic negotiation between software agents is a more efficient approach. They seek to design the rules of interaction for these negotiations such that a society of agents, as a whole, exhibits desirable properties such as stability (e.g. Nash equilibrium), efficiency (Pareto-optimality) and simplicity (strategy-proofness). Agents interact with each other in a decentralized manner obeying the *rules of encounter*. Careful design of such rules prevents the agents from being able to exploit the system and also reduces the overall complexity of interaction for the agents. They show examples of such rules for certain domains but conclude that there is no universal set of rules for general encounters and, therefore, a specific set of rules is required for a specific domain. Our work is in line with this approach, however, our setting is more challenging than the ones they consider since negotiation over energy exchange is multi-issue because agents are required to decide over the amount of energy exchange and also, how this exchange is scheduled across the day. Furthermore, these issues are interdependent as the recipient's utility for any period may depend on the energy received in earlier periods (since energy can be stored in our setting).

¹See the Rural Solar Homes in India (www.tatabpsolar.com), the Solar Homes program in Bangladesh (www.gshakti.org) and the Solar Village program in Ethiopia (www.solarsenegal.com).

More recent work addressing interdependent issues has focused on two tracks. Hindriks et al. [2006] attempt to remove interdependencies by approximating the utility space. However, they conclude that their technique only works with certain classes of utility functions where issues are not highly dependent (which is not the case in the problem of energy exchange) and cannot be applied to more general cases. Hattori et al. [2007], Ito et al. [2007], and Fujita et al. [2010] take a different approach and assume a central authority (a mediator) to which agents provide information about their utility functions. This centre calculates the set of Pareto-optimal solutions, from which the agents choose one. However, these solutions require the presence of an unbiased and independent mediator, capable of carrying out intensive computations. Such assumptions are hard to hold in our decentralized setting where there is no centre and households are required to negotiate directly to each other.

In contrast, in this paper, we address the problem of interdependent multi-issues negotiation to exchange energy, by providing a negotiation protocol which imposes three key restrictions on the offers that agents can make, and effectively reduces this problem to a single-issue negotiation problem but only minimally reduces the efficiency of the solutions found. We show that our approach leads to a Pareto-optimal, dominant strategy equilibrium and makes it easier for agents to compute optimal offers. In more detail, this work advances the state-of-the-art in the following ways:

1. We present a novel negotiation protocol that reduces the complexity of interdependent multi-issue negotiation in energy exchange problem, to a single-issue negotiation.
2. We prove that: (i) this protocol leads to a dominant strategy equilibrium and (ii) the outcome is Pareto-optimal. Offers can be calculated optimally using linear programming and the negotiation stops in two rounds.
3. We empirically evaluate our protocol in an example and show that in this case (i) the energy saved using our protocol can be up to 84% of the energy that can be saved using the Nash bargaining solution (ii) agents can increase their utility via our protocol up to 22% compared to utility without exchange and (iii) agents can maintain their utility with less battery storage (up to 37% less).

The rest of the paper is structured as follows. Our model is explained in Section 2 followed by a discussion of a theoretical solution in Section 3. Section 4 presents our negotiation protocol and its properties. We evaluate our protocol in a specific setting in Section 5 and conclude in Section 6.

2 Energy Exchange In Homes: the Model

In this section, we outline a model of energy exchange in two connected homes, each with a renewable generation unit, some loads and a battery to store electricity. Specifically, let a be an agent with a *generation capability* $\mathbf{k} = (k_1, \dots, k_n)$ representing the energy it can generate in n time periods and a *load* $\mathbf{h} = (h_1, \dots, h_n)$ representing its loads requirements. The battery is characterised by four parameters: a maximum storage capacity, s , a maximum charging rate, c_{max} , maximum discharging rate, d_{max} , and efficiency e . The efficiency describes the loss of energy when the battery is charged. We describe the dynamic state of the battery by: the energy flow into the battery (charge) $\mathbf{c} = (c_1, \dots, c_n)$, the flow going out

(discharge) $\mathbf{d} = (d_1, \dots, d_n)$ and the amount of charge stored in battery at any given time $\mathbf{q} = (q_1, \dots, q_n)$.

The generation capability $\mathbf{k}$ denotes the energy that *can* be generated, however, an agent may reduce its generation if the energy to be generated can neither be used immediately nor stored due to the limited battery flow or capacity. To capture this possibility, we use a *generation*, $\mathbf{g} = (g_1, \dots, g_n)$, to indicate the actual energy generated and *wasted energy*, $\mathbf{w}$, to denote the energy that was not generated or *wasted*. It is obvious that $\mathbf{k} = \mathbf{g} + \mathbf{w}$.

Using the battery, an agent can compute an *energy allocation*, $\mathbf{p} = (p_1, \dots, p_n)$, allocating the generated energy $\mathbf{g}$ to loads $\mathbf{h}$. Finally, the *link flow* $\mathbf{l} = (l_1, \dots, l_n)$ is used to denote the flow on the link between agents.

The utility of agent a at time t is the ratio of load p_t that is powered at time t , to total load required (h_t). The overall utility u^a is the sum of these ratios, given by:

$$u^a = \sum_{t=1}^n \frac{p_t}{h_t} \quad (1)$$

Thus, the goal of an agent is to power as much load as possible to maximise its utility. The battery is useful here as it gives the agent flexibility in deciding when to store and when to use energy and thus, it enables the agent to find an optimal energy allocation, $\mathbf{p}^*$, to maximize utility, given by:

$$\mathbf{p}^* = \operatorname{argmax}_{\mathbf{p} \in \mathbf{P}} \sum_{t=1}^n \left(\frac{p_t}{h_t} \right) \quad (2)$$

This can be transformed to a linear programming model subjected to the following constraints:

Constraint 1: At any given time t , the allocated power $p_t \in \mathbf{p}$ depends on the generated power g_t , battery charging flow c_t , discharging flow d_t and link flow l_t .

$$p_t = g_t - c_t + d_t + l_t \quad (c_1)$$

Constraint 2: The current battery state q_t depends on the last battery state $q_{(t-1)}$, charge $c_{(t-1)}$ and discharge $d_{(t-1)}$. The charge flow $c_t \in \mathbf{c}$ is subjected to the battery efficiency e . Also, the first state of the battery q_1 must equal the last battery state of the battery q_n to ensure there is no net change of battery charge over the day so that the utility remains dependent only on the energy generated in n time periods.

$$q_t = \begin{cases} q_{(t-1)} + e \times c_{(t-1)} - d_{(t-1)} & , t > 1 \\ q_n + e \times c_n - d_n & , t = 1 \end{cases} \quad (c_2)$$

Constraint 3: Allocated power p_t must not exceed load h_t .

$$p_t \leq h_t \quad \forall p_t \in \mathbf{p}, h_t \in \mathbf{h} \quad (c_3)$$

Constraint 4: The battery state q_t must not exceed the maximum capacity s . Also, the battery state cannot be negative, i.e. energy must be stored before it is drawn.

$$0 \leq q_t \leq s \quad \forall q_t \in \mathbf{q} \quad (c_4)$$

Constraint 5: At any time period t , the battery charge flow c_t must not exceed the maximum charge limit c_{max} . Also charge flow is always positive.

$$0 \leq c_t \leq c_{max} \quad \forall c_t \in \mathbf{c} \quad (c_5)$$

Constraint 6: At any time period t , the battery discharge flow d_t must not exceed the maximum discharge limit d_{max} . Also discharge flow is always positive.

$$0 \leq d_t \leq d_{max} \quad \forall d_t \in \mathbf{d} \quad (c_6)$$

Constraint 7: Wasted energy w_t is always positive and cannot exceed the energy k_t that can be generated at time t .

$$0 < w_t < k_t \quad \forall w_t \in \mathbf{w}, k_t \in \mathbf{k} \quad (c_7)$$

Constraint 8: Battery efficiency must be between 0 to 1 (i.e. 0% to 100%).

$$0 \leq e \leq 1 \quad (c_8)$$

Now, an agent can compute an energy allocation $\mathbf{p}^*$ which maximizes its utility via Equation 2 and constraints $\{c_1, \dots, c_8\}$. When an agent's objective is to compute a link flow that can maximize its utility, it can compute it as the following. Given Equation 1, substituting for p_t from c_1 :

$$u^a = \sum_{t=1}^n \left(\frac{g_t - c_t + d_t + l_t}{h_t} \right) \quad (3)$$

The objective is then to find the optimal link flow $\mathbf{l}^*$, such that Equation 3 is maximized:

$$\mathbf{l}^* = \operatorname{argmax}_{\mathbf{l} \in \mathbf{l}} \sum_{t=1}^n \left(\frac{g_t - c_t + d_t + l_t}{h_t} \right) \quad (4)$$

subjected to all constraints $\{c_1, \dots, c_8\}$ as listed above.

Given this model, the problem of finding an energy exchange agreement between two self-interested agents can now be simplified to the problem of finding an agreed link flow, $\mathbf{l}$, that provides each agent with greater utility than when disconnected (i.e., when $\mathbf{l} = 0$). However, the set of all such links is exponential in the number of time periods and thus, it is a challenging task for both agents to agree on a link given that each agent will prefer a different one (i.e., the link which maximizes its own utility).

In the next section, we elaborate on a game theoretic solution to find an agreed link flow between agents. In so doing, we establish the benchmark against which we compare our protocol (which we detail in Section 4).

3 The Nash Bargaining Solution: A Theoretical Benchmark

Given the model above, we now describe a game theoretic solution, known as the Nash bargaining solution [Nash, 1950; 1953], which allows the agents to find a satisfying solution for both. Nash bargaining solution (NBS) is defined by four axioms that must hold true: (i) invariance to utility scales (it is independent of how agents scale their utility) (ii) symmetry (it does not depend on the identity of agents), (iii) Pareto-optimal (the outcome is such that no agent can improve upon its utility without making another agent worse off) and independence from irrelevant alternatives. These axioms define a unique solution that maximizes the product of gains in utility of agents. Note that the NBS does not provide the mechanism by which agents might reach agreement, but it serves as the solution concept against which we will compare our own protocol. In the context of energy exchange, the *gain in utility* comes from the fact that via exchange an agent can

avoid energy storage losses and utilize energy that will be unused otherwise. To be clearer on this point, if an agent has 100% efficient battery and infinite storage, it cannot increase its utility via exchange.

Finding the NBS in the energy exchange problem basically involves finding the link flow between agents that maximizes the product of gains in their utility. Assuming two agents a and b , let d^a and d^b be the utilities that agents can avail when they are disconnected (also called *disagreement utilities*). These disagreement utilities are the maximum utilities that a and b can get when the link flow $\mathbf{l} = 0$, as we explained in Section 2. Let the solution set S_{NBS} be the set of all feasible link flows between a and b . Nash [1953] requires this set to be compact¹ and convex² for the solution to be unique. Since agents will only exchange if they get more utility than their disagreement utilities, $\forall \mathbf{l} \in S_{NBS} : (u^a(\mathbf{l}), u^b(\mathbf{l})) > (d^a, d^b)$. Given the set S_{NBS} , the Nash bargaining solution, $\mathbf{l}_{NBS}$, is obtained by:

$$\mathbf{l}_{NBS} = \operatorname{argmax}_{\mathbf{l}} [u^a(\mathbf{l}) - d^a] \times [u^b(\mathbf{l}) - d^b] \quad (5)$$

This equation is subjected to constraints listed in Section 2 for both agents a and b . Given compactness and convexity, computing the optimal solution is straight forward using convex optimization. However, interdependency between issues gives rise to a non-convex solution set with multiple NBS [Fujita, Ito, and Klein, 2010] whereby stochastic optimization techniques (we use simulated annealing) is needed to find the NBS. In general, negotiation mechanisms which aim to reach the NBS come with the implicit assumption that agents reveal their information truthfully either to each other or to a mediator so that the NBS can be computed. In a society of honest agents, where time and computational power are abundant, such mechanisms can be a good candidate to solve the problem of energy exchange. However, these assumptions do not hold in our setting thus the direct use of NBS is not valid in our setting. However, it can serve as an upper bound of the performance of a negotiation protocol. In the next section, we propose a negotiation protocol which does not require such assumptions and can be used to reach energy exchange agreements in the settings describe earlier.

4 Energy Exchange Protocol

Here we propose our energy exchange protocol (EEP) which allows agents to make offers to each other in order to reach an agreement on the link flow between them. We call these offers *link flow offers* and the agreement, an *exchange agreement*. The protocol imposes restrictions on the offers that agents can make and dictates that whichever offer minimizes the load on the link is selected as the exchange agreement. The restrictions on the offers reduce the set of feasible solutions between agents and ensure that the solution is reached with certain properties (listed in Section 4.2).

Before we define our protocol, we define our assumptions and terminology. We consider each exchange agreement in isolation, independent of the past and future exchange agreements. Also, we consider exchange over finite time (e.g. a

¹ A set is compact if it is closed and bounded.

² A set C is convex if for all x and y in C and all t in the interval $[0, 1]$, the point $(1 - t)x + t(y)$ is in C .

day) which can be divided into *exchange periods*. An exchange period is a unit of time for energy exchange and consists of at least one *time period*. A time period is an atomic unit of time and each entry in a profile (e.g. energy generation) is described against a single time period. Given this protocol, we assume two agents a and b , with utility functions u^a and u^b , disagreement utilities d^a and d^b respectively, and let S_p be the set of possible link flows between them, i.e. $\forall l \in S_p : (u^a(l), u^b(l)) \geq (d^a, d^b)$, the same set as defined in Section 3. The EEP imposes three restrictions, r_1, r_2 and r_3 on the offers and allows agents to make offers at most once such that the negotiation results in a dominant strategy equilibrium and a Pareto-optimal outcome. The outcome (the agreed link flow) is denoted by l^{ex} . Figure 1 describes the EEP in detail.

4.1 Computing the Valid Link Flow Offers

Given the protocol, we now define how agents can compute valid link flow offers. As discussed in Section 2, each agent will prefer the link flow that maximizes its own utility. An agent making an offer can compute a valid link flow offer which maximizes its utility by using Equation 4 with two more constraints, $\{r_1, r_2\}$ (in addition of $\{c_1, \dots, c_8\}$), and an agent making a counter-offer needs to include a further constraint r_3 . In fact, agents can offer *any* link flow provided it is valid, not just the one that maximizes its utility. However, in Section 4.2, we prove that the dominant strategy for agents is to offer the link flow that maximizes its utility, leaving no need for them to strategize.

Before we explain the properties of the EEP, we give an intuitive example to show how it will work in action:

Example 1. *Imagine that in a society of agents, the following are the already agreed conventions:*

1. *The total time of an exchange is 24 hours. The exchange starts at 0600 hours local time and ends at 0600 hours next day.*
2. *This day is divided into two exchange periods, each consists of 6 two-hours-long time periods.*

Given these conventions, an agent a computes a valid link flow offer, $l^a = (2, 2, 2, 2, 2, 2, -2, -2, -2, -2, -2, -2)$, which maximizes its utility. Agent a makes this offer to agent b . However, Agent b finds that its utility is maximized with the link flow, $l^b = (1, 1, 1, 1, 1, 1, -1, -1, -1, -1, -1, -1)$. Since $l^b <^3 l^a$, b makes this offer to a . Agent a accepts this offer and exchange takes place as per l^b .

4.2 Properties of Our protocol

Before outlining the properties, we describe our notation. Let S be the set of *valid link flows* between a and b obeying the rules defined by the EEP. If $\exists l \in S : u^a(l) > 0$ then $\forall l' \in S : 0 < l' < l, u^a(l') > 0$. For example, if agent a prefers an exchange $(2, 2, -2, -2)$, then $(1, 1, -1, -1)$ is also a feasible link flow for a , since $(1, 1, -1, -1) \in S$ and $(1, 1, -1, -1) < (2, 2, -2, -2)$, although $u^a[(1, 1, -1, -1)] < u^a[(2, 2, -2, -2)]$. It actually follows that given a link $l^* \in S : u^a(l^*) > u^a(l) \forall l \in S : l \neq l^*$, then all $l \in S : 0 < l < l^*$ are feasible and u^a is a strictly monotonically increasing function on this range.

³The inequality here is the *vector inequality*.

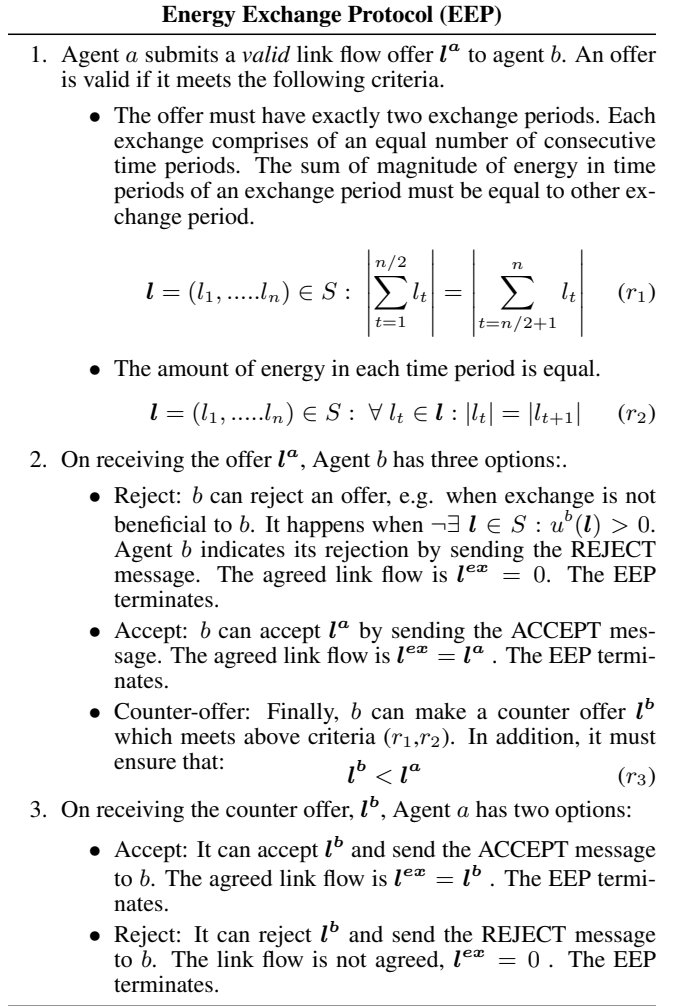


Figure 1: The Negotiation Protocol

This monotonicity arises from the fact that the negotiation has been reduced to a single-issue, i.e. the amount of flow in one time period (since the amount of energy in each time period must be equal - see r_2). Now we prove the properties.

1. Dominant Strategy Equilibrium

Theorem 1. *The agent making an initial offer has a dominant strategy which is to offer the link that maximizes its own utility.*

Proof. Let's assume two agents a and b , and $l^a \in S$ is the link that optimizes the utility of agent a . Let's imagine that a wants to manipulate the protocol by reporting some other link $\hat{l}^a \in S$ to agent b . Table 1 lists all possible scenarios that a faces in this case. It is evident that reporting l^a weakly dominates reporting $\hat{l}^a$. Specifically, in the first two cases, reporting $\hat{l}^a$ gives the same utility as reporting l^a , however it is strongly dominated in the rest of the four cases (unless $\hat{l}^a = l^a$). Thus, an agent maximizes its utility by reporting $\hat{l}^a = l^a$. $\square$

Theorem 2. *An agent making a counter-offer has a dominant strategy to offer the link that maximizes its own*

Case	Agent a reports $\hat{l}^a$		Agent a reports l^a		Result
	Chosen Link $\min(\hat{l}^a, l^b)$	u^a	Chosen Link $\min(l^a, l^b)$	u^a	
$\hat{l}^a \geq l^a \geq l^b$	l^b	$u^a(l^b)$	l^b	$u^a(l^b)$	$u^a(l^b) = u^a(l^b)$.
$l^a \geq \hat{l}^a \geq l^b$	l^b	$u^a(l^b)$	l^b	$u^a(l^b)$	$u^a(l^b) = u^a(l^b)$.
$l^a \geq l^b \geq \hat{l}^a$	$\hat{l}^a$	$u^a(\hat{l}^a)$	l^b	$u^a(l^b)$	$u^a(\hat{l}^a) \leq u^a(l^b) \because \hat{l}^a \leq l^b \leq l^a$ (monotonicity).
$l^b \geq l^a \geq \hat{l}^a$	$\hat{l}^a$	$u^a(\hat{l}^a)$	l^a	$u^a(l^a)$	$u^a(\hat{l}^a) \leq u^a(l^a) \because u^a(l^a) > u^a(l) \forall l \in S : l \neq l^a$.
$l^b \geq \hat{l}^a \geq l^a$	$\hat{l}^a$	$u^a(\hat{l}^a)$	l^a	$u^a(l^a)$	$u^a(\hat{l}^a) \leq u^a(l^a) \because u^a(l^a) > u^a(l) \forall l \in S : l \neq l^a$.
$\hat{l}^a \geq l^b \geq l^a$	l^b	$u^a(l^b)$	l^a	$u^a(l^a)$	$u^a(l^b) \leq u^a(l^a) \because u^a(l^a) > u^a(l) \forall l \in S : l \neq l^a$.

Table 1: Reporting the link, l^a , which maximizes its utility, is the dominant strategy for agent a .

utility.

Proof. Imagine agent a with $l^a \in S$ which optimizes its utility and $l^b \in S$, the offer it has received from agent b . If $\hat{l}^a$ is the link that a wants to report b to manipulate the protocol, then Table 1 lists all the possible scenarios. Again, reporting l^a weakly dominates $\hat{l}^a$. $\square$

It is evident that revealing the link that maximizes its utility, is a weakly dominant strategy for both initial-offer and counter-offer making agents. As a by-product, this property also renders the order, in which agents make offers to each other, irrelevant. The solution is the same no matter which agent commences the negotiation. The result is always a dominant strategy equilibrium.

2. Pareto-Optimality

Let's imagine that $\exists l \in S : u^a(l) > 0$ and $u^b(l) > 0$. As proved above, agent a and b will reveal truthfully the links l^a and l^b , that maximizes their utilities u^a and u^b . Assume that $l^a > l^b$, then the EEP dictates that l^b is the agreed link flow. Since l^b maximizes u^b and it is unique (u^b is strictly monotonic), any other link flow which increases u^a will decrease u^b , hence the protocol is Pareto-optimal.

3. Tractability

Our solution is tractable as each agent needs to make at most one offer. Also, an optimal offer can be computed using standard linear programming solver (Section 2).

5 Empirical Evaluation

Having proven the properties of our protocol, we now set-up a realistic example to demonstrate its practical applicability. To this end, we consider an example of energy exchange between two agents, where one agent has a 1.5kW wind turbine and the other has a 1.75kW solar panel. The energy generation data for the wind turbine comes from a wind farm near Lugo, Northwest Spain (www.sotaventogalicia.com), while the output of the solar panel is estimated to be directly proportional to the daily radiance for the same region (www.re.jrc.ec.europa.eu/pvgis/apps/radday.php). We use data for July 2010, estimate the average generation for a day and scale it to match the output of a 1.5kW wind turbine and a 1.75kW solar panel. At present, the load requirements of homes in remote areas are not available so we use load data, recorded and provided by a UK electric company in low-income homes equipped with smart meters. We also assume that agents have identical batteries that have a maximum capacity, $s = 20\text{kWh}$, a maximum charging rate, $c = 4\text{kWh}$, a maximum discharging rate, $d = -4\text{kWh}$ and efficiency,

$e = 90\%$. We note that the generation data for agents have some disparity and all other data (including loads) and batteries are identical for both agents. Given these profiles, agents can compute their utilities (without exchange) using a linear programming solver (see Section 2). We use IBM ILOG CPLEX, a powerful optimizer, which provides easy and rich methods to model constraints, variables and objective functions and flexible interfaces to many development tools.

