Semantics-Enriched Document Exchange

Jingzhi Guo and Ming Sang Ho
Department of Computer and Information Science, University of Macau
Av. Padre Tomás, Pereira, S.J., Taipa, Macau
+853- 8397-4360

jzguo@umac.mo

ABSTRACT

In e-business development, semantics-oriented document exchange is becoming important, because it can support cross-domain user connection, business transaction and collaboration. To provide this support, this paper proposes a DOC Mechanism to exchange semantically interoperable business documents between heterogeneous enterprise information systems. This mechanism is designed on a layered-sign network, which enables any exchanged e-business document to be independently interpretable without losing semantic consistency.

Categories and Subject Descriptors

D.2.12 [**Interoperability**]: Data mapping; I.7.5 [**Document and Text Processing**]: Document Capture - *document analysis;* K.4.3 [**Electronic Commerce**]: Electronic data interchange (EDI).

General Terms

Design, Languages

Keywords

Document engineering, document exchange, electronic business, sign, representation, semantics, concept, XML Product Map

1. INTRODUCTION

Internet is experiencing a drastic change when it is more applied in electronic business. In this change, the intensive user participation asks for the research of business document exchange [5] to analyze, study, and support business connectivity, business collaboration, and the establishment of electronic and virtual marketplaces. It requires the incorporation of semantics study [1]. *Semantics* can be defined as the machine-computable, human-understandable, and program-reasonable meanings of concepts for Internet. It is one of the most important tasks of document engineering [3], which emphasizes on unifying heterogeneous business vocabularies, documents and processes across multiple business contexts to achieve an integrated semantics-enriched document exchange framework. Its key research issue is how to semantically analyze, represent and exchange business documents for natural business interaction across heterogeneous business information systems, for example, how to exchange a business document like inquiry sheet, order sheet or a contract in a cross-domain document flow but maintain semantic consistency between heterogeneous business system domains.

Three document analysis methods can be found for document analysis for document modeling and exchange. *Document-centric method* analyzes document structures to abstract logical models of documents from various document instances [4][8]. This method is more appropriate for classifying existing documents. *Data-centric method* analyzes data objects of different document instances to build object relationship between analyzed documents and databases [4][9]. This method is flexible for searching similar documents based on a set of data objects. *Concept-centric method* analyzes a document at the semantic level of concepts [5]. It regards a document as a set of hierarchically arranged atomic concepts. This method is more suitable for e-business document engineering, which requires exact meaning understanding and interpretation of the exchanged documents.

It is challenging to achieve the meaning understanding with precise interpretation for the exchanged documents between document sender and receiver. This is because users are creative and have their individual perspectives due to diverse backgrounds of natural languages, cultures, customs and behaviors [2]. To solve this problem, this paper aims at proposing a novel *semantics-enriched document exchange mechanism* (DOC Mechanism) based on a ConexNet theory discussed in another paper [7] by adopting concept-centric method. This mechanism ensures that any received document can be independently interpreted by the document receiver exactly as meant by the document sender without ambiguity.

The rest of this paper is organized as follows. Section 2 proposes a novel DOC Mechanism. Section 3 provides an integrated example to illustrate how semantics-enriched business document exchange can be enabled between heterogeneous business domains. Finally, a conclusion is made.

2. DOC MECHANISM

2.1 ConexNet Theory

ConexNet theory [7] thinks any terms, phrases, sentences and even a document is a sign, which is a tuple of structure (S), concept (C), context (X) and interpretant (◊) such that ∑Sign = (S, C, X, ◊). While any signs are *atomic and independent* (i.e. AISigns), they can be uniquely identified in any space and time and used to construct vocabularies, documents and processes. Based on 7 basic and 2 complex relations between signs, shown in Table 1 and 2, any subset of a *composite sign* CSign (i.e. a list of AISigns) and any subset of a *document sign* DSign (i.e. a tree of AISigns) can be independently represented, exchanged and interpreted without meaning ambiguity between sign originator and sign user.

