Making Workflows Situation Aware — An Ontology-driven Framework for Dynamic Spatial Systems

Stefan Mitsch, Wolfgang Gottesheim
Franz Hermann Pommer, Birgit Pröll,
Werner Retschitzegger, Wieland Schwinger
Johannes Kepler University Linz
Altenberger Straße 69,
4040 Linz, Austria
{firstname.lastname}@jku.at

Robert Hutter
Prologics IT GmbH
Hafenstraße 47-51,
4020 Linz, Austria
robert.hutter@prologics.at

Gustavo Rossi
Universidad Nacional
De La Plata,
alle 50 & 115, primer piso (1900) La Plata,
Buenos Aires, Argentina
gustavo@lifia.info.unlp.edu.ar

Norbert Baumgartner
team Communication Technology Management GmbH,
Linke Wienzeile 4/1/2,
1060 Wien, Austria
norbert.baumgartner@te-am.at

ABSTRACT
Business processes face constantly changing context factors like varying customer behavior or market conditions that force to adapt the underlying workflows to these evolving situations. Information overload induced by the diversity of context factors, however, leads to the inability to provide coherently modeled, comprehensible, and re-usable workflows and the failure to recognize relevant situations in time. The main goal of our research project ProFlow is to leverage situation awareness in all phases of workflow management especially focusing on dynamic spatial systems as encountered, e.g., in the domain of road traffic management. ProFlow thereby bases on a generic ontology-driven framework for situation perception and comprehension. This paper details on the corresponding ontological representations especially addressing extension points that allow developers to extend and configure our framework for their own application domains. This forms the basis for the overall system architecture, which is laid out along its prototypical implementation.

Categories and Subject Descriptors
H.4 [**Information Systems Applications**] – *Workflow management*; D.3.3 [**Programming Languages**]: Language Constructs and Features – *Frameworks*

General Terms
Design

Keywords
Situation-awareness, workflow management, process management, relation calculi

iiWAS2011, 5-7 December, 2011, Ho Chi Minh City, Vietnam.

1. INTRODUCTION
Workflow management is more and more considered key for the overall business success [14][43]. Constantly changing business environments in terms of *internal context factors* (e.g., evolving business strategies, available personnel), and *external ones* (e.g., economic developments, new legal regulations) force, however, to *pro-actively adapt workflows* to these changed situations [44][45]. There is, however, a permanent risk of failing to identify relevant context factors in the induced information overload, stemming from heterogeneous sources and, consequently, being unable to comprehend and in turn project the relevant situations (which are made up of these context factors), processes have to cope with [1][3][25][26][46][50]0. This *lack of situation awareness* endangers to timely and correctly execute workflow tasks, as well as to pro-actively prevent critical situations and escalations, potentially causing significant costs, delays, and quality losses. At the same time, workflow requirements change rapidly and many times these new requirements are volatile, i.e. they are valid for short periods of time, sometimes involving irregular patterns of occurrence. In these cases workflow designers are also stressed by the constraints posed by these cases. Summarizing, workflow designers and agents are currently de-facto scarcely supported by existing systems for achieving *awareness about the situations a workflow is running in*, nor is there a sound conceptual foundation thereof [17].

The main *goal* of our project *ProFlow – Situation Aware Process Management* – is to lay a first *conceptual foundation* for supporting situation awareness during all phases of workflow management, i.e., modeling, execution, and maintenance. In this respect, ProFlow bases on several traditionally disparate research fields comprising *workflow management (WFM)* [30], *situation awareness (SA)* [31], and *context-awareness (CA)* [4]. For demonstrating the applicability of ProFlow we focus on so-called *dynamic spatial systems* [13], as encountered in domains such as road traffic management, which (i) are highly dynamic in terms of a large number of evolving objects, (ii) react in a non-deterministic manner on actions and events occurring within them, and (iii) often cover geographically large environments that are only partially observable.

The paper firstly details on the ontological representations for situation perception and comprehension as well as their extension points in Section 2. The overall system architecture of our ProFlow framework along with its prototypical implementation is sketched out in Section 3. Section 4 compares the approach to related work before, finally, a concluding summary and outlook is given in Section 4.

2. SITUATION PERCEPTION AND COMPREHENSION

Workflow agents are to be supported with appropriate means for situation assessment bringing relevant, high-level situations to their attention. Situations must be machine-interpretable in order to be used by the workflow engine to automatically make decisions on behalf of the agent, which is the pre-requisite for predictive adaptation of workflows. For this, we will base upon our previous work on situation-awareness in the domain of road traffic management pursued in the course of our project "BeAware!"[1] project [7][8][9], where the execution of workflows due to the occurrence of a critical situation plays a major role (e.g., the occurrence of an accident in a tunnel requires certain steps to be carried out, such as informing drivers via variable message signs and radio stations, informing the rescue squad, etc.).

