Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Semantics-oriented approach for information interoperability and governance: towards user-centric enterprise architecture management

Imran GHANI, Choon Yeul LEE, Sung Hyun JUHN, Seung Ryul JEONG
(*School of Business IT, Kookmin University, Seoul 136-702, Korea*)
E-mail: imransaieen@gmail.com; {cylee, juhn, srjeong}@kookmin.ac.kr

Received Aug. 17, 2009; Revision accepted Nov. 13, 2009; Crosschecked Jan. 19, 2010; Published online Feb. 27, 2010

Abstract: Enterprise architecture (EA) efforts focus on business, technology, data, and application architecture, and their integration. However, less attention has been given to one of the most critical EA elements, i.e., users (EA audiences). As a result, existing EA management systems (EAMS) have become old, large, content-centric document-repositories that are unable to provide meaningful information of use to the enterprise users and aligned with their needs and functional scope. We argue that a semantic technology based mechanism focusing on enterprise information and user-centricity has the potential to solve this problem. In this context, we present a novel ontology-based strategy named the user-centric semantics-oriented EA (U-SEA) model. Based on this model, we have developed a user-centric semantics-oriented enterprise architecture management (U-SEAM) system. Our approach is generic enough to be used in a wide variety of user-centric EAM applications. The results obtained show computational feasibility to integrate and govern enterprise information and to reduce complexity with respect to interoperability between enterprise information and users.

Key words: Enterprise architecture management system (EAMS), Ontology, Recommender system, User-centric, Semantic Web, Framework
doi:10.1631/jzus.C0910508 **Document code:** A **CLC number:** TP31; F270.7

1 Introduction

Enterprise architecture (EA) is a discipline that attempts to integrate, govern and analyze enterprise elements. These elements, once properly aligned together, create synergy in achieving enterprise objectives. Consistent with this concept, The Open Group Architecture Framework (TOGAF) defined EA as follows: "EA is about understanding all of the different elements that go to make up enterprise and how these elements inter-relate" (Schekkerman, 2004). This definition, further explicated by Aier and Marten (2006) and Ross *et al.* (2006), describes the important elements of EA including technology, process, business, and people. However, in the EA literature while the first three elements have been in the subject of many detailed studies (Schekkerman, 2004; DoDAF, 2007a; FEA, 2007) the last element (i.e., people or 'users') has been the subject of only a few (e.g., Zachman, 1987). Therefore, this paper concentrates on the 'user element' of EA, which is referred to as the human capital of enterprise. We adopt the term 'users' based on the ISO 13407 model (ISO, 1999). The notion of 'users' includes enterprise planners, analysts, designers, constructors, information asset owners, administrators, managers, customers, etc. (depending upon the enterprise stakeholders). We consider bringing the notation of 'users' into the delivery process of EA with respect to information interoperability as well as information governance, and their relationship with the enterprise users. The user-centricity based policy element of information governance is concerned with matters such as "Who owns information?" Blumenthal (2007) noted that user-centric EA represents a new and improved way of practicing this discipline. We extend this notion and argue that user-centric semantic technology oriented EA management (EAM) creates synergy that extends and expands the goal of actionable EA. It does this by providing useful information to the enterprise users through semantics-based interoperability and governance that is beyond the traditional EA (the classical business-IT alignment story). Semantics-based information interoperability and governance ensures that business missions are mapped accurately to the information products and services developed and that they are useful for users and reusable for the enterprise. This requires a logical integration of IT and business processes mapped with the users' needs (roles, domains, access rights, requirements, and preferences). In essence our aim is to promote collaboration among human actors and, enterprise and information assets, rather than only interoperability among computers (Fig. 1).

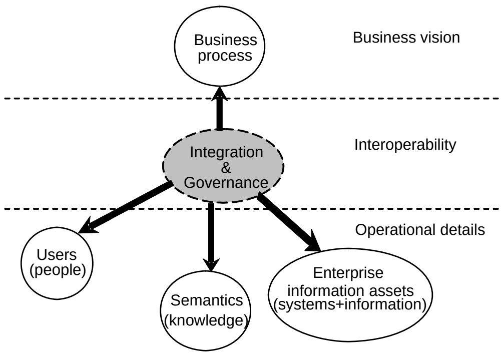


Fig. 1 Overview of vision

Through this approach, EA can be used to decrease the level of complexity to achieve interoperability between IT systems, business processes, and users. It may also help to increase understanding among human actors.

This paper is organized as follows. In Section 2, we discuss current issues. Section 3 highlights some of the limitations in existing EA management system (EAMS) approaches and sheds light on the limitations of conventional EA frameworks and current systems that follow these frameworks. Section 4 focuses on the capabilities of semantic technology. In Section 5, we describe the proposed model and developed architecture. The implementation, including a case study, is presented in Section 6. In Section 7, we briefly compare our proposed approach with existing approaches. Finally, in Section 8, we highlight the potential benefits of our approach and suggest directions for further studies.

2 Existing EAMS and current issues

Existing well-known EAM systems, such as select component architect, casewise and adaptive systems, are content-centric (mashing up heterogeneous data sources) and thus become large repositories. Such approaches are also time-consuming because of the need to manually map information to ensure that it is provided to the correct user. Zachman (2005) considered that time consumption was one of the most important issues in EA. Several questions relating to 'user-centric' focused information interoperability and information governance (ECU, 2008) have not been clearly answered in the existing EAM literature:

• Governance perspective: How can certain top or middle managers (planners, designers, constructors, administrators, managers, etc.) be enabled to understand the needs of enterprise users and the value (usefulness) of cross-functional information assets? How can the functional scope and responsibilities of enterprise users be defined, and classified? (Who is involved in what? Who owns what? And who wants what?)

• Interoperability perspective: How would certain users (information consumers) experience information interoperability among enterprise systems to perform their tasks?

• Interoperability and governance perspective: How can users' needs and expectations about the information assets that an enterprise has to offer or may provide be matched for effective operation? How can information resources (products/services) be accessed by certain users (policies and access rights)? Have alternative information resources been considered for enterprise users across different domains?

Existing EAMSs have limitations in answering these questions:

• Most EAMS models are either too complex or too simple (lacking required features) to implement and do not focus on solving the enterprise users' business problems on a runtime basis.

• Existing EAM methods have an unclear vision about information interoperability and governance with respect to the users; if nobody wants the specific resource, time should not be spent in developing it (Blumenthal, 2007). This highlights the need for semantics-based governance.

• Most of the EAMSs are content-based document repositories which lack practical methods for semantic interoperability at the user's context level. For this reason, users do not have a clear understanding of EA.

These issues are the logical outcome of frameworks that are data- or net-centric such as DoDAF, TOGAF, and Gartner; these frameworks offer guidelines only from a strategic IT point of view and hence only partially mention the notion of enterprise users.

