Automatic Workflow Generation and Modification by Enterprise Ontologies and Documents

Babak Vaziri [1] *, Saeed Parsa[2], Fereidoon Shams[3]
(1) Engineering Department , Science and Research Branch , Islamic Azad University , Tehran , Iran
(2) Computer Department, Science and Technology University, Tehran , Iran
(3) Computer Department, Shahid-Beheshti University, Tehran , Iran
vaziribabak@yahoo.com; parsa@iust.ac.ir; f_shams@sbu.ac.ir

Received: 2011/05/12; Accepted: 2011/07/05 Pages: 63-72

Abstract

This article presents a novel method and development paradigm that proposes a general template for an enterprise information structure and allows for the automatic generation and modification of enterprise workflows. This dynamically integrated workflow development approach utilises a conceptual ontology of domain processes and tasks, enterprise charts, and enterprise entities. It also suggests a document conceptual model for the circulation process in the generated workflows and the display of process-related information and transactions with users in a document-based platform as a document management system (DMS). The conceptual model and suggested ontologies represent an abstraction of two hundred different processes applied in the case of seven different enterprises with different missions, along with three main propounded enterprise ontologies.

Keywords: *architecture, enterprise, document, ontology, service oriented architecture*

1. Introduction

In workflow management systems, the tasks of controlling and coordinating the process implementation have been adapted from human activities. The automation of control functions and the coordination of processes are made possible by structuring the characteristics of workflows in an appropriate and accurate manner. Their tasks and intermediate executive conditions are defined in a very detailed manner by an expert who is competent in the operating environment. Although a variety of methods have been proposed for the configuration, the relevant modeling is still done manually, for which various graphic tools are made, while the construction and reconstruction of workflows and the putting together of relevant components are done manually and statically.

On the other hand, the usual process-based workflow involves a series of specific, static tasks suited to supporting well-structured business processes. These tasks form a workflow with small changes to their possible executive scenarios.

This feature of the present workflow systems causes them to be used for basic, primary applications in accordance with permanent changes in enterprise workflows when environmental changes and needs make such changes necessary. Unfortunately, most enterprise workflows in connection with the present processes are dynamic and diverse in nature [7], making it expensive to cover a variety of scenarios.

The proposed method, which is based on methods for designing knowledge-based workflows, consists of three different ontologies: process, enterprise structure, and entity.

The process ontology is used to formulate enterprise action rules and operations as well as the general and hierarchical structure of processes. The enterprise structure ontology is based on determining the structure of the divisions, the enterprise chart, and the positions included in them. Enterprise entities are taken into consideration to cover important information resources and parts of the enterprise.

This paper presents a model for automatically generating and modifying workflows based on the above ontology and includes an algorithm for generating the necessary workflows.

The remaining parts of this paper are organized as follows: Section 2 presents the previous works and framework for the research. In section 3, the common structure of information within different enterprises is surveyed. Section 4 presents a new ontology for the common parts of enterprises as a chart, process, and entities. In section 5, the document ontology is presented. Section 6 presents the proposed algorithm. A summary of the paper is given in section 7.

2. Related Work

Currently, considering outside enterprise communications, it is possible to automatically generate workflows across multiple enterprises [1,15,3]. The main challenge is to create the necessary conditions for the automatic generation of workflows within the enterprise. For this purpose, general formats for describing the processes, information structure, and documentation to be shared are required. In this regard, comprehensive enterprise models such as TOVE, CYC, and Enterprise Ontology (EO) provide enterprise reference models that cover most of the existing concepts in enterprises [2,12].These ontologies have defined most enterprise concepts and terms in a massive and comprehensive way, taking into account the relations between them. Such ontologies have provided a common language and platform for the terms used in enterprises, allowing the exchange of information between designers/software engineers and users/operating environment experts.

These ontologies, however, do not automatically include the necessary details to implement workflows. In addition, they are regarded more from the data and information perspective than from the operation perspective, to which not enough attention has been paid [8].

The method proposed by Glushko and others [9,11] can be used to determine the structure of information exchange at the enterprise level.

