An Approach for Knowledge-Based IT Management of Air Traffic Control Systems

Fabian Meyer, Reinhold Kroeger
RheinMain University of Applied Sciences
Distributed Systems Lab
D-65195 Wiesbaden, Germany
{firstname.lastname}@hs-rm.de

Ralf Heidger, Morris Milekovic
Deutsche Flugsicherung GmbH
Systemhaus Langen
D-63225 Langen, Germany
{fistname.lastname}@dfs.de

***Abstract*—In this paper, an approach for runtime analysis and automated knowledge-based IT management is presented and applied to the example of an Air Traffic Control (ATC) system. An analysis method has been developed, which combines the strengths of formal ontologies and Complex Event Processing (CEP). Thereby, the configuration of the whole analysis process is derived from one homogeneous knowledge source, a formal ontology model, which contains the system topology, the events that occur in the system, aggregations and Service Level Agreements (SLAs). Ontology rules are evaluated using semantic reasoning, event aggregations are translated into rules for the CEP engine. During runtime, data from the *Managed System* is classified to be either low-frequent or high-frequent and is accordingly either mapped to the ontology or used as input for the CEP engine. The aggregated information from both processes is used as input for the planning of reconfigurations on the Managed System.**

I. Introduction

Modern Air Traffic Control (ATC) systems are organized as large distributed IT applications, consisting of many differnt components (primary system, secondary backup system, interfaces to external systems, radars, trackers, adapters, workstations, etc.) and information models (software models, system configurations, message formats, etc.). From the raise of radar data, data transfer, data processing, right up to the visualization for the controllers, IT is ubiquitous. Not just the functionality but also the Quality of Service (QoS) plays a significant role in the ATC context. Service Level Agreements (SLAs) based on different Service Level Indicators (SLI), such as maximum time to delivery, quality of the tracking process (false tracks, track drops) and fault tolerance, have to be met to ensure a continuously fault-free and well performing operational process. Therefore, administrators with domain specific knowledge have to monitor and analyze the internal state of the *Managed System* constantly and pro-actively plan reconfigurations. Considering the growing complexity of the setup of ATC systems, IT management has become a steadily-growing, time- and cost-intensive task.

Similar problems also exist in other domains. IT applications and the underlying infrastructure have become a significant share for the fulfillment of business goals for companies. The execution of business processes relies on services offered by large IT systems, which in general have grown over years and consist of many heterogeneous components from different vendors. The complexity of the Managed System and the requirements on reliability and robustness of the services are a time-consuming challenge for IT management. Therefore, companies have a high demand for intelligent, automated management tools, which are not product specific, but overarching. Tools for the management of large heterogeneous environments exist (e.g., IBM Tivoli, HP OpenView), but because of the diversity of system models and management interfaces the integration of new components is hard to achieve.

Through the advances of Semantic Web technologies in the last years, ontologies from the field of artificial intelligence have experienced a revival as domain spanning knowledge models and new standards like the Web Ontology Language (OWL) were established. Ontologies define semantics for data, so that reasoners can validate the model and derive new knowledge from existing facts, using description logics. Those characteristics make ontologies a fitting model for the combination of different IT models and thereby for the management of heterogeneous systems. Existing applications of ontologies in IT management (see chapter V) have shown that ontology-based runtime analysis is performing well for small, timeless models. But in reality, systems are much more complex, dynamic and have various timing aspects, which leads to the following problems:

1) The combined reasoning complexity of OWL ontologies is NP-complete (as shown in [1]), which makes large ontologies hard to handle for runtime management.
2) There is no concept of time in ontologies, they just reflect the state for a concrete point in time, so that versioning is required for time-based runtime analysis (as presented in [2] and [3]), which blows up the instance space in the ontology dramatically.

Especially in the context of ATC systems those two characteristics turn the exclusive use of ontologies for IT management infeasible. Mapping the runtime information of a large distributed infrastructure of components that generate thousands of events per second to an ontology makes reasoning nearly impossible. Hence, new methods are needed, combining the strength of domain spanning semantic ontologies with other technologies for the processing of rapidly changing data.