For a comparative analysis of our protocol, we compute the utilities that agents can get (i) without exchange, (ii) by computing the NBS using simulated annealing (see Section 3) and (iii) by using our EEP (Section 4.1). Moreover, we consider how these utilities vary as we reduce the battery flow and storage capacity of agent a as shown in Figure 2.

Consider first Figure 2(a) which shows the utilities of agent a and b , and the total energy saved via exchange, as the battery efficiency of agent a is reduced. All the other factors, (including the battery efficiency of agent b) are kept constant. We can see that agents can increase their utilities via exchange. The amount by which it increases depends on the type of solution used and the battery efficiency. The *total energy saved via exchange* is the energy which would otherwise be lost (either in storage loss or due to limited battery flow or storage - Section 3) without exchange. We also see that as the battery efficiency of agent a is reduced, so does its ability to meet its load, therefore, its utility reduces. While for b , this reduction in the battery efficiency of a does not affect its utility for without exchange and with the EEP, however, for the NBS its utility increases. The reason that b gets better utility via the NBS as the battery efficiency of a is reduced, is because storage loss for a increases and thus, the amount of energy that can be saved in exchange via exchange increases, as shown in Figure 2(a). However, this saved energy is the result of cooperation between agents and the NBS divides it between both agents and hence, the increase in utility of b . Thus, with reduction in the battery efficiency, the energy exchange becomes more beneficial as agents can reduce their storage losses by exchanging energy.

We also note that the NBS gives both agents better utilities compared to the EEP, however, its computation comes with the assumptions discussed in Section 3. To compare the EEP against the NBS, we measure the difference in the energy saved using these solutions. We observe that this difference depends on the battery efficiency of a and that the EEP can save from 29% to 84% (Figure 2(a) and Figure 2(b), respectively) compared to the energy saved with the NBS.

Figure 2(b) shows the utilities of the agents as the battery capacity of agent a is reduced. Again, as in Figure 2(a), agents can get better utilities with exchange. Also evident is the fact that as storage is reduced, energy exchange becomes

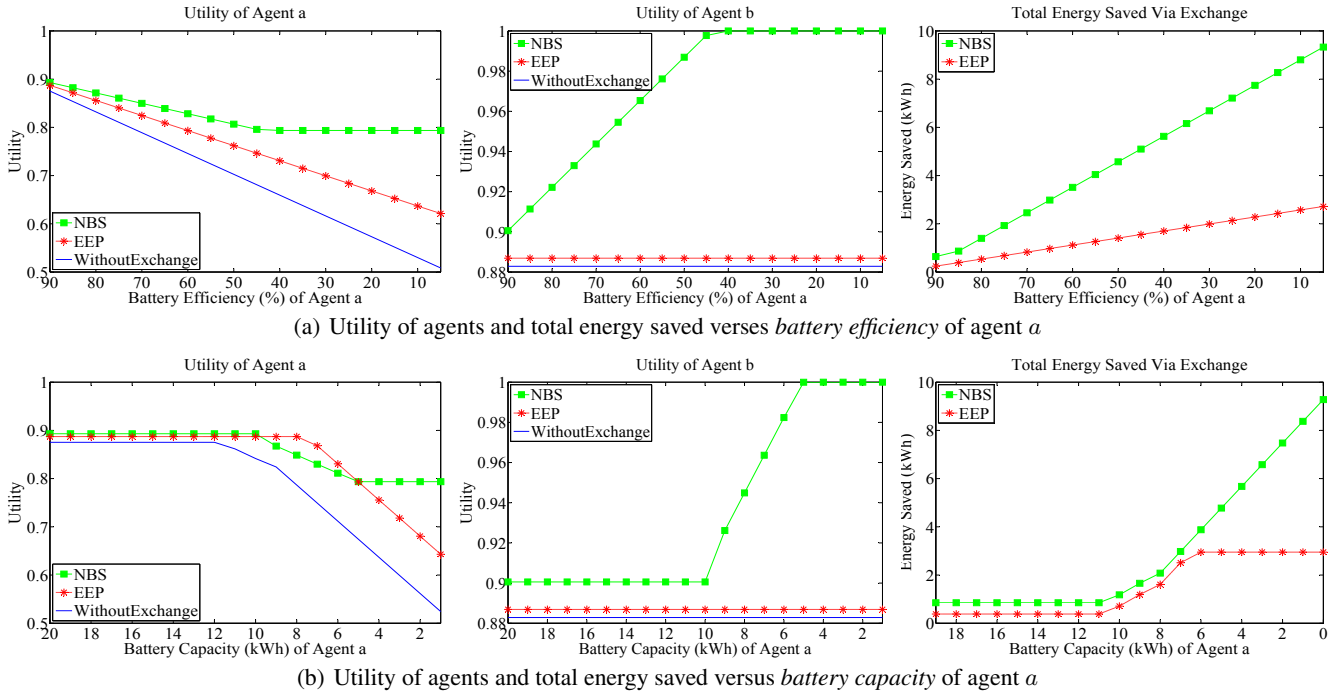


Figure 2: Total energy saved and utility of agents (without exchange, with EEP and with NBS), as battery efficiency and capacity is reduced.

more useful as agents can exchange energy instead of storing it. An important observation here is that an agent can maintain the same level of utility with a smaller battery via exchange that it obtains otherwise. For example, we see that *a* maintains the same level of utility in exchange via our protocol with a 7kWh battery that it gets without exchange with an 11kWh battery. This corresponds to almost a 37% reduction in battery size. Though, the same phenomenon can be seen for battery efficiency, what makes battery capacity more interesting here is the strongly correlated relationship between the battery capacity and cost, and thus, reducing the battery size will significantly reduce its cost.

6 Conclusion and Future Work

The problem of energy exchange in homes is an interdependent multi-issue negotiation problem. Such problems are complex and computationally expensive to negotiate over. We present a novel negotiation protocol, the EEP, which imposes certain restrictions to reduce the complexity of the energy exchange problem and leads to a dominant strategy equilibrium and a Pareto-optimal outcome. We prove these properties of the EEP and demonstrate via example that agents can increase their utility (up to 22%), reduce the required storage capacity (up to 37% less) and that the energy saved using the EEP can be up to 84% of the energy that can be saved using the NBS. In our example, only the energy generation data for agents is different and introducing diversity in other data (e.g. loads) can make energy exchange more useful and lead to even better results. We conclude from our presented work and the empirical results that energy exchange between homes has the obvious potential of better energy management. Although, we have confined our vision to homes in developing countries, energy exchange can also be useful in the smart grid vision. In future work, we aim to extend this model and investigate a similar set of rules to reach a dominant strategy equilibrium in multiagent society with less re-

strictions. We also aim to investigate the cases where energy generation is uncertain and where loads are deferrable.

References

- [Ehtamo et al., 1988] Ehtamo, H.; Ruusunen, J.; Kaitala, V.; and Hamalainen, R. 1988. Solution for a dynamic bargaining problem with an application to resource management. *Jr. of Optimization Theory* 59(3):391–405.
- [Fujita, Ito, and Klein, 2010] Fujita, K.; Ito, T.; and Klein, M. 2010. A secure and fair protocol that addresses weaknesses of the Nash bargaining solution in nonlinear negotiation. *Group Decision and Negotiation*. 1–19.
- [Hattori, Klein, and Ito, 2007] Hattori, H.; Klein, M.; and Ito, T. 2007. A multi-phase protocol for negotiation with interdependent issues. In *Proc. of IAT*, 153–159, Sydney, Australia.
- [Hindriks, Jonker, and Tykhonov, 2006] Hindriks, K.; Jonker, C.; and Tykhonov, D. 2006. Eliminating interdependencies between issues for multi-issue negotiation. *Cooperative Info. Agents, Lecture Notes in Computer Science*. Springer.
- [IEA, 2008] International Energy Agency. *World Energy Outlook*.
- [Ito, Hattori, and Klein, 2007] Ito, T.; Hattori, H.; and Klein, M. 2007. Multi-issue negotiation protocol for agents: exploring non-linear utility spaces. In *Proc. of IJCAI, Hyderabad, India*.
- [Nash, 1950] Nash, J. 1950. The bargaining problem. *Econometrica* 18(2):155.
- [Nash, 1953] Nash, J. 1953. Two-person cooperative games. *Econometrica* 21(1):128–140.
- [Rosenschein and Zlotkin, 1994] Rosenschein, J. S., and Zlotkin, G. 1994. Rules of encounter: designing conventions for automated negotiation among computers. Cambridge, MA, USA: MIT Press.
- [Ruusunen, Ehtamo, and Hamalainen, 1991] Ruusunen, J.; Ehtamo, H.; and Hamalainen, R. 1991. Dynamic cooperative electricity exchange in a power pool. *Systems, Man and Cybernetics, IEEE Transactions on* 21(4):758–766.

Alarm Processing with Model-Based Diagnosis of Event Discrete Systems

Andreas Bauer, Adi Botea, Alban Grastien, Patrik Haslum, Jussi Rintanen

NICTA and the Australian National University
Canberra, Australia

Abstract

Reliable and informative alarm processing is important for improving the situational awareness of operators of electricity networks and other complex systems. Earlier approaches to alarm processing have been predominantly syntactic, based on text-level filtering of alarm sequences or shallow models of the monitored system. We argue that a deep understanding of the current state of the system being monitored is a prerequisite for more advanced forms of alarm processing.

We use a model-based approach to infer the (unobservable) events behind alarms and to determine causal connections between events and alarms. Based on this information, we propose implementations of several forms of alarm processing functionalities. We demonstrate and evaluate the resulting framework with data from an Australian transmission network operator.

1 Introduction

Many views grouped under the broad Smart Grid umbrella agree that the power networks of the future will produce very large volumes of data. Compared to current-day networks, one source of additional data will be the deployment of more sensors, including new types of sensors such as smart meters. Furthermore, future networks will change their configuration more dynamically, to work with wind turbines and other variable-output sources, and to handle overload, voltage, and phase imbalance issues caused by charging electric cars [Ipakchi and Albuyeh, 2009]. This dynamic nature of networks will further increase the volume of data generated.

An alarm is a message that signals a discrete event in the network, such as automatic protection equipment triggering or an analog sensor measurement crossing a pre-defined threshold value. As an example, Figure 1 shows a small excerpt from an alarm log generated by a transmission network. The purpose of alarms is to alert operators to changes, signifying potential error conditions. However, actual faults frequently give rise to “alarm cascades”, where the original fault causes a range of secondary abnormalities, all of which generate multiple alarms, thus quickly overwhelming operators’ attention.

00:00:00	CB 1B A-B	–OPEN–
00:00:00	CB 2B A-B	–OPEN–
00:00:00	CB 2A A-B	–OPEN–
00:00:00	CB 1A A-B	–OPEN–
00:00:01	Line A-B	KV LIMIT LOW
00:00:04	Line C-D	KV LIMIT NORMAL
00:00:15	CB 1A A-B	–CLOSED–
00:00:17	Line A-B	KV LIMIT NORMAL
00:00:17	CB 1B A-B	–CLOSED–
00:00:17	CB 2B A-B	–CLOSED–
00:00:20	CB 2A A-B	–CLOSED–
00:00:20	Line C-D	KV LIMIT HIGH

Figure 1: Excerpt from an alarm log, spanning 20 seconds. (Names have been simplified.)

This problem is widely recognised, and techniques that aim to deal with it are collectively known as “intelligent alarm processing”. Kirschen and Wollenberg [1992] identify three functions that intelligent alarm processors should provide: reduce the number of alarms, provide a clearer idea of the cause of the alarms, and recommend corrective actions to the operator. Adopting a similar view, McDonald *et al.* [1991] argue that the role of alarm processing is to transform raw alarm data into a format that is more digestible to a human operator. The increasing availability of networked sensors, and the increasing dynamicity of the future power grid, will only further emphasize the need for intelligent alarm processing techniques to assist human operators in making sense of the data that systems provide them with.

In this paper we present a model-based approach to analysing the evolution of systems like electricity distribution and transmission networks. We use a discrete-event model that describes the possible behavior of the network, at an abstract level. An online diagnosis process infers comprehensive information about the (recent) evolution and the current state of the network. This information is the basis for multiple alarm processing functions, each of which provides a different kind of meaningful summary that can be quickly grasped by operators. For example, in the log excerpt in figure 1, the group of alarms relating to the “A–B” line can all be explained by one single cause (a transient ground fault on the line), and clearly separated from the independent events (voltage fluc-

tuations) taking place at the same time on the “C–D” line.

The diagnosis engine takes as input a log of alarms, the model of the network, and the state of the network (e.g., closed/open status of switches) at the beginning of the interval to analyse. Taking the system state into account is essential, because the causal links between events often depend on the current status of network components. For instance, where an electrical fault in one part of the network will propagate, and thus which alarms it will lead to, depends on the status of switches. Our model also includes time constraints between events. Few earlier alarm processing techniques do this, notable examples being chronicles [Laborie and Krivine, 1997a; 1997b; Taisne, 2006; Cordier *et al.*, 1998] and temporal constraint networks [Guo *et al.*, 2010]. The result of the diagnosis process is a *scenario*, which is an evolution of the system (i.e., series of states and transitions) that is consistent with the observed alarms. There may be several scenarios that are compatible with the observations, and preference is given to those that contain the fewest unexplained and unlikely events.

The purpose of alarm processing is to extract those pieces of information that are most relevant to the control room operators in a given situation. This can take the form of compact summaries of long event sequences, or highlighting the most important events that the operator should be aware of. The scenarios themselves contain far too much detail to be presented to operators in raw form. Different filtering methods are applied to the generated scenarios, each extracting a different type of summary to be presented to the operator. In this paper, we present four different techniques, namely alarm clustering, root cause analysis, fault-independent alarm highlighting and highlighting of live alarms (cf. Section 4). However, the result of the diagnosis process can be put to more uses than these specific alarm processing functions. It provides a powerful basis for implementing further functions useful in a control room, such as computing plans for corrective action.

We have developed a prototype system, which we tested on a real alarm log, provided by TransGrid, the transmission network operator in New South Wales and the Australian Capital Territory, Australia. In spite of being only a prototype, the diagnosis engine and alarm processing functions are fast enough to process all but the largest alarm cascades in less than a minute. The network model that we developed is also a highly simplified prototype; in spite of this, the system generates meaningful summaries in many instances.

2 Related Work

The literature contains a wide variety of techniques applied to alarm processing and fault diagnosis, including neural networks [Lin *et al.*, 2004], expert systems [Protopapas *et al.*, 1991; McDonald *et al.*, 1991], fuzzy logic [Meza *et al.*, 2001], tabu search [Wen and Chang, 1997], genetic algorithms [Wen *et al.*, 1995], chronicle-based approaches [Taisne, 2006; Cordier *et al.*, 1998], and temporal constraint networks [Guo *et al.*, 2010].

Model-based alarm processing is a recurring theme in the literature. Lin *et al.* [1998] use a power flow relation model. Dahlgren *et al.* [1998] view causal relations between types

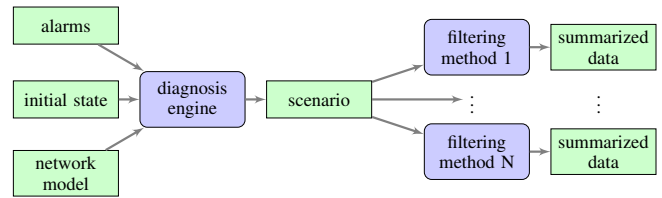


Figure 2: Architecture of the alarm processing system. The diagnosis engine creates a scenario (cf. Sections 3.2 and 3.3), which is the basis for multiple filtering methods. Each filter creates a different kind of summary view of the alarm log, which is presented to the operator. In Section 4, we describe four different filtering techniques.

of alarms as model-based knowledge. A similar approach is followed by Larsson and DeBor [2007; 2002]. More recently, Guo *et al.* [2010] have used temporal constraint networks (TCNs) as a model. This permits specifying temporal constraints on the occurrence of individual events, as well as between cause-effect pairs of events. However, this model is still limited to defining such causal relations statically, i.e., whether or not one event causes another does not depend on the state of the network. This precludes specifying many essential aspects of the behaviour of the network. For example, a circuit breaker opening may cause automatic shutdown of a generator, *if* the opening of that breaker isolated the generator from a significant part of its load; but whether or not this is the case clearly depends on the state of other switches.

A notable work in this context is that of Laborie and Krivine [1997a; 1997b]. They use a more detailed model, very similar to ours, containing the topology of the network together with an automata-based description of the behavior of each component type. The main difference between their approach and ours lies in how this model is used. They use the model for an off-line simulation of the system, which is used to pre-compute a collection of event patterns called chronicles. These chronicles are then compared on-line against the evolving alarm log, and the output of alarm processing is matching instances of chronicles appearing in the log.

Lin *et al.* [1998] present a combination of rule-based and model-based alarm processing. Rule-based reasoning performs message synthesis. A power flow relation model is used to find relations between alarms. The model contains the topology of the system (components and connections) together with information on how changes in the parameters of one component will affect the parameters of other, adjacent components. The program is tested on alarms corresponding to incidents at the Taipei Mass Rapid Transit System.

3 Overview of the Approach

The overall architecture of our alarm processing system is summarized in Figure 2. In this section we discuss briefly each of the main components and the way they work together.

3.1 Network Model

The model we use contains all the application-specific knowledge on which the inferences made by the diagnosis and

```

componenttype Generator {
  var status : { on, off };
  var status_changing : bool;
  var runback : bool;

  // output events (observable)
  observable event runback_alarm;
  observable event runback_reset;
  observable event unit_status_ON;
  observable event unit_status_OFF;

  // output events (for synchronization)
  event isolator_open;
  event isolator_close;

  // shutdown

  transition begin_shutdown
    status = on and status_changing = false
    -> status_changing := true;

  transition shutdown_breaker_open
    status = on and status_changing = true
    -> isolator_open;

  transition shutdown_turn_off
    status = on and status_changing = true
    -> unit_status_OFF;

  transition shutdown_complete
    status = on and status_changing = true
    -> status := off, status_changing := false;

  // startup

  transition begin_startup
    status = off and status_changing = false
    -> status_changing := true;

  transition startup_breaker_close
    status = off and status_changing = true
    -> isolator_close;

  transition startup_turn_on
    status = off and status_changing = true
    -> unit_status_ON;

  transition startup_complete
    status = off and status_changing = true
    -> status := on, status_changing := false;

  // runback

  transition runback_alarm_unexplained
    true -> runback := true, runback_alarm;

  transition runback_reset_unexplained
    true -> runback := false, runback_reset;

  // the runback alarm can reset when the generator is off

  transition runback_reset_when_off
    status = off -> runback := false, runback_reset;
}

```

Figure 3: MMLD model of the Generator component type. The model has been slightly simplified for readability. For example, time constraints have been left out.

alarm processing system are made. The advantage of the model-based approach is reusability: to adapt the software to changes in the network, or to deploy it in a new network, only the model needs to be updated. We designed a specification language, called MMLD, for describing the model. Here, we will not discuss the syntax of MMLD, but focus on the underlying concepts.

The model is a component-based, discrete-event dynamical system [Cassandras and LaFortune, 1999]. (It can be viewed as a factored description of a very large finite state machine, augmented with time constraints.) Power grids are very large, but consist of many instances of the same, rela-

tively few, component types. We separate the model into descriptions of the behavior of component types, and the topology, which enumerates the actual components, of each type, and describes how they are interconnected in the grid. This separation makes the model specification more compact, and further improves reusability since component types are also generic, to some extent. An example of a component type model, slightly simplified, in MMLD syntax, is shown in Figure 3. The component type is a power generator.

The model of a component type describes its dynamical behaviors, both nominal and abnormal, at a relatively high level of abstraction. At this level of abstraction, a component (instance of the type) can be in one of a finite set of discrete states, and changes between states by discrete events. Each component type has a set of (local) transition rules, which define under what conditions it may, or must, change its (local) state. Causal relations between events are enforced by the transition rules preconditions and effects on the component state, and by time constraints: a minimum time constraint specifies that the transition must be enabled for at least a certain amount of time before it fires; a maximum time constraint specifies that the transition must take place if it is continuously enabled long enough. For example, fault current through a protection relay not only enables it to trigger circuit breakers, but forces it to do so within a set time limit.

Components interact by two mechanisms. The first is synchronisation of events. When a certain event takes place in one component, the synchronised event occurs simultaneously in the other component. This is used to model fast interactions, such as, for example, the trip signal from a protection relay to a circuit breaker. The second is that transition rule preconditions may refer to variables in neighbouring components, and to certain global properties of the state of the network, such as, for example, the existence of a conducting path to an active generator.

For the purpose of alarm processing, the most important distinction in the model is between events that can be explained, i.e., that follow logically from earlier events (observed or unobserved), and events that have no visible explanation, i.e., that are simply hypothesised to have occurred. These are analogous to the “root cause” events that explain observed alarms in simpler model-based approaches [Wen and Chang, 1997; Wen *et al.*, 1995; Dahlgren *et al.*, 1998; Guo *et al.*, 2010]. Given a log of observations, the diagnosis engine will find a scenario that contains the fewest unexplained events. We can also assign different likelihoods to unexplained events, to find the most likely explanation.

The level of detail captured in the model represents a trade-off between its explanatory power and the complexity of reasoning. For example, circuit breakers opening to isolate a line (perhaps triggered by a protection relay) may cause a change in the flow of power through other parts of the network, triggering, for instance, a low voltage alarm. Because our model does not include many detailed aspects of power flow (those calculations would be too expensive to carry out as part of the diagnostic reasoning) we may not be able to explain the voltage alarm. But, we can explain, e.g., an alarm indicating that voltage has dropped to zero on the isolated line, because this follows from a simpler, qualitative causal relationship.

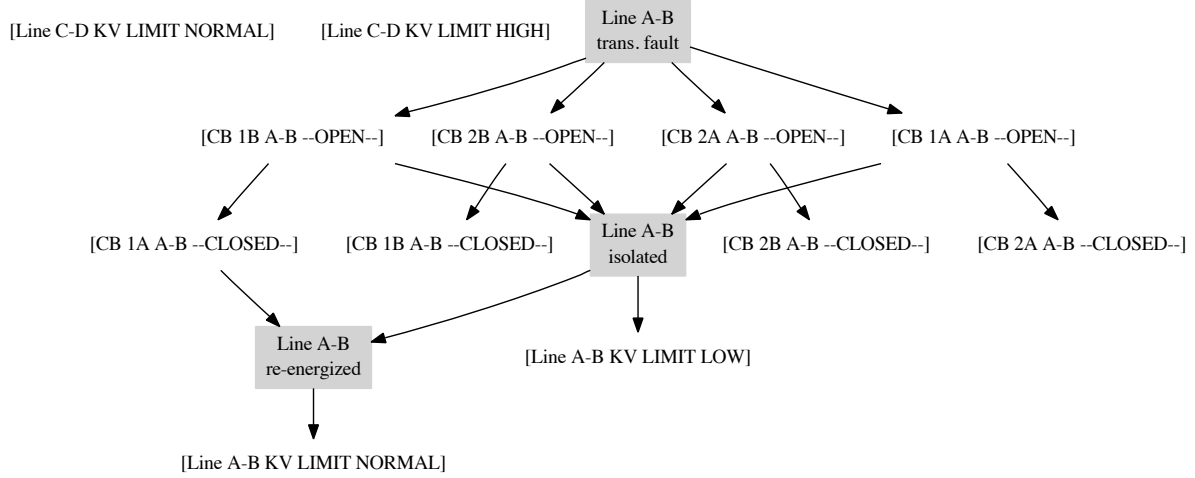


Figure 4: Scenario found by the diagnosis engine (simplified for readability). Shaded events are unobservable. Edges represent causal relationships.