Although numerous researchers have used concepts related to the ontology to implement workflows, it has no ability to automatically generate and modify workflows [14,17,4].

3. Information structure of enterprises

Studies conducted on seven enterprises and two hundred of their processes using three main enterprise ontologies indicate a common information structure between different enterprises and work environments [6]. From the information perspective, an enterprise can be configured by three concepts: entity, process, and enterprise chart.

The concept of entity, or enterprise resource, refers to each major component in the enterprise. Enterprises include a set of entities or resources, processes, and communications between them [10].

The term 'enterprise entity' refers to any concept with considerable information load at the enterprise. According to this definition, concepts such as employees, property, equipment, and customers are a part of enterprise entities.

Enterprise processes include some operational processes in which enterprise entities meet enterprise objectives in relation to each other [16]. The concept of an enterprise structure or chart suggests relationships between enterprise units and positions in each enterprise unit. The descriptions of the duties of the enterprise positions can be obtained from the relationships between enterprise processes and positions.

4. Enterprise ontologies

In regard to representing an enterprise as a set of ontologies, enterprises can be divided into three main categories, as follows.

4.1 Enterprise chart

An enterprise chart consists of an information structure representing units and positions within an enterprise. The current state of knowledge in the enterprise chart is represented by the enterprise structure ontology. This hierarchy of 'caption – sub caption' allows the identification of the information necessary for an enterprise chart. An enterprise position is placed in each of the leaves of the tree.

Definition 1. [Enterprise Chart Ontology]

An enterprise chart ontology (ECO) is denoted by a set of chart topics C = { c1; c2;…}. Each ci ∈ C is denoted by a triple < ID; REL; A >, where ID denotes a chart topic identifier, REL is a set of relationships that ci bears, and A is a set of enterprise agents, {a1, a2,...}, where each ai ∈ A is denoted by a triple of (ID, Actor, T), where ID denotes an enterprise role identifier. The actor is the owner of the role in the enterprise from the entity ontology (EO). T is a set of tasks, {t1, t2,...}, where each ti ∈ T is denoted by a task in the process and task ontology (PO) performed by an identified role.

4.2 Enterprise entities

An enterprise includes a number of resources, which are used during different processes [16,5] and can act in an active manner to perform operations on other resources (e.g. employees) or in a passive manner to accept operations on them by other entities (such as checks). The resources above, known as 'enterprise entities', constitute the main scheme of information in the enterprises.

Each entity in the enterprise has a set of characteristics that maintains information for the instances of that entity and includes a set of methods to specify actions associated with it. Based on studies using the enterprise ontologies as well as seven typical enterprises, the following can be cited as common parts in a variety of enterprises, each of which in turn will also include other entities, depending on its type, function, and mission. For example, an airline can consider an airplane to be one of its major entities.

Definition 2: [Entity Ontology]

An entity ontology (EO) is denoted by a set of entities EO = {e1, e2,...}, where each $e_i \in$ EO is denoted by a triple < ID; EA; EM >, where ID is an entity identifier, EA is a set of entity attributes, and EM is a set of methods that are associated with e_i, shown by EntityName.Methodname.

4.3 Process and task ontology

Enterprise tasks are considered to be the basic operations of an enterprise, that each of them can do by an enterprise position. An operation that is specific to one position shows a description of the duties of that role in the enterprise. For example, the confirmation of an application for insurance is a task dedicated to a specific role in an insurance company. An enterprise workflow consists of an explicit, detailed procedure to specify the relationship between entities, roles, and tasks based on enterprise rules.

The above rules are considered to be enterprise policies to be followed in the run of workflows. The following definition will be used to formulate the process structure, enterprise rules, and tasks.

Definition 3: [Process and task ontology]

The PO ontology is considered to be a set of processes, PO = {p_1; p_2;...}, where $p_i \in$ PO will be taken into account in the form of a triple < ID; REL; T >, where ID is a process identification, REL is a set of relations between P_i and other processes, and the characteristic of REL is of Is_A nature.

