Defining Reusable Administrative Processes Using A Generic Ontology

ALVARO E. PRIETO
Quercus Software Engineering Group, University of Extremadura, Escuela Politécnica, av. Universidad s/n, Cáceres, 10003, Spain
aeprieto@unex.es

ADOLFO LOZANO-TELLO
Quercus Software Engineering Group, University of Extremadura, Escuela Politécnica, av. Universidad s/n, Cáceres, 10003, Spain
alozano@unex.es

Abstract: Administrative processes are a type of business process commonly used in public institutions and large companies. The workflow definitions of these processes are frequently reused within the organizations. The reuse of these workflow definitions can be easier if they are divided into three separate but related definitions: on the one hand, the definition of the data structures to be managed by the process activities, on the other, the users that can perform each activity and, finally, the process activities together with the relationships among the three definitions. The use of ontologies can facilitate these workflow definitions in the three related parts. In this paper, we describe OntoMetaWorkflow, a generic ontology to represent canonical workflow terms in the domain of administrative processes, and the methods for using it in the defining of administrative processes. This ontology is the basis of the WEAPON Model, a complete model supported by tools, for defining and managing administrative process workflows.

Keywords: Business process; Administrative process; workflow; ontologies, OntoMetaWorkflow; WEAPON.

1. Introduction

The Workflow Management Coalition (WfMC*) tries to standardize Workflow Management Systems (hereinafter referred to as WfMS) to facilitate their widespread application. Basically, a WfMS is able to interpret the workflow definition and create and manage the execution of these workflows. WfMS can be applied to different types of business processes such as administrative processes.

Administrative processes are usually characterized by the submission of an application form that will be handled by different users at different stages. These users may access to the current information of the process in order to provide data that will be included in its dossier. All this information will be used by the person responsible for taking a final decision on the application of each process. In general they are processes regulated by laws, rulings of action or protocol executions in public institutions or large companies. Examples could be the management of public contest bids, loan application procedures or a simple holiday application.

These processes are often defined generically in the level of management or governance of the organization. However, these processes must be reused in the lower

* http://www.wfmc.org
levels of the hierarchy dependent institutions, subdepartments, delegations, etc. in order to be executed by them. This reuse may become even more complex when the generic definition of the process is not done within the organization but rather it is determined by rules enacted by an external organization and those rules are mandatory in order to use this process. For example, in many countries the central bank obliges financial institutions to comply with regulations for using processes such as taking deposits or granting loans. These rules, which can range from the types of credits that can be granted to activities that form the process, must be reused following the idiosyncrasy of each bank, but always complying with all the restrictions set by the central bank.

A complex WfMS with advanced characteristics, such as coordination with other external applications, is not usually required by this type of process. Actually, the ideal WfMS would be one that identifies and classifies the activities that form the process, the data elements managed by the activities and the users that must carry out the activities without ambiguities. Because of this, this type of WfMS is easily reusable and adaptable to changes that may occur regularly in this type of process, whether due to changes in laws that regulate it or due to its reuse in similar institutions or companies.

The use of ontologies as a basis for the management of this type of process could be very useful due to their characteristics of complete and precise representation of terms that make the integration of a data scheme easier and require less effort in reuse and adaptation. With this in mind, a WfMS model based on ontologies was proposed in [1]. In some cases, we found that the definition of the generic ontology did not allow an easy reuse of the process. This generic ontology has been restructured. In the new ontology the division of the representation of the process into three separate but related definitions has been improved. Due to this restructuration, it has been necessary to redefine the original model together with its tools and test them in several domains. That is, they have been redesigned with the aim of favoring the precise identification of the data elements managed, the activities that form the processes and the users involved. As a result, OntoMetaWorkflow and WEAPON have improved the objectives of its predecessors.

In this paper, an exhaustive description of the new ontology, called OntoMetaWorkflow, is presented. The methods for defining administrative processes using OntoMetaWorkflow are also presented together with a fictitious and didactical example of a loan application process. In addition to this, a description of the redefinition of the model [1], now called WEAPON (Workflow Engine for Administrative Processes based on ontologies) is also presented together with its tools.