3.2 Diagnosis Engine

We use a generic discrete-event systems diagnosis solver (diagnoser). It takes as input 1) a network model in MMLD format, 2) the (possibly partially known) initial state of the network describing e.g. which switches are open/closed, and 3) the observations (alarms) generated by network devices.

The reasoning performed by the diagnoser is similar to discrete state estimation under incomplete information [Zanella and Lamperti, 2003]. It involves finding a discrete state sequence that is compatible with the observations and that minimizes the occurrence of unlikely events, such as faults and other spontaneous, inexplicable events. This is a hard combinatorial problem, due to the very high number of states the system may be in. For example, in an electricity network with n switches, the open/closed status of the switches alone represents 2^n different possible discrete states. This makes the use of trivial enumerative methods ineffective.

Our diagnoser uses satisfiability (SAT) algorithms as the search mechanism. The SAT problem involves finding an assignment of values *true* and *false* (equivalently, 1 and 0) to a number of Boolean variables so that the set of clauses (Boolean sum of variables and negated variables) representing a Boolean function is satisfied (evaluates to *true*). The Boolean variables correspond to the state variables of the system at different time points, representing a sequence of states, and the events that take place at different time points. Clauses express dependencies between state variables within one time point or between consecutive time points and the observation sequence [Grastien *et al.*, 2007].

A SAT algorithm, enhanced with conflict-directed clause learning [Moskewicz *et al.*, 2001], is used for finding a value assignment to the variables. Minimization of unlikely events is achieved by repeatedly finding solutions with stronger and stronger constraints on the number of such events.

The SAT approach to diagnosis requires a bound N on the number of consecutive states. This bound is obtained from

the number of observations made. Then a propositional formula Φ is defined that constrains the value of each state variable after the i th timestep (for all $i \in \{0, \dots, N\}$) and the event occurrence at the i th timestep (for all $i \in \{1, \dots, N\}$). For instance, if the event `cb_open` can take place only when the state variable `cb_state` equals `closed`, then the formula would include: $(\neg \text{cb_open}@1 \vee (\text{cb_state} = \text{closed})@0)$ which specifies the above property for the first transition. The formula is augmented to enforce the initial state and the observations. Any assignment of the variables satisfying Φ represents a scenario, i.e., a system evolution consistent with the given observations.

The time constraints require specific treatment. The timesteps are defined with respect to the alarms and the alarms are timestamped. Additional constraints are included in the SAT formula to ensure these time constraints are satisfied. Consider for instance that a specific circuit breaker requires at least 1s delay to notice an overload; consider further that the timestep 1 to 6 are stamped as follows: $\tau(1) = 0s; \tau(2) = 0.2s; \tau(3) = 0.5s; \tau(4) = 1s; \tau(5) = 1.3s; \tau(6) = 1.6s$. Then the event `overload_detected` can take place at timestep 5 only if the `overload` state variable equals `true` before 0.3s, i.e. at timestep 2. This is encoded by the following formula for all $t \in \{2, \dots, 5\}$: $\text{overload_detected}@5 \rightarrow (\text{overload} = \text{true})@t$.

To minimise the number of undesirable (i.e., unexplained or unlikely) events, we augment the formula Φ with a cardinality constraint on the number of occurrences of such events [Anbulagan and Grastien, 2009]. This constraint is initially set to 0 in order to find an explanation with no unlikely event and it is incremented until the SAT formula becomes satisfiable, thus exhibiting a preferred scenario.

3.3 Scenarios

A scenario $\mathcal{S} = (\Sigma, \Theta)$ is a double sequence of global states Σ and global transitions Θ , where a global transition $t \in \Theta$ is a collection of synchronized local transitions, and a global

state to a collection of local states. The first state q_0 of Σ is an instantiation of the (possibly partially unspecified) initial state of the instance. Each transition $t_i \in \Theta$ leads the system from state q_{i-1} to state q_i . There are several ways a transition can affect a component state: directly if the triggering event of the transition takes place in the component, through event synchronisation if the triggering event forces an event on the component (for instance, a sensor detecting a fault may force a circuit breaker to open), or through dependant variables (for instance, an isolator electrifying all lines in a feeder).

Figure 4 shows a graphical representation of the scenario found by the diagnosis engine for the small example log in Figure 1. (The picture shows only the events that occur in the scenario, not states.)

4 Scenario Filtering

Scenarios computed with model-based diagnosis are a rich source of information, but are too detailed to present directly to operators. Further processing is required to obtain summarized data that an operator can quickly understand. In this section we discuss four alarm processing functions, each providing different types of information to the operator.

4.1 Alarm Clustering

Alarm clustering partitions an alarm log into causally independent subsets which can be understood more easily.

Clustering is performed on all events in a scenario, including unobservable ones that have been inferred from entries in the alarm log. The clustering of the alarm log is then trivially obtained by filtering away all hidden events. In the rest of this section, we use the term cluster to refer to collections of events including unobservable ones.

For example, in the scenario found by the diagnosis engine for the alarm log in Figure 1, the set of all alarms referring to the A-B line, together with several inferred unobservable events, form a cluster. The two alarms referring to the C-D line, however, are not clustered, because there is no (apparent) causal relationship between them.

Our clustering is based on partitioning by an equivalence relation (transitive, reflexive, symmetric), which expresses causal dependencies between events. Let $\mathcal{S} = (\Sigma, \Theta)$ be a scenario with an ordered set of states and an ordered set of global transitions. A global transition $t \in \Theta$ is a collection of synchronized local transitions. As synchronized local transitions are interrelated, they will all belong to the same cluster. Additionally, transitions connected with cause-effect relations belong to the same cluster.

The clustering algorithm (Algorithm 1) captures both these types of relations between transitions. It processes global transitions from Θ in order. When the i th transition $t \in \Theta$ is processed, `RelatedClusters` computes a collection $\mathcal{R}$ of zero or more clusters which we call a *support cluster set* for t . By definition, a support cluster set is a subset of the clusters built so far with the property that applying their transitions in the same order as in Θ , starting from the initial state, results in a state where the preconditions of t hold. Furthermore, a support cluster set is set-inclusion minimal.

Notice that it is possible that several support cluster sets exist for a transition t . For example, a given precondition of t

could be supported independently by two different transitions t' and t'' belonging to two different clusters c' and c'' . We can add either c' or c'' to a support cluster set. Our algorithm builds only one support cluster set per transition t . The implementation detailed in Algorithms 2 and 3 favors considering as related to t the most recent of transitions t' and t'' .

Depending on the size of $\mathcal{R}$, one of the following outcomes can happen in `MergeClusters`: (i) If $|\mathcal{R}| = 0$, a new cluster is created and t is added to the new cluster. Since t doesn't need any of the previous transitions in the scenario to support its preconditions, t is considered independent from the previous transitions. In this case, t becomes the seed of a new cluster. (ii) When only one cluster is sufficient to support t 's precondition, t is added to that cluster ($|\mathcal{R}| = 1$). (iii) When two or more existing clusters are required to support t 's preconditions ($|\mathcal{R}| > 1$), all those clusters are merged and t is added to the resulting cluster.

Algorithm 1 `Clustering( $\Sigma, \Theta$ )`

```

 $\mathcal{C} \leftarrow \emptyset$  {initialize set of clusters}
for all  $t \in \Theta$  do
   $\mathcal{R} \leftarrow \text{RelatedClusters}(\mathcal{C}, t)$  {return zero or more clusters
    related to transition  $t$ }
   $c \leftarrow \text{MergeClusters}(\mathcal{R})$  {return one (possibly empty) cluster}
  extend  $c$  to include  $t$ 
   $\mathcal{C} \rightarrow \mathcal{C} \cup \{c\}$  {add  $c$  to set of clusters}
   $\mathcal{C} \leftarrow \mathcal{C} \setminus \mathcal{R}$  {remove previously merged clusters}
return  $\mathcal{C}$ 

```

Algorithm 2 `RelatedClusters( $\mathcal{C}, t$ )`

```

 $\mathcal{R} \leftarrow \emptyset$  {initialize set of clusters related to  $t$ }
 $\mathcal{T} \leftarrow \mathcal{C}$  {copy the contents of  $\mathcal{C}$  into temporary data structure}
for all  $c \in \mathcal{C}$  do
  if Related( $\mathcal{T}, c, t$ ) then
     $c \rightarrow \mathcal{R}$ 
  else
     $\mathcal{T} \leftarrow \mathcal{T} \setminus \{c\}$ 
return  $\mathcal{R}$ 

```

Algorithm 3 `Related( $\mathcal{C}, c, t$ )`

```

 $\mathcal{T} \leftarrow \mathcal{C} \setminus \{c\}$  {initialize temporary data structure}
 $\theta \leftarrow$  all transitions from  $\mathcal{T}$  in order
 $\sigma \leftarrow \gamma(\sigma_0, \theta)$  {apply transitions contained in  $\theta$  starting in
  initial state  $\sigma_0$  and store resulting state  $\sigma$ }
if  $t$  is applicable in  $\sigma$  then
  return false { $t$  is independent from  $c$ }
else
  return true { $t$  depends on  $c$ }

```

4.2 Root Cause Analysis

Root cause events are the unexplained spontaneous events, whose preconditions do not necessarily depend on other events. For example, in the scenario depicted in Figure 4,

the transient fault on line A-B is the root cause of the entire large cluster of events. Root cause identification is immediate from the scenario: we simply select those spontaneous events labelled as unexplained or faults.

The combination of alarm clustering and root cause analysis offers an additional option, allowing to focus root cause analysis on a subset of the alarms.

4.3 Alarms Independent of Catastrophic Events

A catastrophic fault can create a cascade of many follow-up alarms. Such follow-up alarms get mixed in the alarm log with alarms that are independent from the catastrophic event. As a result, incident-independent alarms could get lost unless they can be automatically filtered from the alarm log.

Alarm clustering is well suited to provide such functionality. Once a scenario is partitioned into clusters, the set of incident-independent alarms can be computed as the set of alarms from the clusters that do not contain the incident event.

4.4 Live Alarms

Our fourth alarm filtering technique is highlighting what we call “live” alarms. Intuitively, live events are part of a process that has started but not yet converged back to the normal (nominal) state of the involved network components. For example, the alarm “Line C-D KV LIMIT HIGH” in Figure 1 is a live alarm, since it indicates that the component is not in its nominal state. However, subsequence “Line A-B KV LIMIT LOW” followed by “Line A-B KV LIMIT NORMAL” is not live, since the component is at the end back in its nominal state.

Model based reasoning followed by scenario clustering offers a good basis for highlighting live alarms. Selecting only the alarms from clusters whose current last state is not a nominal state will provide the desired functionality.

5 Experiments

TransGrid, the operator of the transmission network in New South Wales, provided us with the topology of their network (i.e., the list of components and connections between components), and a log containing 2246 entries (alarms and commands). The log covers roughly fifteen hours: the first two thirds are routine operation, then a major fault situation arises and the rest of the log chronicles the operators’ efforts to re-configure the network to restore service.

Based on this data we built a model of the TransGrid network, including a discrete state/transition model of each component type, and the network topology which determines event synchronisations between components, as discussed in Section 3.1. We focused on a subset of alarm types, mainly those related to the primary electrical system (e.g., switch gear state changes, high/low voltages, etc). Restricted to these types, the total number of observations (alarms and commands) is only 731.

For the purpose of experimentation, we split the alarm log into (a) fixed size (one minute) time windows, and (b) variable sized windows separated by periods of at least one minutes silence (i.e., where no alarm is generated). This gave us

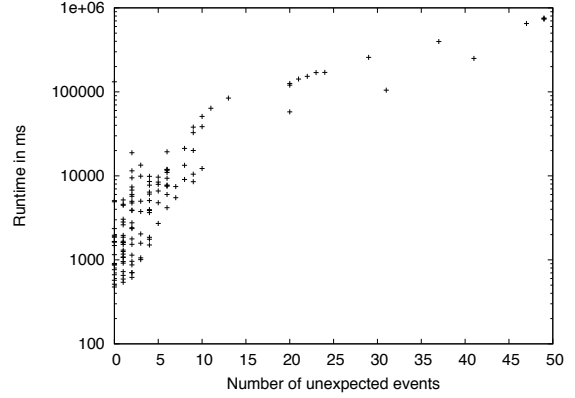


Figure 5: Runtime wrt the number of unexplained events.

129 problem instances. For each of those we extract a corresponding model, which contains only the parts of the network that are potentially relevant to reasoning about this set of alarms. This subnetwork is computed iteratively, starting with the components that emitted the alarms in the instance alarm log. Components that may influence (according to the model) components already in the subnetwork are added, until a fixpoint is reached.

For each problem instance, we ran the diagnoser to generate a preferred scenario, and then used the methods presented in the previous section to generate clusters of alarms. We set a limit of one minute for the entire process (scenario generation and filtering). This is somewhat generous, as operators would probably prefer an answer within a few seconds, but our implementation being only a prototype, we believe it can be easily improved by a factor of 10 or more.

Out of 129, 16 instances were not solved within the time limit. The most difficult of these instances involves 105 components and 146 alarms, and the SAT problem was so large (in terms of boolean variables and clauses in the SAT formula) that it could not even be read by the SAT solver; the simplest one involves 36 components and 16 alarms. However, what really makes the problem hard is the number of unexplained events. Remember that the preferred scenario is defined as the path in the model that matches the observations and minimizes the number k of unexplained events. Therefore, the problem is much simpler when k is small. This is illustrated Figure 5. The model we used in the experiments is quite simple, and in many cases is not able to link a single unexplained event to as many alarms as it should. A more precise model would decrease k , and hence improve both the runtimes and the quality of scenarios.

There are no metrics to determine how good a filtering is: on the one hand, having few clusters shows that many alarms are related, while on the other, having many clusters shows that unrelated events have been successfully taken apart. Nevertheless, the usefulness of the filtering method is obvious in several cases. For the example log in Figure 1, for exam-

ple, a large cluster of related alarms (the four circuit breakers opening to isolate a line and later reclosing), and a single root cause for this cluster, is identified, and the second of the remaining two of unrelated alarms (showing voltage fluctuating in a different part of the network) is highlighted as live. An unrelated event like this can be quite important, while being very easy to miss in the flood of alarms. Our alarm processor can draw the operator's attention to it.

6 Conclusions

Intelligent alarm processing tools are important for control room operators who supervise and manage large-scale systems, such as power networks. Scalable and accurate alarm processing capabilities are becoming even more critical as increasing instrumentation of networks generates a dramatically larger volume of alarm data.

Earlier approaches to alarm processing have been predominantly syntactic, based on text-level filtering or shallow system models. We have presented an approach that uses a comprehensive model of a network, including functional models of all components as well as the global behavior that emerges from intercomponent interactions. Online model-based diagnosis ensures that important information, such as the current state of the network, is taken into account when applying alarm processing algorithms. We demonstrated our ideas on a model representing the network of a leading Australian transmission company.

Future work includes developing a more refined and detailed network model, and analysing several scenarios at a time, for a more comprehensive diagnosis. We are also interested in performing alarm processing incrementally, updating the output as new alarms arrive.

Acknowledgments

We thank TransGrid for permission to use their data.

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

References

- [Anbulagan and Grastien, 2009] Anbulagan and A. Grastien. Importance of variables semantic in CNF encoding of cardinality constraints. In *Symposium on Abstraction, Reformulation and Approximation*, 2009.
- [Cassandras and Lafortune, 1999] C. Cassandras and S. Lafortune. *Introduction to discrete event systems*. Kluwer Academic Publishers, 1999.
- [Cordier et al., 1998] M.-O. Cordier, J.-P. Krivine, P. Laborie, and S. Thiebaut. Alarm processing and reconfiguration in power distribution systems. In *Tasks and Methods in Applied Artificial Intelligence*, volume 1416, pages 230–241. Springer, 1998.
- [Dahlgren et al., 1998] R. Dahlgren, G. Rosenwald, C. C. Liu, S. Muchlinski, A. Eide, and D. Sobajic. Model-based synthesis and suppression of transformer alarms in a control center environment. *Power Delivery, IEEE Transactions on*, 13(3):843–848, 1998.
- [Grastien et al., 2007] A. Grastien, Anbulagan, J. Rintanen, and E. Kelareva. Diagnosis of discrete-event systems using satisfiability algorithms. In *22nd Conference on Artificial Intelligence (AAAI-07)*, 2007.
- [Guo et al., 2010] W. Guo, F. Wen, Z. Liao, L. Wei, and J. Xin. An analytic model-based approach for power system alarm processing employing temporal constraint network. *Power Delivery, IEEE Transactions on*, 25(4):2435–2447, 2010.
- [Ipakchi and Albuyeh, 2009] A. Ipakchi and F. Albuyeh. Grid of the future. *IEEE Power and Energy Magazine*, 7:52–62, 2009.
- [Kirschen and Wollenberg, 1992] D. S. Kirschen and B. F. Wollenberg. Intelligent alarm processing in power systems. *Proceedings of the IEEE*, 80(5):663–672, 1992.
- [Laborie and Krivine, 1997a] P. Laborie and J.-P. Krivine. Automatic generation of chronicles and its application to alarm processing in power distribution systems. In *International Workshop on Principles of Diagnosis (DX'97)*, pages 61–68, 1997.
- [Laborie and Krivine, 1997b] P. Laborie and J.-P. Krivine. Gemo: A model-based approach for an alarm processing function in power distribution networks. In *International Conference on Intelligent System Application to Power Systems (ISAP'97)*, pages 135–141, 1997.
- [Larsson and DeBor, 2007] J. E. Larsson and J. DeBor. Real-time root cause analysis for complex technical systems. In *IEEE 8th Human Factors and Power Plants and HPRCT 13th Annual Meeting*, pages 156–163, 2007.
- [Larsson, 2002] J. E. Larsson. Diagnostic reasoning based on means-end models: experiences and future prospects. *Knowledge-Based Systems*, 15(1-2):103–110, 2002.
- [Lin et al., 1998] Y.-K. Lin, M.-S. Tsai, Y.-Y. Hong, and Y.-C. Tsuei. Application of model based intelligent alarm processor for the power supply system of the mass rapid transit system of taipei. pages 387–392, 1998.
- [Lin et al., 2004] W.-M. Lin, C.-H. Lin, and Z.-C. Sun. Adaptive multiple fault detection and alarm processing for loop system with probabilistic network. *Power Delivery, IEEE Transactions on*, 19(1):64–69, 2004.
- [McDonald et al., 1991] J. R. McDonald, G.M. Burt, and D. J. Young. Alarm processing and fault diagnosis using knowledge based systems for transmission and distribution network control. *Transmission and Distribution Conference, Proceedings of the IEEE Power Engineering Society*, pages 280–286, 1991.
- [Meza et al., 2001] E. M. Meza, J. C. S. de Souza, M. T. Schilling, and M. B. Do Coutto Filho. Exploring fuzzy relations for alarm processing and fault location in electrical power systems. In *IEEE Power Tech Proceedings*, 2001.
- [Moskewicz et al., 2001] M. W. Moskewicz, C. F. Madigan, Y. Zhao, L. Zhang, and S. Malik. Chaff: engineering an efficient SAT solver. In *Proceedings of the 38th ACM/IEEE*

Design Automation Conference (DAC'01), pages 530–535. ACM Press, 2001.

- [Protopapas *et al.*, 1991] C. A. Protopapas, K. P. Psaltiras, and A. V. Machias. An expert system for substation fault diagnosis and alarm processing. *Power Delivery, IEEE Transactions on*, 6(2):648–655, 1991.
- [Taisne, 2006] J. Taisne. Intelligent alarm processor based on chronicle recognition for transmission and distribution system. *Power Systems Conference and Exposition PSCE '06. IEEE PES*, pages 1606–1611, 2006.
- [Wen and Chang, 1997] E. S. Wen and C. S. Chang. Tabu search approach to alarm processing in power systems. *Generation, Transmission and Distribution, IEE Proceedings*, 144(1):31–38, 1997.
- [Wen *et al.*, 1995] F. Wen, C. S. Chang, and D. Srinivasan. Alarm processing in power systems using a genetic algorithm. *Evolutionary Computation, IEEE International Conference on*, 1995.
- [Zanella and Lamperti, 2003] M. Zanella and G. Lamperti. *Diagnosis of active systems*. Kluwer Academic Publishers, 2003.

Cities that offer a customized and personalized tourist experience to each and every visitor: the Smartourism project

Francisco Palao and Luis Castillo

IActive Intelligent Technologies
Granada, SPAIN

Juan Fdez-Olivares

Dept. of Computer Science and A.I
University of Granada

Oscar García

IActive Intelligent Technologies
Granada, SPAIN

Abstract

This paper presents an approach of HTN planning technology devoted to design personalized visits to a city depending on the tourist attractions of the city, the preferences of the visitor, his/her budget, available time and even his/her disabilities. Therefore, the city intelligently adapts to every potential visitor and offer them its best choice becoming a customized tourist experience because it has been exclusively designed for them.

1 Introduction

There is no doubt, the world is changing. People are now placed at the center of the technology more than ever before, providing new customized services adapted to his/her needs and preferences and able to share it with his/her social environment. Most of this technology is also becoming smarter, with expert knowledge playing a central role. Knowledge becomes one more tool, it is instrumented, represented and processed by programs, giving Artificial Intelligence and Knowledge Representation technologies a clear opportunity to bridge the gap to the practical side of the world. These knowledge-based programs would be able to offer added value capabilities similar to those offered by human experts. This does not mean that they are intended to substitute for humans, but to complement and empower them, as Keith Swenson says [Swenson, 2010]: *"We support the worker, but do not replace their decision-making ability"*.

This paper presents an approach based on these assumptions focused on tourism activities within a given city. The core of the approach is an HTN planning engine that has been used before to plan for civil emergencies, e-learning problems [Castillo *et al.*, 2010] or to design oncology treatment plans [Fdez-Olivares *et al.*, 2011] among others. When someone talks about real world applications of planning and scheduling one leans to think of complex artifacts plenty of complex knowledge representations, with a hardwork employed to interconnect the planning engine with some legacy systems. And in many cases it is true.

However it is not the case of the application shown in this paper. This paper features a very simple, but powerful, planning application devoted to the design of personalized visits

to a city depending on the tourist offer of the city, the preferences of the visitor, his/her budget, available time and even his/her disabilities. It is Smartourism¹, a web application that is being used in the tourism portal of the city of Granada in Andalusia (Spain) since october 2010, generating an average of about one thousand of plans per week, and that is currently being implemented in a number of other Spanish cities.

