ONTOLOGY-BASED DYNAMIC SERVICE COMPOSITION USING SEMANTIC RELATEDNESS AND CATEGORIZATION TECHNIQUES

Yacine Rezgui and Samia Nefti
Informatics Research Institute, University of Salford, M5 4WT Salford, UK

Keywords: Web services, Service composition, Ontology, Semantic Relatedness, Categorization.

Abstract: Organizations need to migrate their legacy systems to higher order applications capable of engaging in automated modes of collaboration to support distributed business processes. This requires a change of focus from intra-enterprise system integration through agreed data structures to inter-enterprise business process integration through smart composition of web-serviced applications. The paper presents an approach aiming at supporting ontology-based semantic composition of web-services to support distributed electronic business processes. This new generation of composite services is semantically coordinated in a secure, scalable, and resource-aware environment. Two services, at the heart of the service composition exercise are featured, namely: the semantic compatibility and categorization services.

1 INTRODUCTION

Service-oriented computing is becoming the prominent paradigm for leveraging inter and intra enterprise information systems, creating opportunities for smart organizations to provide value added services and products. The benefits of web services include the decoupling of service interfaces from implementation and platform considerations, the support for dynamic service binding, and an increase in cross-language and cross-platform interoperability (Ferris and Farrell, 2003). This new form of computing should move from its initial "Describe, Publish, Interact" capability to support dynamic composition of services into reinvented assemblies, in ways that previously could not be predicted in advance (Heuvel and Maamar, 2003; Rezgui, 2007a).

However, composed web service applications are not adaptive to change. If the requirements of the application change or need extending, the service composition needs to be re-specified from scratch. It is currently not possible to define and implement a web service composition once and use it in similar designs with some variations in a later stage. A more flexible approach allowing service re-use, extension, specialization should be supported. Also, web service composition methodologies have a focus on syntactic integration and therefore do not support automatic composition of web services. As highlighted in (Sycara et al., 2003), semantic integration becomes crucial for web services as it allows them to (a) represent and reason about the task that a web service performs, (b) explicitly express and reason about business relations and rules, (c) understand the meaning of exchanged messages, (d) represent and reason about preconditions that are required to use the service and the effects of having invoked the service, and (e) allow intelligent composition of web services to achieve a more complex service.

The paper describes a web services infrastructure, developed within the EU funded e-Cognos project (Rezgui and Meziane, 2005) and refined in the follow-up EU funded FUNSIEC project (Barresi et al., 2005), aimed at facilitating and supporting the execution of distributed business processes implemented through a coordinated composition and invocation of web-enabled, service-oriented, Enterprise Information Systems (EIS). This service infrastructure has been specified and implemented for the Construction sector, but with a view to be generalized and used across industry. First, the paper provides a conceptualization of the proposed service infrastructure. The latter establishes the middleware foundation as well as the necessary components, including services, enabling the support of distrib-

Rezgui Y. and Nefti S. (2007).
ONTOLOGY-BASED DYNAMIC SERVICE COMPOSITION USING SEMANTIC RELATEDNESS AND CATEGORIZATION TECHNIQUES.
In *Proceedings of the Ninth International Conference on Enterprise Information Systems - SAIC*, pages 113-120
DOI: 10.5220/0002369301130120

uted business processes. The paper then describes the different services developed to support intelligent service composition. Two of these services are then illustrated, namely: the semantic compatibility and categorization services. Finally the paper discusses the limitations of the research as well as future work.

2 SUPPORTING DISTRIBUTED E-PROCESSES

E-processes are typically designed, developed, and deployed by enterprises that want to compose internal capabilities with third-party capabilities, either for internal use or to expose them as (complex, value-added) e-services to customers. They can be described as the smart aggregation of services that are captured in the form of a Composite Service complying with well-defined business rules. They have the ability to be reused outside their scope and become generic services. Composite services are executed within shared workspace environments. A shared workspace refers to an online web environment involving authorized actors, united for a business or practice purpose. Access to data / information / knowledge from within a shared workspace is organized via dedicated elementary or composite services, invoked by actors through their assigned role(s). Shared workspaces can be defined at different levels of granularity; these can range from supporting collaboration within a complex construction project, to nurturing a small community of practice. Implementing service-oriented shared workspace environments involves three key generic "roles" defined at the service infrastructure level:

<u>Work Space Service Provider (WSSP):</u> This has the responsibility of managing the entire service infrastructure (e.g. servers, computer resources, services, etc…) and allocating shared workspace environments to potential construction clients to host their projects. This involves hosting the core infrastructure through provision of and access to both core services and Third Party Services (TPS). Core services refer to services necessary for the basic operation and management of services, including TPS that have the particularity of being provided by third party entities, namely, third party service providers. WSSPs, through the core services, have the capability to host multiple projects and to make available different services (both core and TPS) to workspaces. WSSPs can also play the role of application integrators, providing help and assistance to migrate legacy applications and legacy systems to Web services.

<u>Third Party Service Providers (TPSP):</u> These represent various companies, including software houses, interested in making their software application(s) accessible through a service-based middleware solution hosted and managed by the WSSP. These companies have their services published within the WSSP UDDI registry. Typically, these services would be geared to serving a particular purpose for the Virtual Enterprise (VE) to which they are being made available. Examples of services offered by TPSP include structural dimensioning service, HVAC simulation service, procurement service, and facility management service.

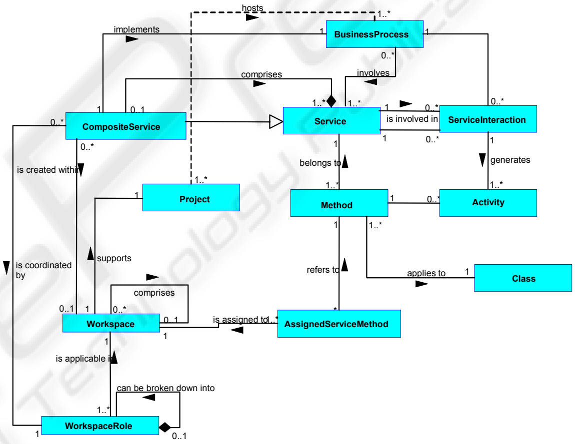

Figure 1: Conceptualization of Service Composition.

These services are offered by organizations that procure the service implementation, supply the description of the service, and provide related technical and business support. These can also represent service aggregators that consolidate multiple services into a new, single service offering.

<u>Work Space Clients (WSC):</u> The shared workspace will involve stakeholders representing construction companies collaborating within the context of a project. This collaboration is supported and enabled through the WSSP platform. While one company, acting in the capacity of Workspace manager, would configure and administer the Workspace, others would make use of the core and TPS services made available to the project.

The proposed conceptualization of service composition (illustrated in Figure 1) is based on BPEL. Business processes interact, on a peer-to-peer basis, with a set of services by invoking one or several of the methods they support. The way messages are exchanged between the business process and the Service Methods is described through the concept of Activity. Activities can then be combined into complex algorithms through the BPEL concept of Structured Activities. The proposed model supports the allocation of such services to Shared Workspace Environments. Methods represent API calls, or functionality, of such services.

3 ARCHITECTURE AND CORE SERVICE DESCRIPTION

In order to support e-Processes a layered service architecture that makes use of established work, initiatives and standards in the web services domain (including BPEL, WS-Security, WS-Coordination and Transaction) has been specified. Each layer represents the main building blocks enabling the project workspace through the three identified roles, namely Work Space Service Provider (WSSP), Third-Party Service Providers (TPSP) and Work Space clients (WSC).