Overall, ProFlow bases on a generic ontology-driven framework for supporting situation perception and comprehension of the underlying situation influencing the workflow. The overall approach pursued by ProFlow for achieving situation awareness heavily bases on semantic technologies in terms of ontologies, (expressed in OWL[2]), which have proven beneficial also in BeAware!. Thereby, occurrences of relevant perceived objects in the context are instantiated as objects with their attributes within the ontological representation. This in turn enables inference of occurrences of concrete situations when checking for the satisfaction of given situation type definitions on basis of objects and their relations. In the following, this paper discusses in detail the ontologies provided along their respective functional subsystems (cf. Section 3) distinguishing between the *perception subsystem* dealing with a generic ontological representation of domain objects and their attributes and the *comprehension subsystem* representing relations between objects and – based on that – situations themselves. In addition, we address extension points that allow developers to extend and configure our framework for their own application domains to provide a familiar vocabulary to workflow agents. To distinguish explicitly between types and instances representing real-world objects, we differentiate between types of the ontology defined in the T-Box and its individuals (i.e., instances describing real-world objects) inside the A-Box.

2.1 Perception Subsystem

A Generic Core SA and WF Ontology. To provide a uniform representation of the concepts found in various domains, we have developed a *core situation awareness (SA) ontology* and a *core workflow (WF) ontology* for capturing information about objects and actions, respectively. Figure 1 depicts a subset of these ontologies sufficient to support the discussion in this section. Actions, which are described by procedures, result in events[3] affecting the attributes of an object — such as its location in a particular spatial region and its lifespan[4] in terms of a temporal period. By reifying actions as objects, we enable the components of ProFlow not only to work with observed objects, but also with actions taken by an operator, which is particularly interesting for *situation assessment* and *situation projection* (not dealt with in this paper). This core ontology reflects and combines various ontologies and approaches from context- and situation awareness research making it possible to map any specific domain-concept onto one or more concepts from the core ontology[5]. Hence, we are able to provide generic components for detecting duplicates in connected data sources, fusing the data, and reconstructing object histories within the framework. These components are defined solely on the basis of core ontology concepts [10] and can at the same time operate on information from any domain. Nevertheless, it is desirable to extend the core ontology with sub-ontologies that provide domain concepts, as described below.

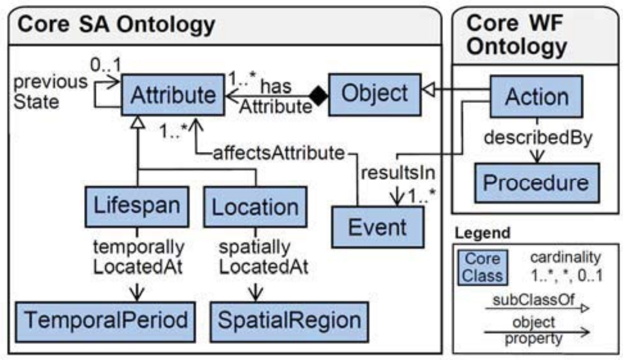

Figure 1: Core SA and WF Ontology

Extension Points for Spatial/Temporal, and WF Sub-ontologies. In order to allow developers to choose from, or extend the set of provided spatial, temporal, and WF sub-ontologies, we generalize the approach outlined by Bateman and Farrar in their survey of spatial ontologies [6] to accommodate temporal and WF ontologies as well. In their approach, objects themselves do not constitute space; rather, they are located at spatial regions, making it possible to disambiguate statements about space (e. g., "being located in a tunnel": is the tunnel an object or a spatial region?), which is not the case in approaches merging objects and locations, such as SUMO [39]. Analogously, objects are decoupled from temporal periods and procedures, resulting in the concepts SpatialRegion, TemporalPeriod, and Procedure (depicted in Figure 1) serving as extension points for integrating concrete spatial, temporal, and WF sub-ontologies into ProFlow. This approach paves the way for assessing relations such as spatio-temporal, causal, or dependency relations in the comprehension subsystem within a concrete reference frame — separated from the objects located there.

Below, we take a closer look at how to extend the SA ontology with a sample spatial graph sub-ontology, depicted in Figure 2, which generalizes concepts from our demonstration domain road traffic management (e.g., junctions could be represented as nodes, whereas the roads connecting them could be represented as edges). In other domains, different sub-ontologies may be required, such as geographic coordinates in air traffic control. For defining the graph sub-ontology, we

[1] www.situation-awareness.net

[2] www.w3.org/TR/owl-ref/

[3] Events capture meta-information, such as transaction times, a concept borrowed from temporal databases [48].

[4] Lifespans resemble the temporal database concept valid time.

[5] In the area of semantic technologies, this approach is known as ontology alignment, whereas in model engineering terms, the core ontology rather defines a meta-model with domain concepts as instances of core concepts.

derive GraphRegion from SpatialRegion, re-using an existing ontology [32] that defines a graph as a non-empty set of contained Edge instances connecting two Node instances. In addition, such a graph region may be oriented, meaning that it occupies a path within the graph, that spans the edges from a start to an end node. The spatial concepts presented so far are still domain-independent; developers may integrate additional characteristics of their domain — for instance, road names as path labels or edge lengths — by deriving from the concepts in the graph sub-ontology, thereby anchoring them in an-other domain-dependent spatial sub-ontology. For example, the road network type RtmNode provides additional information such as the corresponding road (onRoad) or the distance to the beginning of a route (atKm). Analogously, subclasses of Edge (e.g., road elements), GraphRegion (e.g., a concrete road network location), and Graph (e.g., a road network) may define domain concepts more precisely.