3 Limitations of EAM systems/applications

A number of EAM systems partially adopt semantic technologies but do not pay enough attention to user-centric needs. User-centricity and its efficiency are not specific features of these systems (Table 1).

Table 1 Limitations of existing EAM systems

EA systems/ applications	Languages, frameworks	Features with user-centricity
Adaptive EAM	FEAF	User-centricity N/A
Enterprise repository	FEAF	User interaction
Modeling & validation tool set	DoDAF	User-centricity N/A
Corporate modeler enterprise edition	Casewise framework, FEAF, eTOM, DoDAF	Repository-based, Partial users control
Samu	Not specified	Data-centric, User-centricity N/A
Select component architect	Zachman	UML profile & component-based

N/A: not available

4 Semantic technology capabilities—potential toward EAMS

Semantic technology uses ontologies. An ontology is a symbolic logical model of some part of the meanings of the notions used in a field, i.e., those things that are universally true or true by definition. The key relationship in an ontology is 'subsumption' or 'kind-of'. Every instance of a 'subkind' must be an instance of the 'kind', without exception. Typically, ontologies are implemented in logic languages such as resource description framework (RDF) or Web ontology language (OWL) or in frame systems such as Protégé-Frames.

Ontologies have been used for enterprise systems (Cui *et al*., 1999; Qiu, 2006). The required capabilities described in Sections 2 and 3 can be obtained by semantically identifying users' requirements and their relationship with the enterprise information assets and storing their metadata information in the ontologies. An ontology-driven OWL/RDFS (OWL, 2008) semantic interoperability technique can be useful for achieving an enterprise's objectives, such as gaining more customers, reducing operational costs, increasing agility, or improving its ability to measure the benefits generated for end-users. The goal of semantic interoperability is to be able to recognize and process semantically equivalent information homogeneously, even if instances are heterogeneously represented, i.e., if they are structured differently, or are using different terminology systems. This equivalence factor needs to be robustly computable and not just human readable, in order for guidelines, alerting and decision support components to function effectively across enterprise systems. To achieve this, the next section describes our proposed approach, called the user-centric semantics-oriented EA (U-SEA) model.

5 The proposed U-SEA model

Typically, there are two extreme approaches for creating and managing an EA using frameworks: top-down (e.g., Zachman, 1987) and bottom-up (e.g., DoDAF, 2007b). In a mature, medium-to-large organization, both these approaches could take considerable time with little return on investment (ROI). The U-SEA model approach fosters horizontal integration and vertical adaptation in a process-oriented way. Our proposed model (Fig. 2) illustrates employing interoperability and governance as an enterprise process medium and also employing them at a technical level. Our U-SEA model (Fig. 2) focuses on user-centricity and, based on the vision presented in Fig. 1, has the following aims:

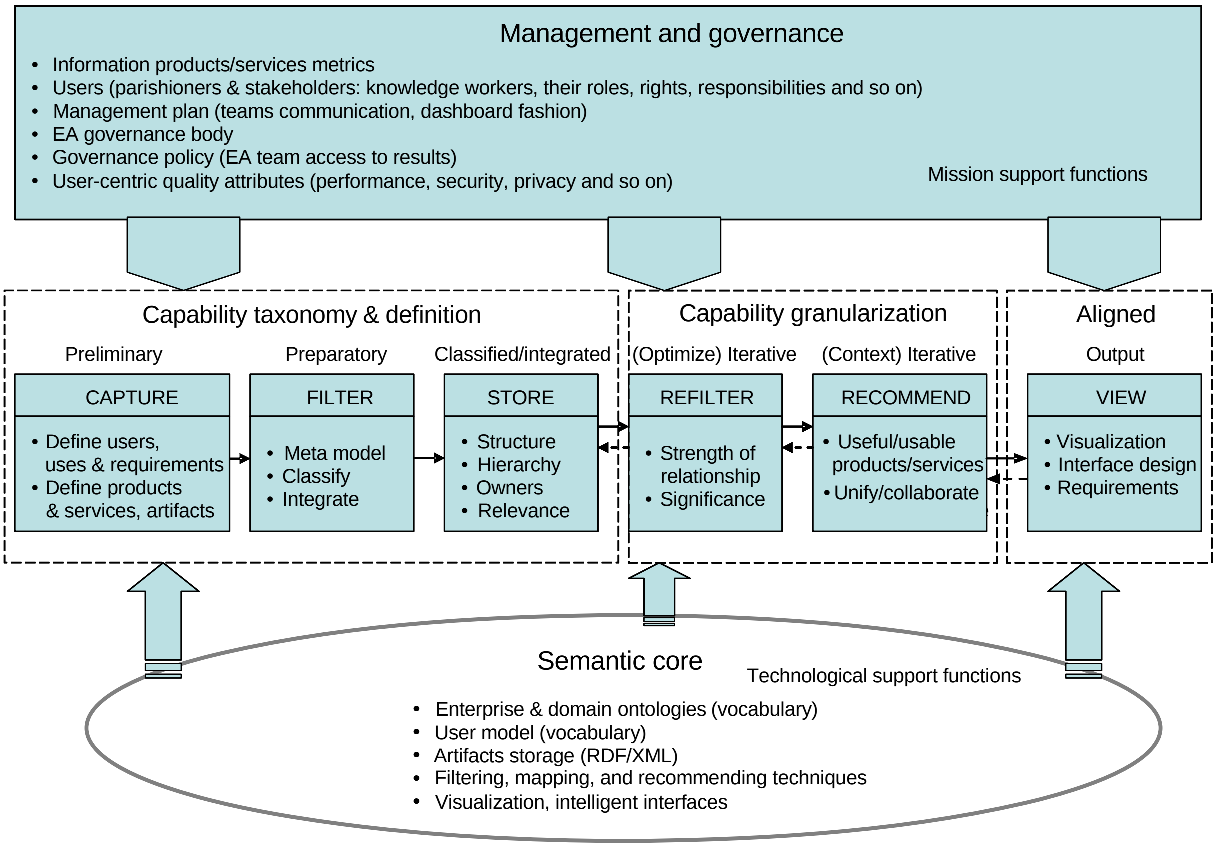


Fig. 2 Schematic of the user-centric semantics-oriented EA (U-SEA) model

• To build and preserve knowledge, by providing different stakeholders with an environment for storing information objects that is secure, scalable, and easy to use.

• To reuse users' information in different projects and not to start analysis from scratch every time, wasting time and cost.

• To reuse stored information about enterprise information assets for decision making.

