A Method for Systematic Adaptation and Synchronization of Healthcare Processes

Álvaro E. Prieto *aeprieto@unex.es*
Quercus Software Engineering Group, University of Extremadura
Cáceres,Spain

Adolfo Lozano-Tello *alozano@unex.es*
Quercus Software Engineering Group, University of Extremadura
Cáceres,Spain

Roberto Rodríguez-Echeverría *rre@unex.es*
Quercus Software Engineering Group, University of Extremadura
Cáceres,Spain

Fernando Sánchez-Figueroa *fernando@unex.es*
Quercus Software Engineering Group, University of Extremadura
Cáceres,Spain

Abstract

International organizations, as the World Health Organization (WHO) and national governments are constantly defining (or modifying) new healthcare protocols and procedures. Those changes have a significant impact, on one side, on the organizational concerns of a great number of healthcare institutions and centers, and on the other side, on their health information systems that need to be adapted according to the new (or modified) procedures. Administrative workflows are commonly defined by a high level entity and they must then be applied on different institutions ruled by this high level entity. Those workflows must then be adapted to the particular circumstances of each institution, complying with the general regulations of the process established at the top level. This problem, called Hierarchical Adaptation Problem, also implies establishing the methods to evolve together the high level regulation. Such methods must maintain the consistency among the different levels by means of the propagation of the changes to all the different adaptations of the original workflow. To solve this problem, this work introduces the Hierarchical Adaptation Method. A method based on ontologies to define the rules that must be satisfied by a generic workflow to be considered adaptable to different application cases and the rules that must be satisfied by its adapted versions. Moreover, it provides the operations to facilitate both adaptation of administrative workflows and propagation of changes.

Keywords: Workflows, Ontologies, Adaptation, Hierarchical Adaptation.

1. Introduction

International organizations, as the World Health Organization (WHO) and national governments are constantly defining (or modifying) new healthcare protocols and procedures. Those changes have a significant impact, on one side, on the organizational concerns of a great number of healthcare institutions and centers, and on the other side, on their health information systems that need to be adapted according to the new (or modified) procedures.

Administrative processes are a type of business processes widely used in public institutions and large companies. These processes are characterized by being governed by laws, regulations or well-defined action rules that define clearly which activities should be performed, their order, who must perform each activity, how each one must be done and how much time is available for doing them [1] [2].

Workflows are used for automatically managing different types of business processes as administrative processes [3]. Workflows for administrative processes (hereinafter referred to as administrative workflows) have distinctive features that distinguish them from other type of workflows. Those workflows are intended for predictable and repetitive processes. And they must have mechanisms that enable the coordination of the users responsible of each activity, including the notification of deadlines for completing them [4] [5] [6]. Normally, these workflows do not have to manage a considerable number of activities, users or data. Problems arise when the organizations need to use them in their different areas and departments, which means that the workflow specification must be adapted to the particular conditions of each area but conforming to the general specification established by the law or regulation that rules the process. This problem is called Hierarchical Adaptation Problem.

The Hierarchical Adaptation Problem (hereinafter referred to as HAP) may appear when an administrative process should be consequently adapted to the specific characteristics of every institution to be applied, but those modifications should not affect the restrictions defined in the original workflow specification. Moreover, HAP implies the necessity of a synchronization strategy between the original workflow and all its different versions, i.e. any change in the original workflow should be properly propagated to every version. This problem implies, on one hand, keeping the consistency of the administrative processes at the different levels of an organization and, on the other hand, if the law or regulation that rules the generic process is changed, establishing the measures that must be taken in order to propagate those changes to the adaptations made in the different areas and departments.

This paper presents an approach to solve this problem called Hierarchical Adaptation Method. This method is based on the specification of administrative workflows by means of ontologies. So, the method establishes the rules that verify the correctness of a workflow specification by means of ontologies, the rules that must satisfy the ontology of a generic workflow to be considered adaptable to different application scenarios and the rules that must satisfy the adaptations of the ontology. Moreover, they provide the set of operations that allow users to make the adaptations correctly and to propagate changes from adaptable workflows to adapted workflows.

As an example to illustrate the stages proposed in the method, the process to define the adaptation framework and to get adapted versions of a real healthcare workflow is presented. This workflow is inspired in the process to apply for inclusion on the Spanish Rare Disease Registry[1]. This is a national level process that each Spanish regional government may adapt to its special circumstances without altering original restrictions.

This paper is structured as follows: section 2 describes the Hierarchical Adaptation Method, section 3 identifies related work, and section 4 presents main conclusions.

2. Hierarchical Adaptation Method

A hierarchical workflow adaptation is basically conformed by four interrelated stages:

1. Generic (original) workflow specification. In this first stage, a generic workflow is specified following the process defined by the general regulation. The specification language should provide the engineers with the proper features to specify activities, their order and deadlines, their data and who is in charge of them. The specification language should also allow defining adaptation restrictions over any element of the workflow. The final product of this stage is the specification of the generic workflow managing the administrative process.
2. Specification of adaptation restrictions. In this second stage, engineers should indicate which elements must be always present in any possible version of the original

[1] https://spainrdr.isciii.es/en/Pages/default.aspx

workflow by defining adaptation restrictions. The final result of this stage is the specification of the adaptable workflow.

3. Adaptation. Taking as input the adaptable workflow and taking into account the specific features of the adaptation case (context), the adapted workflow is defined. The final product of this stage is a properly adapted workflow.
4. Propagation. This is an optional stage. It should be carried out when the generic workflow is modified. Then it is mandatory to modify conveniently the adaptable workflow. And then all those changes should be propagated to every adapted workflow. The final products of this stage are the new version of the adaptable workflow and the new versions of every adapted workflow

The Hierarchical Adaptation Method (hereinafter referred to as HAM) is based on the specification of workflows by means of ontologies. This is due to the several advantages that the use of ontologies can provide to solve the HAP. Firstly, its completeness, flexibility and accuracy for representing workflows, specifying, not only the activities, but also the participants and the data involved together with the characteristics of adaptation of each of these elements in each workflow. Secondly, it allows dividing workflow specifications into two ontologies. One ontology with the relevant data of the domain and the users which can participate in the workflow and another one with the properties of the process and the activities that the process contains, its order, what type of user of the first ontology can perform the activities and what data of the first ontology and what process properties will be managed by every activity. This division will facilitate both the reuse of data and workflows participants from the same domain as the adaptation of workflows to particular cases. Thirdly, the use of ontologies allows a more open notion of adaptation. This is due to the designer of the generic workflow will be who establish the particular adaptation restrictions of each workflow. Lastly, the specification of workflows in ontologies, where each element has a clearly defined purpose, facilitates the development of the set of propagation operations that are necessary to broadcast any change from the generic workflow to its adapted workflows.

