An Ontology-Based Approach for Defining Compliance Rules by Knowledge Workers in Adaptive Case Management

A Repair Service Management Case

Thanh Tran Thi Kim, Erhard Weiss, Alexander Adensamer, Christoph Ruhsam
ISIS Papyrus Europe AG
Maria Enzersdorf, Austria
{thanh.tran, erhard.weiss, alexander.adensamer, christoph.ruhsam}@isis-papyrus.com

Christoph Czepa, Huy Tran, Uwe Zdun
Faculty of Computer Science
University of Vienna
Vienna, Austria
{christoph.czepa, huy.tran, uwe.zdun}@univie.ac.at

***Abstract*—Empowering knowledge workers (KWs) to act more efficiently and flexibly in unpredictable situations is the main focus of Adaptive Case Management (ACM), although ad hoc actions at runtime shall not violate the consistency and compliance of an on-going case. In this paper we discuss how business constraints stemming from regulatory laws or standards are transferred from textual sources to formal specifications in the form of compliance rule objects relating to ACM objects. In order to mitigate the knowledge barriers between business and IT, we apply an ontology-based solution allowing KWs to define compliance rules from a pure business perspective using domain-specific terms. We demonstrate the implementation using a repair service case and discuss the benefit of our approach for the sake of administration of business aspects by KWs without IT involvement.**

Keywords—ontology definition; compliance rules; consistency checking; Adaptive Case Management

I. INTRODUCTION

In this paper, we discuss the challenges of managing compliance rules in Adaptive Case Management (ACM) and propose an ontology-based approach for business users acting as knowledge workers (KWs) to handle compliance rule definitions independently from IT developers.

ACM empowers KWs to work on goals without being bounded by specific predefined process paths [11]. In contrast to rigidly modelled processes, as supported by most of today's Business Process Management (BPM) systems, ACM supports ad hoc changes at runtime which enable KWs to deal with unplanned situations [11]. In our previous work [2, 3, 12, 14] we have addressed the structural as well as behavioral consistency of ad hoc actions at runtime, which is a challenge in ACM systems as the process paths are unknown when the process is started. Ad hoc actions executed by KWs are caught at runtime and verified against a set of compliance rules related to ACM cases to inform about possible violations. Consistency checking applies also at design time when ACM administrators define templates for specific situations.

Compliance rules discussed in this paper are used to reject or even avoid wrong business transactions caused by ad hoc actions taken improperly. All execution events of the ACM application are monitored and analyzed at runtime, so that the system can evaluate the risk of every ad hoc action. The application of compliance checking based on Complex Event Processing (CEP) [5] in the context of ACM can ensure that ad hoc actions are in conformance with the compliance needs of a company.

Business rules and specifically compliance rules in our context are declaratively expressed as sentences in a natural-language-like syntax by business users of a policy department having comprehensive business-specific knowledge. As ACM aims to deliver self-management capabilities for KWs, they should be able to manage the rule definition process with a natural-language-like syntax for the translation into a formal specification and the deployment of rule objects in the ACM system. However, the barrier between business and IT knowledge limits KWs today in handling compliance rules independently from IT developers. Based on the goals for agile business rule development defined in [9], we analyzed the challenges and propose an approach for compliance rule management in ACM as follows:

- *Analyze rules in textual sources and break them into atomic rules:* KWs, particularly in this case are Business Administrators for rule management, discover compliance rules from textual sources and define atomic rules to facilitate the comprehension of rules and the maintenance of the rule collection independent from IT developers.
- *Define rules in business language with a compliance rule editor:* The system should allow KWs to define their business terms through a business-specific-domain ontology. The rule editor interacts with the ontology to specify compliance rules.
- *Connect compliance rules to ACM elements:* KWs assign rules to single ACM entities like cases, goals, or an entire ACM application. The scope of the rules

should be flexibly defined with different levels of impact.

- *Redundancy and contradiction checking:* The system should supply functionalities to check for conflicting rules and redundancies.
- *Implement and deploy a rule set:* A compliance checker integrated in the ACM system controls the rule conformance of performed ad hoc actions on the fly.

Fig. 1 shows an overview of our approach. (i) Domain specific business ontologies are defined by KWs to represent their business model and used terminology in an optimal way. (ii) ACM administrators map with the underlying ACM ontology and the used data model classes. This way, the semantic information is mapped with the structural information of the Papyrus ACM Framework so that ontology queries can retrieve directly the affected data objects. The compliance rule editor accesses the domain specific ontology allowing KWs to transform the compliance specification from a textual source into the compliance rules constraint language. The compliance rule editor is based on an inverse parser concept using a compliance rule specific grammar with temporal and logical operators to combine the data objects referred to by the defined ontology. KWs do not have to learn the grammar as the editor's inverse parsing process also knows about the current context in terms of ontology semantics and thus, can offer in an auto completion style only the correct choices to the users at each point of rule creation. Internally the rules are stored in a table format referencing the ontology items directly. Nesting of constraints is handled by a tree representation. (iii) These rules are assigned by KWs to ACM templates and, after deployment, automatically enacted on the affected ACM cases at runtime. Rules in the constraint language are transformed to an Event Processing Language (EPL) [5] before input to a rule engine for on the fly compliance checking. The implementation of our approach is demonstrated by a repair service management