This paper is structured as follows: section 2 identifies existing works that use ontologies in WfMS, section 3 describes the redefined model WEAPON, section 4 details the ontology OntoMetaWorkflow, section 5 explains the methods for defining administrative processes in ontologies using OntoMetaWorkflow and section 6 describes how WEAPON manages the workflows defined in ontologies.
2. Use of ontologies in WfMS

In recent years, the use of ontologies is becoming ever more frequent for knowledge representation, both in reasoning and knowledge-based systems as well as in traditional information systems. This is due to the fact that ontologies are designed with the aim of their knowledge being easily reusable and shared by users and communities of the same domain. Moreover, the general acceptance of a single language of representation such as OWL b (Ontology Web Language) and the development of tools like Protégé c for the construction of ontologies has favored the wide use of ontologies in many fields, including software engineering [2].

The ontologies, applied to business process definition, provide a complete, precise and shared terminology about a particular domain which facilitates integration and which are easily reusable by the same or another organization. These advantages provide a considerable saving of time and effort in processes and data definition tasks, and in merging methods when similar representations of the same domain exist.

In particular, the application of ontologies to WfMS have been used previously in approaches such as the one of Vieira et al. [3] that proposes a solution to make workflow execution more flexible and is possibly the first work integrating both fields. Also interesting is the work of Gasevic et al. [4] which provides a Petri net ontology. Haller et al. [5] present a multi meta-model process ontology, called m3po, which relates workflow models to choreography models. Abramowicz et al. [6] present a semantically enhanced Business Process Modeling Notation [7], namely the sBPMN ontology. Francescomarino et al. [8] provide a complete business process modeling with semantic annotations in order to formalise and verify constraints on business process diagrams. Groener et al. [9] propose to use OWL-DL representation on process control flow modeling and how to apply query patterns for process retrieval. Savvas et al. [10] present a system that uses an ontology for representing the structure of the Greek administrative authority and which types of documents are produced and sent among its administrative units during the execution of their procedures. Finally, Papageorgiu et al. [11] use ontologies to describe collaboration patterns and employ a reasoning process to match triggering events and conditions with collaboration solutions. We can also mention the approaches presented in [12][13][14][15][16][17][18][19] as examples of integration of both fields and a recent survey about Semantic Business Process Management is available in [20].

Unlike the previous approaches, this paper presents an ontology for representing administrative processes together with their activities, domain data and users involved. Although several consolidated models and languages of workflow representation exist [21][22][23][24][25], the application of ontologies in this field, used directly or as a definition of a metalanguage, can provide the following advantages:

b http://www.w3.org/TR/owl2-overview/

c http://protege.stanford.edu/
• The users, following methodologies for developing ontologies, can obtain complete, precise and shared definitions of process,
• The domain data of a process or the users which participate in it can be changed without modifying the definition of the data managed by activities or the definition of the process,
• Workflow definitions, represented in ontologies, are more easily reusable although the reuse process may involve some effort, mainly in the processes of search, selection, and, in some cases, adaptation to the new system. These factors are discussed in detail in [26].

Also, some of these advantages are described in Dang et al. [27], who proposes the use of ontologies as the basis of medical workflows, although their approach is restricted to the use of a complete and closed ontology of a hospital workflow. However, our ontology, OntoMetaWorkflow, provides a generic framework for representing workflows of administrative processes. Moreover, OntoMetaWorkflow is accompanied by a model, WEAPON, composed of the methods and tools needed for applying it.

3. WEAPON: Workflow Engine for Administrative Processes based on Ontologies

WEAPON (Workflow Engine for Administrative Processes based on ONtologies), is a model that proposes the use of ontologies to define and manage administrative processes and is more reusable than the model proposed in [1]. Concretely, this model is able to manage workflows of administrative processes which are characterized by being initiated by a user and which must be attended to or evaluated by other different users, following a perfectly defined protocol for the order of the activities, the data managed by these activities, the users responsible for performing them and the time frames available for carrying them out.