On this basis, HAM provides the methods and operations needed to deal with each of the four stages that constitute the HAP. For the first stage, it supplies the ontology *OntoMetaWorkflow* and the method that, using this ontology, allows specifying administrative workflows in two ontologies, called *OntoDD* and *OntoWF*. For the second stage, it defines the set of adaptation restrictions that can be established on a workflow specified in *OntoDD* y *OntoWF* ontologies. Moreover, it provides the rules that must be satisfied establishing these restrictions in order to consider the workflow as an adaptable workflow. The method that allows setting these restrictions in a correct manner is also provided. For the third stage, the Method defines the rules that must satisfy a workflow to be considered a properly adapted workflow of an adaptable workflow. Furthermore, it provides the complete set of hierarchical adaptation operations for specifying correctly an adapted workflow from an adaptable workflow. Lastly, for the fourth stage, it provides the complete set of hierarchical propagation operations. These operations allow that any change affecting the adaptable workflow, over time, can be hierarchically propagated to its adapted workflows.

The following subsections describe how to deal with each of the stages of the Hierarchical Adaptation Problem by means of the concepts and operations of the Hierarchical Adaptation Method.

2.1. First stage: specification of the generic workflow

For the first stage, the approach for representing administrative workflows in ontologies proposed in [7] and restructured in [8] has been updated in the Hierarchical Adaptation Method.

Thus, the specification of the generic workflow is going to be made using the OntoMetaWorkflow ontology. This ontology provides the framework for representing administrative workflows defining the common elements of them. Using OntoMetaWorkflow, an ontology engineer must build, in first place, the OntoDD ontology. This ontology will contain the relevant data of a particular domain and the users that could participate in the workflows of that domain. Secondly, the ontology engineer will specify the logic of the administrative workflow in an ontology called OntoWF. OntoWF will contain the particular properties of an administrative workflow together with its activities, the order of execution of said activities, what users, specified in OntoDD, can perform the activities and what properties of the workflow specified in OntoWF and what domain data specified in OntoDD can be shown or modified in an activity.

In order to solve the HAP, *OntoMetaWorkflow* has been extended with a set of elements to specify adaptation restrictions over a concrete workflow. They are used since the second stage and are detailed in the next subsection. The elements of OntoMetaWorkflow used in this first stage are the specification elements. These elements serve to specify the activities of the workflow, the users that will perform the activities and the data used in the entire process. These elements together with the way they are used in *OntoDD* and *OntoWF* ontologies are shown in **Fig. 1**. A detailed explanation of these elements and the method for specifying administrative workflows in ontologies using them is available in [8].

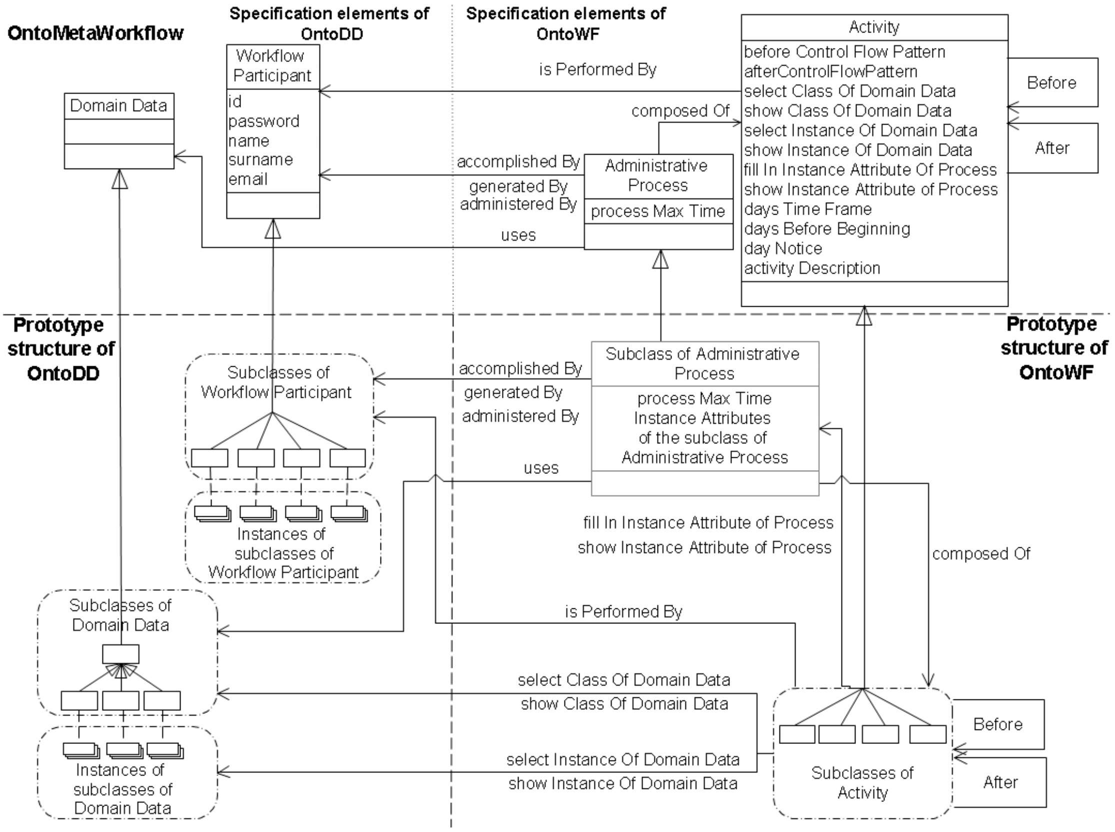
OntoMetaWorkflow
Specification elements of OntoDD
Specification elements of OntoWF
Activity
before Control Flow Pattern
afterControlFlowPattern
select Class Of Domain Data
show Class Of Domain Data
select Instance Of Domain Data
show Instance Of Domain Data
fill In Instance Attribute Of Process
show Instance Attribute of Process
days Time Frame
days Before Beginning
day Notice
activity Description
Before
After
Domain Data
Workflow Participant
id
password
name
surname
email
is Performed By
composed Of
accomplished By
generated By
administered By
uses
Administrative Process
process Max Time
Prototype structure of OntoDD
Prototype structure of OntoWF
Subclasses of Workflow Participant
Instances of subclasses of Workflow Participant
Subclass of Administrative Process
process Max Time
Instance Attributes of the subclass of Administrative Process
accomplished By
generated By
administered By
uses
fill In Instance Attribute of Process
show Instance Attribute of Process
composed Of
is Performed By
Subclasses of Domain Data
Instances of subclasses of Domain Data
select Class Of Domain Data
show Class Of Domain Data
select Instance Of Domain Data
show Instance Of Domain Data
Subclasses of Activity
Before
After

