# Dynamic Context Modeling for Agile Case Management

Manuele Kirsch Pinheiro, Irina Rychkova

**HAL Id: hal-00873994**
**https://paris1.hal.science/hal-00873994**

Submitted on 16 Oct 2013

# Dynamic Context Modeling for Agile Case Management

Manuele Kirsch-Pinheiro and Irina Rychkova

Centre de Recherche en Informatique, Université Paris 1,
Panthéon-Sorbonne 90, rue Tolbiac, 75013, Paris, France
{Manuele.Kirsch-Pinheiro,Irina.Rychkova}@univ-paris1.fr

**Abstract.** Case Management processes are characterized by their high unpredictability and, thus, cannot be handled following traditional process- or activity-centered approaches. Adaptive Case Management paradigm proposes an alternative data-centered approach for management such processes. In this paper, we elaborate on this approach and explore the role of context data in Case Management. We use the state-oriented representation of the process that allows us to incorporate the contextual information in a systematic and transparent way, leading towards agile case management.

**Keywords:** business process agility, context awareness, declarative process modeling.

## 1    Introduction

Davenport [5] [6] defines case management process as a process that is not predefined or repeatable, but instead, depends on its evolving circumstances and decisions regarding a particular situation, a *case*. Case management processes scenarios form dynamically, at run time, and cannot be modeled, managed or analyzed following the traditional BPM approaches [22].

This idea paper builds up on our recent work [19], where we define two forms of agility, leading to more dynamic context-aware business process. The 1st form of process agility is defined as a capacity to handle unpredictable sequences of system events, which results in a dynamically defined order for process activity invocations. The 2nd form of process agility consists in *selecting a right action at the right moment, and with respect to the current situation*. We define it as the ability to monitor and manage *the process context* and to dynamically select and/or alter the execution scenario accordingly. We argue that the second form of process agility is essential for efficient case management. This agility depends heavily on the capability of supporting systems to deal systematically with dynamic process context. Unfortunately, current approaches lack appropriate formalism and mechanisms for context management.

In this paper, we explore the role of context information in the agile case management and propose an extensible meta-model and architecture for representing, capturing and exploring this information in a dynamic and systematic way. We illustrate our findings on the example of crisis management process as defined by the Emergency Plan Specialized on Floods (EPSF) of Hauts-de-Seine [15].

This paper is organized as follows: in Section 2 we discuss the Adaptive Case Management paradigm as opposed to traditional BPM and introduce our example. In Section 3, we define the dynamic context model and illustrate how this model can be instantiated for the crisis management process for process agility. In particular, we focus on the dynamic aspect of the context modeling and discuss the added value of the context model to the case management. In Section 4, we compare our proposals with related works before concluding in Section 5.

## 2    Case Management Process and Adaptive Case Management

The Case Management Process Modeling (CMPM) Request For Proposal released by OMG [12] expresses the practitioners' demand in the case management solutions. OMG defines case management as "*A coordinative and goal-oriented discipline, to handle cases from opening to closure, interactively between persons involved with the subject of the case and a case manager or case team*". Case management processes (CMP) have multiple applications, including "*...licensing and permitting in government, insurance application and claim processing in insurance, patient care and medical diagnosis in health care...*" [12]. The main resource of a CMP is knowledge obtained as a result of communication between multiple actors/users. This knowledge is used for making decisions during the case handling.

Business Process Management (BPM) and Adaptive Case Management (ACM) demonstrate conceptually different views on the system design. Process-centered view adapted by BPM implies that the data emerges and evolves within a process according to a predefined control flow (Fig. 1-a), similarly to a product evolving on a conveyor belt.
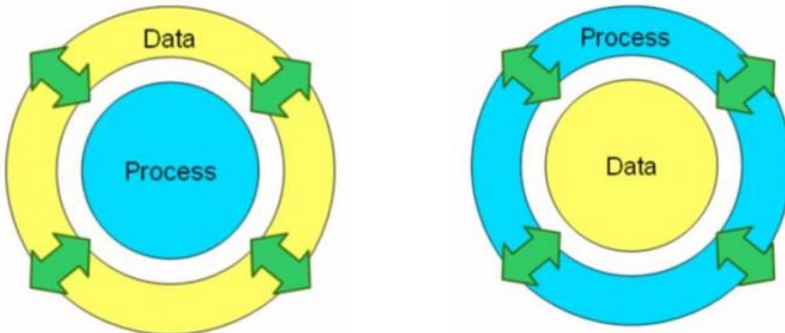


a) BPM – activity centered view on    b) ACM – data – centered view on processes
processes

**Fig. 1.** BPM vs. ACM systems, from [22]

One of the major challenges identified by both practitioners and researchers in the ACM field is the attempts to deal with case management process in the industry the same way as with regular business process - i.e. by applying a process-centered view. In this work, we implement the data-centered view (Fig. 1-b) that is proposed by the

Adaptive Case Management (ACM) practitioners [22]. This view implies that the process shall be adapted at run time, according to evolution of case circumstances and case-related data. This view suits to nondeterministic, knowledge-intensive processes like case management processes [16].