Referring to Figure 2, boxes within the WSSP services represent the essential core services necessary for the setting-up, operation and coordination of a project workspace. The service manager box provides access to the API functions necessary for all aspects of invocation, registration and deregistration of services from third-party service providers, as well as their publication in the local (WSSP maintained) UDDI registry. The Business Process Specification Layer (BPSL) includes the API functions that enable service composition in order to implement a given business process. This is based on the following core services concerned with service coordination, transaction, and security:

- Security Service: This service builds and implements the WS-Security specification. WS-Security defines the core facilities for protecting the integrity and confidentiality of SOAP messages, and is specified in a way that accommodates a wide range of security models (including identity-based security, access control lists, and capabilities-based security) and encryption technologies.
- Coordination Service: This service builds on WS-Coordination, which defines an extensible framework for coordinating activities using a coordinator and set of coordination protocols. This enables participants to reach consistent agreement on the outcome of distributed activities. The coordination protocols that can be defined should accommodate a wide variety of activities, including protocols for simple short-lived operations and protocols for complex long-lived business activities. It provides consistent control of the execution of the services forming the composite service.
- Transaction Service: It is based on WS-Transaction, which leverages WS-Coordination by (a) extending the WS-Coordination context to create a transaction context, (b) augmenting the activation and registration services with a number of additional services (Completion, Completion WithAck, PhaseZero, 2PC, Outcome Notification, BusinessAgreement, and BusinessAgreementWithComplete), and (c) defining two particular coordination types: Atomic Transactions (AT) and Business Activity (BA).

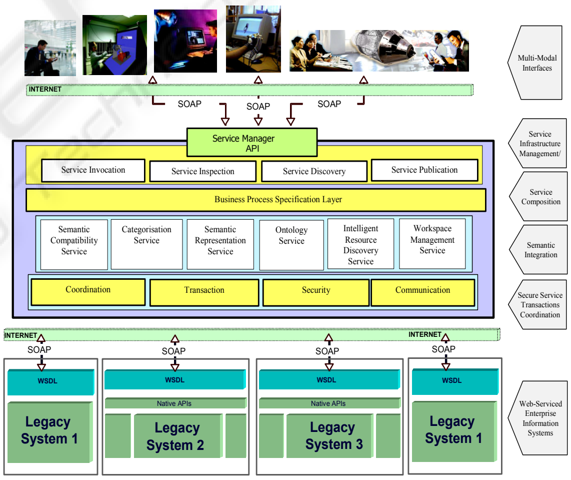

Figure 2: Proposed Service-oriented Architecture.

The Third Party Service Provider Layer (TPSP) represents all web-serviced applications that are ready for invocation and use as part of a service composition exercise in order to implement a business process. As explained in section 2, any existing EIS or legacy application has the potential to be promoted to become a web service. The members of a workspace (WSC) can use adapted software clients made available by the service provider (WSSP) to collaborate and invoke services. They have also the possibility to extend their existing portal (or EIS application) by implementing relevant API functions and/or providing transparent access to relevant core services.

A number of dedicated services have been specified and developed to facilitate and support service composition. The issue of web service composition involves three fundamental aspects: (a) the identification of the required functionality to implement the desired business process; (b) the discovery of the web services that perform the identified functionality; and (c) the management of the interaction between those web services.

The service composition layer includes necessary functionality for the consolidation of multiple services into a single composite service and specifying its business process behavior through the definition of activities and their protocols, including message exchanges that take place between a process and all its partners (represented through WSDL descriptions of their web services). Activities are used to describe ways in which messages are invoked on, and acted upon by, each process partner, and can be combined into complex algorithms using the BPEL structured activity definitions. These include the ability to define a sequence (ordered succession of steps), to define a loop (while), to have a conditional branching (while), or to specify the parallel execution of several steps (flow) as conceptualised in Figure 1. The services developed to support semantic composition include:

- Ontology Service: this provides the functionality required to make the selected ontology available to the other services, which may require it. This is achieved via the e-Cognos Ontology Server (e-COSER) that is devoted to handling all the ontology-related issues within the system (Lima et al., 2005).
- Semantic Representation Service: This semantic representation service makes use of the XML DOM API (Miniaoui et al., 2005), which transforms XML documents into trees. For compatibility detection between two sevices, the XML string data structure is more efficient than the tree structure because it preserves all paths within an XML document and transforms the XML mining problem to a string mining. The string mapping is performed in two stages. The First stage consists in encoding all the edges (tag names) of the XML DOM tree in a digital format. Then, the pre-order pass of the tree allows encoding the tree into a string. Zaki (2002) proposes the encoding of the XML tree into string. His algorithm inserts a character (-1) in the string to indicate the movement in the tree. This latter will be used in our semantic compatibility.
- Semantic compatibility service: this service is used to inspect service definitions at runtime and determine the level of compatibility between services to engage into a composition mode. This makes use of the eCognos Construction ontology (Lima et al., 2005) that takes a pivotal role in the process by checking semantic relatedness between concepts used by each partner service.
- Categorisation Service: This service is primarily used to provide a context and criteria-based categorisation of the information held within the UDDI registry for effective use and mining by potential business partners. As things stand, this can be queried using the native UDDI API. However, this presents serious limitations and inefficiencies in that it relies on data and information defined by business and service providers and do not model the vagueness and the uncertainty of stored information. Furthermore, our approach handles uncertainty in the reasoning using fuzzy concept.

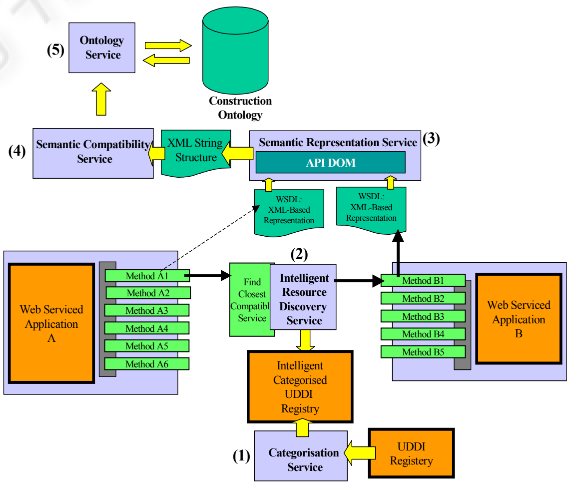

Figure 3: Intelligent resolution of semantic Compatibility for web services.

- Intelligent Resource Discovery Service: this service provides a standardised interface to query and make use of the Intelligent Categorised UDDI Registry generated through the categorisation service described above.

The above-mentioned services are illustrated in Figure 3 detailing the process of composing an e-Process from elementary semantically compatible services.

4 SEMANTIC COMPATIBILITY SERVICE

This service is used to inspect service definitions at run-time and determine the level of compatibility between services to engage into a composition mode. This makes use of the strings generated by the semantic representation service and the eCognos Construction ontology (Lima et al., 2005) that takes a pivotal role in the process by checking semantic relatedness between concepts used by each partner service.

Moreover, services involved in inter-working have to understand each other's when they are supposed to perform meaningful cooperative actions. Service definitions are described in the XML-based WSDL format.

Different mining techniques have been proposed in the literature (Laurent et al., 2005). The main difference between our semantic compatibility approach and the existing ones is that our method handles uncertainty in the reasoning using fuzzy concept (Miniaoui et al., 2005). Given two XML string structures, the main idea used in our approach is to measure the degree of overlap between two strings by introducing the inclusion concept. This means that an XML string is included within another with a certain degree. A degree of inclusion is defined based on notion of relationship between concepts using ontology. This inclusion is measured by using the semantic relatedness between labels (tags name in the strings). Thus, the vector representation of an XML structure provides sufficient statistics for string mining and for computing the level of compatibility or expected overlap and inclusion between XML strings.