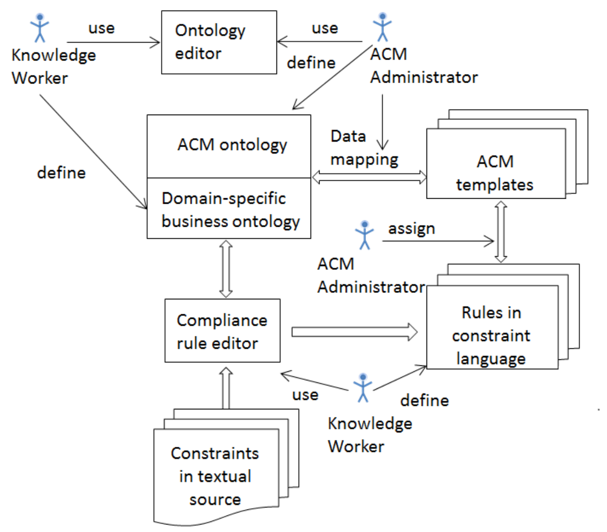

Fig. 1. The integration process of compliance rules in ACM

solution built with the ISIS Papyrus ACM framework [8]. The use case describes the rule definition by KWs starting from textual sources until the application of compliance rules on ad hoc actions associated with the repair service case.

This paper is organized as follows: Section 2 analyzes the challenges and introduces our approach addressing them. A case study described in Section 3 shows the application of our approach and is also used to discuss implementation aspects of the approach. Section 4 discusses the approach, its benefit, and outlines future work and Section 5 concludes the paper.

II. Challenges and Approach

A. *Create compliance rules in a constraint language using a compliance rule editor*

Business rules are mostly expressed in textual format for a business audience having comprehensive knowledge of a business-specific domain [7]. These texts are usually analyzed and translated into a rule specification in a format that is machine readable by using specialized tools such as business rule languages, business rule engines and business rule applications. This work needs IT specialist skills and is not suitable for business users. To eliminate these obstacles, we propose a constraint language that enables KWs to define compliance rules without involving IT developers. The constraint language delivers a simple grammar and a set of relation patterns to specify business rules in a way that is easily understandable for KWs.

Compliance rules considered in our approach are composed of event occurrences and temporal operators. Events exported from ad hoc actions are caught continuously at runtime and used as input data for compliance checking based on the Complex Event Processing (CEP) technique [2, 3, 5, 12]. Thus, the compliance rules are a combination of temporal operators, event states of business entities and data-related conditions.

The constraint language includes a set of temporal patterns used for event processing based on the patterns defined by Dwyer et al. [10]. The main temporal operators used in our approach are:

- *occurs*: Implies the existence of an event. For example, *TaskA.finished occurs* describes the occurring of an event when task A is finished.
- *never occurs*: Implies that an event never happens.
- *leads to*: Implies a response of an event when another event happens. For example, *TaskA.finished leads to TaskB.started*.
- *precedes*: Implies an existence condition of an event when another event happens. For example, *TaskA.finished precedes TaskB.started*.

Compliance rules are classified into two types: state-based rules and data-based rules.

State-based rules relate to states of ACM entities, such as goals and tasks. For example, a business regulation about payment can be specified as *A customer needs to finish payment before he can receive the product*. The temporal pattern *precedes* can be used to specify a temporal precedence relation in our constraint language as follows:

Payment.finished precedes Shipment.started

If a Clerk, a kind of KWs, tries to execute the task *Shipment* at runtime, the consistency checking will notify the KW that the rule would become permanently violated when the task *Payment* is not yet finished.

Data-based rules describe constraints related to conditions based on data values. For example, the rule *Customers living in Europe can only receive products with CE marking* can be specified in the constraint language as:

Product.Mark not equal to CE and Customer.Region equals EU and Shipment.started never occurs

B. Knowledge gap between business and IT-specific domains

Transferring business regulations into a business system is a knowledge migration process from business domain to IT domain. Business users understand the regulations involved in their business whilst IT developers comprehend software engineering techniques and implement business regulations in applications. The challenge is building the bridge between these two domains so that business users (KWs) can define compliance rules independently from IT developers.

Along with using a grammar close to the natural business language of KWs, we use domain specific ontology definitions to support the business users while creating compliance rules. Business Administrators define the ontology describing their business-specific domain and map to the ACM ontology which links ACM objects with entities on the system layer. In our approach we describe the repair service management scenario with a repair service ontology as basis for the compliance rules described in our case study.