Fig. 1. Specification elements of OntoMetaWorkflow.

Moreover, in order to deal with the rest of the stages of the Problem, 24 basic modification operations (not shown in this work), has been developed. These operations can be applied on a workflow specified using *OntoMetaWorkflow* keeping the correctness of the workflow. They allow adding and removing different elements and increasing and reducing time frames of the activities and the entire workflow.

A BPMN diagram with the activities of the Spanish process to apply for the inclusion on the Rare Disease Registry is shown in **Fig. 2**. Moreover, the diagram also shows the process properties used in every activity and the restrictions related to the days available. BPMN has been used in the figures of the illustrative example in order to simplify their understanding.

As can be seen, this workflow is composed of five different activities. In the first one (*Apply for Inclusion*) a patient initiates the process by indicating her symptoms. In the second one, an *Attending Physician* makes a first diagnostic report. Next, this report and the patient symptoms can be assessed by a *Public Consultant* or a *Private Consultant* (as the patient whishes) who must make another diagnostic report. Finally, the *Rare Disease Medical Commission* will issue a final verdict about the inclusion or not of the patient on the registry.

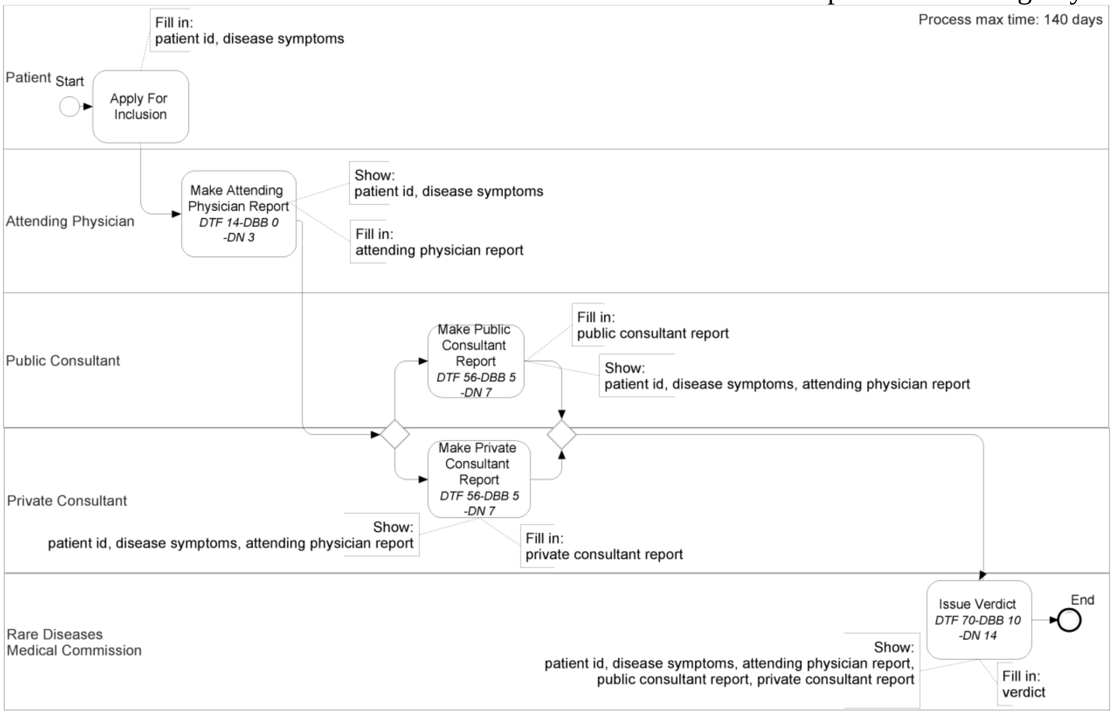
Process max time: 140 days
Patient
Start
Fill in:
patient id, disease symptoms
Apply For Inclusion
Attending Physician
Make Attending Physician Report
DTF 14-DBB 0 -DN 3
Show:
patient id, disease symptoms
Fill in:
attending physician report
Public Consultant
Make Public Consultant Report
DTF 56-DBB 5 -DN 7
Fill in:
public consultant report
Show:
patient id, disease symptoms, attending physician report
Private Consultant
Make Private Consultant Report
DTF 56-DBB 5 -DN 7
Show:
patient id, disease symptoms, attending physician report
Fill in:
private consultant report
Rare Diseases Medical Commission
Issue Verdict
DTF 70-DBB 10 -DN 14
End
Show:
patient id, disease symptoms, attending physician report, public consultant report, private consultant report
Fill in:
verdict

Fig. 2. Rare Disease Registry inclusion process.

2.2. Second stage: specification of adaptation restrictions

At this stage, the HAM uses the adaptation elements of *OntoMetaWorkflow* to establish what characteristics are essentials or not in a workflow. These elements allow indicating the adaptation restrictions of a generic workflow that must be satisfied in any possible adaptation of it. Moreover, the correct use of these elements will ensure that the adapted workflows are valid hierarchical adaptations from the generic workflow.

Three types of adaptation elements have been defined in *OntoMetaWorkflow*: *Mandatory*, Rigid and *Required*.

The first type of elements is used to indicate what activities (with *Mandatory Activities*), participants (with *Mandatory* Participants) and domain data and properties (with *Mandatory Data*) must always be in any possible adaptation of a workflow.