Let us now consider that we are provided with knowledge on the data (strings) given by the semantic representation service. The idea is to find the degree of compatibility between those two strings. The difference between our approach and the existing mining approaches is that in our reasoning the inclusion detection is not only performed on same tag names. We also consider the fact that two different labels could be semantically related even if they don't have the same name. This relatedness can be detected by using semantic relatedness measure based on the Construction ontology (Lima et al., 2005), which allows us then to measure the inclusion between strings structure. The semantic relatedness used is Hirst–St-Onge (Hirst and StOnge, 1998) measure. The idea behind this measure is that two lexicalized concepts are semantically close if their synsets are connected by a path that is not too long and that "does not change direction too often". The strength of the relationship is given by:
$rel_{HS}(c_1, c_2) = C - path\,length \quad -kd$; where d is the number of changes of direction in the path using the ontology, and C and k are constants; if no such path exists, $rel_{HS}(c_1, c_2)$ is zero and the labels are deemed unrelated.
For instance, let's take three cases where two strings S and X could be included within another with different degrees.
Case 1:
String S: ABCDEF-1G-1H
String X: ABYDEF-1G-1H
The string X is similar to S if $rel_{HS}(C, Y) = 1$
The string X is highly included in S if Semantic relatedness is low
Case 2:
String S: AB-1C-1D-1KLMNY
String X: AC-1B-1D
The string X is fully included in S if the $rel_{HS}(C, B)$ *is height (B is synonym of C)*
The string X is partially included (with a certain degree) in $rel_{HS}(C, B) = 0$.
Case 3:
String S: ABC-1D-1F
String X: AE-1D

The string X is fully included in S if the $rel_{HS}(B, C)$ is high and is semantically related to E.

In the string, each label will be represented by fuzzy membership degree describing the ancestor-descendant relationship measured by the semantic relatedness value between given labels.

5 CATEGORISATION SERVICE

The Categorisation service provides a context and multi-criteria based clustering of the information held within the UDDI registry for effective exploitation by potential business partners. As argued above, the native UDDI API presents limitations due to the reliance on service providers for the description of their services. This does not model the uncertainty and the vagueness of stored information. Categorisation techniques employed to date, including hard clustering methods, have shown some limitations. These can be overcome with fuzzy clustering technique where service descriptions are attributed to several clusters simultaneously and thus, useful relationships between businesses and service categories may be uncovered, which would otherwise be neglected by hard clustering methods. Furthermore, services may belong to several clusters at the same time. This is in line with the Fuzzy c-means clustering technique as it adopts a non-binary approach by assessing the membership of a service to a cluster. Each service is assigned a vector of membership degree over all categories or disciplines. Hereafter, a service is assimilated to a text document.

To be able to cluster these text documents one needs to map them to numerical feature vectors. After applying the traditional pre-processing methods (stopwords elimination and stemming using the Porter Stemmer (Porter, 1980), each document is represented by a vector model. Under this representation model, documents are mapped into vectors usually normalized in a high dimension concept space. The axes of such a concept space are different dictionary terms and its coordinates are term weighting values corresponding to dictionary terms.

In its simplest form, each weighting term correspond to the term-frequency (TF) value (Zhao and Karypis, 2002). The resulting document vector will be defined by : $d_{tf} = (tf_1, tf_2, \ldots\ldots\ldots\ldots\ldots\ldots, tf_m)$; where tfi is the frequency of the i th term in the document.

Thus, this vector representation of documents provides sufficient statistics for computing the distribution of terms within each document. Than, each distribution is approximated by a Gaussian. Thus, this new vector representation will allow us to find expected overlap between two vectors in terms of corresponding distributions.

In order to perform the fuzzy clustering of these distributions, one needs to find a new distance measure which can measure effectively the distance between two distributions.

In this service we adapted our Modified algorithm proposed in (Nefti and Oussalah, 2004) to measure the distance between the distributions.

An appealing example consists of clustering a set of Gaussian distributions $G_i(x)$, defined by their mean μ_i and covariance matrix Σ_i ($i = 1..n$, n is the number of distributions)

$$G_i(x) = \exp(-\frac{1}{2}(x-\mu_i)^T [\Sigma_i]^{-1} (x-\mu_i)) \quad (1)$$

However, the distance structure in the product space of Gaussian parameters need not correspond one to one to the inclusion structure between Gaussians, which has influence on the location of the cluster prototypes. Therefore, we constrained the fuzzy clustering algorithm additionally, in order to detect the inclusion of document in the cluster prototype. In (Nefti and Oussalah, 2004), we refocused on standard FCM algorithm when dealing with probabilistic distance structure which incorporates a component of variance-covariance document (see for more details Nefti and Oussalah, 2004),. We showed that the use of such distance leads to optimal solutions in terms of prototype canters and membership function matrix U in a reasonable computational time.