The proposed U-SEA model represents an information interoperability and governance mechanism with respect to enterprise users. The mechanism of information governance is illustrated in the top layer cell while the semantic core is shown in the bottom layer. Processing interoperability is shown in the middle layer. The processes in the middle layer are divided into three core capabilities: capability taxonomy and definition of vocabulary, capability granularization, and capability alignment (aimed at establishing the best practices). The first cell in the processing layer, capability taxonomy and definition, deals with the users and information assets vocabulary capturing, processing, and storage. The second cell, capability granularization, deals with preprocessing of stored information in the context of users. The last cell, capability alignment, deals with the representation of mapped information to the users (see Section 6 for details). The proposed model packages together the business and IT processes that create a synergistic environment while collaborating with each other to benefit the enterprise users. Based on the U-SEA model, a logical architecture called U-SEAM has been developed. We propose that the eight cells (eight sub-component models) representing a directed graph (with arrows), are a good starting place to create and manage a user-centric EAMS.

5.1 U-SEAM

Fig. 3 presents an architecture called user-centric semantics-oriented architecture management (U-SEAM). It shows the interoperability of enterprise information assets with the users through ontological knowledge and machine processable semantic descriptions. The architecture is based on four ontologies: users ontology, enterprise ontology, domain ontology, and enterprise assets ontology.

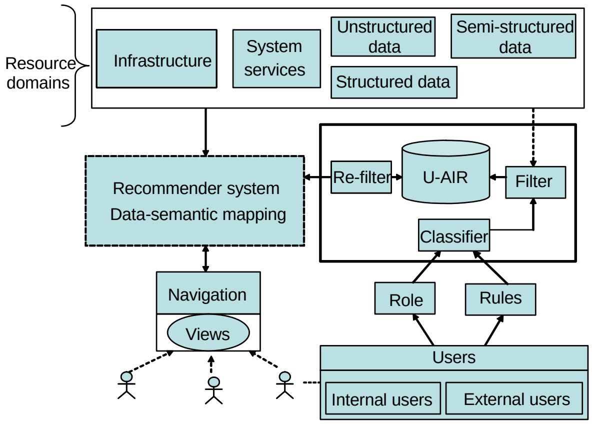


Fig. 3 User-centric semantics-oriented EA management (U-SEAM) architecture

As a first implementation step, we have developed a user model that deals with identifying and capturing metadata information of users (their types, roles, domains, preferences, expected needs, and access rights or policies) and storing this information in an ontology.

5.1.1 User model

To identify and capture the users' needs, we considered the ISO 13407 model (ISO, 1999). This model (Fig. 4a) comprises five stages, four of which are implicitly joined in a loop.

Although this process looks iterative, it need not be so. If required, it may be converted into a waterfall life-cycle model simply by going through only once (in which case, there is simply more focus on users' needs and evaluation of users' requirements than would normally be expected in a conventional system development). However, the true benefit of this model emerges when it is used to guide an iterative development process. In our approach, the information captured from users' requirements is populated in an OWL/RDFS ontology which helps present a shared understanding of the concept 'user' in a common vocabulary format called user model ontology (UMO). The UMO includes user type (internal and external enterprise users), user description, access rights, responsibilities, preferences, and domains to which a certain type of user or set of users may belong. In UMO, the enterprise internal users can be categorized into different sub-types such as top-, middle-, and operational-level management. Similarly, the external users can be categorized as customers, vendors and so on based on the enterprise needs. The UMO format helps in answering the questions such as "Who are the users? To which domain do they belong? What kind of information would different types of users expect to view from many different systems across the enterprise? What are the policies, workflows, and processes that a user needs to know in order to execute a business function? Who will be impacted if there is a technological change that may introduce new functions?" Fig. 4b shows a graphical representation of the user's view in use-case fashion.

After capturing the users' metadata information and storing it in UMO, the next step is to capture metadata about enterprise information assets and to store it in enterprise assets knowledge (EAK).

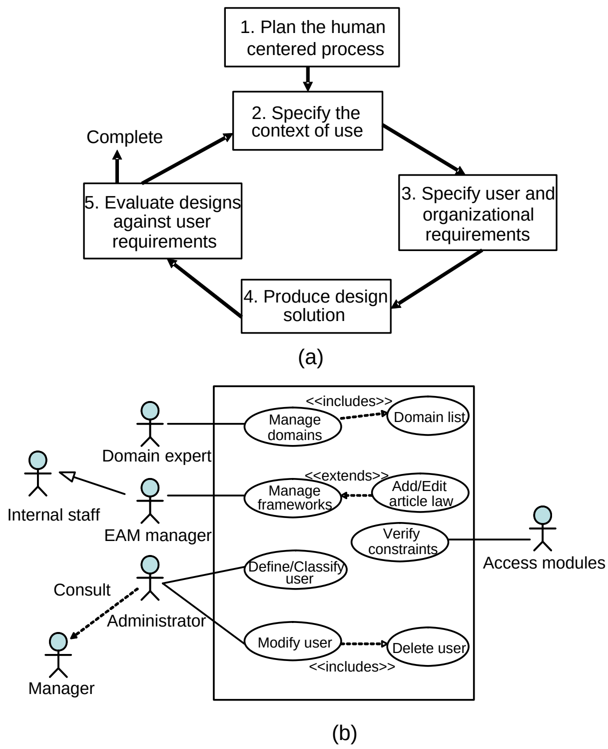


Fig. 4 (a) ISO 13407 model; (b) Use-case view of the user model

5.1.2 Enterprise assets knowledge

We propose the attribute-based approach for storing the available enterprise information assets in semantic form. The attribute-based approach emphasizes the availability of metadata (attributes) about the enterprise information assets. This involves capturing information regarding products, services, resources, applications, software, hardware, projects artifacts and so on. These assets should be provided with the metadata (attribute-based description of the asset) in a centralized knowledge base (U-AIR in our approach) resulting in the accurate description of resources. This knowledge base is further used to recommend enterprise resources to the users. The description depth of a resource can vary depending on its features. The metadata information can be described but the information is not restricted to the content of the asset (e.g., what is in the service package?). It may also include the domain, provenance (history up to the point of capture) and ownership of the asset, rights of access and many other features (depending on the set standards in the organization). Fig. 5 shows a conceptual sketch of the EAK.

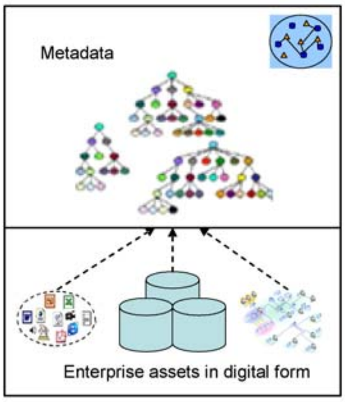


Fig. 5 Enterprise assets knowledge (EAK) model

The operational focus of the EAK is as follows:

• To store information about enterprise digital assets in OWL/RDFS metadata format. This contributes to improving the enterprise user's knowledge.