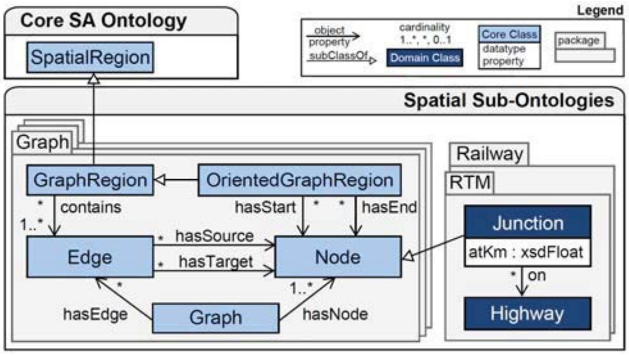

Figure 2: Spatial sub-ontologies extension

ProFlow's default temporal sub-ontology extension is based on OWL time [29] to represent temporal periods with instances of ProperInterval. In OWL time, each proper interval is described by two instances of Instant, one delimiting the beginning of the interval, the other one its end.

In the case of WF sub-ontologies, a multitude of modeling possibilities and requirements from different domains open up, ranging from single actions to complete workflows. They can be represented with languages such as Petri nets [42], UML activity diagrams [28], and various kinds of business process modeling languages, such as YAWL [49].

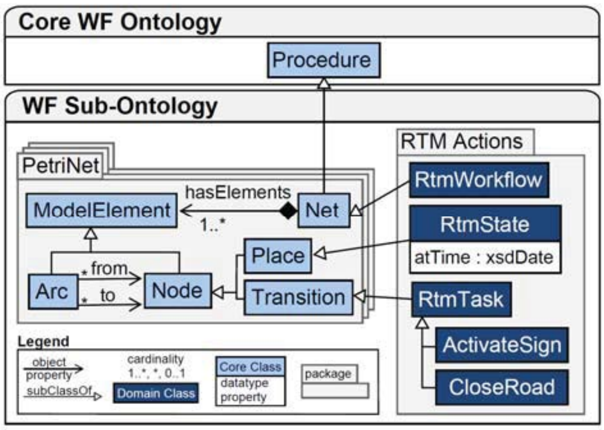

Figure 3: WF sub-ontology extension

ProFlow currently includes the Petri net ontology of Gašević and Devedžić [22] as its default WF sub-ontology extension, as depicted in Figure 3. In the sub-ontology, a Petri net consists of two kinds of model elements: arcs, which specify workflow execution, connect nodes (subdivided into places, representing states in an domain-dependent RTM workflow), and transitions, representing workflow tasks (such as activating a variable message sign, calling the police, or closing a road).

2.2 Comprehension Subsystem

The core situation ontology of the comprehension subsystem, depicted in Figure 4, follows the Situation Theory of Barwise and Perry [5]. It extends the SA ontology with the types Relation — an abstract concept for integrating concrete relation types such as PartiallyOverlapping, which is instantiated in case two objects overlap partially — and Situation for integrating concrete situation types of a domain. To associate additional information (such as valid and transaction times) with relation instances, our ontology represents relations as types rather than as object properties (at the cost of needing to redefine meta-information of OWL object properties, such as symmetry, inverseness, and transitivity for the type Relation).

Spatio-temporal relations. In Situation Theory, major emphasis is put on the relations between objects. The design rationale behind ProFlow is to re-use well-established and tested relation calculi with defined semantics from the field of spatio-temporal reasoning, such as the Region Connection Calculus RCC [16], the Oriented Point Relations Algebra OPRA [19][20], Cardinal Directions [41], Distances [27], the Order of Magnitude Calculus [40], Allen's Time Interval Algebra [1], and Freksa's Temporal Semi-intervals [21].

Figure 4 shows the exemplary spatial relation types ProperPart (i. e., an object is part of another object in spatial terms) and PartiallyOverlapping (i.e., one object's spatial region overlaps another object's region), integrated within RCC. To make such relation types domain-independent, ProFlow distinguishes between primitive relation types and situational relation types [9]. Primitive relation types are characterized by low focus (i.e., they apply to many different object types) and low domain-dependence; most of the relation types in the calculi enumerated above are representatives of this category. Situational relation types, despite being highly domain-dependent, can often be expressed by using one or more of the primitive ones, thus minimizing development effort. Nevertheless, developers can provide a relation interpretation for each relation type, thereby further detailing its meaning in a particular domain. For example, being close to each other may be defined as "within 1 km" in road traffic management, but as "within the same room" in a smart home environment. In order to define subsumption lattices of relation types, we define the semantics of a primitive relation type with respect to other relation types within the framework: for example, the relation type ProperPart of RCC subsumes the relation type SmallerThan of Order of Magnitude Calculus (because, in order for one object to be a proper part of another one it must be smaller). Thus, we can not only assist developers with consistency checks (e.g., issuing a warning if a developer combines ProperPart and LargerThan), but also shortcut relation assessment (e.g., when a particular object is detected as a proper part of another one, we know without checking whether the interpretation of SmallerThan is satisfied that it must be smaller than the second object).