DocEng2010, September 21–24, 2010, Manchester, United Kingdom.

Table 1: Seven Basic Relations between Signs

Name	Notation	Definition
atomicity	$\otimes$	For any sign A, A is *atomic* iff $\otimes$A.
independence	$\parallel$	For any two signs A and B, A is *independent of* B iff A $\parallel$ B.
interpretation	<<	For any sign A and interpretant ◊, A is *interpreted by* ◊ iff A << ◊.
conceptualization	$\Rightarrow$	For any two signs A and B, A *conceptualizes* B iff A$\Rightarrow$B.
contextualization	@	For any two signs A and B, A *is contextualized by* B iff A@B.
equivalence	$\equiv$	For any two signs A and B, A is *concept-equivalent with* B iff A $\equiv$ B.
reification	$\rightarrow$	For any two signs A & B, A *reifies* B iff A $\rightarrow$ B.

Table 2: Two Complex Relations in ∑Sign

Name	Notation	Definition
concealment	$\langle\,\rangle$	For any two signs A and B, A is *concealed from* B iff B$\langle$A$\rangle$, such that: (1) A is set of signs, and B is any sign such that B $\not\subset$ A; (2) For any $a \in$ A, a is an AISign of Lemma 1; (3) P(A) $\parallel$ B, where P(A) is a power set of A.
interface	$\models$	For any three signs A, B and C, A is *interfaced by* B to C iff (1) A $\equiv$ B; (2) B $\models$ C and B $\equiv$ C.

Formally, any sign A is *atomic* if and only if (1) S(A) = S(IID, T, AN, X, {OP}) and (2) C(A) = C(IID $\equiv$ T $\Leftarrow$ AN @ X) << ◊, where the structure S of sign A is composed of a set of elementary structures $\sum S_i$ = (unique concept identifier IID, term T, concept definition AN, context X, optional extensions {OP}). The concept C of sign A is interpreted by an interpretant ◊ as an annotation AN at context of X. The AN is again conceptualized as a term T that is equivalent to unique identifier IID. The uniqueness of IID cross domains is guaranteed by an MD-IID scheme [7]. Besides atomicity, any two sign A and B are *independent* if and only if A and B are uniquely identified as MD-IID$_A$, MD-IID$_B$, and MD-IID$_A$ $\neq$ MD-IID$_B$. A sign A is an *atomic and independent sign* (i.e. AISign) if and only if it is both atomic and independent. An AISign can be referenced by any other signs without any versioning problem, because anything happened has become a history and will be never changed.

Applying the relations of concealment and interface, we can achieve independence of any composite sign. Formally, for any sign A, A is said to be a *composite sign* (i.e. *CSign*) if and only if A can be expanded to a list of AISign such that A $\equiv$ (A$_1$, A$_2$, …, A$_n$), where A is said to be interfaced by (A$_1$, A$_2$, …, A$_n$) to B if and only if B $\equiv$ (B$_1$, B$_2$, …, B$_n$) and A$_1$ $\models$ B$_1$, A$_2$ $\models$ B$_2$, …, A$_n$ $\models$ B$_n$, where A$_1$ $\equiv$ B$_1$, A$_2$ $\equiv$ B$_2$, …, A$_n$ $\equiv$ B$_n$. Further, a sign D is a document sign (DSign or D) if and only if D = (D_1^1, D_i^2, ..., D_i^k, ..., D_i^n), such that D consists of a set of hierarchical AISigns D_1^1, D_i^2, ..., D_i^k, ..., D_i^n, where D_1^1 is the tree root, k is tree level, and i is sibling position. We can prove that any sub-document D' of D is a concealment of $\langle D_1^1 \langle D_i^2 \langle \ldots \langle D_i^k \langle \ldots \langle D_i^{m+1} \rangle\rangle\rangle\rangle\rangle\rangle$, and any sub-document D' of D is interfaced to A by RT such that D' $\equiv$ RT $\models$ A, where A = (A$_1$, …, A$_n$) $\not\subset$ D and RT = (RT$_1$, RT$_2$, …, RT$_n$) $\subset$ D' (see proofs in [7]).