In the era of the internet of things, everything is interconnected, producing an overwhelming flow of information. Tourism is one of the areas experimenting this explosion of information making it more feasible, but also more difficult, to design a plan to visit a city. The web contains large amounts of information that allow us to plan the flight, book hotel rooms, discover opening and closing hours of buildings, monuments, show their main features, stories and even the price of tickets. All this is possible. But it also makes the search slower and complex for potential visitors that have to gather information, select the most appealing visits, arrange the targets, prioritize them, compose a plan, check the budget, look for pathways to move from one place to another and consider the time taken and the opening hours, etc. Smartourism does the search for the visitor, select the most appropriate activities based on visitor's profile, compose an activity plan, and sequence it along a timeline depending on the schedule of every selected visit, design door to door instructions in the case of moving by foot, in a matter of seconds.

This process is carried out for every visitor, so the outcome is a customized experience for everyone. Smartourism is accessed from public civil services web sites, as of cities, or clusters of cities, and gather statistics about the tourist demands that can be used to gain a deeper knowledge about visitors and their preferences and to improve the quality and use of the tourist services offered. In summary, Smartourism, a relatively simple artificial intelligence planning application, may be used to improve the tourist offer and associated resources of a city, give a higher quality welcome to visitors in less time with a lower cost. All the ingredients for a city to become smarter.

2 The problem

Generally speaking, the problem is the selection and arrangement of a sequence of visits to different tourist attractions of

¹ www.smartourism.com

a city. This visit plan has to be designed taking into account the following variables:

1. The resources included in the plan must match the preferences of the visitor, specified in terms of the artistic style, type of activity among others.
2. The opening hours of the resources must coincide with the available time of the visitor. Movements across the city must be planned to fit within these opening hours.
3. The visitor must move from one point to another and must be provided with instructions on how to move, either by taxi or by foot, including door to door instructions (a multimodal transport plan is not available to date in Smartourism).
4. The amount of money spent in the visit must be within the limits allowed by the visitor.
5. Every resource, and in fact the whole plan, must be prepared for mobility disabilities if the visitor declares this impairment.
6. If required, the plan must include a selection of restaurants to have lunch in the nearby during the visit, according to visitor preferences and budget.
7. The visitor must be able to modify the suggested visit plan to adapt it on the fly.

As a result of the problem, Smartourism must suggest, basically, what to do, where to go and at which time, during the whole visit.

3 The technology behind the scenes

From a technical point of view, it is basically a temporal planning problem with some scheduling flavor, that is, how to arrange an undefined sequence of actions along a timeline subject to some temporal constraints given by movements from one point to another and opening hours. The approach presented in this paper builds on previous results obtained in a research project named SAMAP [Castillo *et al.*, 2008] part of which has been successfully transferred for commercial exploitation to IActive Intelligent Technologies, a spin-off of the University of Granada. In particular it refers to the Hierarchical Task Network (HTN) planning approach [Castillo *et al.*, 2006] to this problem that has been redesigned and renamed as Smart Process Management technology (SPM). It allows for the design of processes built on top of a Simple Temporal Network [Dechter *et al.*, 1991] as a flexible temporal plan. These processes must be understood as sequences of actions (in sequence or in parallel) that achieve a certain goal in a given context based on an initial state of the context and the represented expert knowledge that determines how one can act in this context. These are the components of Smart Process Management technology, borrowed from HTN planning:

- Context Model of the environment in which the plan is to be executed, the types of objects, their properties and their relationships. In terms of PDDL, the Planning Domain Description Language [Edelkamp and Hoffmann,

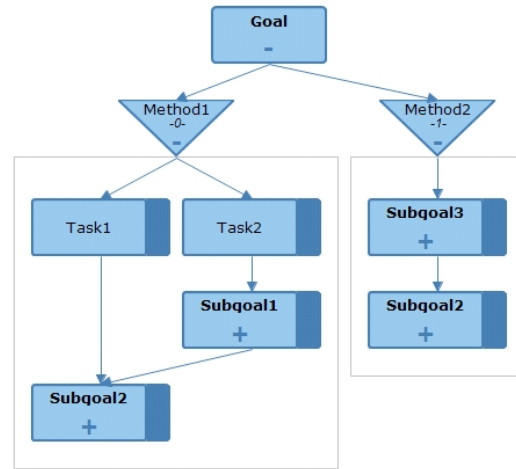


Figure 1: Four basic graphical constructs of Expert Knowledge: compound *goals* that may be decomposed into one or more subgoals by following decomposition *methods*. A goal may be decomposed by following multiple, alternative methods and one or more subgoals may be *tasks* which are non-decomposable, primitive activities. All these elements are related by means of *arrows* to represent precedence constraints.

2004] this is the hierarchy of types of the planning domain and it is represented in UML 2.0 to ease a wider adoption.

- Initial state. Initial state of the objects in the context, that is a dump of the specific values of the properties and relationships of any object in the problem. It is equivalent to the `:INIT` section of PDDL and it may be extracted from corporate databases as an XML file.
- Expert Knowledge (EK). Hierarchical representation of the actions that may be done in the context, in terms of compound actions (named goals) and their constituent subactions (named tasks). EK uses a very simple graphical language with only four constructs (Figure 1). Among other aspects, the EK represents preconditions that must be met for each action and their effects on the properties of the objects it affects to, temporal constraints, metadata associated to actions, etc. In summary, this EK is the HTN domain itself, but following a graphical interpretation [Palao *et al.*, 2011] of the textual language proposed in [Castillo *et al.*, 2006].
- Goal. Description of high level goals to be carried out to solve a problem. They may be specified either explicitly by an expert or implicitly (e.g., "solve all problems encountered in the context" or "visit as many monuments as possible today").
- Process. It is a timed action plan (with an estimated time of start and end for every action) that determines the sequence of activities to be carried out, extracted among those available on the EK, that can be applied in the initial state with regards to achieving a goal.

These components are modeled, represented and validated in an integrated environment whose two main components

are:

- Decisor. It is the intelligent component that is responsible for designing processes to meet the objectives, to arrange the means described in the expert knowledge within the circumstances specified in the initial state of the context. Therefore, Decisor is the temporal HTN planner.
- IActive Knowledge Studio. It is the environment for the modeling and validation of all the knowledge related to the problem:
 - The contextual knowledge(Figure 2.a)).
 - The expert knowledge (Figure 2.b)).
 - The initial state of the context, in the form of a relational database, showing the values of the attributes of every instance of any object in the CM present in the context(Figure 2.c)).
 - The sources of data where the initial state is extracted from.
 - The goal of the problem specified as an ordered sequence of high level tasks from the expert knowledge to be decomposed.
 - The resulting processes. It provides several tools for the visualization and comparison of processes, extraction of statistics, with regards to validate the processes obtained (Figure 2.d)).

The lifecycle of exploitation of Smart Process Management is shown in Figure 3 and it might well be shared with any other planning application. The following are the most important steps in this lifecycle:

1. The consulting team of Smartourism with the help of the of Partner Support Department of IActive, extracts the end-user knowledge of the application and model it within IActive Knowledge Studio. In this case end users are tourism experts of a city and the knowledge is fully related to tourism activities²:
 - (a) The structure of the problem (Context Model).
 - (b) The problem-solving skills and previous experience in designing tourist plans (EK).
2. Thanks to IActive Knowledge Studio, which integrates the HTN planner (Decisor), solution plans may be automatically generated and displayed.
3. These solution processes are validated in an iterative refinement lifecycle by end users and the consulting team. These three steps (knowledge extraction, representation and validation) configure a typical knowledge engineering effort that might well be suited for any planning approach and may be fully carried out with IActive Knowledge Studio. Once the knowledge of the problem has been correctly modeled and validated, it is necessary to integrate the planner together with its knowledge representation and its connections to databases with larger

²This is not a process from scratch. Indeed it is an adaptation and extension of a basic knowledge representation common to all instances of Smartourism

systems, in particular it is integrated with the whole Smartourism system.

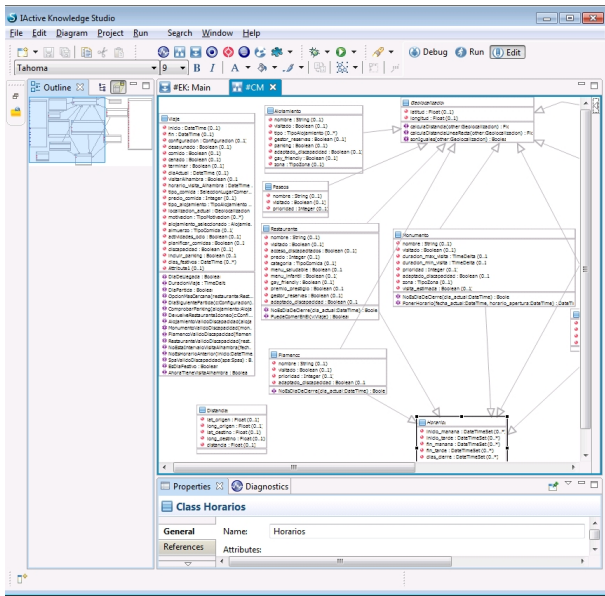
4. The integration of the the planner is done through a deployment operation from Knowledge Studio. A deployment is the generation of a package that includes the intelligent planner (Decisor), the Context Model and the Expert Knowledge together with a programming interface. We name this package a *Smart Module*, since it may be integrated with almost any larger IT system to enrich it with HTN planning capabilities. A deployment may be featured either as a JAR package (Java / Eclipse) or as a Web service depending on the application to be integrated. The communication between the application and the Smart Module may be done either by passing XML files or by including data gateways to corporate databases without going through XML files.
5. End users enjoy the added value of Smartourism by allowing them to perform tasks that were unable to perform before, that is to say, obtaining a customized visit to a city. It is difficult to demonstrate whether this is the best plan or not because this is very subjective to measure and it depends on many variables (adequacy to user preferences, time tightness, number of trips, etc). In this case, the search is driven by a greedy process that uses a ranking function which aggregates these variables and also others corresponding to policies of fostering of tourism at every city. Figure 5 shows a screen intended to be used by authorized administrative staff of cities where every row corresponds to a category of tourist attractions and columns corresponds to different criteria, in this case, the four different seasons of the year. Each cell contains a slider to represent the relative weight of every attraction during every season, as an heuristic evaluation that takes part in the whole ranking function used by Smartourism.

4 Smartourism

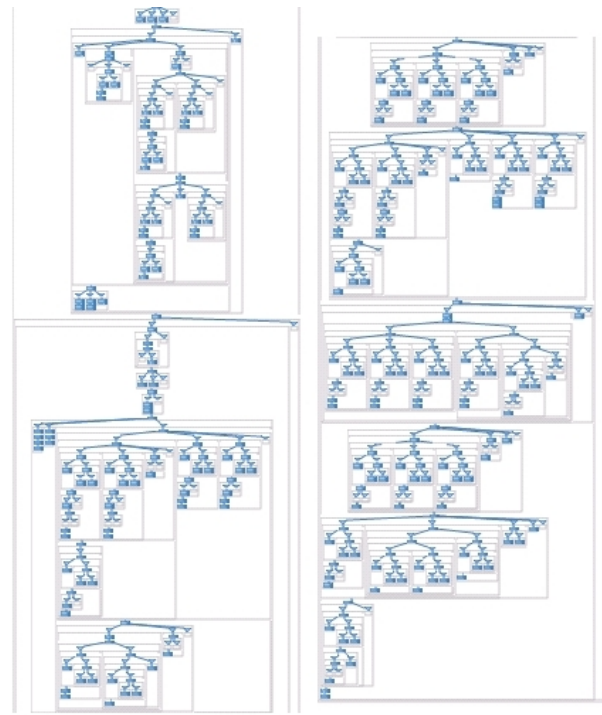
Smartourism is a web application and its architecture is sketched in Figure 4 showing a typical layered structure of web applications. It integrates these two main modules:

- The web application, that includes the database of visitor's profiles and tourist attractions, the graphical user interface and the remaining software platform.
- The Smart Module, the package that integrates the HTN planner (Decisor). In the Smartourism project the Smart System has been deployed as a web service.

Every plan satisfy all the requirements presented in Section 2 and may be obtained in just three steps: (1) specify the date for the visit, (2) specify the location of the hotel of the visitor or select one on the fly and (3) specify the motivation of the visit (Cultural, Religious, Family visit, Leisure or the most popular visits. In order to make the Smartourism experience more satisfactory, plans are displayed in several ways. At the top of Figure 6.d) the plan is displayed as a continuous timeline, from left to right, showing the main stops, walking instructions and time. The bottom of the same figure shows the same plan integrated in Google Maps. To facilitate the



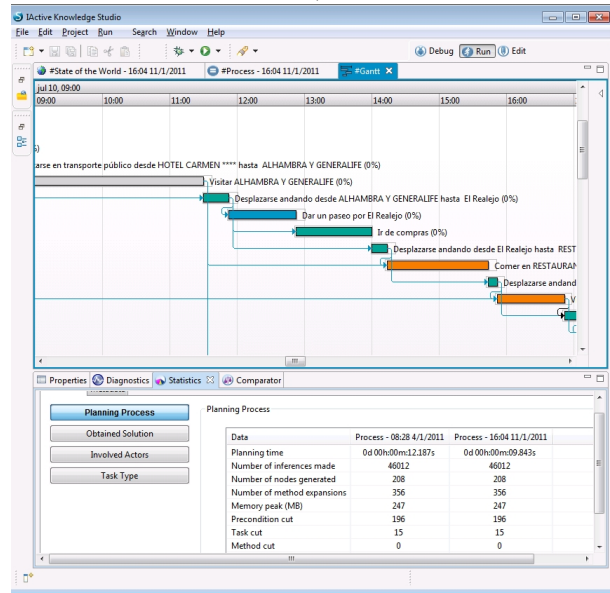
a)



b)

Instances - Monumento(26)	nombre	visitado	duracion_max_visita	duracion_min_visita	prioridad	adapt
Alajamiento(130)	CENTRO CULTURAL CAJA GRANADA "ME...	false	0d 01h00m00.000s	0d 01h00m00.000s	1	true
Configuration(1)	PARQUE DE LAS CIENCIAS	true	0d 04h00m00.000s	0d 03h00m00.000s	1	true
Viaje(0)	CASA MUSEO MANUEL DE FALLA	false	0d 01h00m00.000s	0d 01h00m00.000s	3	false
Monumento(26)	MUSEO DE SAN JUAN DE DIOS	true	0d 01h00m00.000s	0d 01h00m00.000s	4	true
Restaurante(150)	MUSEO CUERVAS DEL SACRAMONTE, CENT...	true	0d 01h00m00.000s	0d 01h00m00.000s	3	false
Paseos(5)	CENTRO JOSÉ GUERRERO	false	0d 01h00m00.000s	0d 01h00m00.000s	4	true
Ocio(5)	FUNDACIÓN RODRÍGUEZ-ACOSTA	false	0d 01h00m00.000s	0d 01h00m00.000s	4	false
Spas(9)	CASA-MUSEO, FEDERICO GARCÍA LORCA, ...	false	0d 01h00m00.000s	0d 01h00m00.000s	3	false
Flamenco(7)	MUSEO ANGEL BARRIOS	true	0d 01h00m00.000s	0d 01h00m00.000s	4	false
RutasTematicas(4)	MUSEO MAX MOREAU, (CARMEN)	false	0d 01h00m00.000s	0d 01h00m00.000s	4	false
Distancias(0)	CARMEN DE LOS MÁRTIRES	true	0d 01h00m00.000s	0d 01h00m00.000s	4	false
	CASA DE LOS TIROS, MUSEO DE ARTES Y C...	false	0d 01h00m00.000s	0d 01h00m00.000s	3	false
	CORRAL DEL CARBÓN	true	0d 00h10m00.000s	0d 00h10m00.000s	4	false
	JARDINES PARQUE FEDERICO GARCÍA LOR...	false	0d 01h00m00.000s	0d 01h00m00.000s	3	false
	JARDINES CUARTO REAL DE SANTO DOMI...	true	0d 01h00m00.000s	0d 01h00m00.000s	3	false
	ALIBRE DEL REY	false	0d 01h00m00.000s	0d 01h00m00.000s	4	false
	CATEDRAL	true	0d 01h00m00.000s	0d 01h00m00.000s	2	false
	CAPILLA REAL	true	0d 01h00m00.000s	0d 01h00m00.000s	2	false
	MONASTERIO DE LA CARTUJA	true	0d 01h00m00.000s	0d 01h00m00.000s	2	false
	MONASTERIO E IGLESIA DE SAN JERÓNIMO	true	0d 01h00m00.000s	0d 01h00m00.000s	2	false
	ABADÍA DEL SACRAMONTE	false	0d 01h00m00.000s	0d 01h00m00.000s	3	false
	BASILICA DE SAN JUAN DE DIOS	false	0d 01h00m00.000s	0d 01h00m00.000s	4	true
	BASILICA NUESTRA SEÑORA DE LAS ANG...	true	0d 01h00m00.000s	0d 01h00m00.000s	4	false
	IGLESIA DEL SALVADOR	false	0d 01h00m00.000s	0d 01h00m00.000s	4	false
	CUARTO REAL DE SANTO DOMINGO	true	0d 01h00m00.000s	0d 01h00m00.000s	4	false
	ALHAMBRA Y GENERALIFE	true	0d 05h00m00.000s	0d 01h00m00.000s	1	true

c)



d)

Figure 2: a) View of IActive Knowledge Studio showing the main classes and their relationships in UML 2.0 as they are represented in the Context Model. b) Birds-eye view of the EK of Smartourism, which contains 141 entities out of goals and tasks and 150 decomposition methods. c) View of the Initial State as a local relational database. d) View of IActive Knowledge Studio showing a sample plan as a Gantt Chart and some statistics and key performance indicators about it

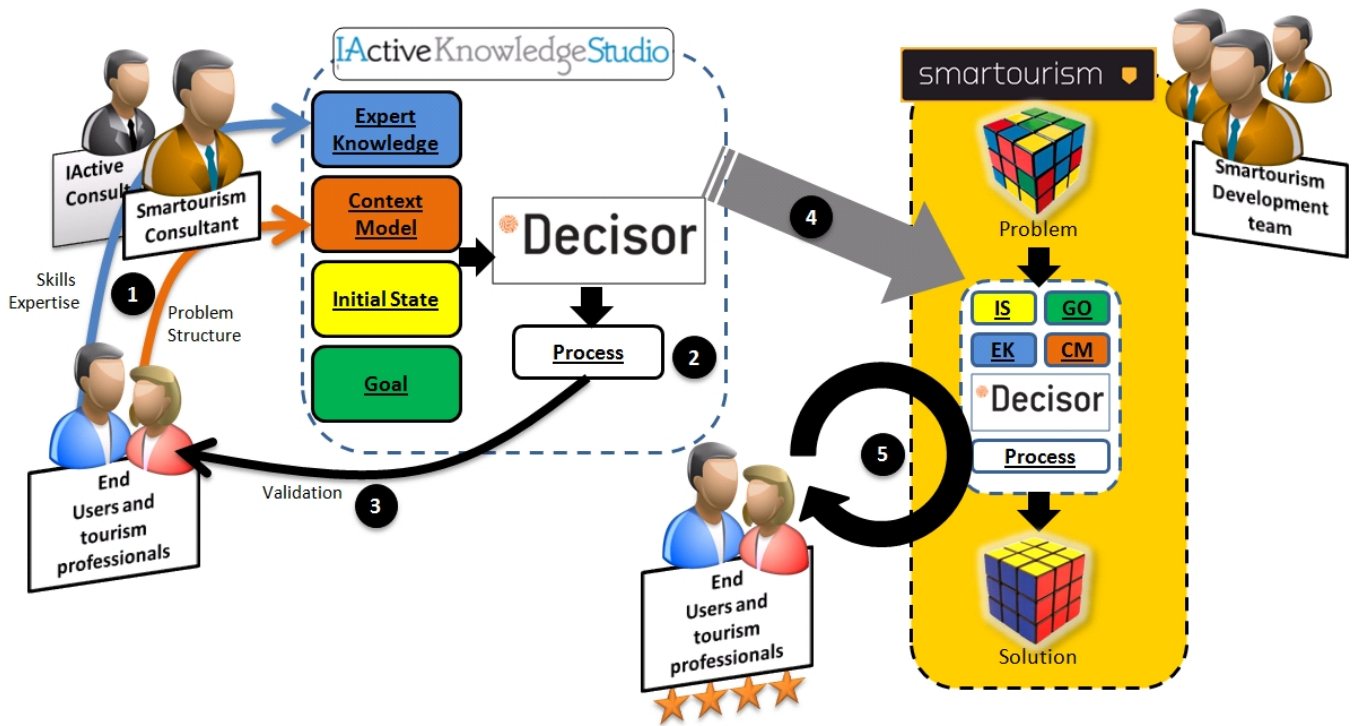


Figure 3: Life cycle of the application of Smart Process Management technology: (1) Knowledge extraction from end users and representation in Knowledge Studio, (2) Plan generation, (3) Domain and Plan validation, (4) Deployment and integration into real applications and (5) Exploitation and commercial use

adaptation to user's preferences, Smartourism allow to edit the plan, reject some activities or request new ones in a interactive refinement process. Therefore, a new alternative plans is designed after each iteration with the end user, until a final plan is obtained that satisfies him/her. This plan may be downloaded or shared with other visitors.

5 An example

Figure 6 illustrates a step by step execution of Smartourism:

Figure 6.a) Select the start date and duration of the visit.

Figure 6.b) Select user's hotel or address. All the dairy plans start and end here. If the user needs an accommodation, he/she may select it from here. There is an option for people with mobility disabilities. When selected, only accommodations prepared for handicapped persons are shown. This option also implies that only attractions prepared for this condition will be selected.

Figure 6.c) Select one or more motivations for the visit: cultural, religious, family, leisure or most popular. Al the above information is easily stored in the Context Model as properties of different types of several classes (visitors and tourist attractions), and they are equivalent to different predicates and fluents in PDDL.

Figure 6.d) Display the plan day to day, with geographical information about the movements and selected attractions and restaurants. From this display, the user may

accept, reject, advance or delay any of the selected attractions and request a new plan accordingly.

The interface has been kept as simple as possible to promote its ease of use and its translation into a specific application for smartphones (to appear).