A compliance rule editor is implemented in the Papyrus ACM system (see Fig. 2). We define the ontologies of the ACM framework and business-specific domains using the ontology editor of the Papyrus system. As explained in Fig. 1 these ontologies are mapped by ACM Administrators to the underlying system objects to create the connection between conceptual and instance levels. The compliance rule editor retrieves the concepts defined in the business ontology and expresses the rule with the grammar defined for the constraint language. With the rule editor, KWs make use of the concepts (i.e. business terms) to create compliance rules in a business constraint language. These rules are the input to a compliance checker for on the fly checking of ad hoc actions.

C. Integration of compliance rule management in ACM

We apply the compliance rule management in combination with consistency checking to ensure the consistency of ad hoc actions in ACM. Each rule discovered from regulative sources is expressed as rule object residing in a compliance rule collection to extend the ACM library with its case, goal or task templates. A rule is composed of several ontology concepts. These concepts represent single or groups of ACM entities that have common features. Based on the references between ontologies and underlying system objects, the rule influences the execution of these objects at runtime.

The scope of a rule is defined in the rule expression at design time by the KW's assignment of rules to specific ACM entities which must be affected by the compliance regulations.

D. Self-management capability of KWs in ACM

We implement an ontology editor including graphical views of ontology diagrams and functions for ontology management so that KWs can define and arrange ontologies by themselves. A rule editor using the ontology-based approach assists KWs to independently create and maintain a rule collection in ACM without IT involvement. This approach supplies KWs with the needed tools to handle the compliance rules, and thus the compliance of ad hoc actions, that are essential for the strength of an ACM system.

We present the implementation of our proposal through a case study and describe the lifetime of a rule from the beginning, when the rule is discovered from regulation sources, until the rule is triggered at runtime.

III. CASE STUDY

The scenario of repair service management was introduced in our previous work [12]. In this paper we present the ACM implementation details of the ontology-based approach applied

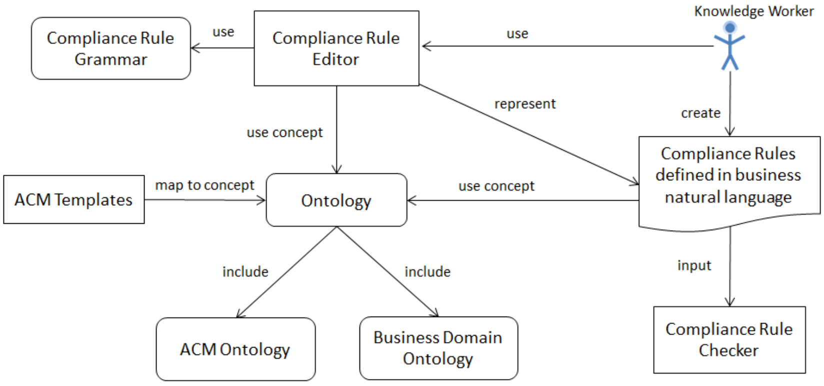

Fig. 2. Compliance rule editor using ontologies

to the United States Environmental Protection Agency's (EPA) Lead Renovation, Repair and Painting Rule (RRP Rule) to show the extended capabilities of KWs through compliance rule management.

A. Use case story

"EPA's Lead Renovation, Repair and Painting Rule (RRP Rule) requires that firms performing renovation, repair, and painting projects that disturb lead-based paint in homes, child care facilities and pre-schools built before 1978 have their firm certified by EPA (or an EPA authorized state), use certified renovators who are trained by EPA-approved training providers and follow lead-safe work practices." [4]

According to the RRP rule, EPA's lead pamphlet must be distributed to the owner and occupants of the affected building and to parents or guardians of children when used as child-occupied facility. A confirmation of receipt or certificate of mailing must be retained for three years following the completion of renovation.

The following sections represent the application implementation in the ISIS Papyrus ACM system [8].

B. ACM configuration

1) Ontology definitions

The ontologies described in the case study are created with an ontology editor integrated into the ACM system. The ontology editor supports KWs in defining concepts and relations as well as the mapping of the concepts to the ACM data objects. The business users can create or remove concepts or relations directly in the ontology diagram. They can also define attributes of concepts or relations in the item details frame. Moreover, different colors (also highlighted as numbers in Fig. 3) and graphical representations provide an interactive interface for business users to create and edit ontologies.

The repair service ontology facilitates the mapping of the domain-specific concepts to the underlying ACM ontology and system data models. The ACM ontology (in green marked with "1" in Fig. 3) describes the elements of ACM and the relations between those elements expressing the common understanding of an ACM system and how an ACM case can be built and executed in the system. The following are the main concepts and relations used in repair service management based on ACM.