In this paper an approach is presented, which combines ontologies from the Semantic Web with Complex Event Processing (CEP) techniques, to realize an automated ontology-based system management based on the Monitor Analyze Plan Execute Knowledge (MAPE-K) Loop introduced in [4], to

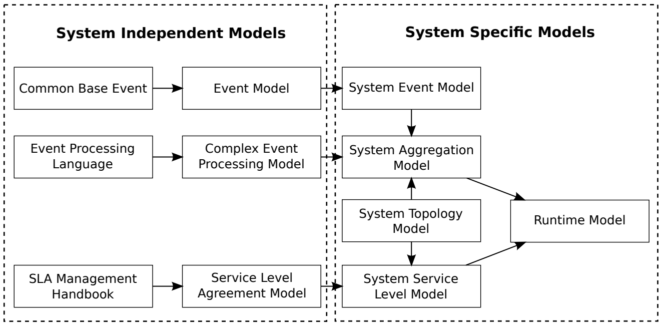

Fig. 1. Structure of the different models for the Knowledge Component of the Management System

be capable of processing data streams with high event rates. Therefore, runtime analysis, knowledge derivation, situation assessment and automated reconfiguration methods are developed.

Section II presents the new approach, Section III applies the approach to a use case from the ATC context, Section IV presents the current state of the project and the planned future work, Section V gives an overview of related work, before a summary is given in Section VI.

II. Approach

The approach presented in this section describes the combination of ontologies and CEP techniques, fulfilling the central requirement that all components of the Management System are configured by one homogeneous model, the ontology. Therefore, a software architecture based on the MAPE-K Loop was developed, where some of the MAPE-K components are melt together or are spread over several components.

The Knowledge Component is a data storage for models used in the management cycle. It is a passive component, which can be queried or updated. The stored models (as shown in Fig. 1) are used for the reasoning performed by the Analysis Component and to configure the components of the system. Domain models stored in the Knowledge Component are divided into models that are either independent of the Managed System or system specific. All those models are combined to a *Runtime (RT)*-model, which is filled with instances and assertions from the system configuration and used as initial input for the Knowledge Component.

For this approach, the following system independent models were defined: (1) The *Event (Ev)*-model defines the basic OWL structure of a system event. Therefore, the Common Base Event (CBE) [5] model is converted into an ontology. Besides payload, the CBE format also defines different meta-aspects (e.g., version, creation time, reporter component, source component). (2) The *CEP*-model defines aggregations that can be performed on the event data streams and the system model to generate new complex events. Basis for the definition of the *CEP*-model is the Statement Object Model (SODA) of the Event Processing Language (EPL) [6]. Common aggregations are arithmetic operators (e.g., minimum, maximum, average) and pattern matching (e.g., $A \wedge B \rightarrow C$). (3) The *SLA*-model defines the structure of SLAs, SLIs and QoS. The Tele Management Forum has published the SLA Management Handbook [7], which presents useful key concepts and definitions. Those concepts are mapped to a formal ontology representation, so that they can be applied to a Managed System.

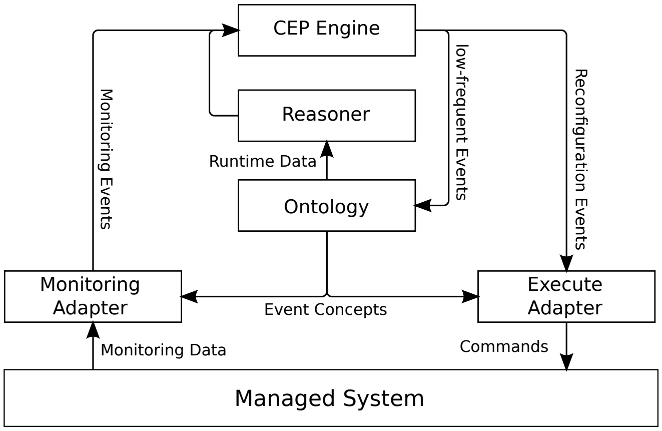

Fig. 2. Data flow between the components of the Management System.