6 Some results and conclusions

As shown throughout the paper, Smartourism is a simple web application built on top of a temporal planner with a relatively simple knowledge representation of 13 classes in the UML representation of the context knowledge, and a domain made up of about 140 compound goals and primitive tasks. Our experience with the application of Smartourism in the city of Granada allow us to state the following. Every plan obtained by Smartourism in seconds is equivalent of around 15 minutes of work of tourist offices staff to help a visitor to design a plan. This might raise up to 20 minutes if the intention is to personalize the plan. Therefore Smartourism may help tourism support staff to give a faster response to visitors (therefore a higher number of visitors may be attended, with a lower cost due to the reduction of the time needed per visitor), to foster a more efficient use of tourist attractions and their associated staff in a city. In summary it allows to do more with less: give a higher quality service in less time and cost. Here are some of the benefits that allow Smartourism to be aligned with mainstream Smart Cities initiatives

- For tourists

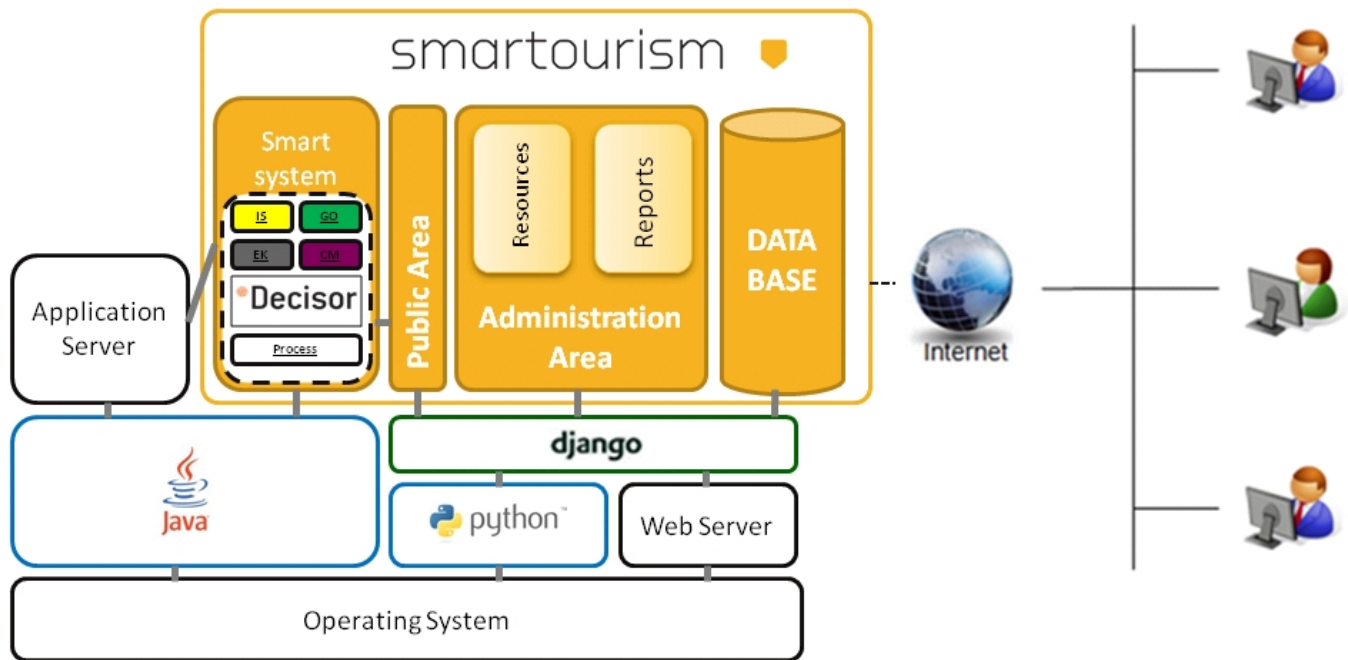


Figure 4: Architecture and modules of the implementation of Smartourism

- Easier design of the visit to a city. In just a few clicks all the information is gathered, matched against visitor's preferences and arranged according the timetable of visits.
- All the visits are customized to every visitor, providing a higher subjective sense of quality. Only possibly interesting places are included in the visit.
- Better use of the available time of the tourist by considering timetables, time taken to move from one point to another and the available time.
- Continuous 24x7 support since visitors can connect from any place at any time.
- Support for people with mobility impairments, since Smartourism only selects those visit adapted for them.
- For city councils and other public organizations in charge of tourism
 - Expected increase on the tourist interest.
 - Better feedback knowledge of the visitors of the city.
 - Better service given to visitors.
 - Improve the management of the tourist attractions, targeted by the selection and opinion of visitors.
 - It allows to configure ad-hoc tourist policies by fostering or promoting certain tourist areas.

In the current state of deployment of Smartourism it is rather difficult to state its influence on real visits. Figure 7 shows the number of plans dispatched in the city of Granada during the last five months. It is not possible to say which, among these plans, are real plans being executed by tourists,

how many tourist have been attracted to the city thanks to Smartourism or their feedback about the quality of plans. However the technical team is working on this by allowing sharing plans through social networks, mainly short links in Twitter and also by sharing and ranking the plan in Facebook. This indirect information would provide feedback on user satisfaction and a lower bound of real visits out of the plans being dispatched. These measures are currently being implemented but they need some time to be effectively measurable and to be reported.

And last, but no least, Smartourism, as an Artificial Intelligence planning application proves that there is life beyond blockworld and logistic puzzles in planning research. It is a very simple planning domain, where planning and replanning are easy, time representation and reasoning are affordable and with a high social and economic impact. And it is not the only one. There are other small, simple problems very sensitive to be approached by AI planning with relatively little effort.

References

- [Castillo *et al.*, 2006] L. Castillo, J. Fdez-Olivares, O. García-Pérez, and F. Palao. Efficiently handling temporal knowledge in an HTN planner. In *Sixteenth International Conference on Automated Planning and Scheduling, ICAPS*, 2006.
- [Castillo *et al.*, 2008] Luis Castillo, Eva Armengol, Eva Onaíndia, Laura Sebastián, Jesús González-Boticario, Antonio Rodríguez, Susana Fernández, Juan D. Arias, and Daniel Borrajo. SAMAP. An user-oriented adaptive system for planning tourist visits. *Expert Systems With Applications*, 35(2), 2008.

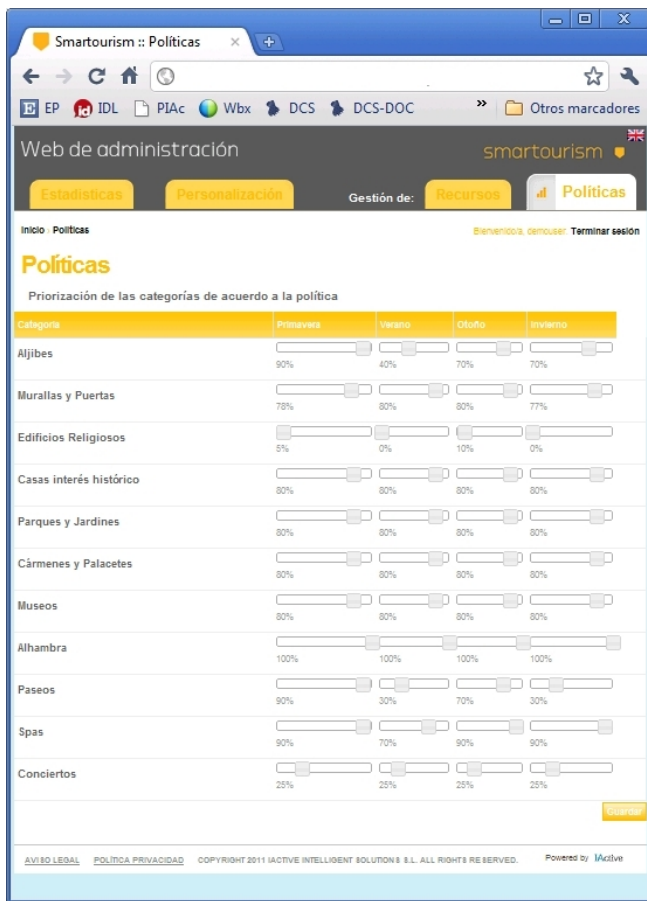


Figure 5: Matrix of policy weights in Smartourism. The administrative staff of a city intuitively encourages or discourages the use of certain attractions (rows) during different seasons of the year (columns)

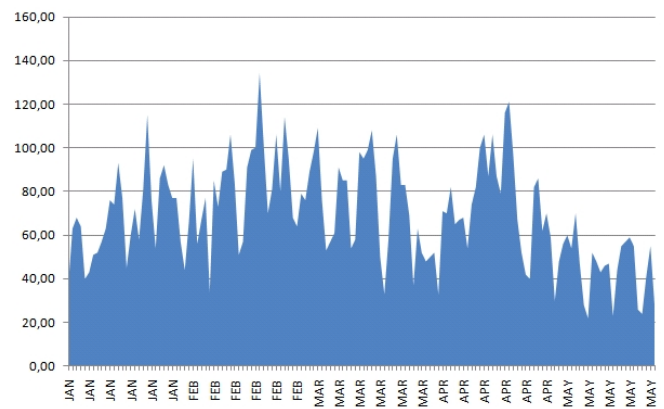


Figure 7: Number of dispatched plans every day in Granada from January 2011 to late May 2011

edge Engineering for Planning and Scheduling, ICAPS 2011, 2011.

[Swenson, 2010] Keith Swenson. *Mastering the Unpredictable: How Adaptive Case Management Will Revolutionize the Way That Knowledge Workers Get Things Done*. Meghan-Kiffer Press, 2010.

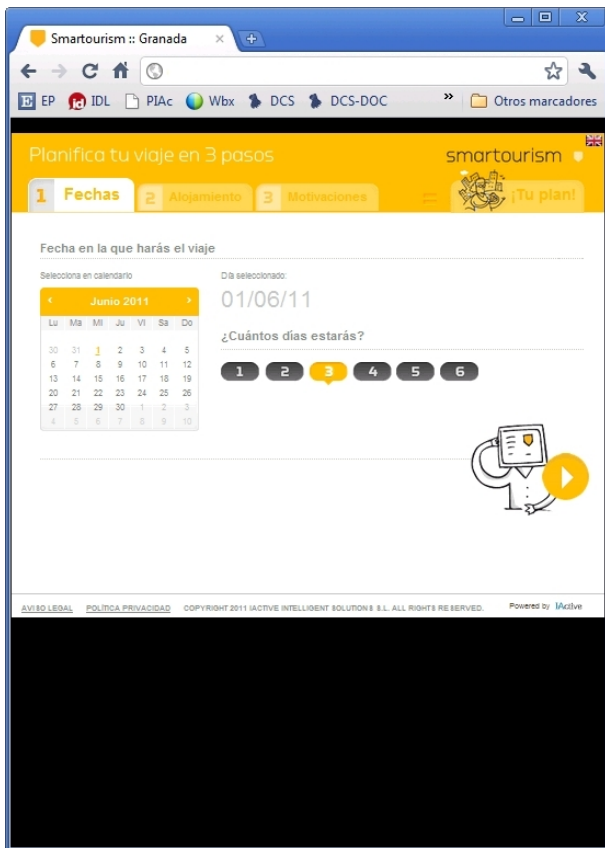
[Castillo *et al.*, 2010] Luis Castillo, Lluvia Morales, Arturo González-Ferrer, Juan Fdez-Olivares, Daniel Borrajo, and Eva Onaindía. Automatic generation of temporal planning domains for e-learning problems. *Journal of Scheduling*, 13(1):347–362, 2010.

[Dechter *et al.*, 1991] R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *Artificial Intelligence*, 49:61–95, 1991.

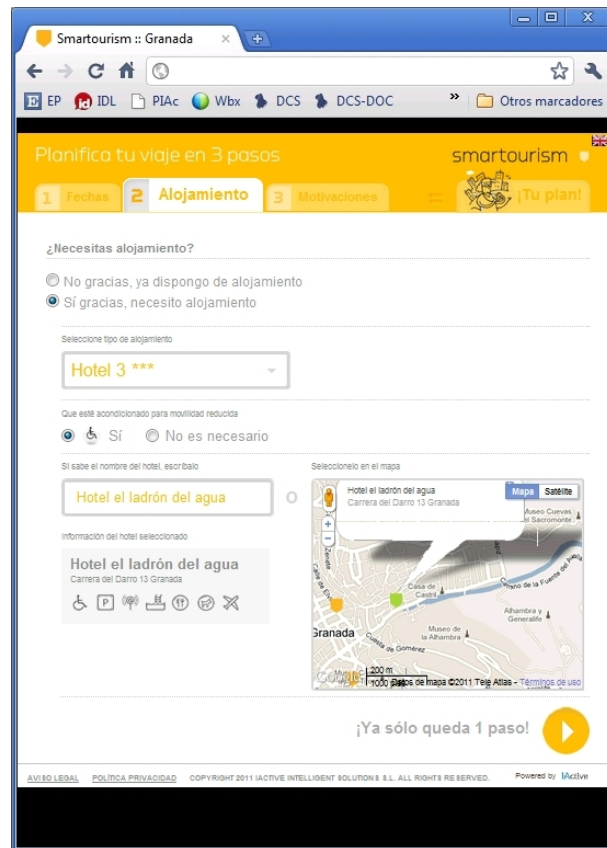
[Edelkamp and Hoffmann, 2004] S. Edelkamp and J. Hoffmann. The language for the 2004 international planning competition. <http://ls5-www.cs.uni-dortmund.de/~edelkamp/ipc-4/pddl.html>, 2004.

[Fdez-Olivares *et al.*, 2011] Juan Fdez-Olivares, Luis Castillo, J.A. Cózar, and Oscar García-Pérez. Supporting clinical processes and decisions by hierarchical planning and scheduling. *Computational Intelligence*, 27(1):103–122, 2011.

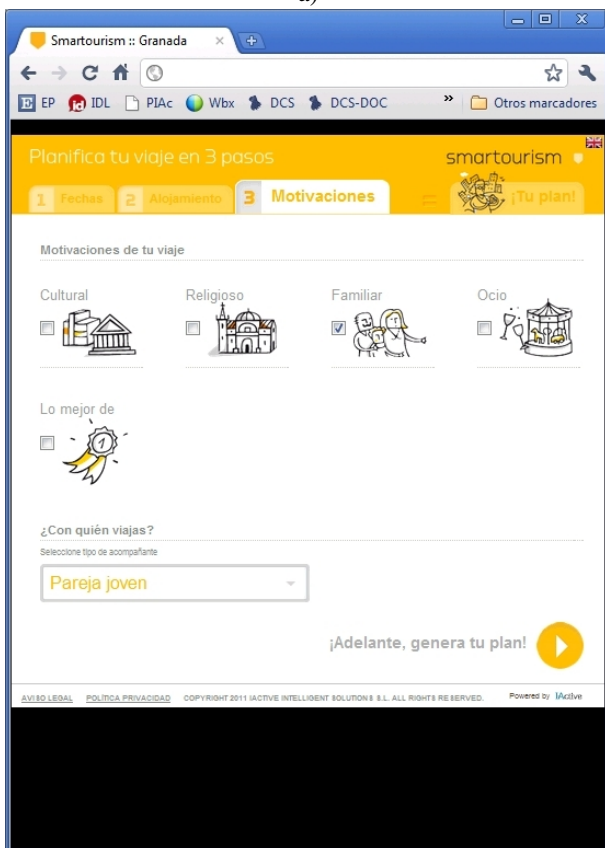
[Palao *et al.*, 2011] F. Palao, J. Fdez-Olivares, L. Castillo, and O. García. An extended HTN knowledge representation based on a graphical notation. In *Workshop Knowl-*



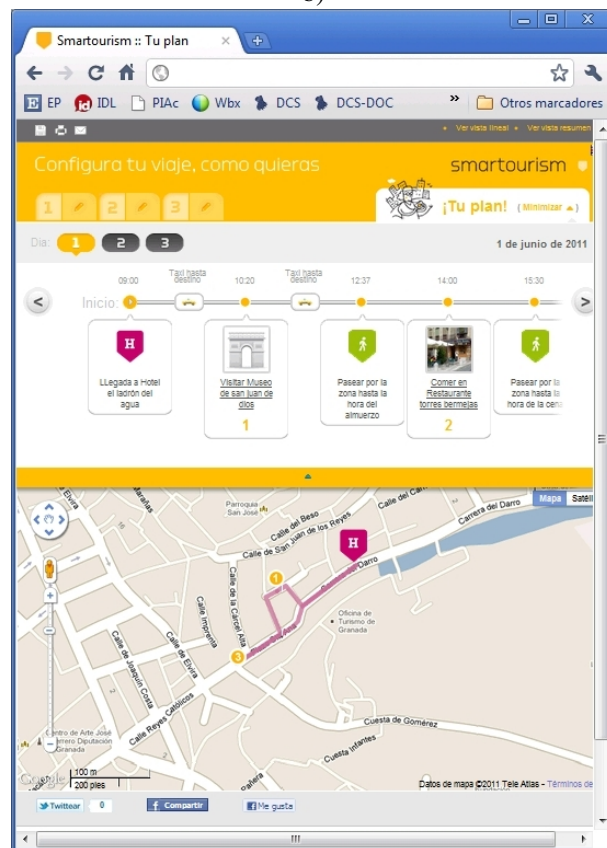
a)



b)



c)



d)

Figure 6: Smartourism storyboard

Building a Highly Consumable Semantic Model for Smarter Cities

Rosario Uceda-Sosa, Biplav Srivastava, Robert J. Schloss

IBM Research

{rosariou@us,sbiplav@in,rschloss@us}.ibm.com

Abstract

Despite enormous progress made in reconciling and simplifying semantic standards, industry-strength models are still few. In this work, we describe our challenges and approach in building semantic models for consumption in assembling large IT solutions in cyber-physical (Smarter Cities) domains. The result, called SCRIBE, is a large-scale semantic model for Smarter Cities based on data gathered from cities worldwide. We will discuss the design tradeoffs, designer/user experience, and some of the pragmatic challenges related to a model that integrates large, complex, heterogeneous data sources.

Introduction

The value of semantic modeling, whether to enable information sharing on the web (Berners-Lee et al 2001, Horrocks 2002) or to build IT systems (McIlraith et al 2001)], has long been articulated. However, the field has been mired in too many competing and often confusing standards (McCool 2005), prompting calls for reconciliation and simplification (McCool 2006). The reality is that industry-strength models are still few.

One area where the promise of semantic web is needed is complex solutions being built for sustainability efforts around the world. In essences, such efforts aim to reduce waste of natural resources like energy, water or air by building cyber-physical systems which involve sensing the data through physical instruments, interconnecting and integrating them from multiple sources and analyzing them for intelligent patterns. Such systems lead to smarter city operations in areas like transportation, energy, water and public safety. In a Smart City, the flow of information coming from sensors, citizens, administrative units, etc. needs to be integrated to generate events that will trigger city services where several city departments will need to cooperate. The volume, complexity and heterogeneity of information are sizable, and they need to be shared with a diverse stakeholder group – other governmental agencies, individuals (citizens), businesses and international organizations.

In this work, we describe our challenges and approach in building semantic models that will be consumed while assembling large IT solutions in cyber-physical (Smarter Cities) domains. A cyber-physical solution involves sensing data through physical instruments, and using Information

Technology (IT) for interconnecting, storing and integrating the sensed data from multiple sources and analyzing them for intelligent patterns.

In this scenario, we believe that small, composable ontologies, with principled inheritance and aggregation relation hierarchies, authored and annotated through a societal editorial process, offer the following benefits:

- Easy understanding and reuse of core city data, transferring learning across projects
- Increase robustness of the data model through its evolution
- Increase productivity, increase the ability to manage city services' quality and stakeholder satisfaction using data

In this paper, we will discuss our experience with SCRIBE, a semantic model and tools for Smarter Cities based on data gathered from cities from around the world, and discuss the design tradeoffs (completeness vs. manageability), designer and user experience, and some of the pragmatic challenges:

- Standards compliance, extension and integration
- Integration of multiple tools and modeling technologies (RDFS, OWL, UML)
- Desirable features for next generation ontology management tools supporting the case where different stakeholder classes who would prefer different "views" of what the ontology is contained can be provided those views easily.

Motivating Example

In any city, day-to-day operations as well as the response to unexpected events require the coordination of local agencies and their services. In practice, this coordination often spans non-local (county, province or state) agencies but we will simplify our scenario by considering the interaction of a few local agencies. Take a city with a central administrative or monitoring agency (e.g., Command Center or Mayor's office), a Public Works Department, a Traffic Department, and a Police Department.

Let us suppose that we are dealing with the emergency repair (e.g., large potholes) of a busy intersection due to inclement weather. The Public Works Department may issue, through the Central Agency, a Work Advisory to other city departments, so they can assess its impact in their operations. The Bus Department (under Traffic Department) will need to re-route two bus lines. The Police

Department may report that two police officers will need to be sent to the intersection at rush hour.

An information model for the *communication* and *coordination* of the actions and decisions of these departments needs to understand events and messages as they progress through the system, and the protocols they refer to, as these are key for the functioning of any city.

In our scenario, the threads of messages, the generation of work items (road work itself, setting warning signals, dispatching of the police cars, etc.) and the processes they refer to, are key to understand, at any given point, what is happening in the city.

Despite standards that aim to capture messages and events CAP (Oasis 2009), NIEM (NIEM 2007), UCore (UCore 2010) or the city organization MRM(MRM 2006), most city departments still deal with ad hoc, cumbersome code to map inputs and outputs from their legacy applications. In fact, these (XML) standards make either extensions or customizations cumbersome or only cover certain aspects of inter-departmental communication, not the full coordination of services needed in a Smart City.

Another key aspect of an information model for Smarter Cities is the enablement of analytics. In our scenario, other departments may need to factor the information of new traffic limitations in their predictive analytics (expected time for an ambulance to reach a location) and their optimization analytics (management of traffic patterns and subway resources) as shown in Figure 1.

For such situations, *any* analytics would need a model of routes, vehicles, time, distance and people. With the SCRIBE model, mapping of department specific data and the building and deploying of analytics is simplified

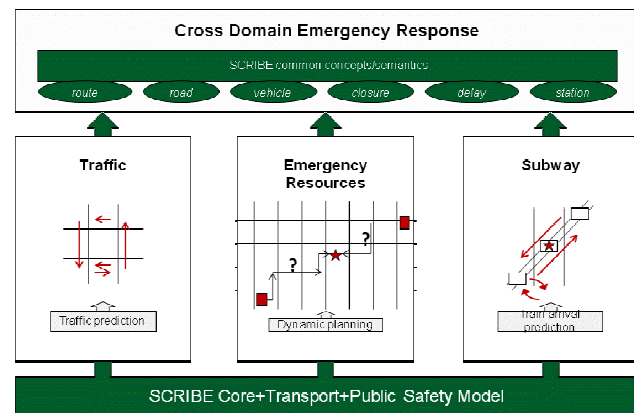


Figure 1: Example of a common analytics – routing

Faced with these challenges, the SCRIBE team aims to provide, not just a common vocabulary, but an *authoritative* information model that captures the *dynamic* aspects of city *services*. Specifically, we believe that an *architected* semantic model will promote the following in Smarter City solutions:

- Reuse of common model patterns at design time, thus reducing errors and increasing quality

- Flexible searching and browsing services so that it's usable to developers and domain experts both at design and run time
- Integration of Smarter City-related standards through well-defined data mappings
- Collaborative authoring that promotes an authoritative model

Related Work

There have been several efforts to standardize data models for inter-agency communication (see table below) with various degrees of maturity and acceptance. Of these, only MRM has been created specifically for cities. We list here their strengths and drawbacks in our scenario above.

Standard	Strengths	Drawbacks
Common Alerting Protocol (CAP)	Concise, Open, well established,	Textual message content, hard to interpret and map programmatically
National Information Exchange Model (NIEM)	Comprehensive, well established, well defined extension mechanism (IEPD)	Complex inheritance mechanism. US-Centric, not focused on cities
UCore	Simple, OWL-implementation	US-Centric, not focused on cities.
Municipal Reference Model	International (initiated by MISA)	Focused on static (administrative) aspects of a city.

The Predictive Model Markup Language (PMML) describes a simple taxonomy of analytic algorithms and their parameters to simplify model scoring. We are currently working to integrate PMML and SCRIBE in order to describe the input and output data of these analytic models in order to simplify deployment and scoring.

Finally, on the tooling and model languages side, a few technologies already exist, each with their strengths and dependencies. In a complex solution like ours, different tool-types are needed at different stages and two key design decisions are (1) to select among these at different phases in the development of the model, and (2) ensure smooth inter-operation across these tools.