Concept *Case* is a container for all ACM elements related to a business case. Concept *Goal* represents a defined achievement that should be gained eventually. Concept *Task* is an activity executed within a case to fulfill a goal or parts of a goal. Attribute *State* represents the state of related concepts at a particular point of time, such as *active*, *processing*, *reached* or *failed*. Concept *Artifact* is a placeholder for content of a case. Concept *Data* belongs to an artifact and represents the information of a case. Relation *Case has Goal* implies that the case is executed by a KW as a Clerk and driven by goals. Relation *Case has Artifact* implies that a case has content, and the relation *Artifact has Data* refers to artifacts that contain data.

The *Repair Service Management* ontology is specified in two ontologies: the *Repair Object* ontology describes the information for a repair service case and the *Renovation & Repair* ontology contains the concepts of activities for a repair service case.

The *Repair Object* ontology (in magenta marked with "3" in Fig. 3) contains the concept *Owner* representing a person or institution owning an object. Concept *Tenant* represents a person or institution occupying space in an object. Concept *Child* refers to an under-aged person occupying space in an object. Concept *Guardian* is a person acting as the official guardian of a child. Concept *Address* is the postal address of a location. Concept *Repair Object* represents the building or structure to be repaired or renovated. Concept *Housing or child occupied facility* refers to the building or portion of a building with residential lodging or being visited regularly by children under 6 years of age. It is implemented as sub-concept of the *Repair Object* where children are living or it is a family house. Relation *Child is guarded by Guardian* represents that a person is being guarded by a person having parental authority. Relation *Owner has Address* expresses that an owner has a postal address. Relation *Tenant has Address* and *Guardian has Address* expresses the same for tenants and guardians. Relation *Repair Object has Owner* represents that a site is being owned by a person or institution. Relation *Repair Object has Child*

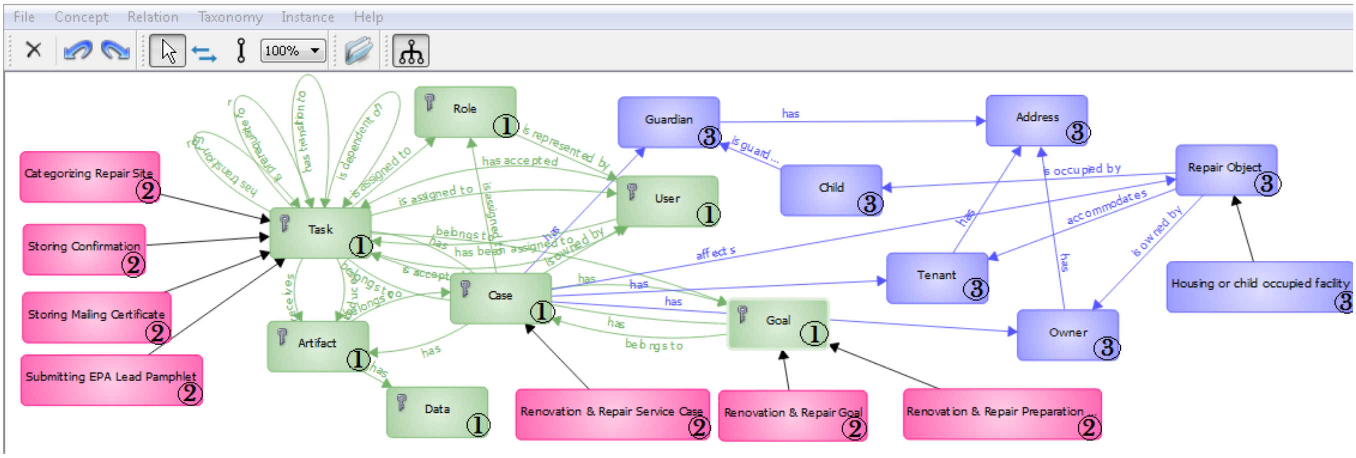

Fig. 3. Ontology diagram created by an ontology editor

considers that a repair object or site can be occupied by an under-aged person. Relation *Case affects Repair Object* expresses that the affected repair object or site is handled by a case. Attribute *Category* of concept *Repair Object* categorizes the repair site, object or structure. Attribute *Year of Completion* of concept *Repair Object* contains the information about the year of completion of the building or structure. Attribute *Street Address* of concept *Address* contains the postal address.

The main concepts and relations of the *Renovation & Repair* ontology (in pink marked with “2” in Fig. 3) include the concept *Categorizing Repair Site* to represent the categorizing activity for a repair object to evaluate the risk of being affected by lead when used for lodging or child-occupation. Concept *Submitting EPA Lead Pamphlet* represents the task submitting the EPA lead pamphlet, for example to an owner of a building. Concept *Renovation & Repair Service Case* represents the business case for the renovation and repair service, when receiving a request from a customer. Concept *Renovation & Repair Preparation Goal* defines the condition to finish the preparation phase whereas the concept *Renovation & Repair Goal* represents the condition to finish the repairing phase of the case *Renovation & Repair Service*. Concept *Renovation & Repair Aftercare Goal* represents the condition to finish the aftercare phase of the case *Renovation & Repair Service*.