Nature Is_A indicates the subtypes of relevant processes. T is a set of enterprise tasks having relationship Has_A with their respective processes, which are in the form of T={t1; t2,...}, so that each $t_i \in$ T is defined as a 6-tuple {ID, PRE, A, RA, RUNAuto, MODE}. ID is the task id and PRE is a set of executive preconditions t_i, which is in the form of a set of process states created during the execution process.

RA is a set of data needed to perform tasks that should be provided either by the operating environment or the process users. These data are kept in the form of a pair (NAME, VALUE), where NAME is the relevant data’s name and VALUE is the value obtained from a user or operation environment.

The agents—whose output is a certain process status, such as issuing an invoice for a sale or validating the outputs of a customer—perform the tasks. RUNAUTO indicates the ability to automatically perform the task through hardware or software, without need for a human agent. Mode determines executive or conditional tasks.

Conditional tasks include those that examine a condition in an operational environment whose result is a possible output added to the list of process conditions for the condition and status of the relevant process. Conditional tasks, along with preconditions of tasks and a list of the current execution states of a running process, implement enterprise policies for a process. Ordinary tasks are like the executive functions common in enterprises, whose implementation adds the state of implementing the relevant task to a status list.

We use this process status list to fire the tasks on the list of process tasks to be done. A conditional task, T_i, is considered to be a binary (O, C), where O is a set of possible outputs for examining the relevant task condition and C is a logical expression, as follows:

Definition 4. [Expression]

Given an alphanumeric string variable v ∈ S (can be NAME of retrieved data from the operational environment or input), a function f ∈ F, which is a set of predefined functions, a relational operation op ∈ OP {<; =; ≥; ≤; >;∈; ≥; ≤; ≠; etc.}, and a literal l ∈ {S;N} for string S and number N (can be the value of retrieved data from the operational environment or input), the expression te is defined as follows:

1. (v op l) is an expression
2. (f(x) op l) is an expression
3. If te1 and te2 are expressions, then: ¬ te1, te1 U te2, and te1 U te2 are also expressions.

Function f represents predefined semantic and temporal functions from enterprise entities (EM), such as the data retrieval of a customer or the activation of a requested method such as SumOfLeaveDays(EmployeeID) ≤ 30.

For example, HasGuarantee has two types of output (has a Guarantee or has no Guarantee), and SumOfLeaveDays (EmployeeID) < 30 has two types of output (leave less than 30 days and leave equal to or greater than 30 days). In Figure 1, the process and task ontology hierarchy of a sample enterprise is shown.