Technology	Strengths	Dependencies
OWL/ RDF	Rich representation of relations	Large developer base has to be trained; stable tools are few
UML	Widely understood; rich tool set	Relies on OCL for complex relationships
Web-based deployment/ configuration tools	Easy to use by developers	Poor integration with OWL/ RDF or UML tools

SCRIBE Design Principles

In SCRIBE, we have focused on making the semantic model *usable, efficient, robust* and *authoritative*.

Usability to both the developers of Smarter Cities solutions and end users (command center or agency operators), is key for a model as complex as SCRIBE, and it has many facets:

- Using rich, flexible technology to author and query assertions, while keeping a balance with scalability and efficiency.
- Reducing the size and complexity of the information users access through small modules (~500 concepts in size), and perspectives with user context awareness.
- Enable the understanding of the semantics of a concept through (1) extensive documentation and metadata, (2) connecting the SCRIBE ontology with Infosphere Business Glossary Terms (IBM BG 2011).
- Allow traceability of concepts' genesis through versioning and detailed explanations of sources.
- Facilitate deployment of customized solutions by disciplined design of packaging and patterns of assembly.

The efficiency and robustness of SCRIBE relies on these principles:

- Utilizing OWL-Lite with boolean combinations of class expressions to speed up inferencing.
- Careful taxonomies of object properties to enable inferencing even in the presence of extensions and customizations.
- Restricting relations between domains and packages to avoid the bloating of the inferred model.
- Keeping the model open to evolution through an iterative development process.

Finally, SCRIBE aims to be an authoritative model through these principles:

- Integrating well-known standards through explicit concept mappings.
- Collecting and implementing scenarios directly from stakeholders.
- Support a collaborative authoring and revision process.

Authoritative semantic models

Large and comprehensive semantic models are qualitatively different than small semantic models. Small semantic models are models that a IT developer or business operations person can review within an hour or two, and where they can make a decision between three outcomes: (1) *"we will use the model as is"* (as-is consumption), *"we will use it after we rework it to our organization"* (consumption with extension/customization) or *"what's missing in the model that we need to have, and what's*

included in the model that we don't need is/are substantial enough that we are better off doing our own model" (no consumption). Large semantic models are less like pure IT software lifecycle artifacts, and are more like business brands (Miller and Muir 2004). That is, they succeed only when:

- a) users and potential users have a coherent set of expectations as to what we be delivered, with what level of quality, completeness and support
- b) the actual product (i.e. the set of concepts and relations in the semantic model) is able to evolve through numerous releases while remaining true to the implicit "brand value proposition"
- c) the producers and consumers of the branded product or model know what content is promised to be included, and know what content will always be excluded
- d) at initial introduction to the "marketplace", there is tangible and specific value to the early adopters
- e) as a hedge against competition, the brand grows to adjacent offerings, which the early adopters get value from as well as attracting new users
- f) users of the model ("the brand") get both specific technical value as well as the "cachet" associated with being part of a larger community of users with shared objectives and criteria of quality
- g) the owners of the brand ("the model") consider the entire experience of the users, from selecting which branded offerings to acquire, to best channels to learn the features and benefits of the different offerings, to customization for individual customer need

Note that branding is used for exclusive products, it is used for open-source offerings, it is used for political factions... and we believe it must be used for large semantic models.

In a project starting in IBM Research, we decided to look at how to build a comprehensive, multi-aspect, evolvable, and modular model of the information produced and used by many agencies of city government, many businesses located in cities, many workers, residents, citizens and commuters in cities. This model is intended to represent information flowing into, flowing out of, and used in a city operations center as well as by numerous agencies. They are enabled by IT systems (particularly software) that carry out city operations as well as by many citizens and businesses in the city which observe, subscribe, report city status. The diversity of information flow vary from things like traffic accidents to discolored drinking water.

The city operations center will operate continuously from the time it opens up. In large cities, significant data about operations in progress will need to be continuously displayed, updated, and computed over. This means that in principle, we need to build our model, and the software based on our model, so that updates to the model can be

injected as the system runs, and will become apparent the next time a calculation is done, the next time a screen is refreshed, the next time a web form is displayed, etc. We believe that because non-semantic approaches have not succeeded in tolerating "as the system runs" schema updates, if we succeed in our work we will attract many software development teams, and that other continuous sense-and-respond real world applications will be attracted to this approach.

The authoring process

We follow a 6-step process to build the SCRIBE models. In the first step, we collect information from cities and standards. In the second step, we collect typical scenarios on Smart City domains like public safety and transportation from major regions of the world. We then select key concepts from these scenarios and make them available for wider contributions and review. In the fourth step, we select and model concepts and their relationships. In the fifth step, we modularize the ontological models to handle reuse and maintenance issues. In the sixth step, we generate IT artifacts (e.g., Service Oriented Architecture service interfaces).

We now present an overview of the ontology modeled in the first release.

The SCRIBE ontology

The current version of SCRIBE is based on a few key design insights:

- The organization of a city is very specific. A Public Safety department in a city may have different competencies from a similar department in another city. What we model is the abstract organization of a city (e.g., departments may have sub-departments, a department has stakeholders that manage, monitor or make decisions) and the city services as a taxonomy of service areas (Fig 2). When a city is instantiated, its departments and competencies use the building blocks already provided by SCRIBE.

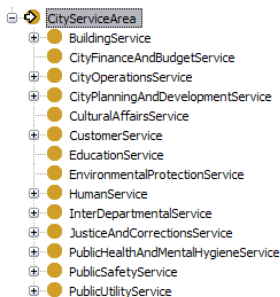


Figure 2. Organization of city services.

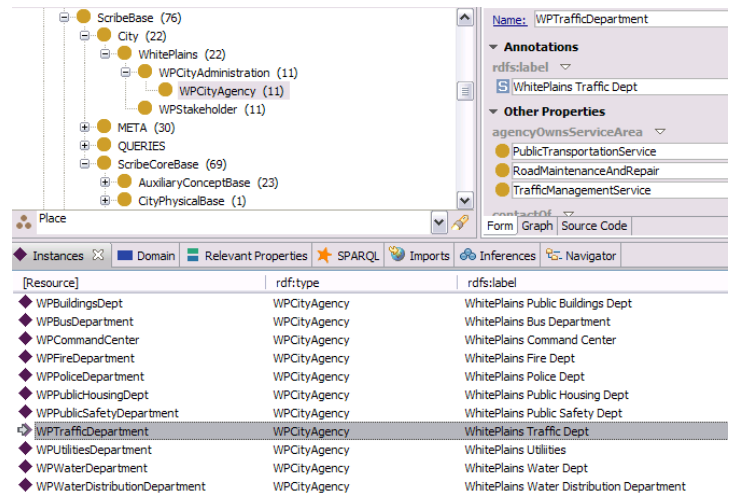
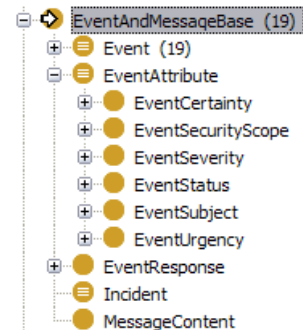


Figure 3. Fictitious Traffic Department

For example, assume that our original Road Work scenario takes place in White Plains, NY. We may model the Traffic Department as having three competencies (in Fig 3). Numbers in parenthesis indicate the number of instances in our initial prototype.

- To model a Smarter City operation is to model the flow of events and messages through the system. Our message and event hierarchy is based in both CAP and NIEM (Fig 4)



- Every concept in SCRIBE is amply documented and its relation with existing standards is captured through annotation properties (Fig 5) and can be viewed with validation queries (Fig 6).

Class Form

Name:

Annotations

classSteward

explanationOfSource

overlapsCAP

superSetOfNIEM

rdfs:comment

rdfs:label

rdfs:seeAlso

owl:versionInfo

Class Axioms

rdfs:subClassOf

Figure 5. Sample Metadata for class Event

☒ Acknowledgement	☒ overlapsCAP	☒ Ack	☒ Acknowledges receipt and acceptance of the message(s) identified in <re...
☒ ActualMessage	☒ equivalentToCAP	☒ Actual	☒ MessageStatus indicating that it's actionable by all targeted recipients.
☒ Alert	☒ overlapsCAP	☒ Alert	☒ Initial information requiring attention by targeted recipients.
☒ AllClear	☒ equivalentToCAP	☒ responseType:AllClear	☒ The subject event no longer poses a thread or concern and any follow on...
☒ Assess	☒ equivalentToCAP	☒ responseType:Assess	☒ Evaluate the information in this message. This value SHOULD NOT be use...
☒ Avoid	☒ equivalentToCAP	☒ responseType:Avoid	☒ Avoid the subject event as per the instruction
☒ CBRNE	☒ equivalentToCAP	☒ CBRNE	☒ Chemical, Biological, Radiological, Nuclear or High-Yield Explosive Threat o...
☒ Cancel	☒ overlapsCAP	☒ Cancel	☒ Cancels the earlier message(s)
☒ DraftMessage	☒ equivalentToCAP	☒ Draft	☒ MessageStatus for a preliminary template or draft, not actionable in its cu...
☒ Environmental	☒ superSetOfCAP	☒ Env	☒ Pollution and other environmental events.
☒ Error	☒ overlapsCAP	☒ Error	☒ Indicates rejection of the message(s) identified in <references>; explana...
☒ Evacuate	☒ equivalentToCAP	☒ responseType:Evacuate	☒ Relocate as instructed in the instruction
☒ Event	☒ overlapsCAP	☒ EventSubject	☒ Occurrence or event that may require some response from the system or...
☒ EventCertainty	☒ superSetOfCAP	☒ Certainty	☒ Denotes the certainty of the subject event of an alert or an actionable it...
☒ EventOfUnknownCertainty	☒ equivalentToCAP	☒ Unknown	☒ Unknown Certainty
☒ EventResponse	☒ superSetOfCAP	☒ responseType	☒ Denotes the type of action recommended for a message. The response co...
☒ EventSecurityScope	☒ superSetOfCAP	☒ scope	☒ the intended distribution of an event or a message (from CAP)
☒ EventSubject	☒ superSetOfCAP	☒ subject	☒ Occurrence or event that may require some response from the system or...
☒ EventUrgency	☒ superSetOfCAP	☒ urgency	☒ Denotes the urgency of a message or notification.

Figure 6: CAP-related concepts

The first version of SCRIBE includes the upper ontology, whose main sub-taxonomies are (Fig 7):

- *CityPhysicalBase*, which includes the physical objects (geospatially and temporally based) in the city: landmarks, roads, sewage networks, etc.
- *EntityRoleBase*, describing organizations, people, items (i.e., entities as in NIEM) and their roles.
- *EventAndMessageBase*, based on CAP, including *ExternalEvents* (like storms, road work), *Message* (a Road Work Advisory), *WorkItem* (Road Work Work Item) and a *SystemEvent* (application alert). All of these are events, since they can be organized temporally and causally.
- *KPIBase*, which organizes KPI's according to whether they are cost related, quality of service or response time.
- *MeasurementBase*, for measurements (height, length) and measurement units (inches, cm).
- *OrganizationBase*, which captures the abstract organization of a city plus the set of service areas (Fig 2)
- *ProtocolBase*, describing the city protocols as a set of protocol steps.
- *SCGeo*, the geospatial core sub-ontology.
- *TimeBase*, an extension of time.owl.

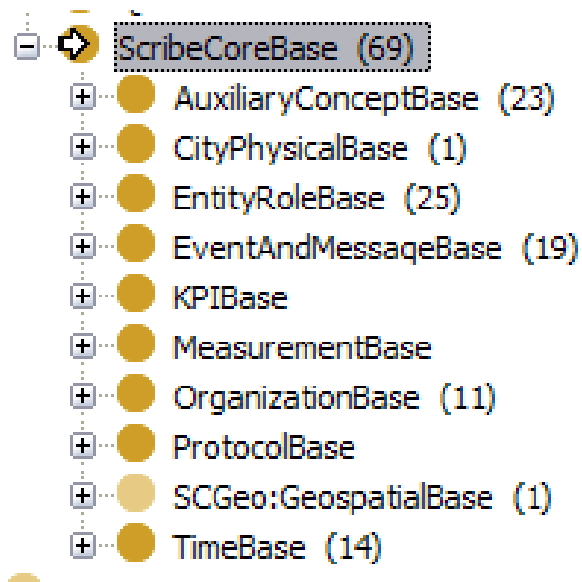


Figure 7: The SCRIBE Upper Ontology

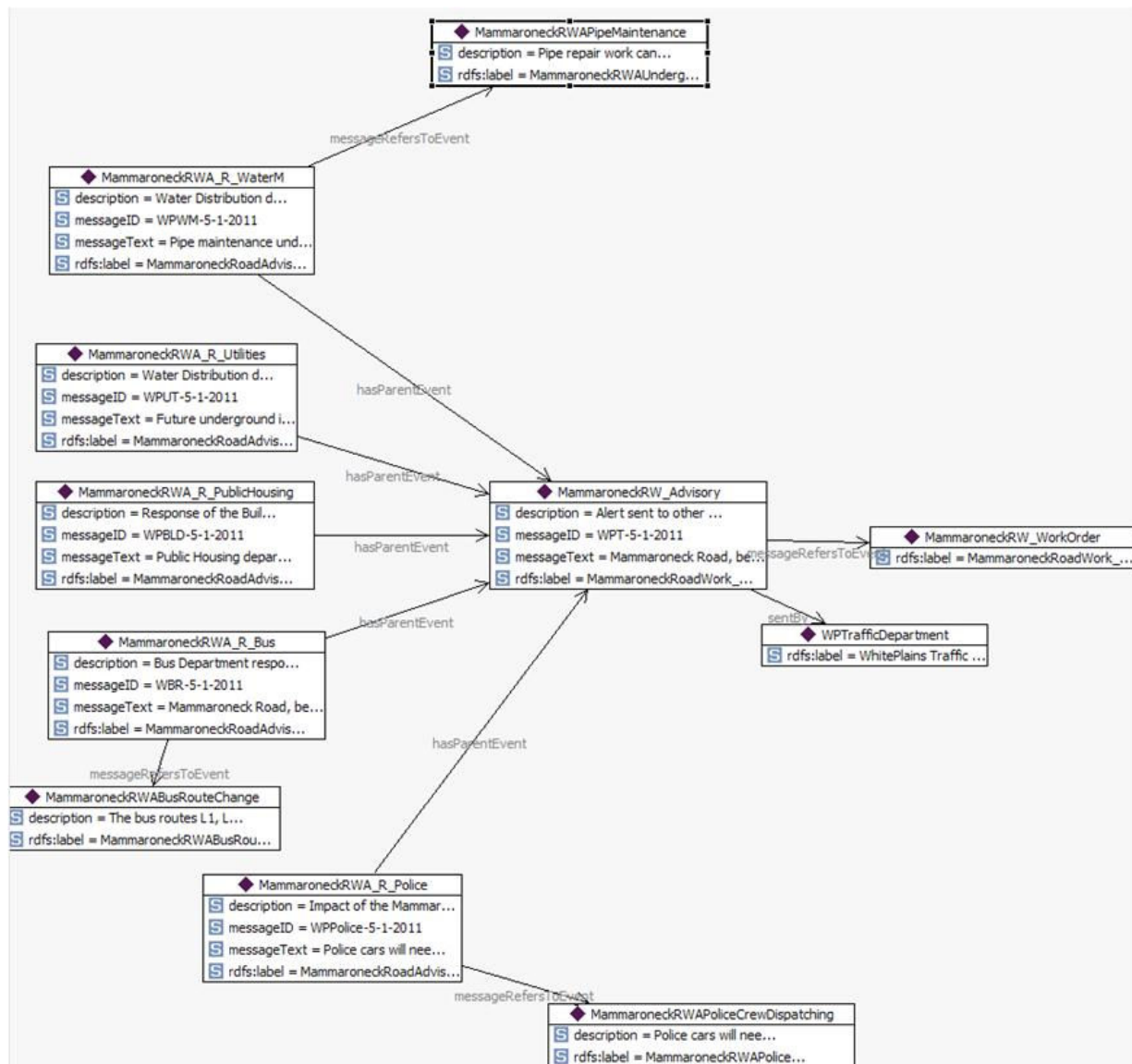
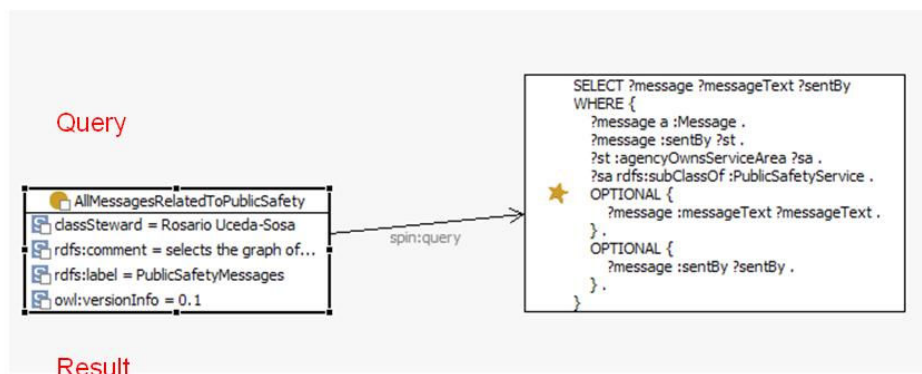


Figure 8. Road Work messages and related events

In our Road Work Scenario, a SCRIBE developer will describe the organization of the departments and their competencies. The thread of messages and events is depicted in (Fig 8).

Once classes have been instantiated, we can issue queries like ‘what are the responses received from Public Safety related agencies?’ This query’s result is depicted in (Fig 9).

Figure 9. Public Safety-related responses.



◆ MammaroneckRWA_R_Fire	◆ No impact to the Fire Department.	◆ WPFireDepartment
◆ MammaroneckRWA_R_Police	◆ Police cars will need to be sent to the end points of the w...	◆ WPPoliceDepartment

Conclusion

In this paper, we presented SCRIBE, a large-scale semantic model for Smarter Cities based on data gathered from cities worldwide. We discussed the design tradeoffs, designer/user experience, and some of the pragmatic challenges related to a model that integrates large, complex, heterogeneous data sources. The work is an engineered attempt towards consumable, industry-strength semantic models for cyber-physical systems and will be extended to different city domains in a phased manner.

Our approaches to consumability include leveraging well-known open standards, using extensive metadata for each class that describes how it relates to concepts in one or more open standards, distributing reporting and validation queries with the model, and a city-specific but very clean upper ontology. In the next phase of our work, three separate groups will develop SOA solutions using this ontology.

Acknowledgments

Our sincere thanks to collaborators and members of the SCRIBE team: Jim Amsden, Ron Baker, Anthony Carrato, Carolyn Hyink, Stacy Hobson, Randy Kubich, Arnaud Le Hors, Pranav Gupta, Pol MacAonghusa, John Meegan, Martin Stephenson, and Keith Wells.

References

- New Form of Web Content that Is Meaningful to Computers Will Unleash a Revolution of New Possibilities, *Scientific Am.*, May 2001, pp. 28–37; www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-4A9809EC588EF21.
- Brickley D. and Guha, R.V. (eds.). RDF Vocabulary Description Language 1.0: RDF Schema, *W3C recommendation*, 10 Feb. 2004; www.w3.org/TR/rdf-schema/.
- Maedche A. et al, Managing multiple and distributed ontologies in the Semantic Web, *XLDB Journal*, <http://portal.acm.org/citation.cfm?id=953238.953240&coll=DL&dl=ACM&CFID=18551261&CFTOKEN=91249095>, 2003.
- Horrocks, I. DAML+OIL : A reason-able Web Ontology Language, *EDBT 2002*, Jensen C., et al., Eds., Springer-Verlag, LNCS 2287, pp. 2-13, 2002

- McCool, R. Rethinking the Semantic Web, Part 1, *IEEE Internet Computing*, Nov-Dec 2005
- McCool, R. Rethinking the Semantic Web, Part 2, *IEEE Internet Computing*, Jan-Feb 2006
- McIlraith, S. A., Son, T. C. and Zeng, H. Semantic Web Services, *IEEE Intelligent Systems*, pp. 46-53, March/April, 2001
- McGuinness D. and van Harmelen, F. (eds.). OWL Web Ontology Language Overview, *W3C recommendation*, 10 Feb. 2004; www.w3.org/2004/OWL/.
- OMG, Unified Modeling Language, <http://www.uml.org/> 2011
- Oasis Common Alerting Protocol (CAP), V 1.2, <http://docs.oasis-open.org/emergency/cap/v1.2/pr03/CAP-v1.2-PR03.pdf>, 2009
- NIEM. National Information Exchange Model (NIEM), http://www.niem.gov/files/NIEM_Introduction.pdf, 2007
- Ucore. Universal Core (Ucore), Common Data Model, <https://ucore.gov/ucore/node/20>, 2010
- MRM. Municipal Reference Model, Municipal Information Systems Association, <http://www.misa.on.ca/en/municipalinterface/municipalinterface.asp>, 2006
- Predictive Model Markup Language, V 4.0, <http://www.dmg.org/pmml-v3-0.html>, 2010
- IBM BG. Infosphere Business Glossary. <http://www-01.ibm.com/software/data/infosphere/business-glossary/>, 2011
- Miller J. and Muir D., *The Business of Brands*, Wiley Publ. ISBN: 9780470862599, 2004.

Using Business Content Explorer for Smarter Projects

Sumana Bhat, Karen Brown, Ankur Jain, Bipin Joshi, Biplav Srivastava,
Srikanth Tamilselvam, Terrence White¹

IBM Research - India, IBM STG - US, IBM GBS - India, IBM GPS - India, US

Abstract

We built a standalone tool, the *Business Content Explorer*, to provide domain-specific content, consisting of concepts, business rules, metrics and business processes that are readily available to IT professionals (software architects and developers). This tool structures content based on the Business Content Hierarchy (BCH). The novelties in the tool are: (a) how the user can use the tool, and (b) how the content is created. The user is able to use the tool to browse both relevant and authoritative content, and also view its evolution over different projects. Specifically, the user can select a specific subset that can be used in their project, enhance it, export it, and then re-import the extended content to reconcile it with the master content. For content creation, we followed a novel, semi-automated, method in which public documents such Request for Proposals (RFPs), tenders and papers are collected; then, content extraction techniques in off-the-shelf tools are used to identify content.

1. Introduction

Domain-specific content is a driving factor in the success of any IT effort, and especially in the nascent Smarter Planet (SP) efforts. In this paper, we will use examples of how this tool affects traffic data collected by a local government. In this case, the government not only needs to communicate with its own agencies, but also with external stakeholders, including citizens, businesses and international organizations. Some of the information shared can be highly technical; yet, much of the information sharing today is informal, ad hoc, and ambiguous. For example, a term like *congestion* is used by traffic personnel, fire departments, electrical utility companies, mayoral offices, citizens, civil contractors, and IT companies implementing Intelligent Transportation Systems.

To formalize content of a domain, we identify concepts (in glossaries), business processes, metrics, and business rules as the key content types that need to be shared. The questions now are 1) how to obtain them, 2)

how to arrange them, and 3) how users will consume them.

The conventional approach to acquire content is to ask the experts. However, this is not always practical. Experts can be hard to obtain, they may be affected by their individual experiences and biases, and using them can be costly and slow. Our approach uses a novel, semi-automated method in which public documents such as RFPs, tenders, and papers are collected; then, content extraction techniques in off-the-shelf tools are used to identify content. Basing our approach on documents allows us to identify important processes that factor into real-world efforts, instead of processes that are merely theoretically possible. In addition, we can more easily obtain candidate processes that can be reviewed by experts. In this way, the process of Business Process Hierarchy (BPH) creation becomes practical.