The next section represents an integration of these ontologies to describe how a repair service case is to be executed in an ACM system for repair service management.

2) Ontology mapping

Table I maps the different domain ontologies that are highlighted by different colors and numbers in Fig. 3 and thus, served as a knowledge bridge between two domains.

TABLE I. ONTOLOGY MAPPING

Renovation & Repair Ontology (2,3)	***ACM ontology (1)***
Renovation & Repair Service Case	*Case*
Renovation & Repair Preparation Goal, Renovation & Repair Goal	*Goal*
Categorizing Repair Site, Submitting EPA Lead Pamphlet, Storing Confirmation of Receipt, Storing Mailing Certification	*Task*
Owner, Tenant, Child, Guardian, Address, Repair Object	*Data*

3) ACM templates

Application developers build the underlying data model from classes to provide the functionality of the repair service management.

ACM administrators build the Repair Service Management Application in the ACM template library. As ad hoc actions are in focus to adapt with the unforeseeable situations that may happen in this business, single task templates are prepared and no predefined processes are needed.

The Repair Service case is driven by three goals representing three phases:

- Goal *Renovation and Repair Preparation* contains the condition to finish the preparation phase of a repair service case.
- Goal *Renovation and Repair* indicates the completion criteria to achieve the repair service work itself.
- Goal *Renovation and Repair After-care* contains the completion criteria of post-processing and follow-up work.

The actions being added on the fly by Clerks in our example are: Task *Categorizing Repair Site*, Task *Submitting EPA Lead Pamphlet*, Task *Storing Confirmation of Receipt* and Task *Storing Mailing Certification*.

Data objects containing the content of this case are *Repair Object*, *Owner*, *Child*, *Tenant*, *Guardian* and *Address* which will be filled during run time with the related information. This can be done by the Clerk and/or through service tasks connecting to backend systems.

4) Mapping Ontology definitions to ACM elements

The mapping of ontology definitions to ACM elements is done by the ACM Administrator who comprehends the ACM and the business application domains, see Table II.

TABLE II. MAPPING CONCEPTS TO ACM OBJECT ELEMENTS

Ontology concepts	***ACM elements***
Concept *Renovation & Repair Service Case*	Case *Repair Service*
Concept *Renovation & Repair Preparation Goal*	Goal *Renovation and Repair Preparation*
Concept *Renovation & Repair Goal*	Goal *Renovation and Repair*
Concept *Renovation & Repair Aftercare Goal*	Goal *Renovation & Repair Aftercare*
Concept *Categorizing Repair Site*	Task *Categorizing Repair Site*
Concept *Submitting EPA Lead Pamphlet*	Task *Submitting EPA Lead Pamphlet*
Concept *Storing Confirmation of Receipt*	Task *Storing Confirmation of Receipt*
Concept *Storing Mailing Certification*	Task *Storing Mailing Certification*
Concept *Tenant*	Data object *Tenant*
Concept *Guardian*	Data object *Guardian*
Concept *Address*	Data object *Address*
Concept *Child*	Data object *Child*

The relations between concepts are defined by relation paths between the underlying data objects.

C. Rule management

1) Rule analysis

To ease the understanding and reduce the effort of implementation and maintenance, compliance rules should be atomic, if possible.

The EPA rule for the repair service is analyzed as follows:

- The category of the repair site will be evaluated in the task *Categorize Repair Site*. (If the site is built before 1978 and if it is a housing or child-occupied facility).
- If the repair disturbs paint and the repair site is critical, then the following compliance rule will be affected.

When the object Address of the repair site's Owner, Tenant or Child's Guardian contains data the tasks Submitting EPA lead pamphlet and Storing Confirmation of Receipt or Storing Mailing Certificate must be finished for these locations before any task of the goal Repair performed may be started.

2) Rule definitions

The compliance rule editor supports the proper rule syntax with an auto completion feature during the composition of a compliance rule. The semantics of rules in a certain business context are defined through the relations and concepts of the used ontologies. Therefore, based on the comprehension of the defined ontologies, KWs understand the business context and compose the corresponding rules with the suggestions from the rule editor (see Fig. 4).

The main constraint for the EPA rule in natural language reads like "If a house or a place occupied by children and built before 1978 is repaired, the owner and tenants need to receive EPA's lead pamphlet". Expressed with the ontology rule editor this results in:

Constraint *Submit EPA Lead Pamphlet* for ***Renovation & Repair Service Case:*** ***affects Housing or Child Occupied Facility*** *Year of Completion* less than 1978 **<u>leads to</u>** ***Submitting EPA Lead Pamphlet*** Completed.