The second type is composed only of Boolean properties that are used within the activities to indicate that some of the relationships between them and the rest of elements cannot be modified in the adapted workflows. So, *Rigid Before* and *Rigid After* will indicate that the activities located immediately before and after of the activity must be always the same. *Rigid Participants* will not allow modifying the participants that can perform the activity. *Rigid Updateable Data* and *Rigid Viewable Data* will indicate that the domain data and the properties that a participant can select, modify or simply read in the activity cannot be changed in the adapted workflows. And, at last, *Rigid Days Time Frame* and *Rigid Days Before Beginning* don’t allow modify the time frame and the days before beginning the activity. This means that if an adapted workflow cannot satisfy these restrictions for any activity, it is necessary to remove such activity from the adapted workflow.

The last type is also used within the activities, but in this case it will only indicate minimum requirements and it will not limit the possibility of adding new elements in the adapted workflows. So, in *Required Before* and in *Required After* will be indicated which activities, at least, must be located before and after the activity and, in *Required Participants*, which participants, at least, must be available to perform the activity. In the same way, *Required Updateable Data* and *Required Viewable Data* will indicate which domain data and properties, at least, can select, modify or simply read a participant during the activity. As with the second type of elements, if any activity of the adapted workflow cannot satisfy these restrictions, it is necessary to remove such activity from the adapted workflow.

These elements together with the way they are used in *OntoDD* and *OntoWF* ontologies are shown in **Fig. 3**.

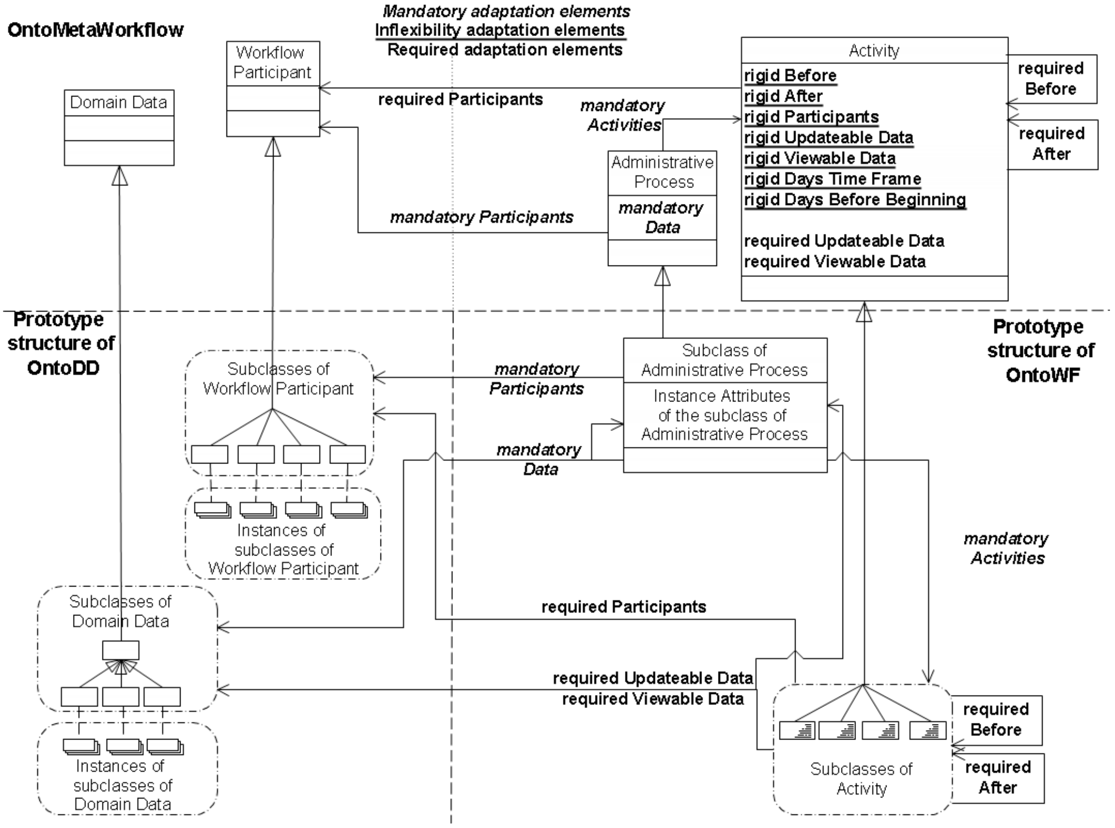

Fig. 3. Adaptation elements of OntoMetaWorkflow.

Using these adaptation elements, at this stage of the HAM is established the formal definition of adaptable workflow. So, an adaptable workflow will be a workflow correctly specified using *OntoMetaWorkflow* and whose values in the adaptations elements satisfy the next restriction: the workflow exclusively composed by the activities, participants, domain data, and process properties indicated in the *Mandatory* elements must be, by itself, a workflow correctly specified using *OntoMetaWorkflow*. This definition implies that it is not possible indicate arbitrary values in the adaptation elements; on the contrary, these values must be coherent. In order to facilitate the correct specification of adaptable workflows, HAM provides a method to correctly specify an adaptable workflow following the next steps:

1. Identifying which activities should be mandatory in any possible hierarchical adaptation. All of them must be included in the *Mandatory Activities* relationship. The initial and the final activity, at least, must be mandatory.
2. Stablishing the degree of adaptability of the position of every activity inside the workflow by means of the attributes *Rigid After* and *Rigid Before*, and the relationships *Required After* and *Required Before*.
3. Specifying optional adaptation restrictions for every activity related to the involved participants (*Rigid Participants* attribute and *Required Participants* relationship), the required data (*Rigid Updateable Data* attribute, *Required Updateable Data* relationship, *Rigid Visible Data* attribute, *Required Visible Data* relationship) and the time constraints considered (*Rigid Days Time Frame* or *Rigid Days Before Beginning* attributes).

The result of applying this method must be a workflow correctly specified using *OntoMetaWorkflow* that is also an adaptable workflow.

Moreover, it should be pointed out that, although the adaptation elements are used only in the *OntoWF* ontology, also involve the *OntoDD* ontology. On the one hand, it is mandatory that the participants included in the *Mandatory Participants* relationship must be in the *OntoDD* ontology of the adapted workflow. On the other hand, it is also mandatory that the domain data included in the *Mandatory Data* relationship must be in the *OntoDD* ontology of the adapted workflow.