Basically, WEAPON provides an ontology for defining workflows of administrative processes called OntoMetaWorkflow. Using this ontology, WEAPON proposes, firstly, how an ontology engineer must define OntoDD: an ontology with the relevant data of the domain and the users which can participate in the workflow. And, secondly, how a workflow designer can define a workflow that will be stored in OntoWF: an ontology with the properties of the process together with the activities that the process contains, its order, what type of user, defined in OntoDD, can perform the activities and what data defined in OntoDD will be managed by every activity. This workflow definition does not contain rules because OntoMetaWorkflow allow workflow designers to describe all possible relationships among data, participants and activities. This ensures that the processes are well defined and are more reusable, in addition, the classes of domain data and the classes of workflow participants can be modified without changing the representation of the workflow process. Moreover, it should be noted that the taxonomies of classes and instances that may be needed as domain data or workflow participants can be defined in ontologies in the organization itself or can be reused from ontology repositories.
Following a bank loan application, as an example of an administrative process, it is necessary to know, among other domain data, the types of loan offered by said bank or the internal classification of types of borrower applied by the bank to the applicants. The ontology engineer could reuse an existing ontology about types of loan or classifications of applicants with a consequent saving of time. Moreover, it is possible to see more advantages if we suppose that in this example the rules for the processes of loan applications are established by the Central Bank of the country. In this case, it would be enough to have the definition of the workflow proposed by the Central Bank and adapt it to each individual bank, simply adding its taxonomy of domain data (types of credit or risk factor) and its taxonomy of allowed participants (customer, credit analyst, credit manager, etc.).

The model presents a series of interrelated components that form its architecture (Figure 1). As described below, our approach provides an ontology as the basis of workflow representation, together with methods (and their respective software tools) to identify and to exploit the workflows of a management process. The software tools are not designed as Semantic Web Services but as ad hoc applications. The elements that constitute the model are:

(i) an ontology for the generic definition of workflows (OntoMetaWorkflow). It is detailed in the next section,

(ii) an ontology of domain data (OntoDD), that must be built by an ontology engineer following the specifications of OntoMetaWorkflow. This ontology will import the elements defined in OntoMetaWorkflow and must contain, firstly, the taxonomy of data which will be used in the domain and, secondly, the taxonomy of the possible workflow participants,

(iii) an ontology of the workflow of the administrative process (OntoWF), that must be built by a workflow designer, following the specifications of OntoMetaWorkflow and OntoDD. This ontology represents the workflow of the particular administrative process that will be managed by the WfMS. The workflow together
with its properties, the activities that it contains, the order of execution of said activities, the domain data of OntoDD that will be shown or modified in an activity and the participants which can carry out the activity are defined in this ontology. Thus, the specifications of elements and possible relationships that may have a workflow and that are represented in OntoMetaWorkflow, and the relevant data of the domain and the possible participants in the workflow defined in OntoDD are used for this representation,

(iv) WEAPON Designer (a fully functional version is available free here\(^d\)) facilitates the development of OntoWF using the graphical representation WF-NET [28]. This tool allows workflow designers to represent, in a graphical way, the activities of the process and their order of execution and save them in a PNML file. But, moreover, it also allows workflow designers to establish relationships among these activities and the users and data defined in OntoDD according to the properties specified in OntoMetaWorkflow and save all as OntoWF ontology in an OWL file. Jena\(^e\) has been used in its implementation and WoPeD\(^f\) has been used as a graphical canvas of the definition of workflows,

(v) WEAPON Manager is a generic web application that manages the administrative processes represented in the OntoWF and OntoDD ontologies by means of web forms. This application allows users to exploit administrative processes as instances of the AdministrativeProcess class of OntoWF and stores them in a database. As well as WEAPON Designer, it has been implemented using the Jena parser together with MySQL, and, in the case of web forms, Java technology. It is explained in more detail in section 6.

It is important to note that, due to the activities of these types of processes being tightly bound to the data managed, in this model, ideas of traditional WfMS as well as some provided by the Case Handling approach have been used \([29]\). Basically, Case Handling is a data-driven approach where each activity is associated with at least one data object of the process being managed. Moreover, it proposes the uses of forms, associated to activities, for managing data. Our model takes this idea but with the difference that the activities are not associated to predefined forms but rather that the forms are built dynamically for each activity using the definitions contained in OntoDD and OntoWF. This provides advantages for independence and reuse of the different representations.