Situation Types. Any two real-world objects are, as described above, in many different relationships with each other[6]. However, only a very small sub-set of the potential relation and object combinations are of interest to a human

[6] In fact, one can find n*(n - 1)=2 pairs among n objects, and when relation calculi are joint exhaustive, the number of relations for each pair is at least as large as the number of relation calculi.

operator (e.g., whether or not a particular tunnel is older than some bridge might be irrelevant). Therefore, to define the relevant situation types of a domain, a developer can extend the core situation ontology by forming relevant sub-classes of Situation. For example, in Figure 4, a situation type S1 is depicted, which can be described informally as the class of "accident in a tunnel causing a traffic jam" situations. Such situations (i.e., instances of a situation type) are identified in terms of the involved objects (i.e., instances of object types) and the relations (i. e., instances of relation types) among them. Situation type definitions extend the core situation ontology with relation clauses and object clauses, which are translated into rules. Figure 4 depicts such a translated rule using a simplified SWRL [55] syntax. In SWRL, an individual ?o being a member of the class C is expressed as C(?o), (e.g., Tunnel(?o1)), whereas two individuals being related by the object property P are expressed as P(?o1, ?o2) (e. g., From(?r1,?o1)).

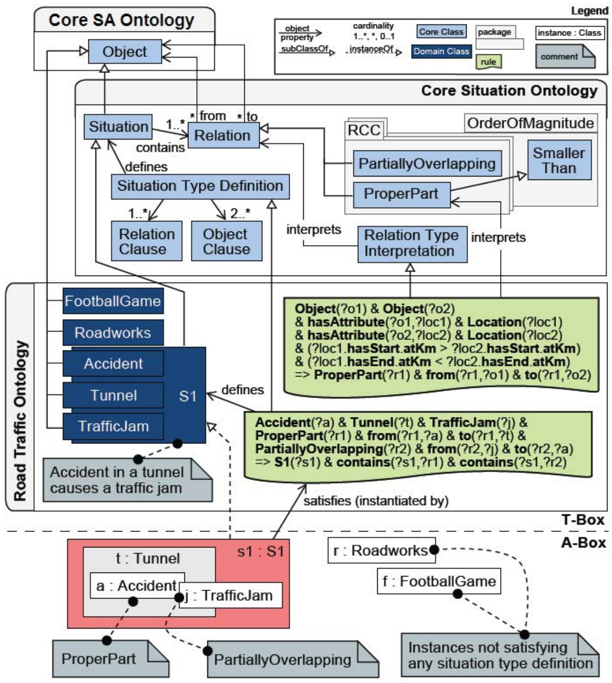

Figure 4: Formalization of a situation type definition on the basis of the core SA ontology

According to the definition in Figure 4, the situation type S1 should be instantiated, if an instance of the domain specific extension (i.e. Road Traffic Ontology) of type Accident (an object clause) and an instance of type Tunnel (another object clause) satisfy the relation interpretation of ProperPart (a relation clause, i.e., the accident is inside the tunnel), and if the same accident individual and an instance of type TrafficJam (third object clause) satisfy the relation interpretation of PartiallyOverlapping (another relation clause, i. e., the traffic jam adjoins the accident). If object instances contained in the A-Box — such as the objects t, a, and j in Figure 4, determined by combined OWL DL and rule reasoning in a forward-backward chaining manner — satisfy such relation interpretations during relation assessment, and consequently satisfy a situation type definition during situation assessment, a new individual, s1 in this case, for the corresponding situation type is instantiated and injected into the A-Box. Note that these rules create situation instances (individuals) in the ontology, which is to date not possible with standard SWRL built-ins. In our prototype, for creating instances we rely on the employed implementation platform with a combination of Lisp and Prolog (cf. Section 3).

Design Support for Relations and Situations. The definition of relations and their interpretation in a particular domain is of major importance for finding interrelated context objects during workflow execution, in order to highlight the occurrence of modeled, relevant situations. Basing on our experience in the development of domain-specific languages (DSL) for model-driven development [33][34][35] we intend to develop a range of design languages for relations, relation families, and situations, aiming at supporting different user interaction preferences including textual DSLs allowing exact declarative and imperative definitions, as well as graphical DSLs supporting sketchy, intuitive definitions. These design languages are to be developed in focus groups together with domain experts from our demonstration domain. For graphical DSLs, a projection of relations and situations onto a graphically presentable, n-dimensional plane – a conceptual space [23], most certainly bounding n by 4 for comprehensibility – is needed.

Figure 5 shows a three-dimensional (temporal size, spatial distance, and mereotopology) conceptual space to exemplify the intended approach: it defines that a critical situation occurred when objects are very close to each other, either partially overlap or one is proper part of the other one, and one object is of longer duration than the other one. By that, a situation type, which is specified through the name associated with the rectangle in the space, as well as its definition, specified through the placement of the rectangle, are given. The conceptual space defines another situation – a warning situation – that could be used to issue early warnings in the workflow of a potentially emerging critical situation (e.g., heavy snowfall along a vital rail cargo route, potentially preventing delivery on time). As a side effect, the distance between the warning and the critical situation can be read off along the metrics of the space.