Regarding our illustrative example, **Fig. 4** shows the adaptation restrictions considered for that workflow. Due to the great number of restrictions, only the most illustrative ones are explained. Basically, the first activity, the second one and the last one are mandatory but it may be possible to remove the *Make Public Consultant Report* activity or the *Make Private Consultant Report* activity or even both. It is also worthy to note that the first activity and the last one allow adding new fields to fill in because they haven't set the *Rigid Updateable Data* attribute as true. However, the rest of the activities do not allow this choice in any possible adaptation. Finally, it is not possible to change the type of participants who can carry out every activity because all of them have set the *Rigid Participant* attribute as true.

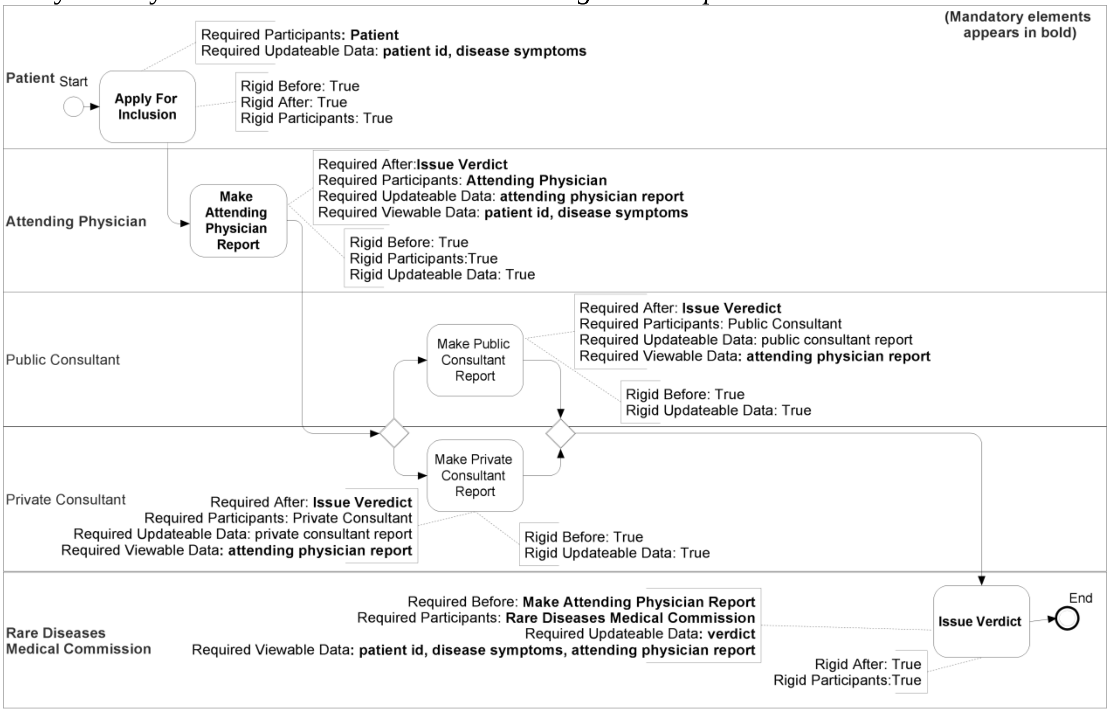
(Mandatory elements appears in bold)
Patient
Start
Apply For Inclusion
Required Participants: Patient
Required Updateable Data: patient id, disease symptoms
Rigid Before: True
Rigid After: True
Rigid Participants: True
Attending Physician
Make Attending Physician Report
Required After:Issue Verdict
Required Participants: Attending Physician
Required Updateable Data: attending physician report
Required Viewable Data: patient id, disease symptoms
Rigid Before: True
Rigid Participants:True
Rigid Updateable Data: True
Public Consultant
Make Public Consultant Report
Required After: Issue Veredict
Required Participants: Public Consultant
Required Updateable Data: public consultant report
Required Viewable Data: attending physician report
Rigid Before: True
Rigid Updateable Data: True
Private Consultant
Make Private Consultant Report
Required After: Issue Veredict
Required Participants: Private Consultant
Required Updateable Data: private consultant report
Required Viewable Data: attending physician report
Rigid Before: True
Rigid Updateable Data: True
Rare Diseases Medical Commission
Issue Verdict
End
Required Before: Make Attending Physician Report
Required Participants: Rare Diseases Medical Commission
Required Updateable Data: verdict
Required Viewable Data: patient id, disease symptoms, attending physician report
Rigid After: True
Rigid Participants:True

Fig. 4. Specification of the adaptation restrictions in the inclusion process.

2.3. Third stage: adaptation

For this stage, in first place the HAM establish the formal definition of an adapted workflow. So, an adapted workflow will be a workflow correctly specified using *OntoMetaWorkflow* that satisfy a set of implicit and explicit adaptation restrictions with respect to a given adaptable workflow.

On the one hand, the implicit restrictions do not depend on the adaptation elements. On the other hand, the explicit restrictions are related to the values set in the adaptation elements of the adaptable workflow.

In summary, a workflow will be a hierarchically adapted workflow from an adaptable workflow if:

- It is a workflow correctly specified using *OntoMetaWorkflow*.
- All the mandatory activities, participants, domain data and process properties of the adaptable workflow are in the adapted workflow.
- All the activities, mandatory or not, included in the adapted workflow, satisfy the restrictions established in their *Rigid* and *Required* attributes.

In order to specify correctly adapted workflows from an adaptable workflow, HAM provides 20 hierarchical adaptation operations, listed in **Table 1**. Those operations may be applied on an adaptable workflow to get the adapted workflow required. These operations use the basic modification operations of the first stage but adding a set of use restrictions that depend on the values of the adaptation elements of the adaptable workflow.

Table 1. Hierarchical adaptation operations.