4. OntoMetaWorkflow: a generic ontology for the representation of workflows

OntoMetaWorkflow\(^g\) contains the terms that form the workflows of administrative processes and their relationships. This ontology is built adapting the definitions of workflow elements provided by the WfMC to the specific characteristics of administrative processes. It has been developed following METHONTOLOGY methodology \([30]\) and it is represented using OWL Language. Because

\(^d\) http://quercusseg.unex.es/weapon/?Designer
\(^e\) http://jena.sourceforge.net/
\(^f\) http://www.woped.org
\(^g\) http://quercusseg.unex.es/weapon/?download=OntoMetaWorkflow.owl
OntoMetaWorkflow is intended to be used as a meta-ontology, their attributes and relationships are not classified as in a usual ontology. Instead, these elements are classified into two types: definition and execution elements. On one hand, definition elements must be established or take a value in the ontologies that represent the specific workflow (OntoDD and OntoWF). These definition elements are shown using UML in Figure 2. A UML representation together with the details of every definition element are available in http://quercusseg.unex.es/weapon/?OntoMetaWorkflow. On the other hand, execution elements will be used to store information about workflow instances during their execution in WEAPON Manager. Each class, relationship and attribute is explained in following subsections:

![Fig. 2. OntoMetaWorkflow in UML.](image)

### 4.1. Definition elements of OntoDD

These elements are used to define the classes and properties that represent the common data and the potential users of all similar processes within a domain. These elements are:

**Classes**
- **Domain Data.** Use: it is the class that stores common data of all instances of an administrative process.
- **Workflow Participant.** Use: it stores the users involved in the process.
- **Root.** Use: it is a subclass of **Workflow Participant** that can manage the WEAPON Manager WiMS.

**Attributes**
- **External Document.** Defined at class: **Domain Data.** Value type: boolean. Use: if a subclass of **Domain Data** refers to an external document, this attribute will store true or otherwise false.

http://quercusseg.unex.es/weapon/?OntoMetaWorkflow
• **Location.** Defined at class: *Domain Data*. Value type: string. Use: if *External Document* is true in some subclass of *Domain Data*, then this attribute stores the URL of said document.

• **Id.** Defined at class: *Workflow Participant*. Value type: string. Use: it stores the id that identifies an instance of *Workflow Participant* in WEAPON Manager WiMS.

• **Password.** Defined at class: *Workflow Participant*. Value type: string. Use: it stores the password that identifies an instance of *Workflow Participant* in WEAPON Manager WiMS.

• **Name.** Defined at class: *Workflow Participant*. Value type: string. Use: it stores the name of an instance of *Workflow Participant*.

• **Surname.** Defined at class: *Workflow Participant*. Value type: string. Use: it stores the last name of an instance of *Workflow Participant*.

• **Email.** Defined at class: *Workflow Participant*. Value type: string. Use: it stores the email of an instance of *Workflow Participant*.

4.2. **Definition elements of OntoWF**

These elements are used to define the classes and properties that represent a particular process, that is, the sequential flow of activities and their relationships with the elements of the domain defined in OntoDD. These elements are:

**Classes**

• **Administrative Process.** Use: it represents the process managed by the workflow system.

• **Activity.** Use: it represents a logical unit of work.

**Relationships**

• **Before.** Source class: *Activity*. Target class: *Activity*. Use: it stores what activities immediately precede an *Activity*.

• **Generated by.** Source class: **Administrative Process**. Target class: *Workflow Participant*. Use: it is used to indicate which *Workflow Participant* can create instances of the *Administrative Process*.

• **Is Performed By.** Source class: *Activity*. Target class: *Workflow Participant*. Use: it is used to indicate which *Workflow Participant* can carry out the activity.

**Attributes**

• **Before Control Flow Pattern.** Defined at class: *Activity*. Value type: string. Use: if two or more activities stored within *before* relationship exist, this attribute contains ‘and’ when those activities are forming a parallel routing or contains ‘xor’ when those activities are forming a selective routing. It is not supported the pattern ‘or’ because our model only manages simple administrative processes.

• **Select Instance Of Domain Data.** Defined at class: *Activity*. Value type: string. Use: it stores the names of subclass of *Domain Data* whose instances will be offered to the *Workflow Participant* who will perform the activity. The Workflow participant must choose one of them during the execution of the activity. The instances chosen during the execution by the Workflow Participant will be stored in the *Selected*
Instances Of Domain Data attribute. Instances that will be offered also include instances of subclasses that the stored classes may have.

- **Show Instance Of Domain Data.** Defined at class: Activity. Value type: string. Use: it stores the names of subclass of Domain Data whose instances will be displayed to the Workflow Participant if said instances were chosen in precedent activities and, therefore, they will be stored in the Selected Instances Of Domain Data attribute.