Content extraction (and the closely related area of information extraction) is a well researched sub-area of information management that has many research prototypes and commercial tools. A good survey for information extraction techniques is *Information extraction*. In: *Foundations and Trends in Databases* [Sarawagi, 2007]. Approaches can vary based on the type of documents considered (for example, plain text, PDF, MS Word, or Excel), the type of prior knowledge assumed (or domain models), the supervision needed (for example, supervised vs. unsupervised), and the type of concepts extracted (for example, plain list vs. hierarchy). We work with IBM SPSS Text Analytics [2], a commercial tool that works with Adobe Acrobat, Microsoft Word, and other popular document formats. The tool performs unsupervised extraction and can generate concept hierarchies. We also used another type of content extractor tool, GATE [3]. GATE is a research prototype that works in a supervised mode with a variety of documents, and generates a list of extracted concepts.

2. Overview

Any system needs information to perform actions; transportation systems are no exception. The key here is how effectively data can be gathered and managed. Traffic Domain experts must interact with the IT professionals to convey the requirements, build the

¹ Authors are listed alphabetically.

system, and validate the system. There is a serious dependency on the domain experts; otherwise, reliability of the system becomes questionable.

Business Content Explorer helps to manage domain information. In a case where we are using traffic data, we identify concepts, processes, metrics and business rules. *Concepts* are the key entities in the domain, *processes* are the activities they perform, *metrics* are the measurement of these activities, and *rules* are the behavioral guidelines used to manage the activities. Business Content Explorer manages this information so that concepts, processes, metrics, and rules are arranged in a hierarchical fashion, in which each element can be explored individually.

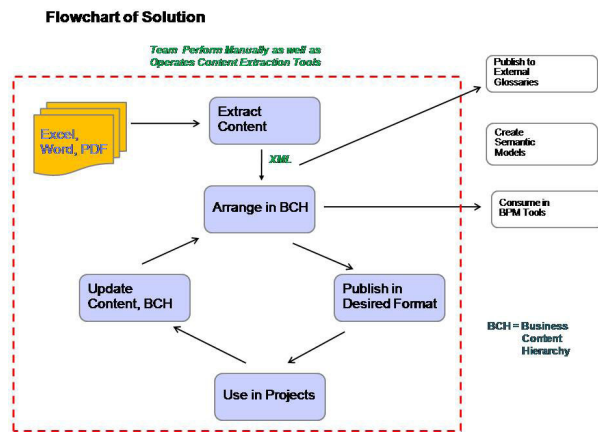


Figure 1: The Business Content Model

3. Extracting Content for Business Content Explorer

We gathered traffic-related information that included RFPs, technical and academic papers, and existing documentation available online. Then, we extracted content for Business Content Explorer, which uses BCH information including domain data processed in .xml format. Before proceeding, we had to extract valuable

data and convert them to BCE consumable format.

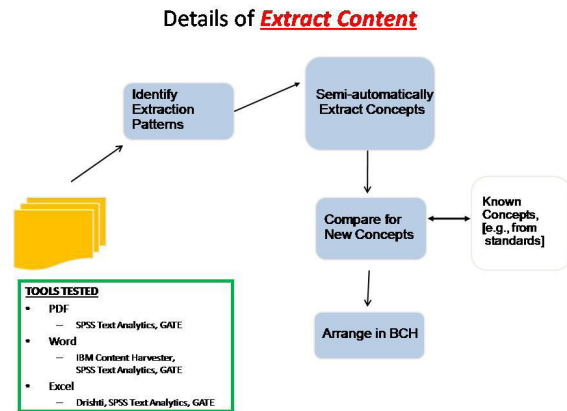


Figure 2. Extracting content

To do this, we used 1) IBM SPSS Text Analytics to obtain concepts (SPSS is an unsupervised extraction method in which you cannot impose constraints), 2) GATE to extract metrics and business rules (GATE is a supervised extraction method that enables you define extraction rules, and 3) the National Information Exchange Model (NIEM), which is a standard for information exchanged between transportation and public safety organizations to extract concepts, rules, metrics, and processes. We also performed manual extraction of concepts and metrics.

4. Supporting Tools for Extracting Content

We wanted to build the tool and be able to update the content, so we had to ensure that duplication did not occur. Having multiple team members conducting various extraction procedures meant an increased risk of duplication. It also made it harder to detect a pattern in the team members' findings. To avoid these difficulties, we built a custom utility with Excel Comparator.

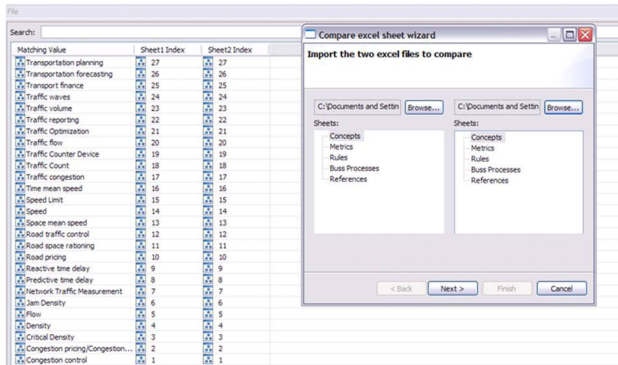


Figure 3: Using Excel Comparator

Using Excel Comparator enabled us to group files together from one location, and enabled us to compare two Excel documents to avoid duplication. This reduced the time needed to manually check for duplication, and enabled us to verify that information was grouped together in one location. As a result, we were able to avoid data loss and concentrate on the process instead of getting lost in extraneous details.

5. Business Content Explorer

Business Content Explorer is a browser-enabled user interface (UI) that enables users to view and manipulate Business Content Hierarchy in an .xml format.

We chose .xml because it is a standard implemented as a plugin, works with Internet Explorer, and uses XSL, XPath, Javascript and HTML technologies. It is packaged as a stand-alone application. See Figure 4 for a description of the activities that BCE enables the user to perform.

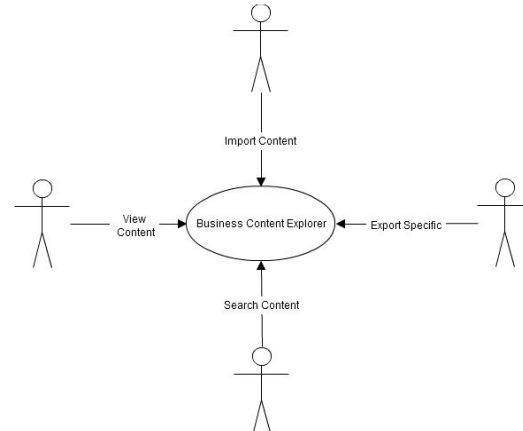


Figure 4: Use-case diagram for business content

BCE translates the BCH XML into DHTML using XSL transformation. BCE allows the content to be displayed in a tree-like structure or alphabetical view. The user can also search for specific content. For example, if a user wants to find information on the financial aspects of traffic projects (specifically, how a traffic system can fund a traffic project with parking revenues), he or she can browse the content or use the search feature (see Figures 5 and 6).



Figure 5: Steps to display content in BCE using BCH.

The Business Content Explorer tool also enables users to export the content from BCH, or import the content into BCH. This is useful if another project needs only a portion of the BCH data. This enables the users from the other project to export the rules, import the rules to their project, and use the rules as a base for their own research. BCE has pagination features that to handle huge concepts. Although numbers will vary by project, they can be quite large. For example, we have approximately 6000 concepts, 130 business processes, 15 metrics, and 10 rules.

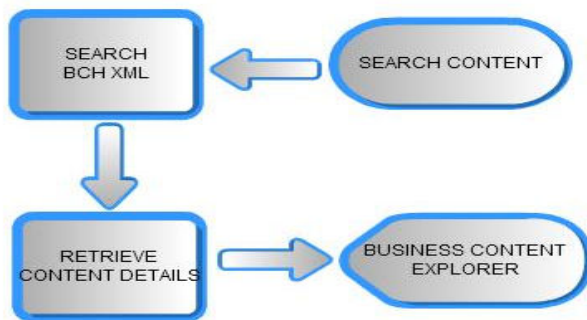


Figure 6: Steps to search content.

Users can view this new BCH using the import feature of BCE (see Figure 7). The import feature allows the user to merge the content into an existing BCH being displayed by the BCE, or to load the content as a new BCH.

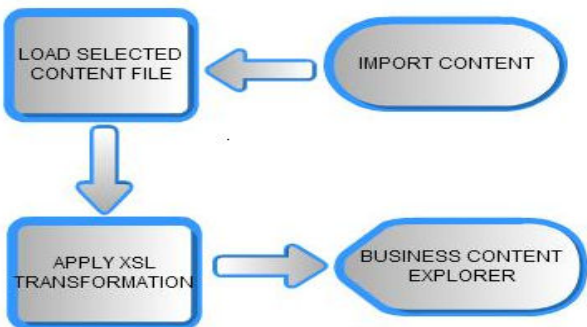


Figure 7. Steps for importing content.

Users can also export the content (see Figure 8).

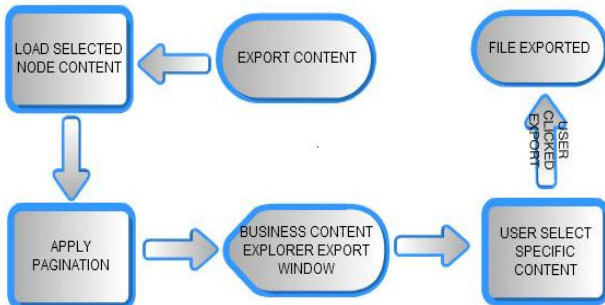


Figure 8. Steps to export content.

BCE also supports annotation feature to arrange information within a category. For example, business processes can be organized under different classification schemes like APQC's Process Classification framework or even the one generated by extraction like SPSS. Both kind of annotation can be found in the tool.

6. Conclusion

We have explored the problem of how to create and provide domain-depended content to a vast set of IT professionals. In response, we have created the notion of a business content hierarchy to organize content, provided a method to create a hierarchy from business documents, and previewed the Business Content Explorer tool to view and consume them.

The tool will released soon. Initial feedback from prospective users has been encouraging, since it meets a critical demand.

References

1. Sarawagi, S.: Information extraction. In: Foundations and Trends in Databases, Vol. 1, No. 3, Pg. 261 to 377 (2007)
2. SPSS, I.: IBM SPSS text analytics. Website (2011), available at: <http://www.spss.com/software/statistics/text-analytics-for-surveys/>
3. GATE: General architecture for text engineering (gate). Website (2011), available <http://gate.ac.uk>

Appendix A

In this appendix, we provide an illustration of a user, Susan, who implements the BCE tool to find content. We discuss what options the user had without the BCE tool, and what benefits the user can now get.

Suppose there is a requirement specialist called Susan who is looking for information on financial aspects of traffic projects and how a traffic system may fund it from parking revenues. Until now, Susan would have looked at reports from previous projects that her organization may have done as well as browsed through requirements in public documents like Request For Proposals (RFPs) initiated by other cities. After spending 3-4 days of effort, she would settle on the information she could find.

With the BCE tool, Susan now has an alternative. She can launch the tool and get a view as seen in Figure 1.

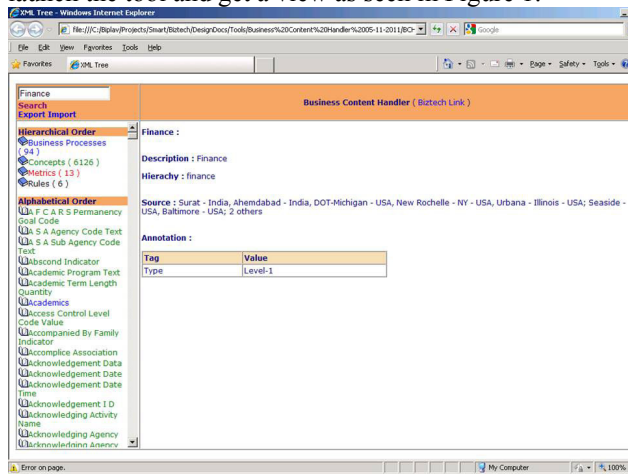


Figure 1: Tool Snapshot: Searching a concept – Finance

She notices that there is a large number of concepts and business processes in the tool and a small number of metrics and business rules. She decides to search for *Finance* and gets the page shown. She notices that this is a high-level business process concept.



Figure 2: Tool Snapshot: Browsing a process - Fare Collection Process.

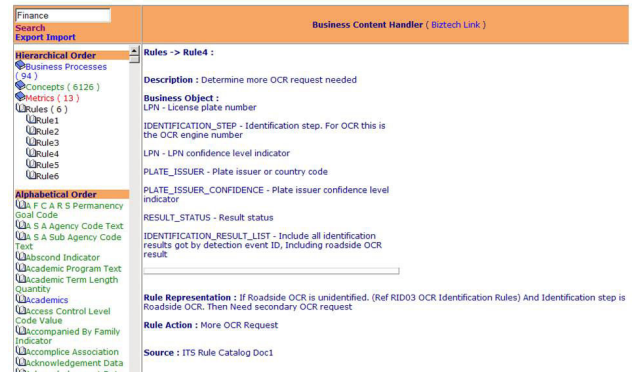


Figure 3: Tool Snapshot: Checking details of a business rule - License Plate Number recognition.



Figure 4: Tool Snapshot: Browsing a metric - Parking Index.

So, Susan browses the business processes section and finds the Fare Collection Process. As shown in Figure 2, it is a fourth-level process under finance, fare and fare processing. She looks business rules in the tool and looks at one on License Plate Number recognition as shown in Figure 3. She now looks at what metrics are related to finance and parking. She notices one on Parking Index as shown in Figure 4.

Once Susan has identified whatever content is of her interest, she can select it by exporting them for her project usage. In Figure 5, she exports only the metrics (7 metrics in the example) and then imports them in the context of any other content she may have already identified. The tool allows import as well as merging of content. Therefore, the same process can be used to merge

the content exported from a completed project back with the content of the reference BCE.

As we see, the BCE tool is able to give Susan a starting point for content exploration in the traffic domain immediately. Where a content exists, it also tells her the source about how the system got the content and various other content details. Thus, Susan is able to improve the quality, speed and efficiency of her work. However, note that the tool in itself is not sufficient to give all the domain information to the user because it relies on the set of documents needed to create the content. The content should be created with a diverse base of documents and continuously refreshed so that it remains relevant over time.

Tool Snapshot: Export from One BCH and Import/ Merge of Metrics to Another

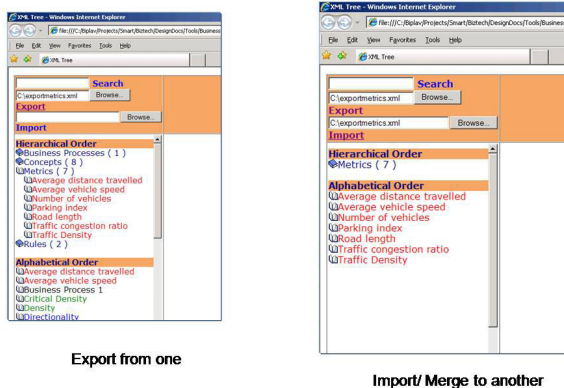


Figure 5: Tool Snapshot: Export and Import/Merge feature of BCE.

Ontology based Qualitative Case Studies for Sustainability Research*

Dong Han, Kilian Stoffel

University of Neuchâtel

Neuchâtel, Switzerland

{dong.han, kilian.stoffel}@unine.ch

Abstract

The overall objective of this paper is to describe the development of methodologies and to present an implementation for supporting researchers in the domains of sustainability. The methodologies are inspired by the approach of qualitative case studies as it is gradually gaining a large acceptance in these domains. In this paper, key elements of qualitative case study research are analysed and the state of the art relevant to these elements is presented. Ontologies, as the core methodology, are proposed as the basic tool to set up and represent knowledge from different perspectives. Original textual data is annotated with ontologies and information is retrieved from these ontologies by different ways for learning, inferring and proposing. Experiments and applications of sustainability scenarios are presented and an outlook to some further design and development possibilities are given.

1 Introduction

These years, qualitative case study has become an important methodology for scientific research and industrial applications in many fields, such as management, economics and environment. Particularly, in the domains of sustainability, it is getting an increasing attention due to its inherent advantages in many aspects. In this paper, we are going to present an ontology based qualitative case study approach for the domains of sustainability including energy and environmental management. This methodology as well as its implementation aim to extract information from original data and analyse it in multiple ways to provide feedback to domain experts. It was motivated by the need in the domains of sustainability but can be extended to other similar domains. The data we are using in this paper is collected in an ongoing interdisciplinary research project and consists of environmental data. The data stems from documents produced over several years in the context of the *Global Report Initiative*[GRI, 2011] and

is analysed from a practical point of view by sustainability experts based on the approaches originated in computer science proposed in this paper.

In accordance with its characteristics, the case study approach provides many advantages which make it very attractive particularly in domains like sustainability management. First, the subject topic can be highly complex while lacking of corresponding theories as for example the study of new service design and environmental issues[Stuart *et al.*, 2002]. Furthermore, our research objective is mainly focusing on the question why the phenomenon emerge which is also well covered by the case study approach[Yin, 2009]. All these characteristics are arguments for using the qualitative case study approach in the domain of sustainability management.

Environmental issues have drawn increasing global attention. Not only international organisations which focus specially on the environment, but IT and financial companies with business closely related to environment, have started focusing on environmental issue as a decisive factor for their plans and operations. One of the usual problems they are confronted with is the large number of documents and reports containing information about environmental issues. Besides, since a high proportion of the original data is presented in a textual form, quantitative analysis is not sufficient. There is a clear need for an effective approach for supporting the analysis of their qualitative information. Qualitative case study, hence, is one of the appropriate methodologies for supporting sustainability management.

Because of their undeniable advantages for knowledge representation, ontologies are adopted in this paper as the core formalism for storing and managing information and knowledge. They allow for representing knowledge from nearly all domains and provide support for learning and reasoning processes. In this paper, we use ontologies for formally modelling the knowledge from the original data. Additionally, we use ontologies to formalise the grounded theory approach — one of the most important methodologies in the qualitative research domain. It enables us to incorporate into our system external knowledge from domain experts as well as from database.

The contribution of this paper focuses on providing a generic methodology and the description of a prototype for qualitative case studies in the domain of sustainability research. As stated in the previous paragraphs, the most sig-

*This work is supported by Swiss National Science Foundation(SNSF) project "Formal Modelling of Qualitative Case Studies: An Application in Environmental Management" (Project No. CR21I2_132089/1).

nificant advance in our work is to allow for analysing very complex data, with a rich and diverse vocabulary. These characteristics caused the lack of standardised approaches readily available. Given this concrete condition, our objective is to successfully extract information and knowledge from the data, to set up a formal model to define a systematic approach, and to provide meaningful results to domain experts. Consequently, we highlight the advantages of using ontologies during our research process while selecting it as the basic tool for knowledge representation. We are thus able to investigate several integrated techniques to provide generic methodologies for the sustainability domains. Our implemented system has been used to contribute to the understanding of environmental management and its impact on financial service firms.

The paper is organised as follows: the second section introduces the state of the art; the third section presents the research methodology; the fourth and fifth sections explicate the system implementation, some experiments and use cases, while the final part is presenting the future work and conclusion.

2 State of the Art

In this section, we are going to present a detailed review on the existing scientific literature related to the main ideas, namely qualitative research, and the concept of ontology as well as their combined utilisation.

2.1 Qualitative Research

Qualitative research [Taylor and Bogdan, 1998] is an academic field dealing with the acquisition and analysis of non-standardised data. Data used in qualitative research can be provided using a wide range approaches, such as historical and phenomenological research. In the context of this paper we will essentially concentrate on the grounded theory approach [Glaser, 1992; Strauss, 1987]. In grounded theory the starting point is the data, which is marked by a series of *codes* which are then grouped together into *concepts* in order to reduce the dimensionality. These concepts are again grouped together to form *categories* which are finally used to build a *theory*. This contrasts the standard scientific method where the starting point consists of a hypothesis which is then tested against specially designed experiments. Based on grounded theory we will develop a formal framework for case study research in the domains of sustainability.

2.2 Ontologies in Computer Science

According to [Buitelaar *et al.*, 2005], in computer science an ontology is an explicit, formal specification of a shared conceptualisation in a domain of interest, where *formal* implies that the ontology should be machine-readable and *shared* so that it is accepted by a group or community. Mostly, they build upon a hierarchical backbone and can be separated into two levels: *upper level ontologies* (which describe the most general entities, contain very generic specifications and serve as a foundation for specialisation) and *domain ontologies* (which aim at describing a subject domain and where entities and relations of a specific domain are sometimes expressed directly in the texts belonging to it).

2.3 Ontology Learning

Ontology learning is concerned with knowledge discovery in different data sources and with its representation through an ontological structure and, together with ontology population, constitutes an approach for automating the knowledge acquisition process [Buitelaar *et al.*, 2005]. According to [Maedche and Staab, 2001], there are two important aspects in ontology learning: *availability of prior knowledge* — if knowledge exists, it must be used to create a first version of the ontology, which is then extended (semi-)automatically through learning procedures — and the *type of input* — according to its nature, input data is classified as structured data, semi-structured data (dictionaries, user tags [Benz and Hotho, 2007]) and unstructured data. The nature of data of interest for this paper is intrinsically non-structured and consists of natural language texts in the form of PDF documents.

3 Research Methodology

Based on our analysis and state of the art given above, the detailed research methodology is presented in this section.

3.1 Ontologies as Knowledge Representation Tool

Grounded Theory (GT) is probably the most systematic methodology currently used in qualitative research. In this section we will describe how we intent to integrate ontologies into this methodology and what the improvements are given this integration, especially from a formal point of view.

The basic idea of GT is to generate a theory starting from data. In our ongoing project, as the way of how we coordinate the research collaboration, GT is essentially regarded as the bridge between domain experts and computer scientists in that it reveals the principles of both specialised sustainability management and knowledge representation and discovery. The two sides supplement each other by fulfilling the implementation of GT from different aspects. The cooperative experts set up their theories by leveraging our methodologies and prototype whilst we take the reference from their fields to establish the domain knowledge. This combination works effectively due to our conformity with the substance of GT and its implementation in the form of ontologies.

Formal Upper Level Ontology

In the first phase of this paper we have developed a formal specification of the GT process.¹ Ontologies are well suited for describing processes (see e.g. [Hepp and Roman, 2007]). The process we have to describe here is very simple compared to a complete business process. However the big challenge and the goal of this ontology would be to formally define the process that describes the three coding schemes used in GT, namely: *open coding*, *selective coding* and *theoretical coding*. Open coding refers to a process where the user is free to select whatever codes he judges relevant to the task under consideration. Selective coding limits the usable codes to a

¹A preliminary analysis has shown that formal knowledge representation aspects can be done using ontologies. We have decided to encode all ontologies in the context of this paper in OWL. Depending on the usage of the ontologies we can use the *lite*, *DL* and *full* dialect.

subset of previously (in open coding) identified codes. Theoretical coding is a top down approach where codes have to be selected in accordance with a previously established theory.