The compliance rule's name *Submit EPA Lead Pamphlet* was defined by KWs in the field reserved for the name. The rule's scope applies to the *Renovation & Repair Service Case*, which is a concept defined in the *Renovation & Repair ontology*. KWs choose the concept from the editor's drop-down list showing the case concepts and it sub-concepts, in this example the *Renovation & Repair Service Case.* In case the list contains a many items, users get filtered results by just typing the first letters of the expected items.

The repair object of the *Renovation & Repair Service Case* is about a house or a child occupied facility. This relation is defined by the relation *affects* in the Repair Object ontology (cf. Fig. 3). The concept *Housing or Child Occupied Facility* is a sub-concept of the Repair Object concept. This sub-concept represents a group of repair objects which are houses or child occupied facilities. Within the ontology definition a constraint concept is derived from a parent concept by specifying its conditions like *Owner.Address is not empty*. The ontology reasoning is applied in our approach to reduce complexity of the rule definition.

The *Year of Completion* is an attribute of the *Repair Object* concept. *Less than* is a logic operator for the numeric attribute *Year of Completion*, which is entered as number *1978* by the business user. *Leads to* is a temporal operator selected from the drop-down list. The *Submitting EPA Lead Pamphlet* is a *Task* concept. *Completed* is the value of the attribute *State* of the *Task* concept.

To ensure that the *Repair Service Case* can be closed in compliance with the EPA rules and the ACM principles, we create another two rules to allow ad hoc actions in this case.

Constraint *Renovation & Repair Categorize Repair Site* for ***Renovation & Repair Service Case: Categorizing Repair Site*** *Completed* **<u>precedes</u>** ***Renovation & Repair Preparation Goal*** *Completed***.**

This constraint implies that for closing the preparation phase of the renovation case, it is necessary to do the *Categorizing Repair Site*. In other words, before completing the *Renovation & Repair Preparation Goal*, the *Categorizing Repair Site* task must be completed.

Constraint *Renovation and Repair Prepared Goal Reached* for ***Renovation & Repair Service Case: Submitting EPA Lead Pamphlet*** *Completed* **<u>leads to</u>** ***Renovation & Repair Preparation Goal*** *Completed***.**

This constraint implies that only when the task *Submitting EPA Lead Pamphlet* is completed, the goal *Renovation & Repair Preparation* can be reached.

The compliance rules can be composed in different ways depending on how KWs transfer the regulations to the constraint specifications. However, the semantic of the rules defined in the system must be the same as the ones defined in the textual language.

D. Select from ad hoc actions at runtime

A company performing renovation and repair work with

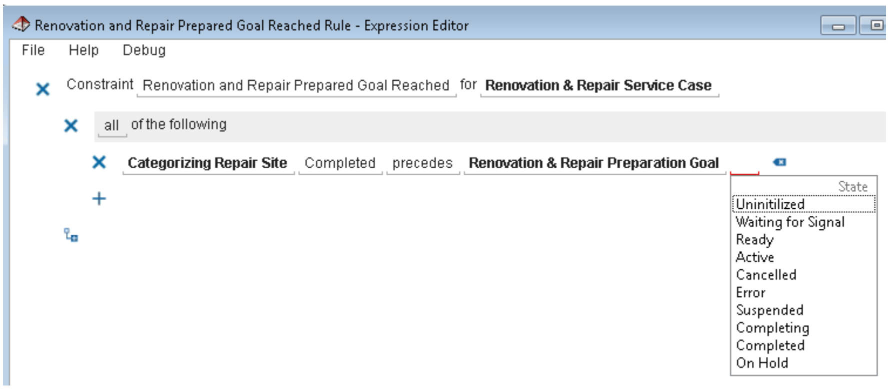

Fig. 4. Compliance Rule Editor

their own staff as well as sub-contractors use the ISIS Papyrus Adaptive Case Management solution and define the compliance constraints in the *Compliance Rule Collection* to manage and administrate their businesses.

A generic case template for renovation and repair cases provides a basic goal structure: *Renovation and Repair Prepared* goal has to be settled before the *Renovation and Repair Performed* goal, before the *Renovation and Repair Aftercare* goal. Established tasks are provided for Clerks by templates related to the predefined goals. Based on their experience and on the given situation Clerks may perform any of the provided tasks or generic ad hoc tasks in their desired order to meet the goal they are working on. To guarantee the compliance of the case execution with the EPA rule for distributing the lead pamphlet, the ACM framework uses the three compliance rules defined above.

Let us consider the first goal *Renovation & Repair Preparation Goal*. A Clerk can perform several tasks, such as the task *Define Customer Requirements*, to prepare a renovation case. However, regarding the compliance rule *Renovation & Repair Categorize Repair Site*, to complete the goal *Renovation & Repair Preparation Goal*, the task *Categorizing Repair Site* must be completed first. Therefore, the rule will be triggered and the Clerk will be notified that the rule is violated and that the task *Categorizing Repair Site* shall be done. The Clerk can accept the suggested task or do another task if wanted. The *Renovation & Repair Categorize Repair Site* rule keeps the violation state until the *Categorizing Repair Site* task is done.