2.2 ConexNode as Layered Sign Framework

DOC Mechanism is a system component within a node of ConexNet [7]. It is designed as an infrastructure to semantically support e-business between heterogeneous systems. ConexNet is comprised by a set of hierarchically connected ConexNet nodes such that ConexNet = ($\mathcal{N}_1^1$, $\mathcal{N}_i^2$, …, $\mathcal{N}_i^k$, …, $\mathcal{N}_i^n$), where i is a sibling and k is level, $\mathcal{N}_1^1$ is a standard node; $\mathcal{N}_i^2$ is common node; $\mathcal{N}_i^k$ is a local node if it is not a standard node or a common node, or a user node without child node. A ConexNet node is designed as a layered sign framework consisted of six layers, shown in Fig. 1.

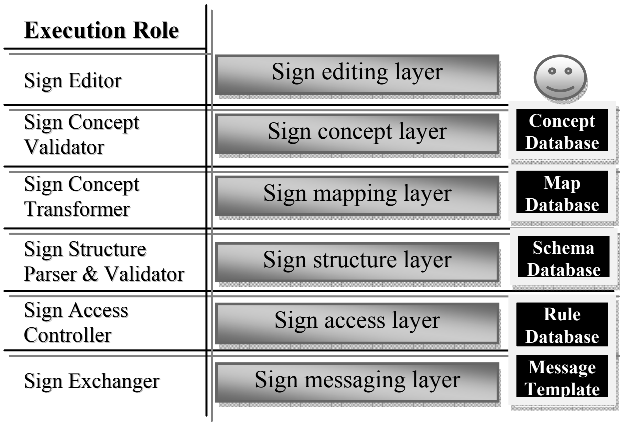


Fig. 1: A ConexNet node as a layered sign framework

Sign editing layer is responsible for sign creation, personalization, and use through sign editor by ConexNet users. *Sign concept layer* store, manage and validate the concept of local signs. *Sign mapping layer* provides the sign mapping between local signs and external signs. *Sign structure layer* describes methods of parsing and validating the structure of all signs. *Sign access layer* is a security layer, which controls and authenticates any incoming and outgoing signs. *Sign messaging layer* packs, unpacks, and routes signs to appropriate destinations.

Particularly, a ConexNet node represents a business information system either owned by an e-marketplace, a service provider or a firm, which creates, edits and exchanges signs.

2.3 DOC Mechanism

As a component of a ConexNet Node, DOC Mechanism, shown in Fig. 2, consists of a Personal AISign Space (S$_P$) as a *personal vocabulary*, a Personal AISign Interface (PSI), a Local AISign Space (S$_L$) as a *local vocabulary*, a Common AISign Localizer (CAL), a DOC generator (DocG), and a DOC Exchanger (DocE) that is interfaced to a set of external Common AISIgn spaces (S$_C$) as *common vocabularies*.

This DOC Mechanism works as follows: many S$_{Ci}$ from common nodes are responsible for providing AISigns to any document exchange users, who localize the common AISigns into an S$_L$ in a *local concealment* (LC) component. This LC is purely internal and conceals any personally developed signs strictly in local space. Any document as a sign is designed and generated by a DOC Generator (DocG) using both AISigns from S$_L$ and S$_P$. Since only AISigns in S$_L$ are known by external users by unique identifiers MD-IID [7] and AISigns in S$_P$ are unknown to external users, all signs from S$_P$ have to be equipped with a PSI, by which the external users can run-time interpret personal AISigns when they receive and interpret a document (DOC) coming from a DOC Exchanger (DocE).