The models presented so far are used as a basis for the defining the following system specific models: (4) The *System Topology (Sys)*-model defines the structure of the Managed System (components, attributes, relationships, configurations and runtime properties). In most cases, the topology can be converted automatically from an existing model, e.g., UML model (see [8]). If no convertible system model exists, the system has to be modeled manually. (4) The *System Event (SysEv)*-model defines the simple events, which are generated when the system performs state transition (e.g., events written as log entries to a log file). In general, there is no formal model for system events, which can be converted into an ontology representation. Hence, those events have to be modeled manually, using the *Ev*-model as basis. (5) The *System Aggregation (SysAgg)*-model uses the concepts defined in the *CEP*-model to describe the aggregation of complex events based on the simple events, defined in the *SysEv*-model in combination with the system entities described in the *Sys*-model. Therefore, aggregations are modeled, referencing either events or system entities as input and complex events as output. Furthermore, the model defines a mapping between system components and simple/complex events using Semantic Web Rule Language (SWRL).

For the Managed System, a *System Service Level Agreement (SysSLA)*-model is defined, representing the SLAs for the Managed System using the *SLA*-model. SLIs for the SLAs are references of the states of the different system components, represented by property assertions in the ontology model. There are models for the definition of SLAs as the Web Service Level Agreement (WSLA) language [9]. If existent, those models can be converted into ontologies. In general, SLAs are just defined as a paper contract between customer and supplier and hence have to be formalized by hand. Furthermore, a reconfiguration aggregation is defined for each SLA, creating an according reconfiguration event, when the SLA is broken.

The structure and data flow of the Management System is show in Fig. 2. The Analysis Component is the basis for the realization of the MAPE-K's analyze and plan components. It analyses the current state of the system, derives new knowledge, performs situation assessment and determines reconfigurations. Based on the information stored in the knowledge base the runtime analysis will be configured automatically, so that these goals can be achieved during runtime. Due to the performance problems for the exclusive use of ontologies for IT management, as described in Section I, a combined analysis mechanism has been developed. Events that describe state changes of the Managed System are classified to be either low-frequent or high-frequent and hence are treated differently for analysis.

In case of low-frequent data, the events are directly added to the ontology (corresponding event instances and property assertions are created) and a reasoning is triggered. Through the mapping between system components and events defined in the *SysAgg*-model the reasoner can evaluate further rules (e.g., assessment and reconfigurations) based on the attributes of the components, which is equivalent to other ontology-based management approached described in chapter V.

In case of high-frequent data, the events are used as input for the CEP engine. The engine is configured (rules are created) using the aggregations defined in the *SysAgg*-model. Since the CEP engine should not just use the event data but also the data stored in the knowledge base, a mechanism is needed to enrich the analysis process. Therefore, a dynamic lookup method is used, extending the event processing engine for knowledge views, which perform a lookup on the knowledge base when executed. Hence, queries can be formulated generically and instance-specific data is aquired from the knowledge base during runtime.

The Adapter Component is the basis for the realization of the MAPE-K's monitor and execute components. Its task is the translation of system specific information to the event format of the Management System and the other way around. Therefore, the implementation of the Adapter Component is specific for each Managed System, so that the corresponding management interface can be used. For each type of monitoring data a mapping to the corresponding event of the *SysEv*-model is defined. The mapping is applied to all monitoring records retrieved from the system, and events are created and fed into the Management System. For each event retrieved from the Management System a mapping is defined, which maps the event to a command, executed on the Managed System.