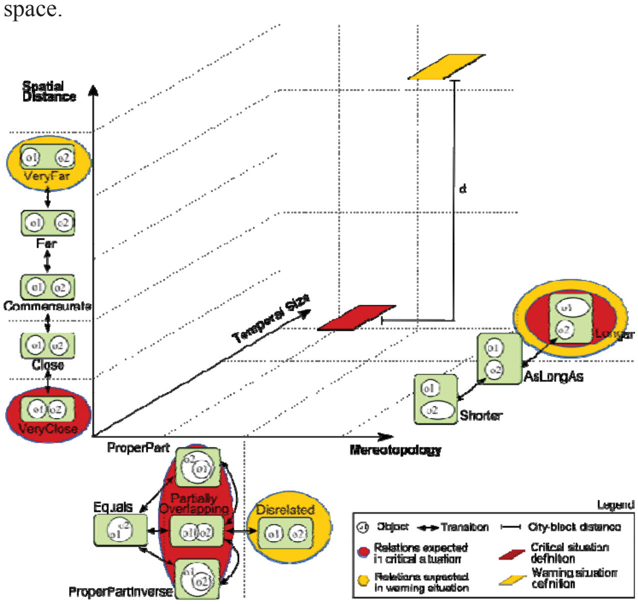

Figure 5: Conceptual space for defining situations

In this realm, situation assessment must not only be able to deal with pre-defined situation definitions provided during the modeling phase, but also with additional situation definitions injected by agents at runtime. Especially to support collaborative and ad-hoc workflows, the problem of communicating implicit assumptions between different agents, as well as between a current and potentially future workflow enactments, can be approached with such participant-generated context information and situation definitions during the execution of a workflow. By that, ProFlow is able to pro-actively notify subsequent agents about critical situations, exceptions, and constraints on the basis of a continuously growing knowledge base. From the context query perspective, we intend to tackle the highly creative nature of collaborative and ad-hoc workflows by supporting agents with extendable reasoning approaches extending or restricting (depending on the query needs) the paths a reasoner searches in a triple graph. For example, in order to search only in situations relevant for the current workflow (instead of all situations), a reasoner can be provided on the basis of core ontologies with rules like

```
(?action resultsIn ?event),
(?event affectsAttribute ?att)
                        -> (?action affectsAttribute ?att).

(?situation contains ?relation),
(?relation derivedFrom ?att)
                        -> (?situation derivedFrom ?att).

(?action affectsAttribute ?att),
(?situation derivedFrom ?att)
                        -> (?situation relevantFor
?action).
```

defining that if we know that a particular "action results in an event" and this "event affects a particular attribute", then it follows that the "action affects the attribute" too. Likewise, if we know that a particular situation contains a relation, and that this relation is derived from a particular attribute, then it follows that the situation is also derived from this attribute. In a summarizing rule, we then combine derived knowledge: if we know, that an action affects a particular attribute, and that a situation was derived from this attribute, then it follows that the situation must be relevant for the action.

3. SYSTEM ARCHITECTURE AND PROTOTYPICAL IMPLEMENTATION

In principle, the ProFlow framework aims at providing a generic solution covering modeling, execution and maintenance of situation-aware workflows being able to integrate with various workflow management systems. In the course of this work, however, we primarily target the workflow management system provided by our industry partner PROLOGICS[7] named FireStart. For this, loose coupling between the ProFlow framework and its surrounding workflow management system (during both, design-time and run-time), as well as abstraction from concrete supporting context subsystems is necessary, realized by three different bridges.

As can be seen in Figure 6, ProFlow integrates with workflow designers, workflow engines, and context sources by means of a *designer bridge*, a *runtime bridge*, and a *context bridge*, respectively. These bridges comprise numerous adaptors mediating between the services provided by ProFlow, and the interfaces of a concrete workflow management system. *Context modeler* and *situation modeler* (developed as part of ProFlow), as well as *workflow modeler* (assumed to be provided by a workflow management system), make up an integrated tool suite for situation-aware workflow modeling. These modeling tools, through adaptors, generate a semantically rich type system in the form of *domain ontologies* extending the framework-provided *core situation awareness* and *core workflow ontology* (allowing implementation of framework functionality independently from the domain, while still being configurable to different workflow and context scenarios). Together, these ontological knowledge bases serve as a repository capturing workflow provenance information, as well as context information as input for the *workflow situation assessor* deriving *high-level workflow situations* on the basis of modeled *situation types*. Such assessed situations are the starting point for the *situation projection subsystem* for projecting possibly emerging *future workflow situations*. Both, current assessed and future projected situations drive execution in the *workflow runtime* and provide concise and comprehensible situation information to agents, thereby supporting situation awareness at the perception, comprehension, and projection level. As cross-cutting component, the *maintainer* extracts information from all levels to close the workflow management lifecycle by feeding back monitored context histories, workflow provenance, and relevant situations into the design tools, resulting, finally, in improved workflow models.