• To provide support for the availability of captured information from several domains' ontologies so as to relate them to the user's needs.

To create the structure for the EAK, we did not consider DOAP ontology (DOAP, 2007) or Dublin Core (DCMI, 1995) since DOAP is limited mainly to software projects while Dublin Core has limited expressiveness (Veltman, 2001).

5.1.3 Filtering process

The objective of a filtering strategy is to discover attributes of enterprise information assets (also referred to as items) that are defined in the EAK and appeal to the users, if defined in UMO. We used the similar direct-matching approach (Mena *et al.*, 1996) with an extension of ontology-based semantic annotation (SA). An ontology-based SA is SA that associates a resource to the concepts of reference ontology (RO) shown in Fig. 6. The filtering phase selects nodes and properties, and maps them with the UMO. We do this using the UMO based SA technique, which is an extension to the technique proposed in LEKS (2008). An enterprise information asset (e.g., an article document related to a new marketing strategy) is relevant if it is semantically associated with at least one concept in the UMO of a user whose role is a marketing strategist.

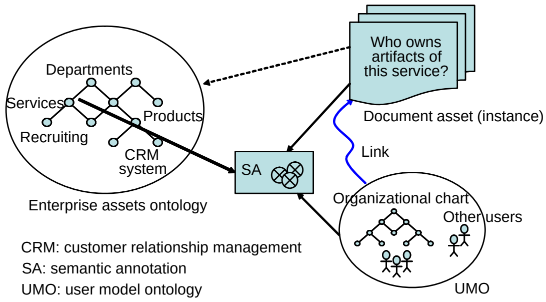


Fig. 6 Semantic annotation

This mechanism works as a concept explorer, as it detects those concepts that are closely related to the user's roles or expected needs by exploring the entities and semantic associations inferred using the following rule: Given a description of two concepts, we measure their semantic similarity by recursively comparing the existing relations among all the concepts involved in the description.

After this filtering phase, the information captured in UMO and EAK is mapped and stored in the U-AIR. Thus, U-AIR is the outcome of the filtering and mapping process.

5.1.4 Users-assets information registry (U-AIR)

U-AIR format is an OWL/RDFS based 'classified knowledge repository'. U-AIR is the centralized source of record in the hierarchical form of metadata that describes records as portfolios. More precisely, each portfolio describes relevance between enterprise information assets and enterprise user types and their related domains. Representing the enterprise information assets in a classified manner is a convenient way to structure and build the relationship repository, which improves processing efficiency.

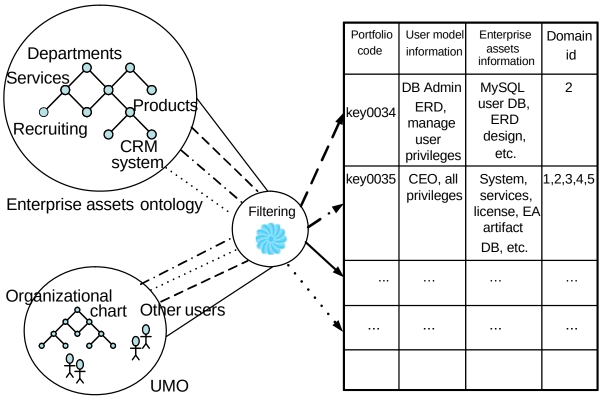


Fig. 7 Structural view of U-AIR
The arrows of the same line type show similarity between concepts in UMO and relevant information in EAK

U-AIR has four main columns. The first column 'portfolio code' represents tuple ID which is an integrated form of the user type and information related to it in the next three columns. In this context, U-AIR can be regarded as a 'centralized knowledge-base' operating within a holistic integrated and governance model. It helps to produce aligned outputs providing enterprise information assets and their relationships to the users. The inferred knowledge is processed in the reasoning and inferring phase.

5.1.5 Reasoning and inferring

We deploy a reasoning and inferring mechanism using OWL language that supports the object's properties with several relationship types such as inverse and transitive. This phase deals with the information assets' access rules that help in recommendations of information. It is based on the system processing automated suggestion algorithms.

• To access information assets and integrate them with identified user types based on their relevance.

5.1.6 Semantics-oriented approach based recommender system

In general, the recommender systems attempt to predict some resources (information products/services) in which a user may be interested. As noted by Cornelis *et al.* (2007), various explicit and implicit data collection methods have been used. Examples of an explicit data collection method in online systems such as amazon.com include: asking a user to rate an item, asking a user to rank a collection of items, asking a user to choose the better of two items, and asking a user to create a list of items that he/she likes. The implicit data collection is achieved mainly by keeping sufficient purchase records and a profile of clickdata. To generate the recommendations, a set of algorithms is developed to analyze the data and generate a list of recommended items. Four main types of recommendation approaches have been identified from the research literature: content-based, CF-based, knowledge-based, and a hybrid approach. Details of each recommender system are beyond the scope of this paper. However, a semantics-based recommendation mechanism has been developed that aligns user information in relation to the enterprise information resources (Fig. 8).

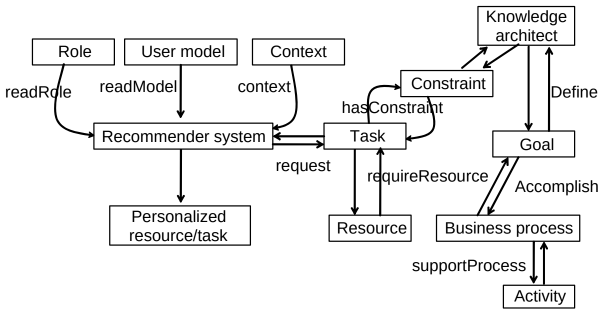


Fig. 8 Unified modeling language (UML) view of the recommendation process

The basic task of a recommendation process is to find all the records that are similar to single or multiple concepts, i.e., having the same attributes as those requested. An 'exact' mapping is the set of all the records that have exactly the same attributes requested. A 'similar' mapping comprises sets of records having some variations between their attributes and those required. In the case of a search, for instance, if the query requests year 2006, an exact match will be the set of all the records found having the year 2006. A 'similar' set will be all the records having 2005 or 2007 (one year distant, etc.). The mapping is done to a set of related attributes.

Given a description of two concepts we measure their semantic similarity by recursively comparing the most similar concept available in the ontologies.

Once the logical infrastructure is equipped, the system can then be executed and interacted with the users and with enterprise information on a real-time basis.

5.1.7 Output (view management)

This phase helps to organize information obtained and to integrate it with the user interface based on their portfolio defined in U-AIR. The operations include:

• Identify "How can information be aligned and presented in various layers of detail in such a way that it is meaningful to a broad array of different users in the enterprise?"