- **Select Class Of Domain Data.** Defined at class: Activity. Value type: string. Use: it stores the names of subclass of Domain Data that will be offered to the Workflow Participant who will perform the activity. The Workflow participant must choose one of them during the execution of the activity. The subclasses chosen during the execution by the Workflow Participant will be stored in the Selected Classes Of Domain Data attribute. Subclasses that will be offered also include subclasses that the stored classes may have.

- **Show Class Of Domain Data.** Defined at class: Activity. Value type: string. Use: it stores the names of subclass of Domain Data that will be displayed to the Workflow Participant if said subclasses were chosen in precedent activities and, therefore, they will be stored in the Selected Classes Of Domain Data attribute.

- **Fill In Instance Attributes of Process.** Defined at class: Activity. Value type: string. Use: it stores the names of the attributes defined for a subclass of Administrative Process that can be filled in or modified in said activity.

- **Show Instance Attributes Of Process.** Defined at class: Activity. Value type: string. Use: it stores the names of the attributes defined for a subclass of Administrative Process that will be displayed in said activity.

- **Days Time Frame.** Defined at class: Activity. Value type: integer. Use: it stores the number of days available for doing the activity.

- **Day Notice.** Defined at class: Activity. Value type: integer. Use: it stores the number of days before the deadline of the activity in which the person responsible for the activity will be warned in order to finish it.

- **Activity Description.** Defined at class: Activity. Value type: string. Use: it maintains an explanation about the work to do within an activity.

### 4.3. Execution elements

These elements are used by WEAPON Manager WfMS to store information about the state of workflow instances during their execution. These elements are:

#### Relationships

- **Assigned to.** Source class: Administrative Process. Target class: Workflow Participant. Use: it stores what instance of Workflow Participant (user) has been assigned to the instance of the Administrative Process.

#### Attributes

- **Selected Classes Of Domain Data.** Class name: Administrative Process. Value type: string. Use: it stores the Domain Data subclasses that have been selected within an instance of an Administrative Process.
• **Selected Instances Of Domain Data.** Class name: *Administrative Process*. Value type: string. Use: it stores the instances of *Domain Data* subclasses that have been selected within an instance of an *Administrative Process*.

• **Current Situation.** Class name: *Administrative Process*. Value type: string. Use: it stores the name of the *Activity* where the instance of a process is.

• **Beginning Date.** Class name: *Administrative Process*. Value type: string. Use: it stores the beginning date of an instance of the process.

• **Max Ending Date.** Class name: *Administrative Process*. Value type: date. Use: it stores the deadline of an instance of the process.

5. **Methods for defining administrative processes using OntoMetaWorkflow**

It is necessary to apply two methods to define administrative processes as ontologies using OntoMetaWorkflow. The first method is for building the domain ontology, OntoDD. The second one is for building the ontology of the process, OntoWF. This last method is more complex but WEAPON Designer can aid workflow designers in applying it more easily. Both methods together with an example of how to build the OntoDD and the OntoWF of a loan application process (see Figure 3) are explained in detail in the following subsections.

Fig. 3. Loan application process in WEAPON Designer
5.1. Method for building OntoDD

OntoDD imports the concepts defined in OntoMetaWorkflow and must contain, firstly, the taxonomy of data which will be used in the domain and, secondly, the taxonomy of the possible workflow participants. The main rule which must be fulfilled by OntoDD is that the root elements of each one of its taxonomies is linked with the superclasses defined in OntoMetaWorkflow, Domain Data and Workflow Participant (or Root if the participant is a WfMS administrator) respectively.

The development of this ontology can be carried out in a simple way with some application for building ontologies in OWL, such as Protégé, carrying out the following steps:

1. Import OntoMetaWorkflow so its elements will be superclasses in OntoDD.
2. Identify those common data taxonomies of the domain in order to define them as subclasses of Domain Data. That is, the taxonomies of data that will be common to all instances of the administrative process must be defined here. In the loan application process, taxonomies such as Credit Types with subclasses such as Loan (credits with a personal guarantee) or Mortgage (credits with a property guarantee) and Risk Factor with subclasses such as High Risk or Low Risk will be subclasses of Domain Data. In Figure 4, the taxonomies of data of the loan application process are shown.

![Fig. 4. Taxonomies of loan application process.](image)

3. Define the common properties of each taxonomy of Domain Data. In particular, in the loan application process, the Credits Types will be classified by having different terms or interest rate ranges while the Risk Factor of an instance of a loan application process will be defined by means of the result obtained in a risk assessment test.