Considering the technologies employed for implementing this architecture, which is currently on its way, we intend to rely on those proven to be suitable in our BeAware! project and employ AllegroGraph[8] as a database backend that stores

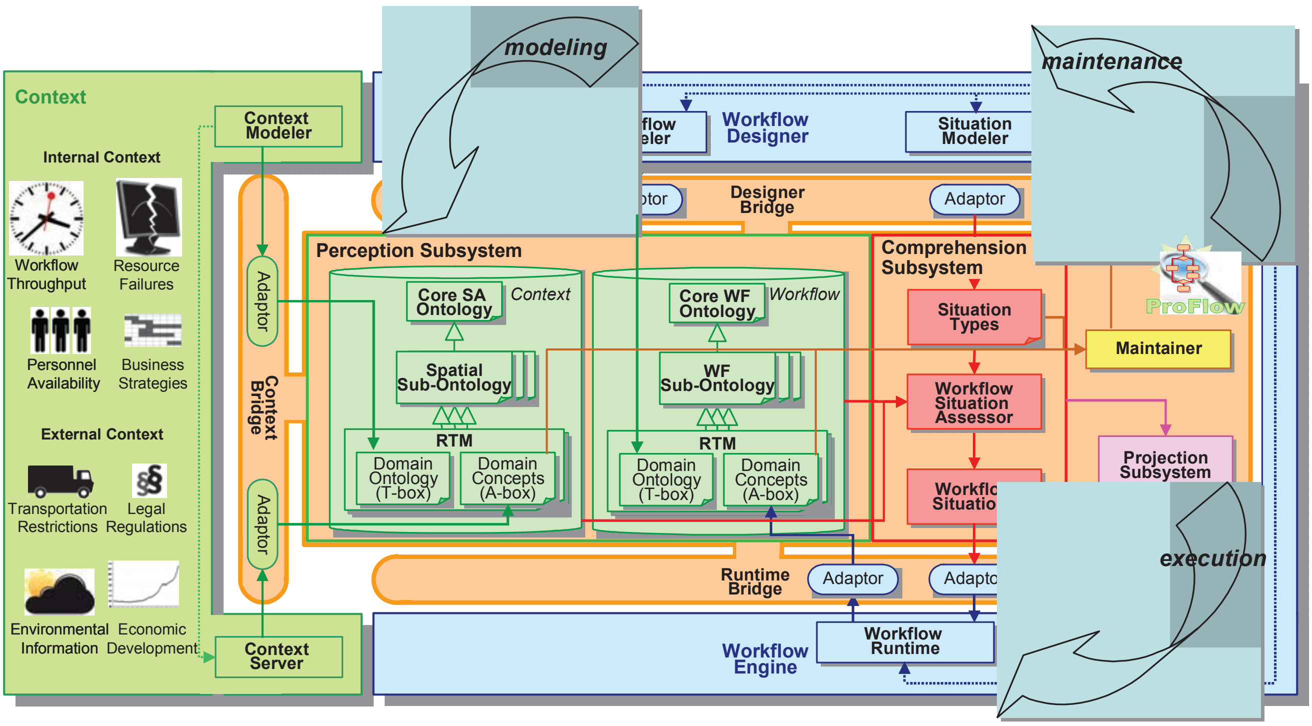

Figure 6: Overview on the System's Architecture

ontologies as well as ontology individuals and provides a performant infrastructure for the implementation of envisioned algorithms. For the realization of the web service interface, we plan to use Java with Jena[9] as a semantic middleware. With AllegroGraph and Jena, we can flexibly exchange the reasoning technique employed and are thus able to tailor the actual reasoner used to the tasks at hand. For example, AllegroGraph provides an integrated RDFS reasoner and can be augmented by, e.g., the commercial reasoner RacerPro to support DL reasoning. Jena's internal reasoner is a hybrid forward-backward-chaining reasoner that can be configured by rules, but DL reasoners such as Pellet can be used as well. The implementation of situation assessment and prediction algorithms will follow a combined DL reasoning and logic programming approach, making use of Prolog and Lisp to ensure high performance.

4. RELATED WORK

Adaptive and context-aware workflows [3][25][26] are of research interest already for some time.

However, ProFlow is going beyond current context-aware workflow systems which provide only uninterpreted, low-level context thus often scarcely supporting existing workflow management systems for representation of and achieving awareness about the situations a workflow is running in. In particular, the ProFlow expands on the state of the art as current workflow modeling languages and standards that either provide only limited context information support [26][49] or do not adequately decouple workflow from context modeling [50], often using proprietary languages such as, e.g., uWDL [25][46], CPDL [36], or pvPDL [15]. Moreover, workflow instances themselves have not yet been recognized as beneficial context entities, although meta-models and ontologies for describing workflows exist [24]. With respect to recognize relevant situations, current approaches focus mainly on deterministic outcomes, providing hardcoded relation interpretations for a particular domain, and supporting situation definitions with hard-to-define rules only [10][16]. ProFlows addresses both deterministic and probabilistic situation assessment algorithms (allowing its application not only in closed-world domains, but also in open-world ones) following the JDL data fusion model [37] to aggregate low-level context to situations. Furthermore, Smanchat et al. [47] revealed that all but one of today's approaches to workflow adaptations only react on context changes, while only [38] support predictive workflow adaptation. ProFlow, adopts situation prediction techniques for qualitative spatio-temporal reasoning [8] for integrated situation and workflow prediction, allowing agents not only to access and browse predicted situations and workflow states, but also to adjust and vary predictions by incorporating domain knowledge, enabling what-if-analysis. Major emphasize is put on overcoming the limitations of related approaches from areas like qualitative neighborhood-based prediction [18], qualitative simulation [2][12] and robot agent control [19], often needing manual modeling on a per domain, or even worse, per prediction basis.