• Develop a high level business description for the end user, e.g., steps to share documents collaboratively, such as between users who interact in a relevant domain.

The output (view management) scenario in Fig. 9 is based on the domain and access levels enabling different types of users to decide at what level to navigate through an EA so as to view the architecture on strategic, functional, information, construction or distribution view, using context-driven interfaces.

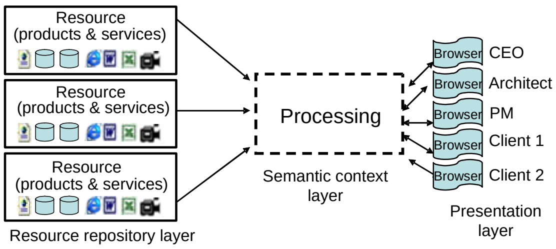


Fig. 9 View of user interaction

6 Prototypical implementation scenario

For our use-case study, we chose a career/job service provider company FemaleJobs.Net in Pakistan. The company has three regional offices in Pakistan and five departments: IT wing, career counseling, customer relationship management (CRM), marketing, and finance. Each department has more than ten silo applications, including at least two in each regional unit. The top management described typical issues: inconsistencies in data, difficulties in creating single user view across multiple silos (interoperability). The CEO commented that some of the expensive enterprise information assets (business procedures, models, services, systems built or bought) were unused in the departments, e.g., the recruiting system. The company was interested in applying an EAM solution to solve the issues of information governance and interpretability to save cost and time. We considered that this was a suitable company to test our approach; when there are 40 or 50 units instead of five, the problem to be solved would be much larger and more complex.

6.1 U-SEA model execution

To solve these issues, we executed the phases of the U-SEA model described in Section 5. For experimental purposes, we asked the teams (domain experts in each domain) to register some of the concepts about enterprise information assets (that are the instances from an EA perspective that the company had) related to each artifact. At the end, each enterprise asset had a unique rdf (an ID which could be resolved into a uniform resource identifier (URI)). Table 2 illustrates the tasks assigned to each domain expert.

Table 2 Team and defined tasks

Activity	Scope
Define users and needs	Admin specific
Define roles/rights (access)/responsibilities	Admin specific
Integrate useful info, usable to users	EA specific
Define user-centric quality attributes	Project specific
Define information assets	Information assets admin

6.1.1 Capture phase (preliminary)

We defined the users' information within the UMO based on the ISO 13407 model. We classified the users' ontology into top management, middle management, operational management, customers and service consumers who interact with the enterprise systems. This gives us an understanding of who are the users and what are their requirements and expectations from enterprise information assets. Afterwards, a user's profile was created that included title, type, job description, access rights, domain, roles (responsibilities), and preferences. The roles, domains, and access rights have a direct effect on the view management features. To create precise rules around users' names, we removed the role from the name and created a subclass of user for each role. Each role has a RoleCode such as rdf:RoleCode.CEO. Removing roles from property is a good way to keep ontologies reasonable for efficient semantic processing. In the context of a job search company, we created a number of subclasses for the user's role in the UMO. They included CEO, enterprise architect, project manager, recruiting manager, consultant, employers, and jobseekers. The following ontology (Fig. 10) represents a UMO format at the instance level.

```
<Person rdf:ID="Person_13">
    <hasMobilePhone rdf:datatype="http://www.w3.org/2001/
       XMLSchema#string">010-2939-1516</hasMobilePhone>
    <hasLastName rdf:datatype="http://www.w3.org/2001/
       XMLSchema#string">Ghani</hasLastName>
    <hasFirstName rdf:datatype="http://www.w3.org/2001/
       XMLSchema#string">Imran</hasFirstName>
</Person>

<owl:FunctionalProperty rdf:ID="hasFirstName">
    <rdfs:range rdf:resource=
       "http://www.w3.org/2001/XMLSchema#string"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/
       07/owl#DatatypeProperty"/>
    <rdfs:domain rdf:resource="#Person"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="hasTitle">
    <rdfs:domain rdf:resource="#Document"/>
    <rdfs:range rdf:resource=
      "http://www.w3.org/2001/XMLSchema#string"/>
    <rdf:type rdf:resource=
      "http://www.w3.org/2002/07/owl#DatatypeProperty"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="hasLastName">
    <rdfs:range rdf:resource=
      "http://www.w3.org/2001/XMLSchema#string"/>
    <rdf:type rdf:resource=
      "http://www.w3.org/2002/07/owl#DatatypeProperty"/>
    <rdfs:domain rdf:resource="#Person"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="hasUnitName">
    <rdfs:range rdf:resource=
       "http://www.w3.org/2001/XMLSchema#string"/>
    <rdf:type rdf:resource=
       "http://www.w3.org/2002/07/owl#DatatypeProperty"/>
    <rdfs:domain rdf:resource="#Organization"/>
</owl:FunctionalProperty>
<owl:FunctionalProperty rdf:ID="hasMobilePhone">
    <rdfs:range rdf:resource=
       "http://www.w3.org/2001/XMLSchema#string"/>
    <rdf:type rdf:resource=
       "http://www.w3.org/2002/07/owl#DatatypeProperty"/>
    <rdfs:domain rdf:resource="#Person"/>
</owl:FunctionalProperty>
```

Fig. 10 Users ontology of a user model ontology (UMO) format

Next, we developed the information assets ontology called EAK. Fig. 11 is the OWL based ontology.