## 2.1    Case Management Process Example: Crisis Management in Case of Floods

Crisis management process is a typical example of case management: it demands interaction between multiple otherwise independent actors (e.g. government, fire brigades, police, public transport, healthcare, electricity and water supplying services, road maintenance etc.). This process is driven by evolved information about the case rather then predefined sequence of activities and, it is thus, suitable for the data-centered paradigm (Fig. 1-b).

In the example below, we consider a crisis management process triggered in case of flood, in the Hauts-de-Seine department of France [15]. A *flood* is an overflow of water that submerges land that is normally dry. It is happening, for example, due to an increase in the flow of a river provoked by significant rainfalls. The risk of a "major flood" is the main natural risk in the Ile-de-France region, particularly during the winter period from November to March. Cities like Paris[1] are confronted to this risk, and, in case of flood, important damages can be expected, affecting thousand of people. In the Hauts-de-Seine department [15], the risk of flood is considered as particularly important since 1910.

The EPSF (Emergency Plan Specialized on Floods) is triggered immediately when the water level rises until 5.5m at the Austerlitz Bridge and will keep rising according to forecasts. Due to its high dynamic nature, this process cannot be handled by a workflow-based approach. Indeed, activities during crisis management are not necessarily predictable and depend on numerous factors that can dynamically change. We list just a few of these factors: watercourse levels, weather conditions and forecasts, electrical outages, traffic conditions, actors' location, equipment availability, etc.

For example, depending on the water level, the crisis management process may require specific traffic control and deviation, partial or complete disabling of public transport lines (SNCF Paris Rive Gauche, RER C, RATP), blocking the highways and principal roads (A86, A14, N14, etc.). The towns affected by the flood may require regular external drinking water supply, in more severe cases - evacuation and temporary accommodation for people, healthcare and childcare facilities. Moreover, the crisis management shall be assured under condition of possible overload or instabilities in telecommunication network, inaccessible or blocked due to heavy traffic roads etc. For example, a traffic jam can put at risk the evacuation of people or deployment of other emergency services. These situations require alternative scenarios such as traffic deviation or preparation of evacuation by air etc. Such (dynamically changing) information defines the crisis management process *context*.

---

[1] See http://cartorisque.prim.net/dpt/75/75_ip.html

More formally, context can be defined as any information that can be used to characterize the situation of an entity (a person, place or object) that is considered relevant to the interaction between a user and an application [7].

We claim that the capacity to timely observe and measure the context information in a systematic way, in order to select and assemble process activities at run time is indispensible for adaptive case management and for our example of crisis management in particular. Thus, we adapt the data-centered paradigm from Fig. 1-b by including into its core the *contextual data*. In the next section, we introduce the concept of dynamic context modeling and illustrate this concept on the example of crisis management process presented above.

## 3      Dynamic Context Modeling

### 3.1      Context Meta-model

The way context information can be exploited for business process flexibility depends on what information is observed and how it is represented. According to Najar et al. [11], the formalism chosen for representing context model determines the reasoning methods that can be used to perform system adaptation to the observed context. A context model (i) ensures the definition of independent adaptation processes and (ii) isolates this process from context acquiring techniques. The same applies to context-aware business process or case management. We claim that the process context information should be acquired, modeled and formally analyzed at run time in order to adapt process execution and to ensure its flexibility.