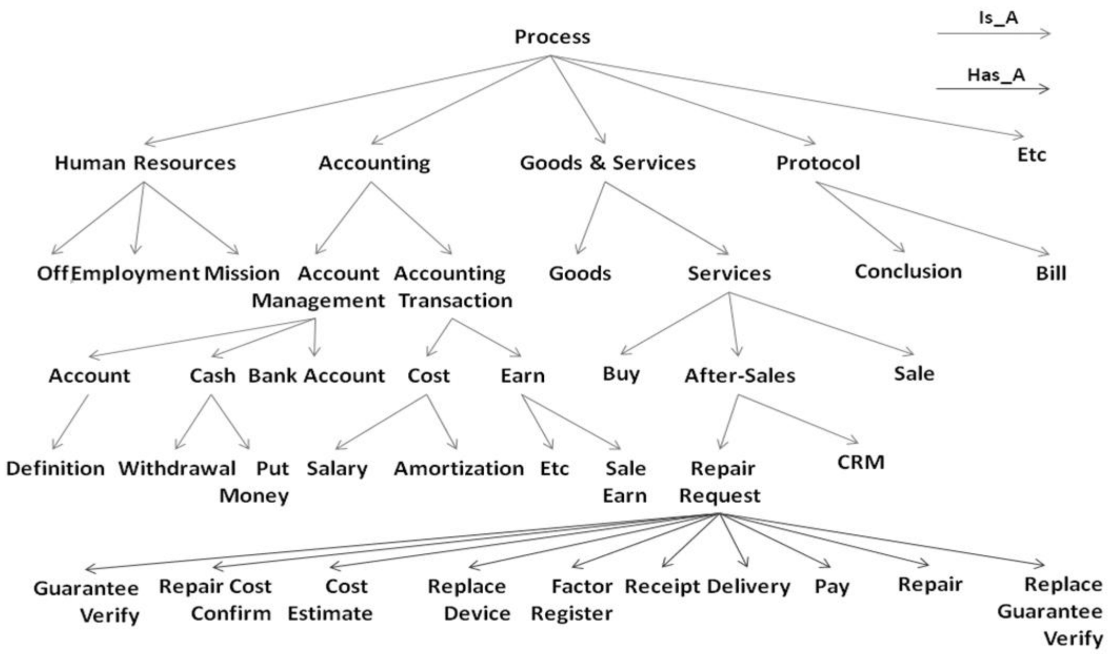

Figure 1. Process and task ontology hierarchy.

5. Documents in enterprises

Enterprises and companies most often utilise documents as a means of storing information. Documents may be used to preserve information about enterprise resources and important concepts. Documents are also applied as carriers for enterprise workflows and processes. The circulation of documents amongst enterprise agents, such as staff and working units, represents the kinetic structure of the enterprise processes. In the course of this circulation, each of the enterprise working units and agents, depending on his/her position and authorities, performs such actions as filling out informational items, adding annexes and files, producing new documents, confirming, signing, adding footnotes and descriptions, and sending the document to the next working units. In fact, documents as information carriers preserve the initial information for enterprise processes as well as carry the status and processing information throughout the enterprise processes and workflows.

Considering documents as information carriers within enterprise workflows and processes, a new framework to develop, manipulate, manage, and circulate documents within enterprises is proposed in the remaining parts of this paper. This framework maintains records of general enterprise process and workflow activities. By applying these general activities, mission-based processes and workflows can be easily implemented within enterprises.

5.1 Document ontology

A study was done on the processes in seven enterprises with different fields of activity (including air travel, shipping, port management, sales and distribution of medical equipment, insurance, car rental, and taxi service) for nearly two hundred processes and related documents, based on the type of relationship chart and the overall information structure of the document that were extracted, after the extraction of relevant documents, routines, and semantic rules of the environment.

There are various types of documents in enterprises in terms of their functions and users, which have a variety of functional ranges [9]. In this structure, each type of document has a series of specific operations and actions that are performed by staff, automated processing systems (computers), or a combination of both.

The fields of the document may be presented on the document or on the rows in that document. They are simple or reference-type: a simple field includes a caption and a value, and reference fields are a reference to a sample of enterprise entities. These fields support the conceptual model of the document to be linked to entities within the enterprise ontology.

Definition 5: [Document Ontology]

The ontology of document D is defined as a set of documents D = {d1, d2,...}, in which each $d_i \in D$ is defined as a triple {ID; P; DS}, where ID is a document ID and P is an ontology-related process whose executive carrier is the document. DS is a set of parts relating to a document as S = {s1, s1,...}, in which each $S_i \in DS$ is defined as a triple {TYPE, F, T}, where TYPE specifies the type of the relevant part, i.e. either DB or DR. The former indicates the main body of the document, and the latter is a detailed part of the document (document rows).

F is a set of data elements for the document in the format F = {f1, f2,...}, where $f_i \in F$ is shown by a 4-tuple {ID; NAME; VALUE; EKEY}, where ID, NAME, and VALUE are the identity, name, and value of the information item, respectively, and EKEY is the key of an instance of typical enterprise entities as a pointer.

DR is a set of rows in the document in the form of G = {g1, g2,...}, where each g_i is a set of information items. T is a set of enterprise tasks that is derived from the ontology of the process, and it acts on the DR or DS.