Once we had captured the users and enterprise assets information in ontologies, we executed the filtering phase to map the relevant information.

```
<owl:Ontology rdf:about=" "/>
<owl:Ontology rdf:about="http://purl.org/dc/elements/1.1/"/>
<owl:AnnotationProperty
   rdf:about="http://purl.org/dc/elements/1.1/title"/>
  <rdfs:label xml:lang="eng-US">Title</rdfs:label>
  <rdfs:comment xml:lang="eng-US">keyword normally
  </rdfs:comment>
  <rdfs:description xml:lang="eng-US">list of keywords
  </rdfs:description>
  <rdfs:isDefinedBy
     rdf:resource="http://purl.org/dc/elements/1.1/"/>
</owl:AnnotationProperty>
<owl:AnnotationProperty
   rdf:about="http://purl.org/dc/elements/1.1/description"/>
  <rdfs:label xml:lang="eng-US">Description</rdfs:label>
  <rdfs:comment xml:lang="eng-US">detail about assets
  </rdfs:comment>
  <rdfs:description xml:lang="eng-US">useful for annotation
  </rdfs:description>
  <rdfs:isDefinedBy
     rdf:resource="http://purl.org/dc/elements/1.1/"/>
</owl:AnnotationProperty>
<owl:AnnotationProperty
   rdf:about="http://purl.org/dc/elements/1.1/type"/>
  <rdfs:label xml:lang="eng-US">Resource Type</rdfs:label>
  <rdfs:comment xml:lang="eng-US">classification of asset
  </rdfs:comment>
  <rdfs:description xml:lang="eng-US">category of asset
  </rdfs:description>
  <rdfs:isDefinedBy
     rdf:resource="http://purl.org/dc/elements/1.1/"/>
</owl:AnnotationProperty>
<owl:AnnotationProperty
   rdf:about="http://purl.org/dc/elements/1.1/identifier"/>
  <rdfs:label xml:lang="eng-US">Resource Identifier</rdfs:label>
  <rdfs:comment xml:lang="eng-US">ID/Code of asset
  </rdfs:comment>
  <rdfs:description xml:lang="eng-US">Asset Code
  </rdfs:description>
  <rdfs:isDefinedBy
     rdf:resource="http://purl.org/dc/elements/1.1/"/>
</owl:AnnotationProperty>
<owl:AnnotationProperty
   rdf:about="http://purl.org/dc/elements/1.1/coverage"/>
  <rdfs:label xml:lang="eng-US">Coverage</rdfs:label>
  <rdfs:comment xml:lang="eng-US">scope of asset
  </rdfs:comment>
  <rdfs:description xml:lang="eng-US">Domain</rdfs:description>
  <rdfs:isDefinedBy
     rdf:resource="http://purl.org/dc/elements/1.1/"/>
</owl:AnnotationProperty>
<owl:AnnotationProperty
   rdf:about="http://purl.org/dc/elements/1.1/rights"/>
  <rdfs:label xml:lang="eng-US">Rights Management</rdfs:label>
  <rdfs:comment xml:lang="eng-US">information about rights on
     assets</rdfs:comment>
  <rdfs:description xml:lang="eng-US">domain related users
  </rdfs:description>
  <rdfs:isDefinedBy
     rdf:resource="http://purl.org/dc/elements/1.1/"/>
</owl:AnnotationProperty>
<rdf:RDF/>
```

Fig. 11 Information assets ontology of enterprise assets knowledge (EAK)

6.1.2 Filter phase (preparatory)

We ran the filtering algorithm recursively to extract each user's attributes information from UMO and properties from EAK to build taxonomies. We then stored this knowledge in the U-AIR. The class instances and their properties (in OWL/RDFS format) that are relevant to the users were explored in traversal fashion:

• Role instances (there can be more than one role assigned to a user type) that are extracted from the UMO as they are successively traversed.

• Information assets ontology instances extracted as they are successively traversed.

In both traversals we ran the same sequence of instructions to explore the classes and their instances with different parameters of the user's roles and the enterprise assets:

• If a node is relevant then continue exploring its properties.

• Otherwise, disregard the properties linking the node reached to others in the ontology.

To implement the common attributes mapping process, we adopted the set theory:

$$U_i=\{u_{1i}, u_{2i}, \ldots, u_{ni}\}, \quad (1)$$

where i denotes the user and n_i varies depending on the user's roles.

$$D_j=\{u_{1j}, u_{2j}, \ldots, u_{nj}\},$$

where j denotes the assets and n_j varies depending on the assets related to a domain.

$$U_i \cap D_j=\alpha,$$

where α is any number of common attribute lists.

• The stronger the relationship between a node N and the user's portfolio, the higher the relevance of N.

• The relevance value was measured by ontology-dependent filtering criteria: (1) the length of the chain of properties established between N and class instances in the user's profile; (2) the implicit relationships detected by set theory concepts.

A particular concept such as a 'Web service' can be mapped. It has a concept 'Owner' who is always a 'Person'. An inverse property of the relations shows that 'Person' has the property 'isOwnerOf' which helps in finding the relevant resource or owner of similar services from the knowledge-base for the purpose of recommendation. In our model, EAMS essentially considers the information system's evolvement. Its inferring feature deals with the semantic association rules between relevant information services and assets to link the common characteristics obtained from heterogeneous business domains. The knowledge phase contains:

• Classes and properties instances.

• Semantic associations between specific user roles and enterprise assets (objects).

This knowledge is stored in U-AIR.

6.1.3 Store phase (classified U-AIR)—interoperability purpose

U-AIR stores and organizes the metadata (about enterprise artifacts, products, and resources) in a hierarchical order based on their relevance to the users in a tuple. This supports the integration of UMO and EAK metadata. This integration enables efficient interoperability of enterprise resources and users based on the relevance of metadata. The following ontology (Fig. 12) is a representation of U-AIR, which is a set of relevant concepts.

```
<owl:Class rdf:ID="Person"/>
<owl:Class rdf:ID="Organization"/>
<owl:Class rdf:ID="Document"/>
<owl:ObjectProperty rdf:ID="hasAuthor">
   <owl:inverseOf>
      <owl:ObjectProperty rdf:ID="isAuthorOf"/>
   </owl:inverseOf>
   <rdfs:range rdf:resource="#Person"/>
   <rdfs:domain rdf:resource="#Document"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasEmployer">
   <rdfs:range rdf:resource="#Organization"/>
   <rdfs:domain rdf:resource="#Person"/>
   <owl:inverseOf>
      <owl:ObjectProperty rdf:ID="hasEmployee"/>
   </owl:inverseOf>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="hasEmployee">
   <rdfs:domain rdf:resource="#Organization"/>
   <owl:inverseOf rdf:resource"hasEmployer"/>
   <rdfs:range rdf:resource="#Person"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="isAuthorOf">
   <owl:inverseOf rdf:resource"hasAuthor"/>
   <rdfs:range rdf:resource="#Document"/>
   <rdfs:domain rdf:resource="#Person"/>
</owl:ObjectProperty>

<preference rdf:ID="work_preference">
  <preference_options>emergency call service
  </preference_options>
</preference>
```

Fig. 12 U-AIR ontology

Since there are no standards to map the ontological data, we defined our own mechanism for experimental purposes. For the purpose of parsing metadata from ontologies, we categorized the data into tuples (subjects, predicates, and objects) with a domain category tag associated with each tuple (fourth column value in EAK). This aim of this step is to find the candidate instance collection for users. Next, the data in the domain category (which is not presented in this paper) was added into U-AIR. It showed that a few classes shared common knowledge in the domain ontology and thus were considered associated, such as DB admin role and associative EAK (Fig. 13b).

6.1.4 Re-filter phase (iterative)

The re-filtering phase is a one-step-further process to determine information objects related to the user's context (browsing behavior and search queries) for recommendation purpose. We used pre-defined constraints (mainly on access rights and domain associations) that guide the recommender system in applying logical filtration from EAK.