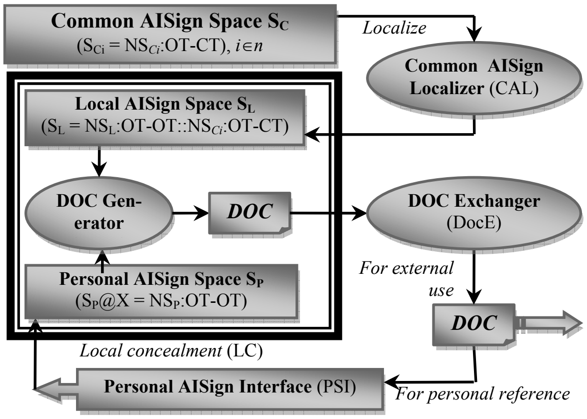

Common AISign Space S_C
($S_{Ci} = NS_{Ci}$:OT-CT), $i \in n$
Localize
Common AISign Localizer (CAL)
Local AISign Space S_L
($S_L = NS_L$:OT-OT::NS_{Ci}:OT-CT)
DOC Generator
DOC
DOC Exchanger (DocE)
Personal AISign Space S_P
(S_P@X = NS_P:OT-OT)
For external use
DOC
Local concealment (LC)
Personal AISign Interface (PSI)
For personal reference

Fig. 2: DOC Mechanism

In DOC Mechanism, any exchanged document (DOC) is a *reified document* (DR), which is concealed and interfaced. Particularly, DR can be defined in the following format:

$$DR = (DR_1^1, DR_i^2, ..., DR_i^k, ..., DR_i^n) \qquad (1.1)$$

$$DR_i^k = (U, V), \text{ such that } U \rightarrow V. \qquad (1.2)$$

$$S(U) = S(MD\text{-}IID, RT, T, PID, \{OP\}) \qquad (1.3)$$

$$C(U) = C(T \equiv MD\text{-}IID \equiv RT\ |=) \qquad (1.4)$$

$$S(V) = S(MD\text{-}IID, RT, T) \rightarrow (OP, Dt, RV) \qquad (1.5)$$

$$C(V) = C((OP, Dt, RV\ |=)@(T \equiv MD\text{-}IID \equiv RT\ |=)) \qquad (1.6)$$

where any DR_i^k in DOC mechanism may carry a value sign (or a CSign) V identified by MD-IID interfaced by RT = ∑MD-IID, which becomes the context of V instances. OP is an operand, Dt is data type, and RV (self-interfaced to external ∑MD-IID) is the V instance. For example, C(V) = C((IS, string, "orange ≡ NS1:777-777")@NS1:555-666 |= NS2:888-888 ≡ "color").

It is apparent that DR is semantically interpretable by any users as long as they are in ConexNet scope no matter whether these users know with each other or not. This interpretability is, in fact, realized by a principle of local sign concealment and interface, which states that for any personal document, the external interpretation is always available in ConexNet e-marketplace through DOC Mechanism.

2.4 XPM Document Exchange Specification

The key of implementing DOC Mechanism is to represent any interpretable document that follows ConexNet theory [7] in an XML specification. This paper inherits the research on XML Product Map (XPM) [5][6] to model an exchangeable document in Table 3, where any element and #PCDATA are signs. In this DTD, a sign is a document. When #PCDATA of "value" element is not given, the DTD is used to design a document template (DT) such that an instance of the DTD is a document template including only abstract signs. When #PCDATA of "value" element is given, the DTD instance is a reified document (DR) including both abstract and concrete signs. All signs used to describe a complex document are AISigns specified in S_L and/or S_P vocabulary. Any vocabulary has a namespace (NS), which identifies S_L, S_P or S_C that maintains the vocabulary database. For each document, it is also regarded as a small contextual domain identified by a namespace "myns" under "sign/head", for example, myns = "00000001voc29:Lk5xyyy848-Lk5xyyy848". In this example, "00000001voc29" is a vocabulary namespace NS and "Lk5xyyy848-Lk5xyyy848" refers to "OT-CT" (*see* Table 2 of [7]). This MD-IID scheme ensures that any document is unique and exchangeable as long as it is Internet-accessible.