Operation
Adaptation 1. Add New Activity
Adaptation 2. Remove Activity
Adaptation 3. Add New Workflow Participant
Adaptation 4. Remove Workflow Participant
Adaptation 5. Add New Domain Data
Adaptation 6. Remove Domain Data
Adaptation 7. Add New Process Property
Adaptation 8. Remove Process Property
Adaptation 9. Add Workflow Participant to Activity
Adaptation 10. Remove Workflow Participant from Activity
Adaptation 11. Add Domain Data to Select Class of Domain Data of Activity
Adaptation 12. Remove Domain Data from Select Class of Domain Data of Activity
Adaptation 13. Add Process Attribute to Fill In Instance Attributes of Process of Activity
Adaptation 14. Remove Process Attribute from Fill In Instance Attributes of Process of Activity
Adaptation 15. Add Domain Data to Show Class of Domain Data of Activity
Adaptation 16. Remove Domain Data from Show Class of Domain Data of Activity
Adaptation 17. Add Process Attribute to Show Instance Attributes of Process of Activity
Adaptation 18. Remove Process Attribute from Show Instance Attributes of Process of Activity
Adaptation 19. Reduce Days Time Frame of Activity
Adaptation 20. Increase Days Before Beginning of Activity

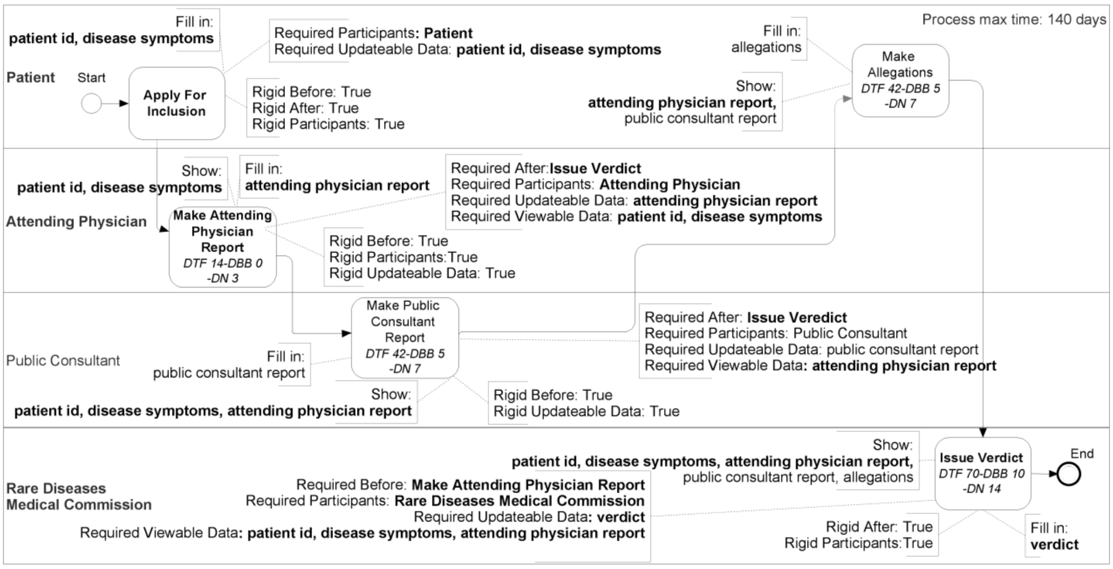
Process max time: 140 days
Patient
Start
Apply For Inclusion
Fill in: patient id, disease symptoms
Required Participants: Patient
Required Updateable Data: patient id, disease symptoms
Rigid Before: True
Rigid After: True
Rigid Participants: True
Fill in: allegations
Show: attending physician report, public consultant report
Make Allegations DTF 42-DBB 5 -DN 7
Attending Physician
Show: patient id, disease symptoms
Fill in: attending physician report
Make Attending Physician Report DTF 14-DBB 0 -DN 3
Required After:Issue Verdict
Required Participants: Attending Physician
Required Updateable Data: attending physician report
Required Viewable Data: patient id, disease symptoms
Rigid Before: True
Rigid Participants:True
Rigid Updateable Data: True
Public Consultant
Make Public Consultant Report DTF 42-DBB 5 -DN 7
Fill in: public consultant report
Show: patient id, disease symptoms, attending physician report
Required After: Issue Veredict
Required Participants: Public Consultant
Required Updateable Data: public consultant report
Required Viewable Data: attending physician report
Rigid Before: True
Rigid Updateable Data: True
Rare Diseases Medical Commission
Show: patient id, disease symptoms, attending physician report, public consultant report, allegations
Issue Verdict DTF 70-DBB 10 -DN 14
End
Required Before: Make Attending Physician Report
Required Participants: Rare Diseases Medical Commission
Required Updateable Data: verdict
Required Viewable Data: patient id, disease symptoms, attending physician report
Rigid After: True
Rigid Participants:True
Fill in: verdict

Fig. 5. Adapted inclusion process.

In the Rare Disease example, it may be possible for a Spanish region to apply the *Remove Activity* operation on the *Make Private Consultant Report* activity because the defined restrictions are satisfied. It may also be possible the incorporation of a new activity that allows patients to make new allegations on the diagnostic reports before the final verdict by using the *Add New Activity* operation. In **Fig. 5** a process with these adaptations is shown.

2.4. Fourth stage: propagation.

For this last stage, the HAM provides 60 hierarchical propagation operations that allow propagating the changes of an adaptable workflow to its adapted workflows. These operations are composed of two kinds of actions, firstly, the actions to change correctly the adaptable workflow and, secondly, the actions to propagate correctly the changes to the adapted workflows. These operations guarantee both the adaptable workflow and its adapted workflow will retain their status after applying them. These operations are not listed and nor explained in this work for the sake of brevity.

Regarding to the rare disease example, these operations may serve, firstly, to change the national adaptable workflow if the regulation that rules the process is changed and, secondly, to propagate those changes to the adapted workflows of the different regions (change synchronization). So, for example, if the national regulation changes the restrictions of the national process to state that *Issue Verdict* activity should always follow *Make Public Consultant Report* activity or *Make Private Consultant Report* activity, it would be necessary to apply the *Set Rigid After of Activity as True* operation on the two former activities. The application of this operation would imply to remove the *Make Allegations* activity added in the stage three to the adapted workflow as can be seen in **Fig. 6**.