6.1.5 Recommendation phase (context)

This step is a context-based information recommendation phase, which follows the same steps as described by Ittoo *et al*. (2006). In user-centric EAMS the recommender system should provide personalized advice and enterprise assets to users about information products (items) or services in which they may be interested. Two main operations are performed: (1) Identify the users' types and extract information from the portfolio; (2) Analyze the user's browsing behavior, search queries, and recommend relevant information about available resources.

We identified two sources for collecting the user's profile information: (1) the knowledge about the user's browsing and search behavior, originating from the navigation of the EAM system; (2) the user's explicit portfolio identified in the EAK in the form of concepts, relations, and instances. The recommendations are generated by triggering rules whose conditions match the terms in users' inputs. To accomplish this process, the recommendation phase has two sub-functions as follows.

1. Metadata manipulation. This sub-function deals with the scanning of contents and information extraction. To enhance the efficiency of recommendation generation, the unstructured users' inputs are first transformed into structured representations. In the case of a search, all irrelevant terms, punctuation marks, and stop-words are removed. Alphabetical characters are converted to a single case, and words that have the same roots are normalized to their common roots, using a stemming algorithm. For instance, a user's input "preferred documents characteristics: recruiting process authored by co-worker recently" will be converted to "recruiting service document in 2009".

2. Computing relevance. To compute the relevance we defined *D* as the total number of product documentations describing resources R with type P products or type S services. The product documentations contained the access constraints C to be applied to the users with R roles. To determine if A is attribution of a resource R, the system first estimates the accuracy with which the set of product documentations for type P resource represents the entire corpus of documentation. The set of documentations for type E resource that contain A represents the collection of all the documentations containing A and is computed by the system. The next step is to compute the accuracy with which the set of documentations for type E resources that do not contain A, represents the collection of all documentations that do not contain A. The attributes are used to formulate classification rules of the form: IF <condition> THEN <action>, domain=*n*. The condition is a single word or word sequence, corresponding to an information asset's attribute. The action part is the product that the features in the condition represent. The rules indicate the accuracy of the condition that represents the asset in the action part, for example, the information gained by representing document Y with attribute 'jobs' and domain 'recruiting service'. After acquiring knowledge on document features, the recommendation system can perform its recommendation functions.

6.1.6 View phase (output)

For the example company, we classified the output views into the following order. The top-down description of view management represents the views in a hierarchy such as enterprise view, organization view, unit view, team view, and self view. These views are classified into the following categories: top management view, middle management view, and operational view.

• Top management information view: for the executive decision-maker who needs to capture a great deal of information in a visual way (models and diagrams) that can be quickly grasped, analyzed, and used to identify problem areas or to make decisions. This view will support the aggregation of information in order to create one big picture (overview), essentially hiding detailed information that is not relevant for decision making (Babu *et al.*, 2008). It is intended for CEO, directors, enterprise architects, enterprise program managers, strategic planners, etc.

• Midlevel information view: for the midlevel manager. The relationship between middle-level management and enterprise information is displayed. For example: how functions interrelate, how systems interoperate, and how information is exchanged among users. The project manager (PM) may, for example, want to develop a segment transition strategy. In that case, the midlevel information view can be provided.

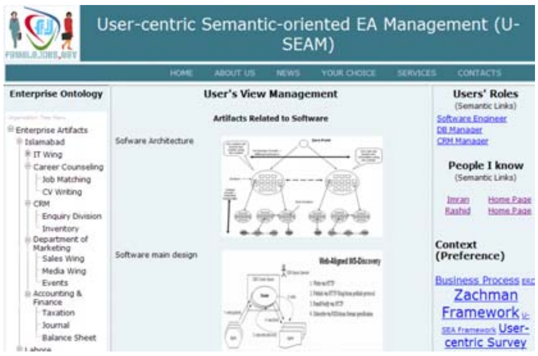
• Operational (detailed) view: drill down for products, services, inventories or catalogs to view information about each item. This view is intended for implementers, developers, employers, and job-seekers (external users).

Fig. 13a illustrates the output screen view of an EA manager in HTML format. The EA manager can see the enterprise and domain ontology (Fig. 13a, left menu) and their related artifacts (Fig. 13a, center). He/She can also see the user types related to those domains. In the meantime, the recommender system provides advice (keywords that are associated with some other enterprise assets; Fig. 13a, bottom right) in the form of a cloud that may be of relevance to the EA manager based on his/her assigned roles.

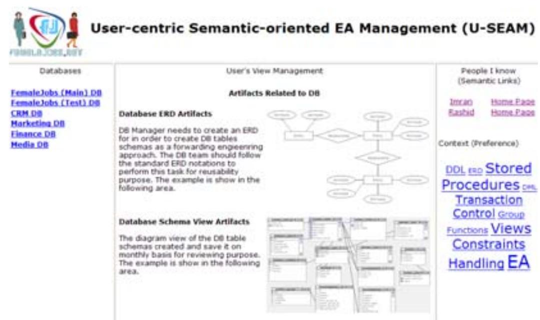
The view in Fig. 13b is shown to the DB admin based on his/her portfolio illustrating artifacts related to his/her domain.

One common feature between Figs. 13a and 13b is the right-side menu. The right-side menu in the browser is the information recommended by the recommender system. This information is based on the user's portfolio and browsing behavior and the search context.

During our experiments, the enterprise users noted that the proposed user-centric semantics-oriented EAM is useful because of its ability to help measure the benefits generated for the enterprise users. The focus on user-centricity (user's access, navigation, sharing views, and interoperability with relevant information) is the key to driving a high valued business since these people (as users or stakeholders) use systems and make decisions. Thus, their satisfaction as a 'happy user' is an important criterion to measure the benefits of EA.



(a)



(b)

Fig. 13 Output screen of (a) EA manager view and (b) DB manager view

7 Comparison with existing systems

Although other ontology-based techniques have been proposed, our UMO, EAK and U-AIR approaches differ greatly from these existing systems. For example, the foremost difference between our technique and the Samu system is not the information interoperability and information governance levels but is related to the stored knowledge used for EA modeling. Samu uses taxonomy based categories for representing the resources, whereas our technique makes use of the semantic relationships formalized in the ontology, enabling the system to carry out reasoning processes in order to extract additional information about the users' roles and enterprise domains. The similar limitation can be found in the system proposed in the Altova Enterprise Suite, which focuses on modeling. There, knowledge is discovered based on analyzing the hierarchical relationships, making the inference process even more complex.

8 Advantages of the U-SEAM approach to EAM

Our approach has the following advantages:

1. Responsive to the user's needs. The enterprise users may first want to ascertain what architecture and enterprise assets are available that may be relevant to their problem or situation. The users should have the following options available for finding architecture and enterprise data assets of interest:

• Ability to find information assets (including all artifacts data) that may be relevant or potentially applicable to a new architecture, software or business project.