Table 3: A Simplified XPM Document DTD

```
<!ELEMENT sign (head, body)>
<!ATTLIST head       myns CDATA #REQUIRED>
<!ELEMENT body ((concept)+)>
<!--Define atomic sign as an abstract concept in hierarchy.-->
<!ELEMENT concept (concept*, value*)>
<!ATTLIST concept                tid CDATA #REQUIRED
    ptid CDATA #REQUIRED     term CDATA #REQUIRED
    refs CDATA #REQUIRED>
<!--reified concept in a certain reified concept array-->
<!ELEMENT value (#PCDATA | value)*>
<!ATTLIST value                  tid CDATA #REQUIRED
    ptid CDATA #REQUIRED     term CDATA #IMPLIED
    refs CDATA #IMPLIED      op CDATA #IMPLIED
   dt CDATA #IMPLIED>
```

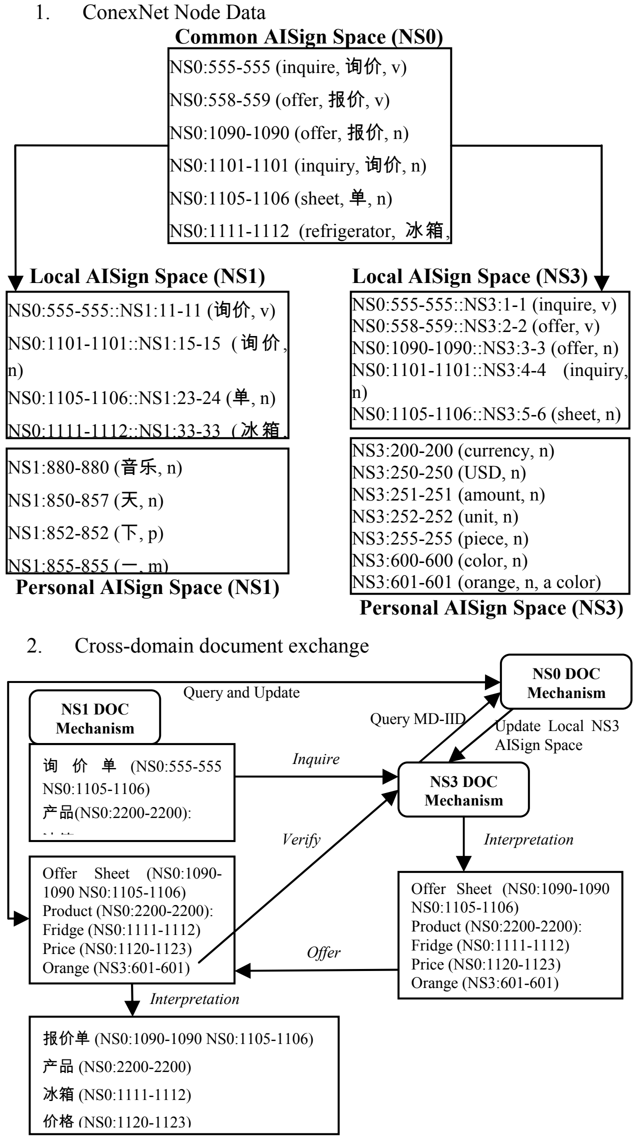
In this DTD, a concept and/or a value together with its descendants form a sub-sign tree (i.e. a sub-document), which is uniquely indentified by its sub-tree root. This enables any sub-document to be retrievable for document exchange.