5. CONCLUSION AND OUTLOOK

Workflow designers and agents are currently de-facto scarcely supported by existing systems for achieving *awareness about the situations a workflow is running in*, nor is there a sound conceptual foundation thereof [17]. ProFlow aims at making workflows situation aware by providing an ontology-based framework for supporting modeling, execution and maintenance of situation-aware workflows, which can be adapted to the needs of different domains.

For demonstrating the feasibility of our proposed approach, a prototypical implementation of the ProFlow framework is currently on its way realizing the ontology-based techniques presented herein. As an exemplary workflow engine using the ProFlow framework, we will adapt FireStart as a representative for a workflow management system.

[9] http://jena.sourceforge.net

6. ACKNOWLEDGMENTS

This work has been partly funded by the Österreichische Forschungsförderungsgesellschaft (FFG) under grant 830505 and the österreichische Agentur für internationale Mobilität und Kooperation in Bildung, Wissenschaft und Forschung (OeAD) under grant AR 21/2011.

7. REFERENCES

[1] J. F. Allen: Maintaining knowledge about temporal intervals. Comm. of the ACM 26 (11), 1983.

[2] K. R. Apt, S. Brand: Constraint-based qualitative simulation, Proc. of Int. Symposium on Temporal Representation and Reasoning, 2005.

[3] L. Ardissono et al.: A Framework for the Management of Context-Aware Workflow Systems, Proc. of Int. Conf. on Web Information Systems and Technologies, 2007.

[4] M. Baldauf, S. Dustdar, F. Rosenberg: A Survey on Context-Aware Systems, Int. Journal of Ad Hoc and Ubiquitous Computing (4), 2007.

[5] J. Barwise, J. Perry: Situations and Attitudes, MIT Press, 1983.

[6] J. Bateman, S. Farrar: Spatial Ontology baseline. internal report, Collaborative Research Center for Spatial Cognition, 2004.

[7] N. Baumgartner, W. Gottesheim, S. Mitsch, W. Retschitzegger, W. Schwinger: BeAware! - Situation Awareness, the Ontology-Driven Way. Data & Knowledge Engineering, 69/11, No. 2010.

[8] N. Baumgartner, W. Gottesheim, S. Mitsch, W. Retschitzegger, W. Schwinger: On Optimization of Predictions in Ontology-Driven Situation Awareness. Proc. of Int. Conf. on Knowledge Science, Engineering and Management (KSEM), Vienna, Austria, 2009.

[9] N. Baumgartner, W. Retschitzegger: Towards a Situation Awareness Framework Based on Primitive Relations. Proc. of the Conf. on Information, Decision, and Control, 2007.

[10] N. Baumgartner, W. Retschitzegger: A Survey of Upper Ontologies for Situation Awareness. Proc. of IASTED Int. Conf. on Knowledge Sharing and Collaborative Engineering, 2006.

[11] W. Beer, B. Moser, W. Kurschl, S. Mitch, F. Matusek: Application Development and Management of Smart Camera Networks. Smart Cameras, Springer, 2010.

[12] M. Bhatt et al.: Qualitative Simulation: Towards a Situation Calculus Based Unifying Semantics for Space, Time and Actions. Proc. of Conf. on Spatial Information Theory, 2005.

[13] M. Bhatt. Commonsense Inference in Dynamic Spatial Systems: Epistemological Requirements. AAAI Press, 2010.

[14] Business Process Management (BPM) Market Opportunities, Market Forecasts, and Market Strategies, 2007 – 2013, Wintergreen Research, 2007.

[15] S. Chen et al.: Toward Context-Awareness: A Workflow Embedded Middleware, Proceedings of Int. Conf. on Ubiquitous Intelligence and Computing, 2006.