The monitoring adapter reads the monitoring data from an interface of the Managed System and translates it into input events of the Management System, using the event definition of the knowledge base. Those events are sent to the Analysis Component, where they are used as input for the CEP engine. The engine is configured with the aggregation rules, which are defined in the knowledge base and use the events and the runtime information from the knowledge base for the derivation of new complex events. All output events from the engine, classified as being low-frequent, lead to an update of the knowledge base, triggering the semantic reasoning. Reconfiguration events are received by the Adapter Component and translated into commands on the Managed System.

III. Application on the ATC System

As mentioned in Section I, the context of the project is the automated knowledge-based management of an ATC system. Hence, in this section the application of the approach on a specific use case of the Air Traffic Management system PHOENIX (see [10]) of Deutsche Flugsicherung GmbH is presented: The monitoring of the SLIs of radar systems, the assessment of their QoS based on its *Probability of Detection (POD)* (high *POD* $\equiv$ well performing radar) and the reconfiguration of the system, if the SLAs are not met anymore. Therefore, a *Sys*-model and an *Ev*-model are created, aggregations, SLAs and reconfiguration actions are defined and system adapters are implemented. For a better understanding, models used in this example are severe simplifications.

The definition of the ATC-specific *Sys*-model is derived from the XML-based PHOENIX configuration schema, which is a conceptual definition of the different components. Needed parts of the schema are automatically transformed into an ontology model, using a developed model converter. The resulting ontology contains the following concepts: A *radar* has an *id*, a *position* in latitude/longitude coordinates and a *range* in nautical miles. Radars are assigned to *tiles* by a *radarAssignment* property. A *tile* represents an observation area on the map. It is defined by an *origin* and a *dimension* in latitude/longitude coordinates as well as an *id*. Events that occur in the ATC system are *POD*-events, generated by a statistic component of the ATC system for every measurement of a *radar* as described in [11], and *Radar Assignment (RA)*-events, which are commands for the association of *radars* to *tiles*.

Based on the *Sys*-model, the *SysAgg*-model and *SLA*-model are defined. Therefore, the model is extended by the following aspects: (1) An *Average POD (AvgPOD)*-event, representing the complex event of the average *POD* for each *radar*. It is aggregated by the average value of the *POD*-event for each *radar* over ten minutes. (2) A *Prone Radar (PRad)*-event, representing the complex event of a *radar* malfunction. It is raised if two following *AvgPOD*-events for a single *radar* violate an defined threshold. (3) A *Prone Tile (PTile)*-event, representing the complex event that a *tile* is not configured sufficiently for operational service. It is raised if a *tile* has not a *radarAssignment* of at least two non-prone *radars*. (4) A *potentialRadar* property, representing potential relations between *tiles* and *radars*. (5) A SWRL rule, stating that a *radar* can potentially be assigned to a *tile*, if the area of the *tile* is completely covered by the *radar* and it is not yet assigned to the *tile*. A reconfiguration aggregation is defined, which generates a *RA*-event, if a *tile* is prone and there are other potential *radars* for the *tile*.

IV. Status

The modeling of the ontologies is partially done. The software has been implemented prototypically ,using the OWL API for ontology handling, the Pellet OWL reasoner for ontology reasoning, and the Esper CEP engine for event processing. The communication of the ATC system is realized by a message-oriented middleware. To extend the system by a new process, the API of the middleware can be used to subscribe to message topics (monitoring) or to send messages to other processes (commanding). Therefore, an adapter process was implemented, using the middleware to serve as a bridge between the ATC and the Management System. On the ATC side, the adapter process subscribes to the *Radar Status (RS)*-messages, which are translated into *POD*-events for the internal use. On the Management System side the adapter receives *RA*-events, which are translated into *Radar Assignment (RA)*-command for the ATC system. The prototype was integrated into the ATC's test environment to give a proof of concept for the automated adaptation of the aggregations modeled in the ontology, the handling of runtime data and the accessing of the ontology during query execution.

Complete versions of the models, the implementation of the generic framework and the wider evaluation of the concept are the next steps. As a basis for the implementation of the architecture the OSGi component framework will be used, which offers a loosely coupled, service-based inter-component communication.