6. Proposed Algorithm

The following algorithm automatically generates and edits the workflows associated with the processes inside the enterprises the enterprises using inside the enterprises using the proposed ontologies .

```
Algorithm 1. [Generate customised workflow]
Input: enterprise ontologies
Output: workflow of enterprise processes
BEGIN
 StatusList := {};            /* list of process states
 TaskList :={};               /* list of process tasks
 RequiredAttributeList :={}    /* with paired format (name, value)
 Pi := ROOT                  /* root of process and task ontology
 ProcessList := Rel(Pi);       /* display below nodes of Pi for selection by user
 While ProcessList is not null
   TaskList := TaskList + T(Pi) ;  /* identify all tasks of Pi
   RequiredAttributeList := RequiredAttributeList + AttributeList(Pi)
   Answer := ShowList(processList)
      /* get answer from current user from a list of possible processes */
   Pi := answer
   StatusList := StatusList + Pi;            /* add process topic to state list
   ProcessList := Rel(Pi);    /* display below nodes of Pi for selection by user
 End while
 Doc := Generate a new Document();  /* generate a new Doc instance with main header
 For each A ∈ RequiredAttributeList
   If A is input parameter
     A.value := GetFromUser(A.Name)
    Else            /* for required attributes, which are derived from enterprise entities */
      Begin
       EKey := Entity.find(EntityType, EntityInstanceKey)   /* find required EntityInstance
       A.Value := GetFromEntity(A.name, EntityInstance);
      End
   /* attribute is an entity field or output of an entity method */
   DocHeader.AddField(A.Name, A.value, EKey);
 End for
 Doc.Location = CurrentUser; /* put process (document) on current user desktop
/* start process */
 While exist ti execute
   For each ti ∈ TaskList
    If ti.precondition is true
     RequiredAttributeListOfTask := RequiredAttributeList(pi);
               /* precondition of pi is true, pi fired */
     For each R ∈ RequiredAttributeListOfTask
      /* from required attribute list of process and task ontology */
       If A is input parameter    /* attribute is an entity field or the output of an entity method
        A.value := GetFromUser(A.Name)
       Else            /* for required attributes, which come from enterprise entities */
         A.Value := GetFromEntity(A.name, entityInstance);
       DocHeader.AddField(A.Name, A.value);
     End For
```

```
If Pi.Runauto = True
   /* Agent is an automatic agent (hardware, software, or external service)
  If Pi.mode = TASK then
   Pi.run;
   StatusList := StatusList + Pi;
  Else
   StatusList := StatusList + Eval(Pi);
/* the evaluate function returns an output from a list of results, which can be obtained from a verification of
the rule
Else
  Actor := FindInECO(pi);        /* find a person that can execute the required task based ECO ontology
(Enterprise
                          Chart) */
  If currentuser <> Actor        /* agent for execution of pi
   Doc.Location = Actor;       /* change location of process (document)
  If pi.mode = TASK then
    Pi.run;
    StatusList := StatusList + Pi;
   Else
     /* get answer from current user from a list of possible condition outputs */
     Wait();             /* for user action to take place
     Answer := ShowList(Pi);
     StatusList := StatusList + answer;
   End If
 End If
End While
End
```

When the algorithm is first run, while displaying the hierarchy of the processes and tasks, a series of operations are performed to discover the processes needed for using the process/task ontology with the presentation of the possible choices to the users, along with the data needed for this route. At the end of the respective loop, the executive part of the workflow begins for the actions and data collected, and appropriate tasks for the execution are then determined and performed.

These tasks may require data for Execution, which are obtained either from their operating environment or by asking the user.

After the proper position is diagnosed, the related person is chosen for the algorithm and the document is sent to his/her desktop. Once the required tasks are selected in accordance with the statuses in the process, the tasks whose prerequisites are established are executed, which in turn creates new states. This implementation adds new states to the process and causes the process to be continued. However, the required data are also requested from users for each stage of the process selection and task execution.

There are numerous general operations (such as approval and sending) that are used for documents, but some are especially defined for the document, such as the estimated cost of a repair. To implement this document, we can consider a service (with respect to SOA architecture) for each operation needed (which often has nearly the same structure in different enterprises), and we can implement it singly or in combination with other tasks.

In this way, when an enterprise has a difference in compliance between the operation process and the operational base of the document, the task section is changed according to the standard format of documents and easily replaces with the previous service.