Several context models have been proposed in the literature [11] [1]. Even if they vary in the adopted formalism (key-values, object-oriented, ontologies, etc.) and in the represented elements, we can generalize some common terms. First of all, most of context models consider a given subject that is observed, typically the user. For instance, Reichle *et al.* [17] and Najar *et al.* [11] consider both that a given "entity" (the user, a device, etc.) is observed. In other models, such as [7], the subject is implicit, since the user is considered as the main observed thing. This subject plays a central role on context modeling, as pointed out by [3] [4], since it is precisely the context of this subject that is currently been observed. Everything we observe is related to this subject. Around it, several elements can be considered. Reichle *et al.* [17] call these observed elements "scope", while Kirsch-Pinheiro *et al.* [9] call these "*context elements*". In both cases, it corresponds to what we really observe from the subject: its location (for a user), the available memory (for a device), etc. When observing such context elements, we obtain *values* corresponding to their status on given moment and that will probably evolve over the time. For instance, by observing the *context element* 'location' for a *subject* 'user', we may obtain *values* for latitude and longitude, corresponding to the current user's location. Besides, some models associate meta-data describing these values. For instance, Reichle *et al.* [17] propose to describe which representation is used for an observed value (*e.g.* a location can be described using latitude and longitude pair or through a postal address). Vanrompay *et al.* [23] consider representing as meta-data a "certainty measure" indicating the reliability of the observed values.

Based on these common terms we identified in the literature, we define a common meta-model presented in Fig. 3. In this meta-model, we consider context as a set of *context elements* that are observed for a given *subject* (e.g. an actor, a device, a resource, etc.). Each subject can be associated with multiple context elements (location, status, etc.); for each context element, we observe *values* that can dynamically change and that can be described by meta-data.

The proposed meta-model must be instantiated in an appropriate model, according to its application domain. This means to transpose and to affine elements from the meta-model in a precise model. Such model can use different formalisms, but ontologies appear as the most interesting to represent context elements [11]. They provide ways to semantically describe these elements and their relationships, as well powerful reasoning mechanisms, notably inference rules. According to Bettini *et al.* [1], "*ontological reasoning can be executed for inferring new context information based on the defined classes and properties, and on the individual objects retrieved from sensors and other context sources*". The meta-model and an ontology model form a complementary approach, allowing a better understanding of the context modeling.
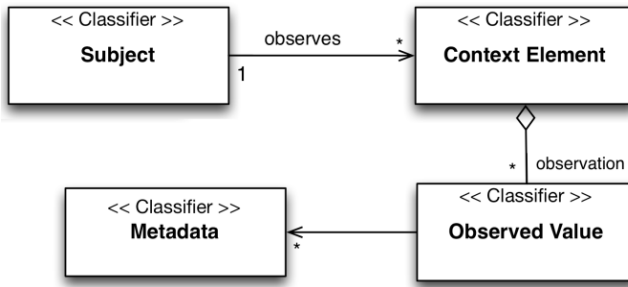


**Fig. 2.** Context meta-model considering context as a set of context elements

Thus, subject and context elements can be semantically described using ontologies. Choosing what elements will be represent depends on the application domain, in our case, the activity domain related to our case management. For instance, a context ontology for the crisis management process will describe subjects such as the different teams and actors' roles, and context elements such as the location, the availability of a given equipment, etc. Based on such context ontology, we propose to represent *events* as logical constraints over context concepts and observed values. We formalize the context of a subject $s$ in a time $t$ as follows:

$$Context(s,t) = \{ Element(s, ce) \},$$

where *Element(s,ce)* represents the observed value of the context element *ce* for the subject *s*.

For the crisis management, we can monitor different actors (subjects) involved into the process (e.g. police brigades, fire brigades, evacuation teams, etc.). The main parameters (context elements) to observe are location and resources available.

*Context (team1, t) = {Element (team1, #location), Element (team1, #vehicle), Element (team1, #agent), Element (team1, #firstAidKit)}*

Along those lines, we can observe weather conditions (subject), with temperature, humidity or rainfall prevision (context elements); general traffic conditions (subject) with the current deviations, traffic jams (context elements); main healthcare and childcare facilities (subjects) with their heating, water, electricity supply, accessibility by the roads (contextual elements indicating the need of evacuation), etc.

## 3.2     Marrying the Dynamic Context Model with the Agile Process Model

Defining and managing an exhaustive context model along with the process model seem to be a challenging task that would potentially raise the complexity and compromise the usability of the overall model. With an appropriate modeling formalism, however, this complexity can be substantially reduced.

In [19], we model a business process as a finite state machine (FSM) [14], where each state represents a process situation at a given time and state transitions define the possible process scenarios. The triggering events specify the underlying process semantics, i.e. conditions for state transitions.