To finish the *Categorizing Repair Site* task, the Clerk has to enter the information about the repair object, such as the building year of the object , the address of the owner and if the object is a residence or an office building, etc. When the Clerk has finished the task, the rule is permanently satisfied. If the information of the object confirms that the building has tenants or is occupied by children and built before 1978, another rule *Submit EPA Lead Pamphlet* will be triggered. The Clerk is notified that this rule is temporally violated and the task *Submitting EPA Lead Pamphlet* should be done. When the Clerk finishes this task, another rule *Renovation and Repair Prepared Goal Reached* is triggered and the Clerk is notified that the goal *Renovation & Repair Preparation Goal* can be completed.

A Clerk can freely execute ad hoc actions in the *Renovation & Repair Service* case as long as the rules related to the case are not violated. The compliance checking observes the execution of the case and triggers the related compliance rules when objects mentioned in the rules are executed. This way, the system can ensure the conformance of ad hoc actions executed by Clerks and support them to close a case in a compliant way.

IV. DISCUSSION AND FUTURE WORK

Our approach and the presented implementation of our approach in the ISIS Papyrus ACM system have certain implications for practice, which are discussed in this section.

A. *The ontology implementation in the ACM system*

Ontologies provide and share a common understanding of the business situation and thus, close the gap between the business domain (in the presented situation for repair service management) and the abstract ACM domain. Consequently, ontologies can support the education and training provided for business users.

The ontology definition in the ACM system benefits from different roles for KWs [13], either acting as Clerks or as Business Administrators having also ontology and/or compliance rule definition rights. With the ontology editor integrated into the ACM system, KWs are encouraged to define their business ontologies themselves. A KW understanding the business domain and knowing ACM can act with a special role as an ACM Administrator or as a Business Administrator, having the responsibility for the mapping between the ontology concepts of the business application and the underlying system objects. The ontology implementation in ACM provides KWs with both an abstract and an operational level in the same working environment. Moreover, ontologies defined in another application, framework or by standardization organizations could be directly imported into the ACM system and used as base for the business specific application model.

The ontology management needs further considerations on how the data mapping can be simplified and be subject to standard change management processes for deployment into production. The mapping on conceptual level between ontologies and from concepts to underlying objects should support users in analyzing and recognizing suitable objects for a certain concept which will be considered for future work.

B. *Explicit compliance rules defined by KWs*

Compliance rules are most of the time implicitly implemented directly inside the business processes or encoded by IT developers [6, 7, 9]. In our approach, we consider compliance rules as explicit entities loosely linked through the ontology concepts with the related ACM elements and applied to different scopes of applications. This flexibility allows assigning a compliance rule to a certain ACM template, a group of templates defined by a concept, an application tenant or an entire framework.

The compliance rule editor represents the context of the business via the concepts and relations suggested to KWs. The constraint language used to express compliance rules uses terms and grammar understood by business users. With the rule editor, KWs can manage compliance rules in the system independently from IT developers.

In future work we aim at addressing usability aspects of the rule editor’s graphical interface and will look at optimizing the natural language syntax used for the rules expression. We are working on several surveys with different types of users, from typical business users to technical IT users, to experiment on user capabilities. The results are expected to influence the syntax to come as close as possible to the logic thinking of business users.

Another future work aspect is to evaluate how the compliance of ad hoc actions at run time should be visualized in a graphical interface of KWs so that they can easily recognize temporarily violated tasks, related compliance rules and tasks suggested to compensate the current violations. The communication between KWs and the compliance rule checking is also considered in our future work.

The system provides KWs functionalities to manage the rule collection, such as to generate a new rule, to edit, to delete, to check conflicting and the redundancy of the collections. We are still working on this feature to support KWs with additional productive functions in the rule collection.

C. The impact of compliance rules on the configuration of an ACM system

The compliance of ad hoc actions in ACM can be controlled by compliance rules observed at runtime. KWs (Clerks) are not restricted by predefined processes. Based on their experience, they are able to freely evolve a case by ad hoc actions that suite each situation best. Users can be supported by a User Trained Agent [15] to propose best next actions. We are also working on a solution to provide non-compliance conflicting proposals. All elements determining the flexibility of an ACM system, ad hoc actions and compliance rules, are now in the hands of KWs. Overall, applying compliance rules in ACM not only reduces the effort of template design but also enhances the flexibility for KWs at run time.

However, the existence of compliance rules in the system requires some effort from KWs, especially caused by the knowledge gap between business and technical domain. The ontology-based ACM approach facilitates an ontology editor for ontology management and a compliance rule editor for creation and management of rules. The development of compliance rules, and thus the ACM system, can be handled by KWs without IT involvement. This self-management capability of KWs will help to speed up the design cycles of ACM cases.