The proposed modified FCM algorithm for Gaussians is based on the probabilistic distance measure. Consider a set of distributions, represented by Gaussians in the form of (1). This set may be interpreted as a set of points (μ_i, Σ_i), $i = 1..n$, in m-dimensional functional space with metrics, defined by (2), μ_i is a column mean vector $(m \times 1)$ and Σ_i is a diagonal covariance matrix $(m \times m)$. Let Δ_{ij} designate the BHATTACHARYYA distance from datum to cluster center (μ^*_j, Σ^*_j):

$$\Delta_{ij} = \frac{1}{8}(\mu_i - \mu^*_j)^T \left[\frac{\Sigma_i + \Sigma^*_j}{2}\right]^{-1} (\mu_i - \mu^*_j) + \frac{1}{2} ln \frac{\left|\frac{1}{2}(\Sigma_i + \Sigma^*_j)\right|}{\left[|\Sigma_i| \, |\Sigma^*_j|\right]^{1/2}} \quad (2)$$

Then, by analogy with standard FCM method, the proposed algorithm optimises objective function J, which may be expressed in the following form

$$J = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^{\alpha} \Delta_{ij} + \sum_{i=1}^{n} \lambda_i \left(\sum_{j=1}^{c} u_{ij} - 1 \right) \quad (3)$$

where λ_i are Lagrangian multipliers. The necessary conditions for optimality of the objective function J may be found by setting its partial derivatives to zero; namely:

$$\frac{\partial J}{\partial u_{ij}} = \alpha u_{ij}^{\alpha-1}\Delta_{ij} + \lambda_i = 0\,, \tag{4}$$

$$\frac{\partial J}{\partial \lambda_i} = \sum_{j=1}^{c} u_{ij} - 1 = 0\,, \tag{5}$$

$$\frac{\partial J}{\partial \mu_j} = -\frac{1}{4}\sum_{i=1}^{n} u_{ij}{}^{\alpha}\left[\frac{\Sigma_i + \Sigma*_j}{2}\right]^{-1}\left(\mu_i - \mu*_j\right) = 0 \;, \tag{6}$$

$$\frac{\partial J}{\partial \Sigma_j} = \sum_{i=1}^{n} u_{ij}{}^{\alpha}\left\{-\frac{1}{4}\left[\Sigma_i + \Sigma*_j\right]^{-2}\left(\sum_{k=1}^{m}\left[\left(\mu_i - \mu*_j\right)^T e_k\right]^2 E_k\right) + \right\} = 0, \tag{7}$$

where e_k and E_k are column vectors $(m \times 1)$ nd matrices with the only non-zero components $\{e_k\}_k$ and $\{E_k\}_{kk}$, respectively. Equations (4) and (5) may be used to derive

$$u_{ij} = \sum_{k=1}^{c}\left(\frac{\Delta_{ij}}{\Delta_{ik}}\right)^{\frac{1}{1-\alpha}} \tag{8}$$

$$\mu*_j = \left(\sum_{i=1}^{n} u_{ij}{}^{\alpha}\left[\Sigma_i + \Sigma*_j\right]^{-1}\mu_i\right)\left(\sum_{i=1}^{n} u_{ij}{}^{\alpha}\left[\Sigma_i + \Sigma*_j\right]^{-1}\right)^{-1} \tag{9}$$

From (9) we may infer, that

The detail of this calculus, based on the derivative of (8-9) with respect to its parameters and the proof of the existence of uniqueness of the solution is omitted here (Nefti and Oussalah, 2004).

Table 1: Proposed modified fuzzy clustering algorithm.

Step 1	Fix c, $2 \le c < n$; fix α , $1 \le \alpha < \infty$; fix ε ; initialise $U^{(0)}$ by using standard FCM algorithm
Step 2	Calculate the c fuzzy cluster prototypes $v_j(\mu_j, \Sigma_j)$ solving (7), (9) numerically
Step 3	Calculate $U^{(1)}$ using (11) and $v_j(\mu_j, \Sigma_j)$, obtained at Step 2
Step 4	If $\left\| U^{(1)} - U^{(0)} \right\| \le \varepsilon$ STOP, otherwise $U^{(0)} := U^{(1)}$ and return to Step 2

Finally, the proposed modified fuzzy clustering algorithm can be given as described in Table 1.