4. Set the property values for every subclass of the Domain Data and, if instances of the subclasses exist, also for these instances. In the loan application process, the taxonomies defined as domain data do not contain instances. Thus, it is only necessary to set property values for subclasses. Typical of an administrative process with instances in the Domain Data is an incident management process. In this case,
a subclass of Domain Data as Involved Material will contain instances of every printer or workstation that the company may have.

(5) identify the different types of users that will participate in the workflow as subclasses of Workflow Participant. In the loan application process, there will be classes such as Customer, Credit Analyst or Credit Manager. This step also includes:
(a) define at least one of the classes as a subclass of Root. In the loan application process, Credit Manager will be a subclass of Root,
(b) create the instances of every real user. It is necessary to set the value for Id, Password, Name, Surname and Email properties for every instance.

The different Workflow Participants defined for the loan application process are shown in Figure 5.

![Fig. 5. Workflow Participants in loan application process.](image)

The product of applying this method is an ontology with the domain data that can be used by the instances of the workflow and the users which can carry out it. As an example, the OntoDD ontology, for the loan application domain, is available¹.

5.1.1. Reusing in OntoDD

Existing domain ontologies can be reused in OntoDD adding them as subclasses of Domain Data.

Ontologies with the users of the organization can be reused as Workflow Participant. In this case it is necessary to take the next simple steps:
(1) set every type of user in the ontology as a subclass of Workflow Participant,
(2) it is necessary to define some of the classes as subclasses of Root,
(3) for every instance of reused classes, it is necessary to set the value for Id, Password, Name, Surname and Email definition attributes.

5.2. Method for building OntoWF

OntoWF represents the workflow of the administrative process that will be managed by the WfMS. The workflow together with its properties, the activities that it contains, the

¹ [http://quercusseg.unex.es/weapon/?download=OntoDD_LoanApplication.owl](http://quercusseg.unex.es/weapon/?download=OntoDD_LoanApplication.owl)
order of execution of said activities, the relevant data of OntoDD that will be shown or modified in an activity and the participants which can carry out the activity are defined in this ontology. Thus, the specifications of elements and possible relationships that may have a workflow and that are represented in OntoMetaWorkflow, and the relevant data of the domain and the possible participants in the workflow defined in OntoDD are used for this representation.

WEAPON Designer is used in the design of the OntoWF Ontology for a specific administrative process. However, it is possible to build OntoWF manually without the use of this tool, with a great effort. In any case, these steps must be followed:

1. import the OntoDD defined in the domain and OntoMetaWorkflow,
2. define the process to manage as a subclass of AdministrativeProcess. In our example, a class called Loan Application will be a subclass of AdministrativeProcess,
3. define the properties of the process. These properties will be displayed or modified for each instance of the process. In the Loan Application process, personal and financial data of the applicant or the different reports that will be made during the process will be properties of each instance of this process. In Figure 6, all the properties defined for the Loan Application process are shown,
4. indicate what subclasses of Workflow Participant can create instances of the process using the Only universal restriction on Generated By definition relationship. In the Loan Application process, only a Customer can apply for a loan,
5. define each activity of the process as a subclass of Activity. In the Loan Application process, there will be eight activities: Gather Credit Info, Credit Scoring Assessment, Advanced Analysis, Assets Statement Analysis, Liabilities Statement Analysis, Advanced Assessment, Request Approval and Send Notification.
6. Additionally, for each subclass of Activity it is necessary to:
(a) indicate which activities precede it, using Only universal restriction on Before definition relationship.

(b) if the Before definition relationship of the activity is restricted to two or more activities, then indicate whether these activities have been carried out in parallel or conditionally. The Before Control Flow Pattern definition attribute will take the value and in the first case and xor in the second one. This property cannot take both values in the same activity,

(c) indicate what subclasses of Workflow Participant can carry out the activity using the Only universal restriction on Is Performed by definition relationship. This relationship should always be restricted to at least one subclass of Workflow Participant,

(d) if within the activity it is necessary to choose among different subclasses of Domain Data, then indicate the name of the root class of these subclasses using the Value restriction in the Select Class Of Domain Data definition attribute,

(e) if within the activity it is necessary to show some subclasses of Domain Data that has been selected in a previous activity of the administrative process, then indicate the name of the root class of these subclasses using the Value restriction in the Show Class Of Domain Data definition attribute,