Fig. 3 illustrates the FSM for our example: here process states may evolve from Flood Vigilance (that corresponds to 2.5m water level at the Austerlitz Bridge) to Flood Alert (3.2m) and to Execution of Emergency plan (5.5m). We assume (although this is not documented in [15]) that Federal Alert state can be triggered in case the emergency plan execution is insufficient. Also, at any time, the system can get back to normal state. This FSM is an abstract representation of the process that can be further detailed by refining states and/or adding component state machines showing how separate case-related elements will be handled (road blocks, evacuation process, etc) as described in [20].

According to EPSF [15], at each process state, various activities must/can be executed in order to protect people and goods and to reduce the consequences before, during and after a flood (e.g. the public transport suppression, preparation and executing the evacuation, road blocking, provisioning water, electricity etc.). Due to natural evolution of the crisis situation (e.g. the water level keeps rising) or other conditions (e.g. not enough people/equipment, electricity outage, no road access etc.) execution of some of these activities becomes impossible and alternative sets of actions need to be executed in order to fulfill the crisis management objectives.

Each process state in Fig. 3 can be defined with a (set of) *subject* and its *contextual elements* to observe. Emergent conditions can be modeled as *context events* and expressed using logical conditions on observed values of contextual elements. For example the following condition triggers a hospital evacuation:

*Element(#hospital, #heating) = "out of order" OR Element(#hospital, #electricity)="out of order" OR Element(#hospital, #access)="not available"*

Thus, context events are expressed referring elements from context ontology. This way, event definition can take advantage of reasoning capabilities offered by ontologies. Besides, a context query language, such as [17], allows the expression of rich context conditions. This is particularly important since context information is naturally imprecise and often incomplete. Process states or events can then be defined in terms of context elements and acceptable interval or sets of values.

Therefore, context information plays the following roles: (i) it is a trigger for process state transitions; (ii) it is a part of process state definition; and (iii) it is a boundary condition for execution one or another process activity.
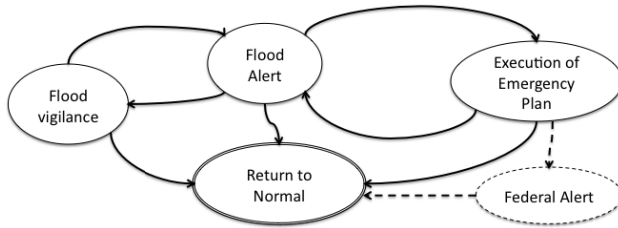


**Fig. 3.** FSM for Crisis management process in case of Flood

## 3.3     Towards Context-Aware Case Management

In [19], we present architecture for CAPE – Context-aware Agile business Process Engine – that defines the following elements for agile process management (Fig. 4):

- Dynamic context monitor
- Activity repository (Process specification)
- Navigation manager.

Here the Activity repository provides a set of activities that can (but not necessarily will) be executed during the process; the Navigation manager provides a context-based guidance, offering an appropriate activity (or a list of alternative activities) to execute from the repository; and the Context monitor is in charge of observing context elements during process execution, in order to enable this context-based guiding. Similar to [13], it recognizes context elements using *plugins*, which feed the monitor with dynamic information about a given context element from a subject. Each plugin observes a given context element. It keeps Context Monitor updated concerning changes in observed context values. Such values dynamically observed by context plugins define the current position of the process in its state space.

To support agility on process definition, we consider that context elements observed for a given process might vary during the process executions. As a result, new context elements to be observed and new respective contextual events are dynamically added to the model.

In practice, the Context monitor can be dynamically extended by adding new plugins or sensors for observing new context elements and subjects. Fig. 4 shows the CAPE model for the crisis management process and illustrates the dynamic context management: a new sensor measuring rainfall level is added through a new context plugin (on the left side of the figure), extending the process definition with new events and, possibly, states and triggering conditions.
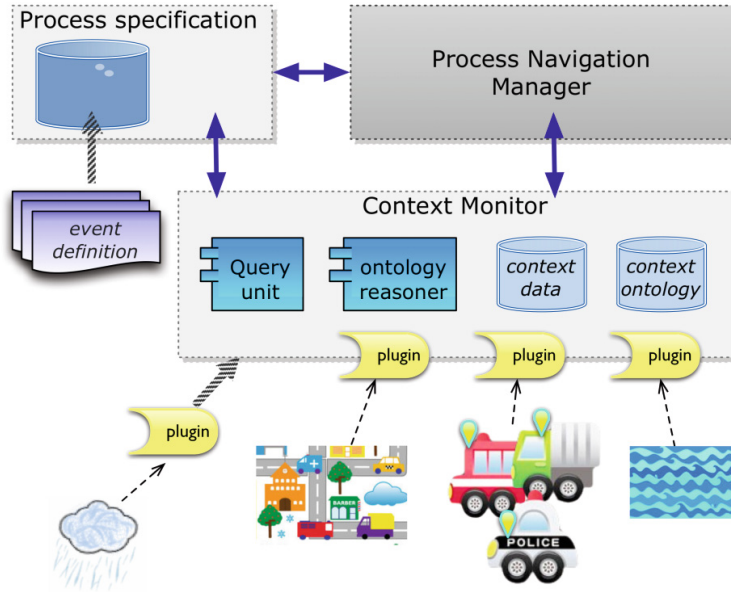
**Fig. 4.** Context Monitor organization in CAPE architecture