6 CONCLUSIONS

The paper argues that the ability to integrate disparate, heterogeneous Enterprise Information Systems to implement a distributed business process is facilitated by the loosely coupled nature of Web services. These provide the ability to combine and aggregate services into sophisticated, higher order, composite services. Moreover, the proposed service-oriented computing paradigm migrates the traditional standalone-hosting model to a networked one, by allowing web services to dynamically discover and hook-up to web services offered by different providers. The approach promotes the creation of domain-specific vertical libraries of services that are modular, well documented, implementation-independent, and interoperable.

The paper proposes an approach aiming at supporting ontology-based semantic composition of web-services to support electronic business processes. This has been implemented to provide the middleware foundation as well as the necessary components, including services, enabling the support of distributed business processes. Two services, at the heart of the service composition exercise, have been featured, namely: the semantic compatibility and categorisation services.

However, while the testing and validation work (not included in the paper due to length limitations), was highly encouraging, the approach requires large scale trials in order to better apprehend the business, technical as well as process implications. Also, given the emerging nature of the web service model, web compliant services are yet to be developed to form a critical mass of services that would encourage businesses to migrate to this new mode of collaboration (Rezgui, 2007b). From a technical point of a view, the algorithms presented in the paper need adapting to accommodate very high dimensional data with acceptable processing times. These issues are currently under investigation and will be reported in future publications.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the support of the European Commission under the eContent FUNSIEC project

REFERENCES

Barresi, S., Rezgui, Y., Lima C., Meziane, F., 2005. Architecture to Support Semantic Resources Interoperability, ACM 14th Conference on Information and Knowedge Management (CIKM), Bremen.

Ferris, C and Farrell, J., 2003. What are web services, Communications of the ACM, Vol.46, No6.

Hirst, G and St-Onge, D., 1998. Lexical chains as representations of context for the detection and correction of malapropisms, In Fellbaum, pp. 305–332.

Laurent, A., Poncelet, P. and Teisseire, M., 2005. Towards a Fuzzy Approach for Mining XML Mapping Schemas. Fuzzy Logic and the Semantic Web Workshop, Marseille.

Lima, C., El-Diraby, T. and Stephens, J., 2005. Ontology-based optimisation of knowledge management in e-Construction, ITcon Vol. 10, pg. 305-327.

Miniaoui, S., Wentland, and Forte, S., 2005. XML mining: from trees to strings, in proc. of ICICIS 2005, Cairo, Egypt March.

Nefti, S., Oussalah, M., 2004. Probabilistic-Fuzzy Clustering Algorithm, IEEE International Conference on Systems, Man and Cybernetics, The Netherlands, October.

Porter, M., 1980. An algorithm for suffix stripping», Program, Vol. 14, no. 3, p.130-137.

Rezgui, Y. and Meziane, F., 2005. A Web Services Implementation of a User Centred Knowledge Management Platform, International Journal of Intelligent Information Technologies, Vol.1, No 4.

Rezgui, Y., 2007a. Exploring Virtual Teamworking Effectiveness in the Construction Sector, Interacting with Computers, Vol. 19, no 1, pp. 96-112.

Rezgui, Y., 2007b. Role-based Service-Oriented implementation of a Virtual Enterprise: A Case Study in the Construction Sector, Computers in Industry, Vol. 58, no 1, pp. 74-86.

Sycara, K., Paolucci, M., Ankolekar, A. and Srinivasan, N., 2003. Automated Discovery, interaction and composition of web services, journal of web semantics, 1 ,27-46.

Van Den Heuvel, W.J. and Maamar, Z., 2003. Moving Towards a Framework to Compose Intelligent Web Services, Communications of the ACM, Vol.46, No10.

Zaki. J., 2002. Efficiently Mining Frequent Trees in a Forest, 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Zhao, Y. and Karypis, G:, 2002. Criterion Functions for Document Clustering: Experiments and Analysis, http://www.cs.umn.edu/˜karypis