V. Related Work

Ontologies for IT management are not yet widely used, but there are several publications, which consider the topic. In [12], a mapping of the Structure of Management Information (SMI), the Guidelines for the Definition of Managed Objects (GDMO), Managed Object Format (MOF) and the Common Information Model (CIM) to OWL is defined. The resulting ontology models are combined to a joint model, so that products of different vendors can be managed together.

[13] describes how several abstraction layers of a system are split into a hierarchical structure of ontologies, where often used ontologies are at the bottom of the hierarchy. The reuse of components and models is an important topic in IT systems, and especially for ontology-based automation. The paper shows that OWL is capable of organizing several abstractions of a system in ontologies and reuse defined components in higher layers.

A real-world management application on ontology-based IT management is shown in [14] where ontologies are used to manage a network infrastructure, based on Policy-based Network Management (PBNM) [15]. SWRL rules, evaluated periodically during runtime, are defined to reconfigure the system in case of a blackout. A management component observes the ontology and maps newly added facts to management operations to adjust the system. In a case study, the system is applied to a backup service of a network provider. In case of a broken connection new resources are assigned. The authors state that the developed concept can be applied to other IT management problems easily.

[16] presents an architecture for automated knowledge-based IT management. A formal ontology model is used as knowledge component. Monitoring data from the Managed System is mapped to instances and assertions of the formal model and new knowledge is derived using semantic reasoners (exact reasoning) and Bayesian networks (probabilistic reasoning). Based on the results of the analysis phase, management decisions are made and executed as commands on the Managed System.

The combination of ontologies and data stream mining was also addressed in some publications. In [17], an approach for a context-aware data mining framework is presented, where context information is modeled using an ontology. During the data mining process the model is accessed through knowledge operators in the event processing language. The events in the data stream are not part of the domain model and the aggregation rules are defined externally.

An architecture for a knowledge-based data mining framework is presented in [18]. The framework allow the connection of different mining operators into a dynamic mining network. A Domain Specific Language (DSL) is developed for the configuration of the network, which also includes so called tagging operators that extend the data stream by information acquired from the knowledge base. The structure of the events is part of the DSL, but no formal mapping to the topology of the underlying Managed System is defined.

Panov et. al. present an ontology for data mining in [19] called OntoDM. The ontology contains concepts for data mining specific datatypes, datasets, tasks, algorithm and components, but no automated translation to rules for CEP engines or connections to the context entities are defined. Some of the presented publications consider the combination of data mining and ontologies, but none of them uses a formal model, containing the system topology, the events and aggregations, to automatically derive the configuration for a combined runtime analysis.

VI. Conclusion

In this paper, an approach for automated knowledge-based IT management was presented. Ontologies from the Semantic Web and their reasoning capability were combined with CEP techniques to achieve a new anylsis method, solving the problem of time modeling and state space explosion, in contrast to the exclusive use of ontologies. Therefore, a semantic model is defined, containing a topology model, system events, aggregations and SLAs, which is then used as a basis for the derivation of rules for runtime event aggregation in the CEP engine.

During runtime, monitoring information from the system is divided into low-frequent and high-frequent data, and hence either being used as input for the ontology or for the CEP engine. For the evaluation of the complex event aggregation rules, the events as well as the information stored in the ontology are used. Reconfiguration events, generated by the engine, are translated into system reconfiguration commands, which are executed on the Managed System.

The approach was applied to a use case from the management of an ATC system. An example was presented, where an automatic radar reconfiguration was performed, based on a formal system model of the ATC system and its low level events.

It has been shown that the combination of ontologies and CEP facilitates an analysis method, relying on the strong semantics of ontologies to describe the aspects of heterogeneous systems in a homogeneous way, but is also capable of handling runtime data with high events rates.