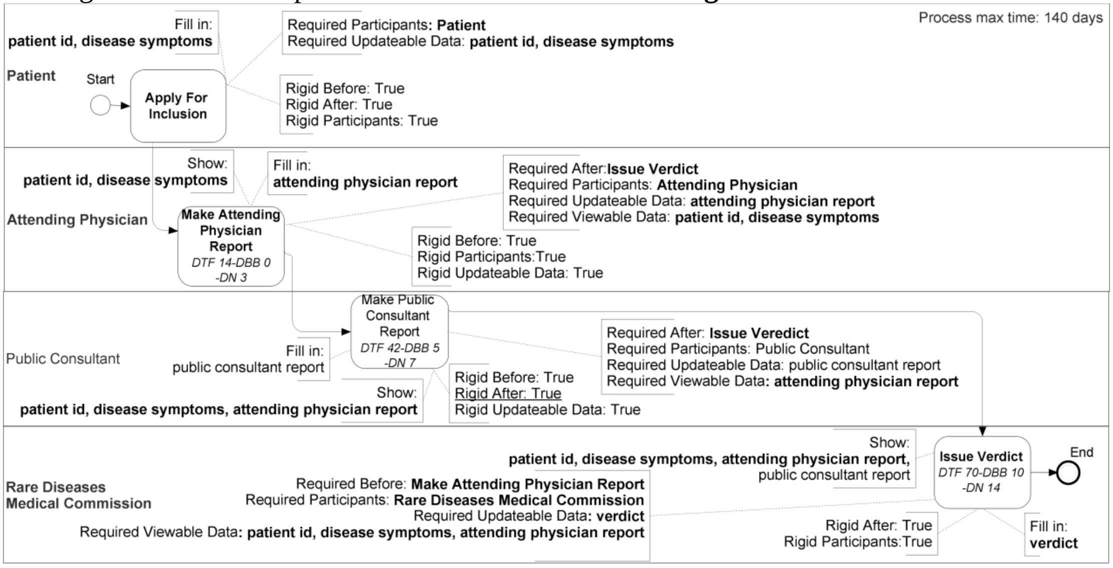
Fill in:
patient id, disease symptoms
Required Participants: Patient
Required Updateable Data: patient id, disease symptoms
Process max time: 140 days
Patient
Start
Apply For Inclusion
Rigid Before: True
Rigid After: True
Rigid Participants: True
Show:
patient id, disease symptoms
Fill in:
attending physician report
Required After:Issue Verdict
Required Participants: Attending Physician
Required Updateable Data: attending physician report
Required Viewable Data: patient id, disease symptoms
Attending Physician
Make Attending Physician Report
DTF 14-DBB 0 -DN 3
Rigid Before: True
Rigid Participants:True
Rigid Updateable Data: True
Make Public Consultant Report
DTF 42-DBB 5 -DN 7
Required After: Issue Veredict
Required Participants: Public Consultant
Required Updateable Data: public consultant report
Required Viewable Data: attending physician report
Public Consultant
Fill in:
public consultant report
Rigid Before: True
~~Rigid After: True~~
Rigid Updateable Data: True
Show:
patient id, disease symptoms, attending physician report
Show:
patient id, disease symptoms, attending physician report,
public consultant report
Issue Verdict
DTF 70-DBB 10 -DN 14
End
Rare Diseases Medical Commission
Required Before: Make Attending Physician Report
Required Participants: Rare Diseases Medical Commission
Required Updateable Data: verdict
Required Viewable Data: patient id, disease symptoms, attending physician report
Rigid After: True
Rigid Participants:True
Fill in:
verdict

Fig. 6. Adapted process after the propagation of the change.

3. Related Works

The Hierarchical Adaptation Problem suggests the review of approaches dealing with workflow inheritance and specialization. A summary of the literature review performed is presented next.

One of the first approaches was named *Maximal Execution Set Semantics* [9] [10] [11]. Basically, it proposes to define a generic process as an abstract process containing all the different possible variations of a workflow. Then every workflow adaptation can be conceived as a specialization of such process. From a practical point of view, this approach cannot be applied to deal with the HAP, mainly because that implies to know every possible adaptation scenario beforehand.

One of the most well-known approaches is van der Aalst [12] [13] [14], named *Workflow Inheritance approach*. It is focused on the migration of running instances of a modified workflow. A workflow is considered a specialization if a new activity is added to the original workflow and its behavior remains unaltered when such activity is hidden or blocked. This approach is then focused in runtime, while the HAP should be resolved on definition time.

Wyner y Lee [15] proposed an interesting extension of the *Workflow Inheritance* approach best suited to deal with HAP. The authors introduced the concept of “frozen” elements that constrains the space of possible extensions of the original workflow. The HAP solution method (HAM) proposed herein extends that concept in the sense that different restriction levels can be established during original workflow design. The HAM provides the engineer with the necessary features (1) to set the required activities, data and participants in any possible adaptation of the original workflow; (2) to indicate the original workflow characteristics that should remain unaltered in any possible adaptation; and (3) to indicate the original workflow characteristics that should appear in any possible adaptation but open to modification. As Wyner and Lee [15] points out, the main issue to apply those ideas is the limited expressiveness of traditional workflow specification languages, as WF-net. That is the rationale behind the use of ontologies why the HAM, instead of common workflow languages, to represent administrative processes.

Schrefl and Stumptner [16], inspired by the concept of object life-cycle, define a specialization by means of the concepts of subnet, observation consistency and invocation consistency. The main objective of that approach is to keep the substitutability of the workflows, i.e. the original workflow may be seamlessly substituted by the adapted workflow and vice versa. That property, though interesting, is not always a main concern on the definition of administrative processes. Such kind of processes is often adapted to change the type of data required (or produced) or to change the type of their participants. And those adaptations may derive on not interchangeable workflows.

Other authors [17] [18] use the concept of non-monotonic inheritance as the base for their approach of process specialization. The concept of non-monotonic inheritance has also been considered in the HAM, although somewhat limited by the adaptation restrictions defined for every workflow.

Choppy et al. [19] define a process specialization according to its activities and the partial order they define to specify a concrete runtime path, called a process run. Their concept of partial order has also been introduced into the HAM. The generic workflow designer can indicate which activities should precede (or succeed) a concrete activity.

In conclusion, as far as we know, none of the reviewed approaches may be suitable to solve the Hierarchical Adaptation Problem. All of them are just focused on behavioral aspects and they are not concerned about data and participant adaptation issues. Most of them specified a set of predefined adaptation rules so the designer has not the possibility to define a customized specialization based on the concrete domain of the process. And, finally, none of them specify how to keep in sync the different adapted workflows with the original (generic) workflow when changes are introduced into the generic workflow.