Following the theorems of document concealment and interface [7], any element of "concept" or "value" or any #PCDATA of "value" element is uniquely indentified and interfaced. This guarantees that any sub-document is interpretable by external users. The particular method of achieving it in this DTD is as follows: (1) The attribute "tid" of concept or value element uniquely identify a concept by means of "OT-CT" scheme. (2) The attribute "term" of concept or value element is uniquely identified by "tid" such that "term ≡ tid", where "term" is a set of atomic terms defined in AISign spaces (i.e. vocabularies). (3) The attribute "refs" of concept or value element uniquely identifies "tid" such that "tid ≡ refs", where "refs" is a set of MD-IID interfaced to external or internal AISign spaces, such that refs = {md-iid$_1$, md-iid$_2$, …, md-iid$_n$} ≡ {term$_1$, term$_2$, …, term$_n$}. (4) For any #PCDATA, PCDATA is a list of atomic terms (i.e. AISigns), which can always be found in the defined vocabularies. Thus, we can always find their corresponding MD-IID for reified document exchange.

The XPM Document DTD specified in Table 3 can well implement the exchanged document for meaning understanding.

3. AN EXAMPLE APPLICATION

DOC Mechanism can design and use any document by following ConexNet theory [7]. This mechanism makes a business document semantically interoperable between different business information systems. To illustrate the use of this mechanism for semantics-enriched document exchange, we provide an integrated example shown in Fig. 3, where a ConexNet consists of three nodes such that NS0 is a common node, and NS1 and NS3 are local nodes (see Fig. 3-1). In common node NS0, there is a vocabulary S_C. In local nodes NS1 and NS3, there are local vocabulary S_L and personalized vocabulary S_P, respectively. In this example, NS1 attempts to inquire a product "冰箱" from an unknown seller NS3 by using its own semantically-encoded inquiry sheet. Through the DOC Mechanism, Fig. 3-2 shows that NS1 finally successfully received semantically interpretable offer sheet from NS3. For the XPM documents, please see Appendix in http://www.sftw.umac.mo/~jzguo/pages/pub/doceng10a.pdf.


1. ConexNet Node Data

2. Cross-domain document exchange

Common AISign Space (NS0)
NS0:555-555 (inquire, 询价, v)
NS0:558-559 (offer, 报价, v)
NS0:1090-1090 (offer, 报价, n)
NS0:1101-1101 (inquiry, 询价, n)
NS0:1105-1106 (sheet, 单, n)
NS0:1111-1112 (refrigerator, 冰箱,
Local AISign Space (NS1)
NS0:555-555::NS1:11-11 (询价, v)
NS0:1101-1101::NS1:15-15 (询价, n)
NS0:1105-1106::NS1:23-24 (单, n)
NS0:1111-1112::NS1:33-33 (冰箱,
NS1:880-880 (音乐, n)
NS1:850-857 (天, n)
NS1:852-852 (下, p)
NS1:855-855 (一, m)
Personal AISign Space (NS1)
Local AISign Space (NS3)
NS0:555-555::NS3:1-1 (inquire, v)
NS0:558-559::NS3:2-2 (offer, v)
NS0:1090-1090::NS3:3-3 (offer, n)
NS0:1101-1101::NS3:4-4 (inquiry, n)
NS0:1105-1106::NS3:5-6 (sheet, n)
NS3:200-200 (currency, n)
NS3:250-250 (USD, n)
NS3:251-251 (amount, n)
NS3:252-252 (unit, n)
NS3:255-255 (piece, n)
NS3:600-600 (color, n)
NS3:601-601 (orange, n, a color)
Personal AISign Space (NS3)
NS0 DOC Mechanism
Query and Update
NS1 DOC Mechanism
Query MD-IID
Update Local NS3 AISign Space
询 价 单 (NS0:555-555 NS0:1105-1106)
产品(NS0:2200-2200):
Inquire
NS3 DOC Mechanism
Verify
Interpretation
Offer Sheet (NS0:1090-1090 NS0:1105-1106)
Product (NS0:2200-2200):
Fridge (NS0:1111-1112)
Price (NS0:1120-1123)
Orange (NS3:601-601)
Offer
Offer Sheet (NS0:1090-1090 NS0:1105-1106)
Product (NS0:2200-2200):
Fridge (NS0:1111-1112)
Price (NS0:1120-1123)
Orange (NS3:601-601)
Interpretation
报价单 (NS0:1090-1090 NS0:1105-1106)
产品 (NS0:2200-2200)
冰箱 (NS0:1111-1112)
价格 (NS0:1120-1123)