Lower Level Ontologies

In this paper we are developing the formal framework that is needed to specify the lower level ontologies which will allow the implementation of a concrete case study research project in the sustainability domains. The set of lower level ontologies is composed of three types of ontologies, namely project, reference and coding ontologies.

The first set of ontologies (project ontologies) defines which GT approach will be used as well as the set of reasoning mechanisms that are allowed. Following this schema, all qualitative case studies will have the same structure even if they are following different views in respect to the GT approach. This will guaranty a much higher comparability between different projects even if they would have all their own structure. This set of lower level ontologies will define a concrete instance of the upper level ontologies which are only defining the formal constraints.

The second set of ontologies (reference ontologies) contains ontologies that are used as reference systems. Domain experts may want to impose a specific view on a given subject and impose a codification based on this reference system. As a minimum an environmental ontology is included as we are focusing on environmental management. This would lead to a *theoretical coding* approach.

The third set of ontologies (coding ontologies) is created by the users while coding the raw data. In our approach the results of this coding process will also be stored in ontologies, as a simple fragment of the ontologies shown in Figure 1. This allows for a direct application of the reasoning mechanisms implemented in the higher level ontologies. Coding ontologies are direct successors of project ontologies as well as reference ontologies and they are thereby inheriting their properties. In this way we guaranty that coding is done accordingly to the constraints imposed by the project and reference ontologies which in their turn have to conform to the upper level ontology.

Ontologies of Sustainability

As far as sustainability is concerned, inheriting from the previous part, a number of ontologies are incorporated as references specifying the domain knowledge. Taking the example of environmental performance indicators, a group of indices to evaluate the performance issues related to environment, we have established ontologies including energy, emissions, materials and transportation. Each indicator, for example, energy, comprises a list of subcategories, such as gas, water, electricity, etc. The knowledge represented in the form of an ontology is readable for both domain experts and machines. It identifies the most important concepts, in the form of a taxonomy, and a hierarchical structure of the sustainability domains. It is contributing to the interface operations, learning, inferring and proposing. Ontologies of the sustainability domains participate seamlessly in all the phases of a project and demonstrate the advantages of our methodologies for a concrete case.

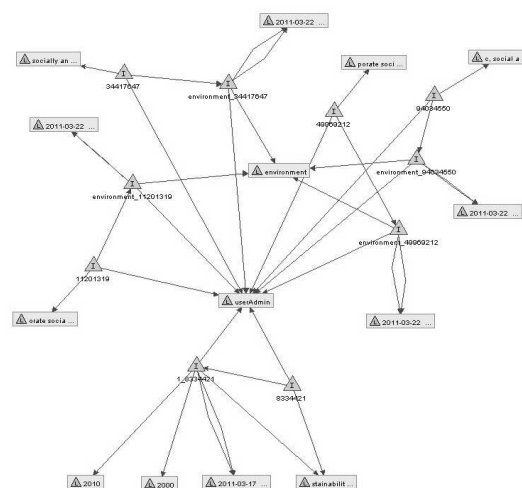


Figure 1: Ontology fragment

With the ontologies of sustainability, the process of case studies is facilitated and amended for two aspects: on the one hand, domain experts are able to employ the knowledge from these ontologies and together with the system we have implemented to do analysis for their case studies in depth. More patterns of the data, which are not intuitive for humans, can be revealed and presented. On the other hand, as their investigation to the cases gets more advanced, the experts are going to perceive more elements from different cases and these elements will be accordingly converted to ontology based knowledge to be navigated by the system.

3.2 Annotating with Ontologies

Based on the formal specification acquired as described previously we have developed a prototype implementation with a solid foundation completely based on an ontological approach. This prototype has been used to implement most of advanced functionalities outlined before.

As benchmark for the prototype we chose to focus on systems currently used by our project partners to analyse "Green Services". Derived from the idea of grounded theory, a number of commercial software products have been released for qualitative applications. Among them ATLAS.ti [ATLAS.ti, 2011] is the most prominent one. It emphasises elementary concepts and their implicit relationship.

ATLAS.ti has been designed on the basis of a *group of concepts*. Beyond all, primary documents are the original data source explored though the domain knowledge. A marked section or piece of text, named quotation, represents the significant or valuable parts of primary documents. These quotations can be exploited by adding codes with initiative names selected by the users or with the words in the text. Another concept, called memo, can be used to add comments about concepts using annotations.

In our prototype implementation, an operational graphical interface providing the functionalities for coding and exporting of information exists together with the mechanisms for

learning and inferring. The whole process follows the design of ontologies on different levels in the way described above as a consistent integration by linking knowledge representation and functional operations.

3.3 Ontology Learning

Ontology learning deals with discovering and acquiring new ontology elements, and integrating them in an existing ontology in a consistent and coherent way. In this paper, ontology learning is the basic process for constructing the coding ontologies. This important functionality is implemented based on the following assumptions:

- Input data for the learning process is represented by unstructured, natural language documents.
- The ontology concepts are represented by latent topics.
- Each document is annotated by the users with a set of labelling concepts (or codes), which may either be provided from the process/reference domain ontologies or from a predefined user language (in the form of a "bag-of-concepts").

As Figure 2 illustrates, the learning process starts from the annotation phase, followed by the term extraction phase. The matrices created using labels, documents and terms become the input for the label-topic model, and the tags (or codes) are analysed to estimate possible relations and to construct the hierarchical structure.

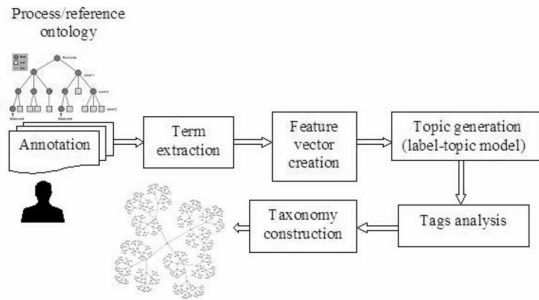


Figure 2: Learning process

More specifically, the stages of the ontology learning process are proposed as follows:

1. *Annotation* - The marked sections of the primary documents are coded (labelled) by the users, according to the constraints imposed by the project and reference ontologies. More than one label can be used to tag a piece of text (considered as an annotated document).
2. *Term extraction* - Each document of the corpus of annotated documents is treated as a bag-of-words. Stop-words are detected using simple, language independent statistical techniques and are then removed.
3. *Feature vector creation* - For each document d , the vector (w_d, l_d) is created. Two matrices, one recording the frequency of each term in each document, and the other recording the frequency of each label in each document, are calculated and used as input for the label-topic model.

4. *Topic generation* - The number of topics (K) is fixed (but approaches using a variable number of topics may also be considered) and a generative probabilistic model is applied to model the correspondence between the documents and the labels.
5. *Tags analysis* - The possible relations between labels, seen as distributions over topics, are estimated using different divergence measures. The considered relations are Subordinate (or hypernym-hyponym relation), Merge (or synonymous relation) and Keep. The divergence measures are calculated based on the KL-divergence or on the conditional independence.
6. *Taxonomy construction* - The hierarchical structure of the tags (codes) is constructed by applying different operations, such as merging two similar tags or creating the hypernym relation between two tags. In this hierarchy, the result of a minimising or a keep operation is always a virtual concept (tag). The resulting taxonomy must be created by minimising the information loss.

3.4 Proposition System

The coding process, whatever schema is used, requires the decision of a user regarding the "best" code for annotating a given document. One of the most important advantages of the label-topic model is its capacity to predict the topic distribution of a new, unannotated document, and — based on this distribution — to find the most likely labels adapted for this document. Concretely, after the model is trained, we can infer information from the existing knowledge in the form of ontologies. The procedure of fold-in is fast, since it only deals with a single document instead of the whole corpus. In this context, the proposition system can be used to estimate the quality of the coding process for a given user, by calculating the "distance" between the codes selected by the user and the "best" ones proposed by the system.

4 System Implementation

Based on the methodologies proposed in the previous section, we have implemented a prototype as our application platform enforcing a systemic utilisation of our approach by the sustainability practitioners.

4.1 Architecture

As shown in Figure 3, the system, named *Qualogier*, is composed of several components - a PDF tool, an ontology manager, an inference engine, a proposition system and an exporting tool.

4.2 PDF Tool

The Portable Document Format(PDF) is an open standard widely used and most of the other data formats, such as Word and web pages, can easily be converted to PDF. This is the reason why we selected PDF as the default format for the original documents. Consequently, assigning annotation and all other essential functionalities for supporting qualitative case study research have to be added to basic PDF documents. We use ICEPdf[ICEpdf, 2011], a Java-based project, as the tool for PDF operations and extend it with extra functionalities. In ICEPdf, basic PDF operations are provided. Users can

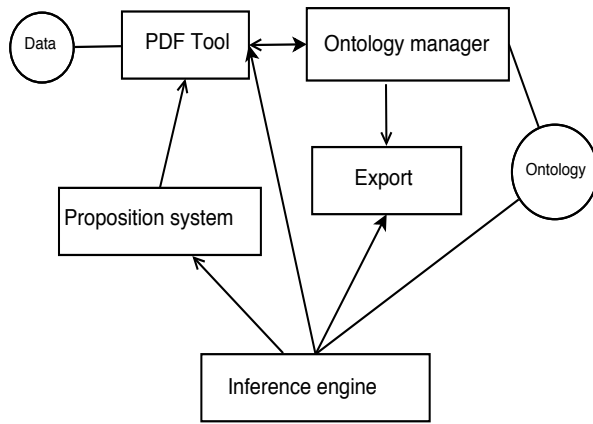


Figure 3: System architecture

open a PDF document, turn pages in different ways, zoom and print. Based on these utilities, as shown in Figure 4, the following functionalities have been added and are implemented in the prototype described in this paper:

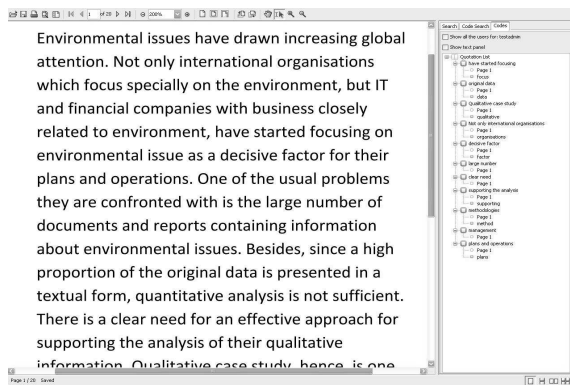


Figure 4: System screenshot

1. **Profile and project management** - For the sake of multiple users and projects, each active instance in the system is located in a scope characterised by its owner and the containing project. User name and project name are displayed at the menu bar of the system as a key. In Figure 5 all the projects previously created are listed for the users to select.



Figure 5: Project list

2. **Quotation** - Users are able to choose some of the words, sentences or paragraphs from the primary document as a quotation. Each quotation contains as information its text, creation and modification time and date and the text size. Figure 6 indicates a quotation highlighted in the original text.

Environmental issues have drawn increasing global attention. Not only international organisations which focus specially on the environment, but IT and financial companies with business closely related to environment, have started focusing on environmental issue as a decisive factor for their

Figure 6: Quotation in the text

3. **Coding** - As demonstrated in Figure 7, once a user has selected some quotations from the document, he can add codes to the quotations as tags. As an important definition in grounded theory, codes are the key elements later used in the analysis provided for qualitative case studies.

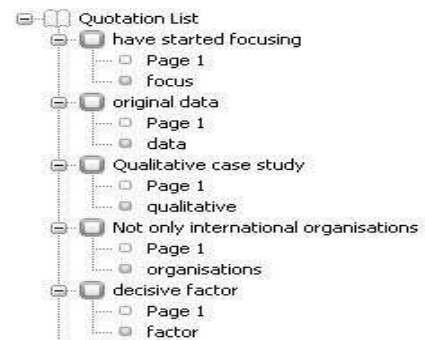


Figure 7: Code tree

4. **Code Search** - A user can specify interesting codes. Likewise he can also exclude words he does not want to find in the result set. Multiple key words are supported.
5. **Code Family** - In some cases, several codes are linked due to their inherent relationships, defined as a code family. Users are able to create a code family with existing codes, and then modify or delete them.

An ontology manager is used to store all the knowledge defined above in different ontologies. They are providing the necessary information for further analysis.

4.3 Inference Engine

We use Jena[Jena, 2011], a Java-based ontology API containing a rule engine, as our ontology inference system. Supported inference mechanisms are forward chaining, backward chaining and hybrid chaining. With the inference engine, we can learn rules and ontologies in order to generate new facts. The results of the inference process can be used in order to make propositions or they can be used by the exporting tool.

4.4 Proposition System

Since we have already extracted the knowledge from annotations in the original documents, the next step is to carry out

ontology learning and exploit inference to provide extended functionalities for qualitative case studies.

Once some codes are given by the users manually, they should be able to get proposition according to the manual coding in an automatic way. Different mechanisms can be used to provide this functionality. Let's assume a code is given (named *codeN*) to a quotation. In our system, a proposition is generated in several ways: first, the system can propose codes which appear together with *codeN* in other quotations. Second, other codes from the same code family as *codeN* are proposed. Third, if the quotation itself contains some key words which are of interest, they are also proposed.

Moreover, data mining algorithms are applied to the data in our system. Currently we have integrated association rule approach, which builds all the item sets by learning and inferring from the ontologies and then adopts a-priori and predictive-a-priori algorithms to generate rules. With these rules, propositional codes are produced for the users according to what they have already coded. The proposition system is a typical case in which ontology learning and inferring are made use of.

4.5 Export and Visualisation

In qualitative research programs it is often important to be able to export the collected data to allow further analysis e.g. in statistical tools. In the standard GT approach this is difficult to realise as codes, concepts, categories and theories are stored in different places. The ontology approach overcomes this difficulty because all the information is stored in one single system and in a common format. Therefore we are able to define and implement an export mechanism based on ontological inference. This is provided by using the OWL query language which will allow a user to specify dynamically which elements of the ontology he wants to export. The data is exported in a portable spread sheet format in order to ensure an easy interoperability with other analysis tools.

Another form of export is to visualise the existing knowledge. For instance, users of the system can give certain ratings to the quotations and codes in each page. They will then be visualised in the form of a rating distribution over the whole document in terms of page numbers. In Figure 8, the X-axis lists all the pages in the document and the Y-axis illustrates the ratings provided by the users. The full mark of the rating can be modified in the configuration file of the system.

4.6 Comparison

As stated above, ATLAS.ti is the most popular software package used in a case study setting and it serves also as the research benchmark of our system. Here we are going to compare the main features of ATLAS.ti and of our system.

From the comparison in Table 1 we can observe that Qualogier has the advantages of learning extracted knowledge to produce analytical results and output them. These are global advantages frequently used by the users and furthermore all these features can be personalised to meet each user's specific needs.

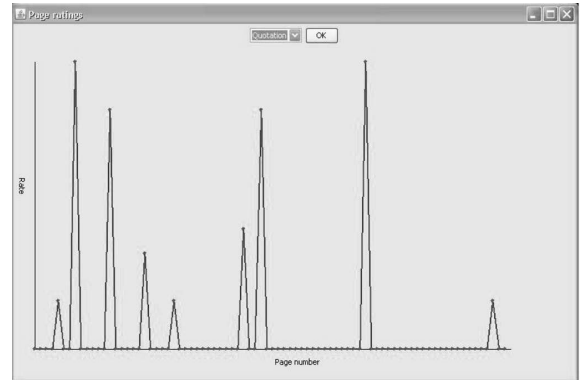


Figure 8: Rating distribution

Features	ATLAS.ti	Qualogier
Basic PDF functionalities	Yes	Yes
Quotation and code labelling	Yes	Yes
Code rating	No	Yes
Code time range	No	Yes
Code proposition	No	Yes
Code family	Yes	Yes
Flexible export	No	Yes
Node network	Yes	No

Table 1: Comparison table

5 Experiment and Use Cases

One of the big challenges in using ontologies is the issue of scalability. When data is getting larger, computational complexity is increasing accordingly. In order to test the efficiency of the system, we have carried out an experiment for adding new codes. A group of ontology files is generated containing a number of quotations varying from one to one thousand. We are going to add new codes to these files to verify the processing time required by the system for automatically iterating over all the files.

While adding a new code, we need first to search for its corresponding quotation in the file, insert the relation between this code and its quotation, and then add the code instance. For each file, we added one hundred codes and calculated the average time required. The experiment was conducted on a PC running Windows XP SP3 with a 2.80GHz CPU and in total of 4GB physical memory. In Figure 9, the X-axis represents the number of existing quotations in each file and the Y-axis indicates the processing time used for adding a new code in millisecond.

From the experimental result, we can see that in the case where a large amount of existing knowledge is already stored in the ontology files, the system still runs efficiently for supporting new coding.

Currently, Qualogier has been deployed to the research team specialised in the sustainability domains, with a focus on environmental analysis. They are conducting research case studies and experiments using this system to facilitate data organising, inferring and exporting. Compared to the

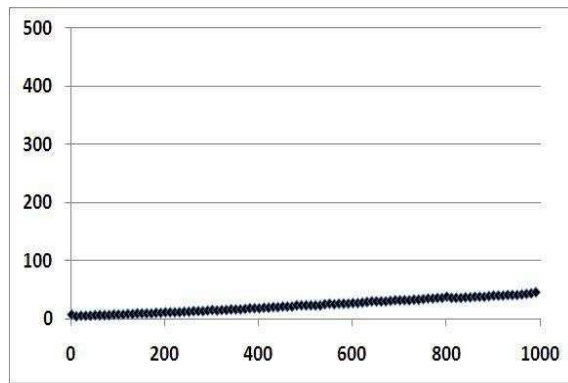


Figure 9: Coding experiment

traditional ways which was used before, ontology shows its advantages in that it is readable for both machines and humans, resulting a more efficient integration between domain analysis and computational output. Data from the original documents is well structured for different types of usage and the codes and concepts established can be maintained in a more consistent way. The system has been proven to be entirely satisfactory for their project.

Furthermore, it is discovered that the methodologies proposed in this paper are specifically feasible and practical for the environment-like topics. As we know, the field of environment contains a large group of concepts with multiple branches. Each of them have influence on each other, either in an implicit or explicit way. The existing way, like relational database, is not sufficient to represent this kind of characteristics. Also, the phenomenon of environmental topics are often resulted by chains of factors. With the reasoning mechanisms of ontologies, the analysis to these processes is more effective. At the same time, different forms of reports and outputs are demanded by the experts for cooperative or conclusive purpose. The query and export capacity of ontologies fits well these utilities.

6 Future Work and Development

As an ongoing interdisciplinary project, it is far-reaching to have reasonable design of the development plan in the coming steps to improve the methodologies and their implementation from different perspective. Based on our previous analysis and description, we will focus on the following aspects:

First, regarding the mechanisms of ontology learning and proposition, we have found that a trade-off is significant between efficiency and accuracy. Since the data is undoubtedly large and diverse, computational duration is usually long. On the other hand, the users of the system desire precise results transparent from the data structure and scalability. We are thus going to find the balance between these two factors to improve our proposed methodologies.

Then, from the technical point of view, a more dynamic and flexible architecture of the system is in demand. Currently the prototype is designed without consideration of reusability and adaptation. As the applications evolve, we need the system suitable for more scenarios not only limited to sustainability

domains and particular data structure. Compositional modularisation is of necessity for the next development.

In addition, it is also significant to set up the evaluation criteria to compare our system and its benchmark. This methodology of the evaluation is supposed to be generic and feasible for all the qualitative software. Besides, we need to design an entire and complete business flow to support and demonstrate how our methodologies are integrated into the processes involving data, system and experts.

Moreover, we will formulate a more advanced research framework that permits for describing and interpreting the links between environmental causes and effects. More domain knowledge will be brought into our project from different sources.

Finally, we will also set up more sophisticated approaches for output evaluation and validation. This is in great necessity because we are supposed to know how the methodologies proposed in this paper and their implementation meet the demand of the users, especially domain experts. We are currently designing some experiments to fulfil this initiative.

7 Conclusion

The goal of this paper, on the one hand, is to make a contribution in the field of case based studies by adding ontologies into the overall process, and on the other hand to integrate the approach of building theories from case studies to the field of ontologies in computer science. The approach would then be validated using real case study research, projected to develop an overall understanding of environmental management orientations and their impact on financial service firms.

In order to achieve this goal we first develop a formal model for case study research based on grounded theory. This model is formalised in the form of ontologies. The ontology approach allows for the validation of a user's work in accordance with the project, domain experts and grounded theory ontologies which are used as constraining reference systems. Based on this model, a prototype implementing the ontology layers and supporting supplementary functionalities has been created and used as a working tool for the case studies in environmental management. In general, we are applying ontologies to enable and facilitate the complete theory development, especially in sustainability domains.

Acknowledgments

The authors wish to thank Swiss National Science Foundation (SNSF) for supporting this work and the reviewers for their time and efforts for reviewing this paper. We would like to thank our cooperative research team for their collaborative work. Thanks to Paul Cotofrei and Christophe Künzi for their help and advice to this work.

References

- [ATLAS.ti, 2011] ATLAS.ti. ATLAS.ti: The Qualitative Data Analysis Software. <http://www.atlasti.com>, 2011.
- [Benz and Hotho, 2007] Dominik Benz and Andreas Hotho. Ontology learning from folksonomies. In *Lernen* -

- Wissen - Adaption, Workshop Proceedings (LWA 2007)*, pages 109–112, Halle, Germany, September 2007. Martin-Luther-University Halle-Wittenberg.
- [Buitelaar *et al.*, 2005] Paul Buitelaar, Philipp Cimiano, and Bernardo Magnini. Ontology learning from text: An overview. In *Ontology Learning from Text: Methods, Applications and Evaluation*, pages 3–12, Amsterdam, Netherlands, July 2005. IOS Press.
- [Glaser, 1992] Barney G. Glaser. *Basics of grounded theory analysis*. Sociology Press, Mill Valley, California, 1992.
- [GRI, 2011] GRI. Home Page of Global Reporting Initiative. <http://www.globalreporting.org>, 2011.
- [Hepp and Roman, 2007] Martin Hepp and Dumitru Roman. An ontology framework for semantic business process management. In *Proceedings of Wirtschaftsinformatik 2007*, pages 423–440, Karlsruhe, Germany, February–March 2007. Universitaetsverlag Karlsruhe.
- [ICEpdf, 2011] ICEpdf. Home Page of ICEpdf. <http://www.icepdf.org>, 2011.
- [Jena, 2011] Jena. Jena Semantic Web Framework. <http://jena.sourceforge.net>, 2011.
- [Maedche and Staab, 2001] Alexander Maedche and Steffen Staab. Ontology learning for the semantic web. *IEEE Intelligent Systems*, 16(2):72 – 79, March 2001.
- [Strauss, 1987] Anselm L. Strauss. *Qualitative analysis for social scientists*. Cambridge University Press, Cambridge, United Kingdom, 1987.
- [Stuart *et al.*, 2002] I. Stuart, D. McCutcheon, R. Handfield, R. McLachlin, and D. Samson. Effective case research in operations management: a process perspective. *Journal of Operations Management*, 20(5):419–433, September 2002.
- [Taylor and Bogdan, 1998] Steven J. Taylor and Robert Bogdan. *Introduction to Qualitative Research Methods*. Wiley, Hoboken, New Jersey, 1998.
- [Yin, 2009] Robert K. Yin. *Case study research: Design and methods*. Sage Publications, Inc, Thousand Oaks, California, 2009.