[16] A. G. Cohn, B. Bennett, J. M. Gooday, N. Gotts: RCC: A Calculus for Region Based Qualitative Spatial Reasoning. GeoInformatica 1 (1997).
[17] D. Distante, G. Rossi, G. Canfora, S. R. Tilley: A Comprehensive Design Model for Integrating Business Processes in Web Applications. Int. Journal of Web Engineering and Technology (IJWET) 3(1):43-72, 2007.
[18] F. Dylla, R. Moratz: Exploiting Qualitative Spatial Neighborhoods in the Situation Calculus, Proc. of Spatial Cognition IV, Reasoning Action Interaction, 2005
[19] F. Dylla, J. O. Wallgrün: On Generalizing Orientation Information in OPRAm. Proc. of German Conf. on AI, 2007.
[20] F. Dylla, J. O. Wallgrün: Qualitative Spatial Reasoning with Conceptual Neighborhoods for Agent Control. Intelligent Robotic Systems, 48(1), 2007.
[21] C. Freksa: Temporal Reasoning Based on Semi-intervals. Artificial Intelligence 54 (1), 1992.
[22] D. Gašević, V. Devedžić: Petri Net Ontology. Journal of Knowledge-Based Systems 19 (4), 2006.
[23] P. Gärdenfors: Conceptual Spaces: The Geometry of Thought, 2000.
[24] A. Haller et al.: An Ontology for Internal and External Business Processes, Proc. of the 15th Int. Conf. on World Wide Web, 2006.
[25] J. Han et al.: Context-Aware Workflow Language Based on Web Services for Ubiquitous Computing. Proc. of Int. Conf. on Comp. Science and Its Applications, 2005.
[26] M. Heravizadeh et al.: Making Workflows Context-Aware: A Way to Support Knowledge-Intensive Tasks, Pro. of the 5th Asia-Pacific Conf. on Conceptual Modelling, Wollongong, Australia, 2008.
[27] D. Hernández, E. Clementini, P. D. Felice: Qualitative Distances. Proc. of the Intl. Conf. COSIT '95, 1995
[28] M. Hitz, G. Kappel, E. Kapsammer, W. Retschitzegger: UML @ Work, dpunkt Verlag, 2005.
[29] J. R. Hobbs, F. Pan: An Ontology of Time for the Semantic web. ACM Transactions on Asian Language Processing (TALIP): Special issue on Temporal Information Processing 3 (1), 2004.
[30] G. Kappel, W. Retschitzegger, S. Rausch-Schott, M. Sakkinen: Rule Patterns – Bottom-up Design of Active Object-Oriented Databases and Application to WFMS. Comm. of the ACM (CACM), Vol. 44/4, April 2001.
[31] M. M. Kokar, C. Matheus, K. Baclawski: Ontology-based Situation Awareness. J. of Information Fusion (10), 2009.
[32] B. Krieg-Brückner, U. Frese, K. Lüttich, C. Mandel, T. Mossakowski, R. J. Ross: Spatial Cognition IV. Reasoning, Action, and Interaction, chap. Specification of an Ontology for Route Graphs, Springer, 2005.
[33] W. Kurschl, S. Mitch, J. Schönböck: Model-Driven Prototyping Support for Pervasive Healthcare Applications. Pervasive and Smart Technologies for Health Care: Ubiquitous Methodologies and Tools, 2010
[34] W. Kurschl, S. Mitch, J. Schönböck: Modeling Distributed Signal Processing Applications. Proc. of Int. Workshop on Body Sensor Networks, 2009.
[35] W. Kurschl, S. Mitch, J. Schönböck: Modeling Wireless Sensor Networks based Context-Aware Emergency Coordination Systems. Proc. of Int. Conference on Advances in Mobile Computing & Multimedia, 2008.
[36] J. Li et al.: FollowMe: On Research of Pluggable Infrastructure for Context Awareness, Proc. of the Int. Conf. on Advanced Information Networking and Applications, 2006.
[37] J. Llinas et al.: Revisiting the JDL Data Fusion Model II, Proc. of Int. Conf. on Information Fusion, Sweden, 2004.
[38] R. Müller et al.: AgentWork: A Workflow System Supporting Rule-Based Workflow Adaption, Data & Knowledge Engineering, 51, 223-256, 2004.
[39] I. Niles, A. Pease: Towards a Standard Upper Ontology. Proc. of Int. Conf. on Formal Ontology in Information Systems, 2001.
[40] O. Raiman, Order of Magnitude Reasoning, Artificial Intelligence 51 (1-3), 1991.
[41] M. Ragni, S. Wölfl: Temporalizing Cardinal Directions: From Constraint Satisfaction to Planning. Proc. of 10th Intl. Conf. on Principles of Knowledge Representation and Reasoning, AAAI Press, 2006.
[42] W. Reisig, G. Rozenberg:Lectures on Petri Nets I: Basic Models - Advances in Petri Nets. chap. Informal Introduction to Petri nets, Springer, 1998.
[43] C. Richardson et al.: Forrester TechRadar for B&A Pros: Business Process Management Suites, Forrester Research, 2009.
[44] M. Rosemann, J. Recker: Context-Aware Process Design: Exploring the Extrinsic Drivers for Process Flexibility, Proc. of the 7th Workshop on Business Process Modelling, Development, and Support, 2006.
[45] M. Rosemann, J. Recker, Ch. Flender, P. Ansell: Understanding Context-Awareness in Business Process Design, Proc. of Australasian Conf. on Information Systems, 2006.
[46] K. Shin et al.: A Workflow Language for Context-Aware Services, Proc. of the Int. Conf. on Multimedia and Ubiquitous Engineering, 2007.
[47] S. Smanchat et al.: A Survey on Context-Aware Workflow Adaptations, Proc. of Conf. on Advances on Mobile Computing & Multimedia Mobile, Linz, 2008.
[48] R. T. Snodgrass: Developing Time-Oriented Database Applications in SQL. Morgan Kaufmann Publishers, Inc., USA, 1999.
[49] W. M. P. van der Aalst, A. H. M. ter Hofstede: Yawl: Yet Another Workow Language. Information Systems 30(4), 2005.
[50] M. Wieland et al.: Towards Context-Aware Workflows, Proc. of the CAiSE'07 Workshops and Doctoral Consortium, 2007.
M. Wieland et al.: Context Integration for Smart Workflows, Proc. of IEEE Int. Conf on Pervasive Computing and Communications, Hong Kong, 2008.