Fig. 3: An integrated example

In Fig. 3, we have illustrated that the solution to a semantics-enriched document exchange relies on NS0's common AISigns and the accessible personalized namespaces for personalized AISigns. The key to the solution is the MD-IID of AISign identification scheme such that MD-IID = NS_x:OT-CT::NS_y:OT-CT defined in [7]. The working mechanism of this example can be described in following steps:

Step 1: Designing AISigns

(1) The common ConexNet node is responsible for designing the common AISigns in S_C, using MD-IID scheme.

(2) The local ConexNet nodes are responsible for localized local AISigns in S_L from the common AISigns of common ConexNet node, using MD-IID scheme.

(3) The local ConexNet nodes are responsible for designing personalized AISigns in S_P, using MD-IID scheme.

Step 2: Design and generate semantics-enriched documents

(1) The local ConexNode at sender side is responsible for designing both document templates and creating reified documents by using common AISigns, local AISigns and/or personalized AISigns.

Step 3: Semantics-enriched document exchange

(1) The local ConexNet node at receiver side is responsible for mapping received common AISigns onto their local AISigns.

(2) The local ConexNet node at receiver side is responsible for verifying the received personalized AISign by the sender's personal AISign interface (PSI) to create new AISigns.

(3) The local ConexNet node at receiver side transforms incoming document into the locally interpretable document.

The above steps ensure that any semantically non-interoperable business documents can be semantically interpretable through DOC Mechanism. For example, semantically heterogeneous inquiry sheet and offer sheet can be understood by any receivers in ConexNet. Thus, legal consequences will follow.

4. CONCLUSION

Semantics-enriched business document exchange mechanism (DOC Mechanism) allows business document users to connect and collaborate by accurate document exchange without meaning ambiguity in a wider ConexNet-based e-marketplace. It guarantees that any sent document can be well-received and exactly-interpreted by the document receiver through a newly-developed XPM document specification. This is illustrated by the example, which shows that DOC Mechanism is correct and workable in implementation. In another paper, we will discuss the implementation of DOC Mechanism together with the introduction to the ConexNet PM vocabulary that is going to be publicly available through a newly launched Website for document engineering.

5. REFERENCES

[1] Atencia, M. and M. Schorlemmer (2008) I-SSA: Interaction-Situated Semantic Alignment. *OTM 2008*, Part I, LNCS 5331, pp. 445–455.

[2] Fischer, G., Resnick, M., Jennings, P., Shneiderman, B. and M. Maher (2009) Creativity Challenges and Opportunities in Social Computing. In: *ACM CHI 2009* (April 4-9, Boston, USA) pp. 3283-3286.

[3] Glushko, R. and T. McGrath (2005) *Document Engineering*, MIT Press, USA.

[4] Glushko, R. and T. McGrath (2002) Document Engineering for e-Business. In: *Proc. of ACM DocEng'02*, pp. 42-48.

[5] Guo, J. and C. Sun (2003) Context Representation, Transformation and Comparison for Ad Hoc Product Data Exchange. In: *Proc. of ACM DocEng'03*, pp.121-130.

[6] Guo, J. (2008) *Collaborative Concept Exchange*. VDM Verlag, Germany.

[7] Guo, J. (2010) ConexNet: A Collaborative Concept Exchange Network. http://www.sftw.umac.mo/~jzguo/pages/pub/ConexNet.pdf.

[8] Lee, K., Choy, Y. and S. Cho (2003) Logical structure analysis and generation for structured documents: A syntactic approach. *IEEE Transactions on Knowledge and Data Engineering*, 15(5), pp. 1277-1294.

[9] Schmidt, A., Waas, F., Kersten, M., Carey, M., Manolescu, I. and R. Busse (2002) XMark: a benchmark for XML data management. In: *Proc. of VLDB'02*, VLDB Endowment, pp. 974-985.