## 4 Related Work

The notion of context is not totally new on the BPM field. Other researches [18] [10] [21] have already pointed out the use of context information on process definition. Roseman et al. [18], for instance, consider that external context elements may influence business process (e.g. weather influencing a call center process in an insurance company). They propose to incorporate such elements into business process modeling. Saidini et al. [21] also consider context on business process definition, particularly on the roles played by each actor. They assume that role affectation can be influenced by context information.

Unfortunately, in BPM field, context information is not always modeled appropriately. Quite often works lack of formalisms in representing context concepts and properly managing them. Besides, context models, when they are present, are not general enough, consider only a limited set of context information and focus manly on workflow based process needs. Indeed, the way context information can be exploited for business process flexibility depends on what information is observed and how it is represented. The meta-model we proposed here allows dynamic modeling of context information, supplying the high level formalization necessary for process flexibility.

Identifying context information that can affect a business process execution remains problematic. Similar to [18] [11], we consider context information on concentric levels, from internal context directly related to process actors and activities till external environment context. In this sense, context can be seen as a recursive notion, a 'local' context referencing a more global one [3]. Such concentric and

recursive vision of context information leads us to represent internal process events and external context events indistinctly. In this way, context information becomes naturally part of the process definition and can be considered under multiple levels. Nevertheless, it is worth noting that we cannot enumerate which aspects of all situations are important, as these will change from situation to situation [7]. In other terms, it is impossible to exhaustively describe all context information needed in a general way. Model extensibility becomes then a key element for successfully representing and managing context information.

Such extensibility depends not only on the context model and its capabilities to be extended with new concepts, but also in the capability of observing and acquiring such new concepts during execution time. Even if some works [18][21] have considered context information on business process definition, at design time, they rarely consider context information at run time, during process execution. Often no architecture is proposed in order to dynamically acquire and take into account context information during process execution. CAPE architecture proposes to overcome this issue by adopting an extensible plugin approach, in which new context elements can be easily observed by connecting the appropriate context plugin.

Finally, a context-aware business process does not necessarily means an agile business process. Although context information may bring an interesting contribution for process flexibility, it does not guarantee that the process will be able to react to unexpected situations. Indeed, works such as [18] still describes process as a predefined and explicit sequence of activities, making it difficult to respond to unpredictable situations. We argue that, in order to support process agility, it is necessary to think business process differently, in terms of events and states instead of in terms of activities.

## 5    Conclusion

Crisis management process is an example of a case management process, with unpredictable scenarios. Activities are chosen according to actors' expertise, previous experiences and to multiple input events moving the situation from a state to another. It is a typical example of process that cannot be handled in a satisfactory way using traditional workflow-based approaches. A more dynamic approach is necessary.

For better efficiency, especially in the situation where the time matters and complicated decision making may cost even lives, any automated guidance becomes important. Our approach represents the process as FSM, defined by its multiple states and transition events. This representation allows integrating the contextual events and contextual parameters into process definition: both process events and contextual events are handled uniformly in order to provide an automated guidance and to prescribe/suggest an activity that suits best with respect to the process objectives.

In this paper, we went one step forward in modeling context information for business process. We proposed a context meta-model allowing a high level formalization of the context information, which can be combined with powerful query and ontology based reasoning mechanisms, for more process flexibility. We are

currently specifying a first implementation for CAPE architecture, including Context Monitor elements. A first prototype is expected soon.

The next step of this research is definition and implementation of techniques for automated user guidance. Formal Concept Analysis (FCA) and Galois lattices [2] [8] is one of the techniques we consider. In [19], we illustrate how Galois lattices can be used to classify/organize the case-related information providing a user with suggestions about which activity to execute at a particular situation. Along those lines, the use of Galois lattices for context analysis provides flexible guidance to end-users at run time and supports them with an expertise required for the process handling.