The application of ontologies in ACM conceptualizes the system architecture into different levels corresponding to different application layers. This conceptual communication based on ontologies may simplify the business software engineering, as it provides the architectural interface between the underlying system and different application domains, see Fig. 1. Another benefit of ontologies in our approach is that KWs can use the business terms defined in their business domain to compose compliance rules with the compliance rule editor. Consequently, compliance rule management is handed over to KWs.

V. CONCLUSION

The ontology-based approach described in this paper aims to support KWs with an environment guaranteeing business compliance in ACM for the sake of a compliant case evolution within the boundaries described by the rules composed in business language by KWs. Rules must be understandable to business people who don't set variables and don't call functions and thus, must be formulated in domain specific language, not in IT terms. The benefits are that ACM based business applications are under control of business departments where the domain specific knowledge is located and the typical overhead of IT projects with complex round trips between departments and release cycles is eliminated. The combination of ontologies with compliance rules enables a robust ACM system which satisfies management needs for business compliance whilst business users can respond to customer needs and focus on customer experience, one of the important forces of the digital business era.

ACKNOWLEDGMENT

This work was supported by the FFG project CACAO, no. 843461 and the Wiener Wissenschafts-, Forschungs- und Technologiefonds (WWTF), Grant No. ICT12-001.

REFERENCES

[1] "Business Rule Group." [Online]. Available: http://www.businessrulesgroup.org/defnbrg.shtml. [Accessed: 10-May-2016].

[2] C. Czepa, H. Tran, U. Zdun, S. Rinderle-Ma, T. Tran, E. Weiss, and C. Ruhsam, "Supporting Structural Consistency Checking in Adaptive Case Management," in International Conference on Cooperative Information Systems (CoopIS) 2015, 2015, pp. 311–319.

[3] C. Czepa, H. Tran, U. Zdun, T. Tran, E. Weiss, and C. Ruhsam, "Towards Structural Consistency Checking in Adaptive Case Management," in 4th International Workshop on Adaptive Case Management and other Non-workflow Approaches to BPM (AdaptiveCM 15), 2015.

[4] EPA, "Renovation, Repair and Painting Program." [Online]. Available: https://www.epa.gov/lead/renovation-repair-and-painting-program. [Accessed: 10-May-2016].

[5] Esper Tech., "Complex Event Processing (CEP)." [Online]. Available: http://www.espertech.com/products/esper.php. [Accessed: 11-Mar-2016].

[6] Forrester, "Best Practices In Implementing Business Rules." [Online]. Available: https://www.forrester.com/report/Best+Practices+In+Implementing+Business+Rules/-/E-RES46339. [Accessed: 10-May-2016].

[7] G. Governatori and A. Rotolo, "Semantic Web Rules: International Symposium, RuleML 2010, Washington, DC, USA, October 21-23, 2010. Proceedings," M. Dean, J. Hall, A. Rotolo, and S. Tabet, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 194–209.

[8] ISIS Papyrus, "Adaptive Case Management," 2016. [Online]. Available: http://www.isis-papyrus.com/e15/pages/business-apps/acm.html.

[9] J. Boyer and H. Mili, "Agile business rule development process, architecture, and JRules examples." Springer, Berlin; Heidelberg; New York, 2011.

[10] M. B. Dwyer, G. S. Avrunin, and J. C. Corbett, "Patterns in Property Specifications for Finite-state Verification," in Proceedings of the 21st International Conference on Software Engineering, 1999, pp. 411–420.

[11] M. J. Pucher, "The Strategic Business Benefits of Adaptive Case Management," in How Knowledge Workers Get Things Done - Real-World Adaptive Case Management, L. Fischer, Ed. Future Strategies Inc., 2012.

[12] T. T. K. Tran, E. Weiss, C. Ruhsam, C. Czepa, H. Tran, and U. Zdun, "Embracing Process Compliance and Flexibility through Behavioral Consistency Checking in ACM - A Repair Service Management Case," BPM 2015 4th Workshop on ACM and Other Non-Workflow Approaches to BPM, pp. 1–12, 2015.

[13] T. T. K. Tran, M. J. Pucher, J. Mendling, and C. Ruhsam, "Setup and maintenance factors of ACM systems," Lecture Notes in Computer Science, vol. 8186 LNCS, pp. 172–177, 2013.

[14] T. T. K. Tran, E. Weiss, C. Ruhsam, C. Czepa, H. Tran, and U. Zdun, "Enabling flexibility of business processes by compliance rules - A case study from the insurance industry," BPM (Industry Track), pp. 30-43, 2015.

[15] T. T. K. Tran, C. Ruhsam, M. J. Pucher, M. Kobler, J. Mendling, "Towards a Pattern Recognition Approach for Transferring Knowledge in ACM," EDOC Workshops, pp. 134-138, 2014.