4. Conclusions

The Hierarchical Adaptation Method has been presented in this paper to solve the Hierarchical Adaptation Problem. This problem arises in healthcare institutions whose processes are often defined by a higher-level entity (regional or national governments, the WHO…). It happens when it is necessary to adapt administrative workflows from the higher levels of the organizations to the particular conditions of the lower levels where they will be used, but satisfying the restrictions established in the higher levels.

As a result of the study of the literature, it can be argued that, as far as we know, this problem has been mainly addressed using the experience and intuition of the engineers responsible for solving them without the support of any formal systematic method. The Hierarchical Adaptation Method has been specifically proposed to cover that necessity.

This method takes advantage of the benefits of specifying workflows by means of ontologies using the *OntoMetaWorkflow* ontology. So, it proposes a formal way to specify the adaptation restrictions of the administrative workflows together with the operations needed to adapt them to the particular characteristics of every application case, complying with these adaptation restrictions. Furthermore, this method also provides the operations needed to propagate every possible change in the generic workflows to their adapted versions. It is important to note that, using this method, the restrictions that must meet a generic workflow and their adapted versions will be determined by the law or regulation that rules the process, and not by a previous rigid definition of adaptation that does not take into account the particular characteristics of every process. Thus, the Hierarchical Adaptation Method solves the Hierarchical Adaptation Problem providing a flexible adaptation method.

A prototype implementation of this method has been developed in order to validate the approach in real scenarios; although its integration on real Health Information Systems needs further research. The prototype has given us the chance to apply the Hierarchical Adaptation Method on different domains, such as banking and health information systems. The initial results have been successful, but we are currently performing a thorough analysis to drive the next steps of the approach.

References

[1] M. S. Feldman and and A. M. Khademian, “Managing for inclusion: Balancing control and participation,” *Int. Public Manag. J.*, vol. 3, no. 2, pp. 149–167, 2000.

[2] J. Moore, J. Stader, A. Macintosh, A. Casson-du Mont, and P. Chung, “Intelligent task management support for new product development in the chemical process industries,” in *6th International Product Development Management Conference (PDM 99)*, 1999, pp. 787–796.

[3] D. Hollingsworth, “The Workflow Reference Model Document Number TC00-1003 Document Status - Issue 1.1,” Winchester, UK, 1995.

[4] S. McReady, “There is more than one kind of Workflow Software,” *Computerworld, November 2*, pp. 85–90, 1992.

[5] D. Georgakopoulos, M. Hornick, and A. Sheth, “An overview of workflow management: From process modeling to workflow automation infrastructure,” *Distrib. Parallel Databases*, vol. 3, no. 2, pp. 119–153, Apr. 1995.

[6] G. Alonso, D. Agrawal, A. E. Abbadi, and C. Mohan, “Functionality and Limitations of Current Workflow Management Systems,” *IEEE Expert Intell. Syst. Their Appl.*, vol. 12, no. 5, pp. 1–25, 1997.

[7] Á. E. Prieto and A. Lozano-Tello, “Use of Ontologies as Representation Support of Workflows Oriented to Administrative Management,” *J. Netw. Syst. Manag.*, vol. 17, no. 3, pp. 309–325, 2009.

[8] A. E. Prieto and A. Lozano-Tello, “Defining Reusable Administrative Processes Using a Generic Ontology,” *Int. J. Softw. Eng. Knowl. Eng.*, vol. 22, no. 2, pp. 243–264, 2012.

[9] G. M. Wyner and J. Lee, “Applying specialization to process models,” *Proc. Conf. Organ. Comput. Syst.*, pp. 290 – 301, 1995.

[10] T. W. Malone, K. Crowston, J. Lee, B. Pentland, C. Dellarocas, G. M. Wyner, J. Quimby, C. S. Osborn, A. Bernstein, G. Herman, M. Klein, and E. O’Donnell, “Tools for Inventing Organizations: Toward a Handbook of Organizational Processes,” *Manage. Sci.*, vol. 45, no. 3, pp. 425–443, Mar. 1999.

[11] J. Lee and G. M. Wyner, “Defining specialization for dataflow diagrams,” *Inf. Syst.*, vol. 28, no. 6, pp. 651–671, 2003.

[12] W. M. P. Van Der Aalst and T. Basten, “Life-Cycle Inheritance: A Petri-Net-Based Approach,” in *Proceedings of the 18th International Conference on Application and Theory of Petri Nets*, 1997, pp. 62–81.

[13] W. M. P. Van Der Aalst and T. Basten, “Inheritance of workflows: an approach to tackling problems related to change,” *Theor. Comput. Sci.*, vol. 270, no. 1, pp. 125–203, 2002.

[14] W. M. P. Van Der Aalst, “Inheritance of Business Processes: A Journey Visiting Four Notorious Problems,” in *Petri Net Technology for Communication-Based Systems - Lecture Notes in Computer Science*, vol. 2472, H. Ehrig, W. Reisig, G. Rozenberg, and H. Weber, Eds. Springer Berlin / Heidelberg, 2003, pp. 383–408.

[15] G. M. Wyner and J. Lee, “Applying Specialization to Petri Nets: Implications for Workflow Design,” in *Business Process Management Workshops. BPM 2005 International Workshops, BPI, BPD, ENEI, BPRM, WSCOBPM, BPS, Nancy, France, September 5, 2005. Revised Selected Papers*, 2005, vol. 3812, pp. 432–443.

[16] M. Schrefl and M. Stumptner, “Behavior-consistent specialization of object life cycles,” *ACM Transations Softw. Eng. Methodol.*, vol. 11, no. 1, pp. 92–148, 2002.

[17] A. Bernstein and B. N. Grosof, “Beyond Monotonic Inheritance: Towards Semantic Web Process Ontologies,” 190, 2003.

[18] S. Ferndriger, A. Bernstein, J. S. Dong, Y. Feng, Y.-F. Li, and J. Hunter, “Enhancing Semantic Web Services with Inheritance,” in *Proceedings of the 7th International Conference on The Semantic Web*, 2008, pp. 162–177.

[19] C. Choppy, J. Desel, and L. Petrucci, “Specialisation and Generalisation of Processes,” in *Proceedings of the workshop on Petri Nets and Software Engineering (PNSE’11), Newcastle, UK*, 2011, pp. 109–123.