# References

1. Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., Riboni, D.: A survey of context modelling and reasoning techniques. Pervasive and Mobile Computing 6(2), 161–180 (2010)
2. Birkhoff, G.: Lattice Theory, 1st edn. Amer. Math. Soc. Pub. 25, Providence (1940)
3. Brézillon, P.: Expliciter le contexte dans les objets communicants. In: Kintzig, C., Poulain, G., Privat, G., Favennec, P.-N. (eds.) Objets Communicants, Hermes Science Publications, Paris, pp. 293–303 (2002)
4. Coutaz, J., Crowley, J., Dobson, S., Garlan, D.: Context is the key. Communication of the ACM 48(3), 49–53 (2005)
5. Davenport, T.H., Nohria, N.: Case management and the integration of labor. Sloan Management Review 35, 11–11 (1994)
6. Davenport, T.H.: Thinking for a living: how to get better performances and results from knowledge workers. Harvard Business Press (2005)
7. Dey, A.: Understanding and Using Context. Personal and Ubiquitous Computing 5, 4–7 (2001)
8. Egho, E., Jay, N., Raissi, C., Napoli, A.: A FCA-based analysis of sequential care trajectories. In: 8th Int. Conf. on Concept Lattices and their Applications (2011)
9. Kirsch-Pinheiro, M., Gensel, J., Martin, H.: Representing Context for an Adaptative Awareness Mechanism. In: de Vreede, G.-J., Guerrero, L.A., Marín Raventós, G. (eds.) CRIWG 2004. LNCS, vol. 3198, pp. 339–348. Springer, Heidelberg (2004)
10. Mounira, Z., Mahmoud, B.: Context-aware process mining framework for Business Process flexibility. In: iiWAS 2010, Paris (2010)
11. Najar, S., Saidani, O., Kirsch-Pinheiro, M., Souveyet, C., Nurcan, S.: Semantic representation of context models: a framework for analyzing and understanding. In: Information and Ontologies, CIAO 2009, European Semantic Web Conference, ESWC 2009, pp. 1–10 (2009)
12. Object Management Group (OMG): Case Management Process Modeling (CMPM) Request For Proposal (September 23, 2009)
13. Paspallis, N.: Middleware-based development of context-aware applications with reusable components, PhD Thesis, University of Cyprus (2009)
14. Plotkin, G.D.: A structural approach to operational semantics (1981)
15. Préfecture des Hauts-de-Seine: Plan de secours spécialisé sur les inondations Hauts-de-Seine, SIDPC (November 21, 2005), `http://www.ville-neuillysurseine.fr/files/neuilly/mairie/services_techniques/plan-secours-inondation.pdf`

16. Pucher, M.J.: The Elements of Adaptive Case Management. Mastering the Unpredictable. How Adaptive Case Management Will Revolutionize The Way That Knowledge Workers Get Things Done, pp. 89–134 (2010)
17. Reichle, R., Wagner, M., Khan, M.U., Geihs, K., Valla, M., Fra, C., Paspallis, N., Papadopoulos, G.A.: A Context Query Language for Pervasive Computing Environments. In: 6th IEEE International Conference on Pervasive Computing and Communications, PerCom 2008, pp. 434–440 (2008)
18. Rosemann, M., Recker, J., Flender, C.: Contextualization of Business Processes. Int. J. Business Process Integration and Management 1(1), 46–60 (2007)
19. Rychkova, I., Kirsch-Pinheiro, M., Le Grand, B.: Context-Aware Agile Business Process Engine: Foundations and Architecture. To appear in 14th Working Conference on Business Process Modeling, Development, and Support (2013)
20. Rychkova, I.: Exploring the Alloy Operational Semantics for Case Management Process Modeling. In: Proceedings of 7th IEEE International Conference on Research Challenges in Information Science, RCIS 2013 (2013)
21. Saidani, O., Nurcan, S.: Towards Context Aware Business Process Modeling. In: 8th Workshop on Business Process Modeling, Development, and Support, BPMDS 2007, CAiSE 2007 (2007)
22. Swenson, K.D.: Mastering The Unpredictable: How Adaptive Case Management Will Revolutionize The Way That Knowledge Workers Get Things Do, 354 (2010)
23. Vanrompay, Y., Mehlhase, S., Berbers, Y.: An effective quality measure for prediction of context information. In: 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), pp. 13–17. IEEE Computer Society (2010)