• Ability to browse and search a catalog of enterprise assets available in all enterprise repositories.

2. Effective information governance beyond the traditional EAM. Based on our proposed framework, we achieve the user-centric based policy element of the information governance that is concerned with matters described in AIIM (2008), such as "Who owns information? Who wants what?" In this context ownership means 'Who is responsible and accountable for?' Thus, an example of information governance policy on ownership might be as follows: when a document is first created, it is owned by the author, or in the case of an email from outside the organization, the recipient. Later, after the document has been reviewed and accepted, the manager who commissioned its creation may become the owner. Once it is declared a record, it is owned by the chief information officer (CIO) on behalf of the business. The U-SEA model is actionable by all enterprise users, not just IT people, which helps to enhance business planning and decision-making. One of the primary missions of the U-SEA model is to improve IT planning and information governance with respect to the enterprise policy. The vision is to make information transparent to enable better decision-making. This is the value proposition that helps in developing and maintaining organizational information assets.

9 Conclusion and future work

As enterprise architectures (EAs) are often overly complex for their intended target audiences, the U-SEA model aims to decrease complexity and to increase accuracy and efficiency of information interoperability and governance. The proposed U-SEA model approach simplifies communications between the EAM's users including strategists, managers, analysts, designers, and constructors through flexible, customizable view management interfaces. The stakeholders can expand their expertise to move from 'view everything' to 'interact with relevant services' and into high-performance OWL/RDFS semantic technologies. The ultimate objective is to increase user productivity and satisfaction. In future, we intend to study the efficient quality attributes perspective in user-centric EAMS as well as to establish a user reference model (URM).

References

Aier, S., Marten, S., 2006. Evaluating Integration Architectures: A Scenario-Based Evaluation of Integration Technologies. Springer Berlin Heidelberg, p.2-14.

AIIM, 2008. What Are the Components of an Information Governance Framework? Available from http://www.informationzen.org/group/governancecompliance/forum/topics/2043787:Topic:306 [Accessed on Sept. 28, 2008].

Babu, M., Aziz, S., Obitz, T., 2008. Findings from the Enterprise Architecture Survey. Executive Summary. Infosys. Available from http://www.infosys.com/offerings/IT-services/architecture-services/ea-survey/Documents/EA-survey-2008.pdf [Accessed on Apr. 25, 2009].

Blumenthal, N.A., 2007. User-Centric EA. US Coast-Guard, DoD Presentation Session. Available from http://www.information-management.com/infodirect/2008_70/10001224-1.html [Accessed on Mar. 1, 2009].

Cornelis, C., Lu, J., Guo, X.T., Zhang, G.Q., 2007. One-and-only item recommendation with fuzzy logic techniques. *Inform. Sci.*, **177**(22):4906-4921. [doi:10.1016/j.ins.2007.07.001]

Cui, Z., Tamma, M., Bellifemine, F., 1999. Ontology management in enterprises. *BT Technol. J.*, **17**(4):98-107. [doi:10.1023/A:1009655311119]

DCMI, 1995. The Dublin Core Metadata Initiative. Available from http://dublincore.org/ [Accessed on Feb. 15, 2009].

DOAP, 2007. Description of a Project. Available from http://trac.usefulinc.com/doap [Accessed on Dec. 12, 2008].

DoDAF, 2007a. Architecture Data Description, Volume III. Available from www.defenselink.mil/cio-nii/docs/DoDAF_Volume_III.pdf [Accessed on Aug. 4, 2008].

DoDAF, 2007b. Definitions and Guidelines. Version 1.5. Available from http://www.defenselink.mil/cio-nii/docs/DoDAF_ Volume_I.pdf [Accessed on Nov. 18, 2008].

ECU (Economist Intelligence Unit), 2008. The Future of Enterprise Information Governance. White Paper, The Economist. Available from http://www.emc.com/leadership/business-view/future-information-governance.htm [Accessed on Jan. 15, 2008].

FEA, 2007. FEA Consolidated Reference Model Document. Version 2.3. Federal Enterprise Architecture Program Management Office, OMB. Available from http://www.whitehouse.gov/omb/assets/fea_docs/FEA_CRM_v23_Final_Oct_2007_Revised.pdf [Accessed on Oct. 23, 2008].

ISO, 1999. Human-Centered Design Processes for Interactive Systems. Available from http://www.iso.org/iso/catalogue_detail.htm?csnumber=21197 [Accessed on Oct. 24, 2008].

Ittoo, A.R., Zhang, Y., Jiao, J., 2006. A Text Mining-Based Recommendation System for Customer Decision Making in Online Product Customization. Proc. IEEE Int. Conf. on Management of Innovation and Technology, p.473-477. [doi:10.1109/ICMIT.2006.262208]

LEKS, 2008. Enterprise Ontology: A Position Statement. Available from http://crinfo.univ-paris1.fr/Missikoff-EnterpriseOntology-Introd.pdf [Accessed on Mar. 7, 2009].

Mena, E., Kashyap, V., Shet, A., Illarramendi, A., 1996. OBSERVER: An Approach for Query Processing in Global Information Systems Based on Interoperation across Pre-existing Ontologies. Proc. 1st IFCIS Int. Conf. on Cooperative Information Systems, p.14-25. [doi:10.1109/COOPIS.1996.554955]

OWL, 2008. Ontology Web Language. Available from http://www.w3.org/TR/owl-guide/ [Accessed on Nov. 17, 2008].

Qiu, R.G., 2006. Towards ontology-driven knowledge synthesis for heterogeneous information systems. *J. Intell. Manuf.*, **17**(1):99-109. [doi:10.1007/s10845-005-5515-z]

Ross, J.W., Peter, W., David, C.R., 2006. Enterprise Architecture as Strategy Creating a Foundation for Business Execution. Harvard Business School Press, Boston, Massachusetts, USA.

Schekkerman, J., 2004. How to Survive in the Jungle of Enterprise Architecture Frameworks? Trafford Publishing, Victoria, Canada.

Veltman, K.H., 2001. Syntactic and semantic interoperability: new approaches to knowledge and the semantic Web. *New Rev. Inform. Netw.*, **7**(1):159-183. [doi:10.1080/13614570109516975]

Zachman, J.A., 1987. A framework for information systems architecture. *IBM Syst. J.*, **26**(3):276-292.

Zachman, J.A., 2005. Enterprise Architecture Artifacts vs Application Development Artifacts. Zachman Institute for Framework Advancement. Available from http://www.zifa.com/cgi-bin/download.cgi?file_name=ZIFA05.pdf&file_path=/home/stagingArea/zifa.com/friends/ZIFA05.pdf&size=164365 [Accessed on Aug. 25, 2009].