REFERENCES

[1] B. Motik, B. C. Grau, I. Horrocks, Z. W. A. Fokoue, and C. Lutz. (2012) Owl 2 web ontology language profiles (second edition). [Online]. Available: http://www.w3.org/TR/owl2-profiles/

[2] C. Welty, R. Fikes, and S. Makarios, “A reusable ontology for fluents in OWL,” *Frontiers in Artificial Intelligence . . .* , 2006.

[3] J. Bao, L. Ding, and D. McGuinness, “Semantic history: Towards modeling and publishing changes of online semantic data,” *The 2nd Social Data on the Web . . .* , 2009.

[4] IBM Corporation. (2006, June) An Architectural Blueprint for Autonomic Computing, Technical Whitepaper (Fourth Edition).

[5] D. Ogle, H. Kreger, and A. Salahshour, “Canonical Situation Data Format: The Common Base Event V1. 0.1,” *IBM Corporation*, 2004.

[6] EsperTech, “Esper Reference,” EsperTech, Tech. Rep., 2012.

[7] Tmforum, “SLA Management Handbook,” *Tele Management Forum*, 2012.

[8] D. Gasevic, D. Djuric, V. Devedzic, and V. Damjanovi, “Converting uml to owl ontologies,” in *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, ser. WWW Alt. ’04. New York, NY, USA: ACM, 2004, pp. 488–489.

[9] H. Ludwig, A. Keller, A. Dan, R. King, and R. Franck, “Web service level agreement (WSLA) language specification,” *IBM Corporation*, pp. 1–110, 2003.

[10] R. Heidger, “The PHOENIX White Paper V2.0,” 2009, publicly available on request.

[11] K. Engels and R. Heidger, “An infrastructure for online tracking quality control,” *2008 Tyrrhenian International Workshop on Digital Communications - Enhanced Surveillance of Aircraft and Vehicles*, pp. 1–7, Sep. 2008. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4649053

[12] J. E. L. D. Vergara, V. A. Villagr, and J. Berrocal, “Applying the web ontology language to management information definitions,” *IEEE Communications Magazine*, vol. 42, pp. 68–74, 2004.

[13] J. E. L. De Vergara, A. Guerrero, V. A. Villagrá, and J. Berrocal, “Ontology-Based Network Management: Study Cases and Lessons Learned,” *Journal of Network and Systems Management*, vol. 17, no. 3, pp. 234–254, September 2009.

[14] V. A. Villagr and J. E. L. D. Vergara, “Ontologybased policy refinement using swrl rules for management information definitions,” in *in OWL. In: Proc. 17th IFIP/IEEE International Workshop on Distributed Systems, Operations and Management (DSOM*, 2006, pp. 227–232.

[15] A. Westerinen, J. Schnizlein, J. Strassner, M. Scherling, B. Quinn, S. Herzog, A. Huynh, M. Carlson, J. Perry, and S. Waldbusser, “Terminology for policy-based management,” United States, 2001.

[16] A. Textor, F. Meyer, and R. Kroeger, “Semantic Processing in IT Management,” in *Proceedings of the Fifth International Conference on Advances in Semantic Processing (SEMAPRO)*, P. Lorenz and E. Ammann, Eds., Lisbon, Portugal, November 2011.

[17] S. Singh, P. Vajirkar, and Y. Lee, “Context-based data mining using ontologies,” in *Conceptual Modeling - ER 2003*, ser. Lecture Notes in Computer Science, I.-Y. Song, S. Liddle, T.-W. Ling, and P. Scheuermann, Eds. Springer Berlin Heidelberg, 2003, vol. 2813, pp. 405–418.

[18] A. Textor, F. Meyer, M. Thoss, J. Schaefer, R. Kroeger, and M. Frey, “An Architecture for Semantically Enriched Data Stream Mining,” in *Proceedings of the First International Conference on Data Analytics*, S. Bhulai, J. Zernik, and P. Dini, Eds., Barcelona, Spain, September 2012.

[19] P. Panov, S. Dzeroski, and L. Soldatova, “OntoDM: An ontology of data mining,” *Data Mining Workshops, . . .* , 2008.