7. Conclusion

In the plan presented, a comprehensive Platform was considered for automatically producing and modifying workflows, covering the common operational and information-related parts of an enterprise, and providing the highest possible level of reuse.

On the other hand, given the historical platform for using documentation and the role of documentation in the circulating processes within the enterprise, a conceptual form of data and ontology is available in exchange for documents, and this conceptual form of data and ontology allow the relevant platform to circulate the relevant processes.

In addition, an algorithm was presented for generating and modifying workflows. A service-based architecture was presented in exchange for the platform, which provides the possibility of editing operational needs for various enterprises, even in exchange for common parts of the platform. This platform, meanwhile, can be extended to inter-enterprise communications, and it can be used as a platform for communication.

8. References

[1] Andersson.M,Bergholtz.A.(2006),Towards a Reference Ontology for Business Models. Lecture Notes in Computer Science, Vol. 4215, pp. 482-496 .

[2] Berger.E, Ellmer. G, Quirchmayr. A. (1997), Zeitlinger.Comparison of Workflow methods, Evaluating Workflow Management Systems.In Proc. 8th International Workshop on Database and Expert Systems Applications, IEEE publication, pp. 412.

[3] Chun.V, Atluri. N.(2002),Domain Knowledge-based Automatic Workflow Generation. Database and Expert Systems Applications, Lecture Notes in Computer Science, Vol. 2453/2002, pp. 778-838.

[4] Corcho.A,Gómez.A.(2003),López-Cima.Automatic Generation of Knowledge Portals for intranets and Extranets. Lecture Notes in Computer Science, Vol. 2870, pp. 802-817.

[5] Dai , P., Daniel , M. and Weld, S.,(2010), Decision-Theoretic Control of Crowd-Sourced Workflows. 24th AAAI Conference on Artificial Intelligence.

[6] Eckert, K., Niepert, M. and Buckner, C.(2010), Crowd sourcing the assembly of concept hierarchies. 10th annual joint conference on Digital libraries – JCDL.

[7] Fadel.Gruninger.(1994),Infrastructure for Collaborative enterprises. In Proc. IEEE Enabling Technologies , pp. 117-128.

[8] Fiskvik.(2006), WISEMOD: A Framework that Combines Workflow, Ontology and Semantic Web Service Technologies. PhD Thesis.

[9] Glushko.McGrath.(1990),Document Engineering: Analysing and Designing Documents for Business. MIT Press.

[10] Hartig, O. and Zhao, J. , (2010) . Publishing and Consuming Provenance Metadata on the Web of Linked Data. 3rd International Provenance and Annotation Workshop (IPAW).

[11] Jang.A,Fekete.Greenfield,Nepal.S.(2006),An Event-Driven Workflow Engine for Service-based Business Systems. Enterprise Distributed Object Computing Conference, EDOC '06. 10th IEEE International, pp. 233-242.

[12] Lenat. R,Guha.(1990), Building Large Knowledge Based Systems: Representation and Inference in the CYC Project. Addison Wesley Pub.

[13] Lieg.(2008),Business Documents in a Service Oriented Context. In Proc. 10th International Conference on Electronic Commerce (ICEC08).

[14] Manolesco.(2001),Micro-workflow: A Workflow Architecture Supporting Copositional Object-Oriented Software Development, PhD Thesis.

[15] Meng.H,Lam.A,Helal.J,Xian.X.Liu.Y. (2006) .DynaFlow, a dynamic inter-organisational workflow management system. International Journal of Business Process Integration and Management, (IJCIS), Vol. 1, Number 2, pp. 101–115.

[16] Strohmaier. (2005) , B-KIDE: A Framework and a Tool for Business Process Oriented Knowledge Infrastructure Development. PhD thesis.

[17] Roger.B ,Luciano . A . (2006) Digiampietri.Automatic Generation of Workflow Provenance. Provenance and Annotation of Data Lecture Notes in Computer Science, Vol. 4145, pp. 1-9.
[18] Rosemann.W. (1993) , Enhancing the Expressiveness of the Bunge–Wand–Weber Ontology. Information Systems Journal, Vol. 3, Issue 4, pp. 217–237.