(f) if within the activity it is necessary to choose among different instances of Domain Data subclasses, then indicate the name of the root class of the subclasses that contain the instances using the Value restriction in the Select Instance Of Domain Data definition attribute,

(g) if within the activity it is necessary to show some instances of Domain Data subclasses that have been selected in a previous activity of the administrative process, then indicate the name of the subclasses that contain the instances using the Value restriction in the Show Instance Of Domain Data definition attribute,

(h) indicate the process properties (defined in step 3 of this method) that can be filled in or modified within the activity, using the Value restriction in the Fill In Instance Attributes of Process definition attribute,

(i) indicate the process properties (defined in step 3 of this method) that can be displayed within the activity, using the Value restriction in the Show Instance Attributes of Process definition attribute,

(j) indicate the number of days available for doing the activity using the Value restriction in the Days Time Frame definition attribute,

(k) indicate, using the Value restriction in the Day Notice definition attribute, the number of days before the deadline of the activity in which the person responsible for the activity will be warned in order to finish,

(l) optionally, explain the actions that must be done within the activity by the participant assigned to the activity using the Value restriction in the Activity Description definition attribute.

The description of the Request Approval activity in WEAPON Designer is shown in Figure 7 as an example of what happens when these steps are taken.
Check if OntoWF defines a well-formed process. This verification is made automatically by WEAPON Designer. It is necessary to verify the next conditions:

(a) there is only one initial activity. Check if OntoWF has one and only one activity without restrictions defined on Before definition relationship. This one will be the initial activity,

(b) there is only one final activity. Check if OntoWF has one and only one activity that is not contained in the Only restriction on Before definition relationship of any other activity. This one will be the final activity,

(c) all activities are accessible. Check if all the activities, except the initial one, have an Only restriction on Before definition relationship with at least one activity,

![Fig. 7. Description of the Request Approval activity.](image)

The product of applying this method is an ontology with the definition of the workflow of the corresponding process. This definition contains its properties and its activities together with all the relationships among activities, domain data, process properties and workflow participants that are necessary to carry it out. As an example, the OntoWF ontology, for the loan application process, is available.

5.2.1. **Reusing in OntoWF**

The reuse in OntoWF is derived by the ease of reuse in OntoDD and how the activities in OntoWF use the OntoDD elements (Domain Data or Workflow Participants). In the Loan Application process, a customer can choose among different credits because the Gather

1 [http://quercusseg.unex.es/weapon/?download=OntoWF_LoanApplication.owl](http://quercusseg.unex.es/weapon/?download=OntoWF_LoanApplication.owl)
Credit Info activity has a Value restriction in the Select Class Of Domain Data definition attribute with Credit Types. If another bank wants to reuse this process with its own credits, they only have to change the subclasses of Credit Types in the original OntoDD by its own credit subclasses and the process works on. This idea also applies if it is necessary to change the Workflow Participant of the process.

6. Managing workflows defined in ontologies

WEAPON Manager is the web application that reads OntoDD and OntoWF and generates the web forms and the database that manages the workflow of the administrative process. It is necessary to install it in an Apache Server. After the installation, it is only available for a root user who must go into the configuration menu of WEAPON Manager and indicate what OntoDD and OntoWF will be used to build the web forms and the database. Figure 8 shows this configuration menu from the WEAPON Manager for the loan application process.

![WEAPON Manager configuration menu](image)

Once the root user indicates the ontologies and saves the configuration, WEAPON Manager reads the attributes of the administrative process defined in OntoWF and generates, using Jena, a database in MySQL where it will be possible to store each new instance of the administrative process. After this, WEAPON Manager comes into operation and it is available for the users stored in OntoDD.

Basically, WEAPON Manager has two modes of operation: to create and/or manage instances of the Administrative Process class that is defined in the OntoWF.

In the first case, WEAPON Manager checks, using OntoDD, if the user can create new instances. If so, it reads the “Select..” and “Fill..” properties of the first activity of the workflow in OntoWF and generates a web form with one field for each data stored in these properties. In our example, a web form, as the one shown in Figure 9, is generated.
for the first activity, *Gather Credit Info*. In this web form, the customer must indicate the concrete data of his loan application.

In the second case, a web page with a work list is presented to the user. This work list contains the instances that are assigned to the user. The user must choose one of them. Then, WEAPON Manager generates a web form that depends on the activity where the instance is. This web form will contain the fields that must be shown to the user (those ones contained in the “Show..” properties of the activity) and the fields that must be selected (those ones contained in the “Select..” properties of the activity) or filled in (those ones contained in the “Fill..” properties of the activity) by the user. In the bank example, if a Credit Analyst connects to WEAPON Manager, a list with the loan application instances that are waiting for his reports will be shown. Then, when he has chosen one, a web form is generated by WEAPON Manager. This web form will contain the fields that must be shown to the Credit Analyst to help him and the field that the Credit Analyst must fill in with his report.

![Loan Application](image)

Fig. 9. Part of the web form generated by WEAPON Manager for the activity Gather Credit Info.

Unlike other WfMS models, the web forms are built dynamically from the information specified in OntoDD and OntoWF. This means that if it is required to change either the sequence of activities, the elements involved in each one of them or the participants of the activity, it is only necessary to update the corresponding ontology, save the configuration with the new ontology, and those changes will be automatically transferred to the system. That is, any change that could affect the workflow of the administrative process, in particular its domain data, the user involved or the number and order of its activities, can be specified and WEAPON Manager will accept these changes without the need to modify anything else. In this way, it is possible to modify OntoWF, where the process is defined, or modify OntoDD, which contains the required information for the activities of the workflow, and WEAPON Manager rebuilds the web forms that work as WfMS.
7. Conclusions

In many administrative processes, the classification of the data managed by the activities and the categorization of the workflow participants in every activity have great importance because they are reusable terms in similar processes within organizations. Ontologies can be very appropriate as representation support for WfMS in these types of processes, because they provide significant advantages such as ease of use and the provision of information which is complete, consistent and can be shared both in data and processes. Moreover, the use of ontologies facilitates reuse, adaptation and integration of the processes and the data managed by every activity.

The WEAPON model (Workflow Engine for Administrative Processes based on ONtologies) has been presented in this paper. This model establishes how to define and manage an administrative process, and it is supported by complete software tools. Its basis is the OntoMetaWorkflow ontology which specifies the elements and rules that define workflows according to the standards and recommendations of the WfMC. Based on this generic ontology, it is possible to represent the relevant data of the corresponding domain and the users involved in each particular administrative process or simple business process in an ontology called OntoDD. Using the elements, relationships and restrictions defined in OntoMetaWorkflow and OntoDD, WEAPON Designer allows users to define workflows and store them as an OntoWF ontology. Finally, WEAPON Manager generates dynamically the corresponding web forms that will work as the WfMS of the workflow represented in OntoWF and OntoDD.

OntoMetaWorkflow and the WEAPON model have been checked in several fictitious domains in laboratories and, currently, WEAPON tools are being tested in two simple administrative processes in the Escuela Politécnica de Cáceres (University of Extremadura, Spain): holiday applications and classroom allocation requests. Also, an application of computer hardware incident management based on ITIL is in design stage.

Although the creation or adaptation of OntoDD ontologies requires considerable effort initially, these ontologies are usually reused by the same organization in other administrative processes. We will evaluate the suitability of the WEAPON model and their tools gathering opinions of designers and users by means of questionnaires. However, initial feedback from workflow designers who have used the WEAPON Designer, reveals that specifying the activities and participants for an administrative process is complicated and takes time. However, they state that the tool is fairly intuitive and aids in each design step. Users who have used the WEAPON Manager assert that web forms are filled quickly, and have improved the corresponding administrative process.

As mentioned previously, the method and tools are not designed for use with processes that need complex queries to database, internal calculations or the use of external applications. However, they work properly with administrative processes that are fully oriented to humans and, especially, in those processes that involve submitting some type of application to be considered at different stages, where different participants need
to handle current information of a dossier in order to provide new data in the corresponding activity.

Although OntoMetaWorkflow is the basis of the WEAPON model and tools, it has been designed with the intention that it can be reused in other projects or with other tools by those researchers interested in using ontologies to manage workflows.

8. Running Heads

Defining reusable administrative processes using a generic ontology

Acknowledgments

This work has been developed with the support of Ministerio de Ciencia e Innovacion (Spanish Contract MIGRARIA - TIN2011-27340), FEDER, Junta de Extremadura and Plan de Iniciacion a la Investigacion, Desarrollo Tecnologico e Innovacion 2010 de la Universidad de Extremadura (ACCVII-04).

References


