Extensible Dynamic Form Approach for Supplier Discovery

Yan Kang, Jaewook Kim, and Yun Peng
Department of Computer Science and Electrical Engineering
University of Maryland, Baltimore County
{kangyan1, jaewook2, and ypeng}@umbc.edu

Abstract

In this paper we propose a new form architecture called eXtensible Dynamic Form (XDF) to facilitate data collection process of supplier discovery. This architecture provides users with flexibility to dynamically extend the base form structure with new form components, or to reuse the existing form components by intelligent search. With XDF, the supplier's capability information can be entered in the structure and format that fit different intentions of individual users. Experimental results demonstrate that the proposed architecture is valuable for facilitating the supplier discovery process.

Keywords: Extensible Dynamic Form, Supplier Discover, Semantic-based Search, XML Matching

1. Introduction

In today's dynamic manufacturing industry, *supplier discovery* is essential for building a flexible network of suppliers in supply chain. In general, the supplier discovery involves two steps. The first step is to collect supplier capabilities and customer requirements – henceforth *collect function*. In the traditional e-marketplaces, online forms are typically used for the *collect function* [1]. These forms are mostly fixed and pre-defined, so they are not flexible enough to capture a variety of requirements and capabilities. As the result, some information is entered in an unstructured way such as free texts. In addition, Different suppliers (or customers) often use different terminologies and structures with their own semantics to represent their own capabilities (or requirements). Ununified semantics often causes ambiguity when attempting to match capabilities to requirement.

The second step is to find suppliers – henceforth, *search function* – whose capabilities best meet the requirements specified by a customer. Existing e-marketplaces rely on simple keyword or category based search with poor precision. Several approaches have been proposed to enhance the search (e.g., semantic-based search). They mostly reply on the structured data models such as XML, RDF, and OWL. These advanced search approaches cannot be applied to unstructured or semi-structured information collected by traditional forms.

In this paper, we mainly focus on the *collect function* of supplier discovery. We propose a new form architecture called eXtensible Dynamic Form (XDF) to help capturing requirements and capabilities in a better structured way. XDF architecture provides several key innovations, including: 1) architecture for users (suppliers or customers) to *dynamically extend* the base form structure with new form components, or to reuse the existing form components by intelligent search; 2) structural representation of the requirements and capabilities as XML instances; and 3) support advanced suppliers search function by semantic-based XML instance matching we proposed earlier [2].

The rest of the paper is organized as follows. Section 2 provides a background for supplier discovery. The architecture of XDF is described in Section 3. Section 4 explains basic operations of XDF. Section 5 reports the experiments and results. Section 6 concludes with lessons learned.

2. Background

Traditionally e-marketplaces have three roles: provision of institutional infrastructure, supplier discovery by matching customers and suppliers, and facilitating the transaction [3]. In this paper, we focus on the supplier discovery role.

Many approaches have been proposed for matching customers and suppliers in the supplier discovery [4, 5]. Most of them are based on similarity-based retrieval of textual descriptions. In such retrievals, the customer requirement usually consists of a number of keywords or phrase for text retrieval. However, these approaches often ignore semantics in the textual descriptions.

To overcome these problems, several knowledge-based approaches have been developed for manufacturing domain [6, 7]. Most of them employ ontologies to capture and represent semantic information. This requires that the ontology is comprehensive and it is shared and agreed upon by both suppliers and customers. However, ontology-based approaches face many challenges due to immaturity of technologies in semantic representation, measuring, and reasoning.

Another approach to enhance the search capabilities is to utilize XML data representations which are widely used in the e-business industry to represent the structured information. Although XML is not a formal semantic model, its structure and the English words for the labels contain rich semantic information. Many XML matching approaches have been proposed [8, 9], most of which analyze the similarity between two XML schemas or two XML instances based on their syntactic, linguistic and structural information. Some applications [10] [11] have been developed to generate XML schema based web forms to capture user's information and produce XML instances as output. However, these applications support limited XML schema features. And the web forms they generated are fixed.

To address these limitations of existing approaches, we propose our eXtensible Dynamic Form (XDF) architecture.

3. Architecture of XDF

The proposed XDF architecture allows users to extend the base form with new components and to reuse the existing form components by intelligent search. It helps to represent the semi-structured or unstructured information in requirements and capabilities in a better structured way. Figure 1 shows basic architecture of XDF.

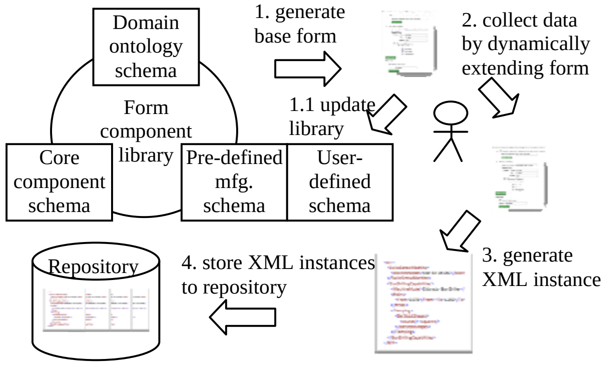


Figure 1. Architecture of XDF

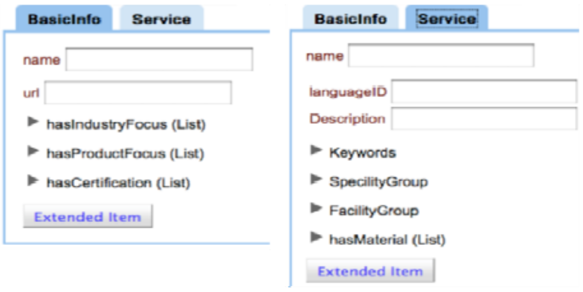
The basic architecture of XDF relies on a form component library to provide a collection of reusable form components, each of which are defined by several XML schemas: Domain ontology schema is generated based on ontology defined in [12]. Core components schemas come from Open Application Group's Integration (OAGi) [13] which was proposed to increase interoperability for enterprises. Since the ontology schema only defines a small set of manufacturing concepts and core component schemas are mainly focus on the general concepts of e-business, we created manufacturing schemas specifically focusing on manufacturing industry but not defined by the ontology. Details of XDF operations are given in the next section.

4. XDF operations

This section describes the core operations of XDF as illustrated in Figure 1. These operations include 1) base form generation, 2) data collection by dynamically extending the base form, 3) creating new form components or 4) reusing the existing form components by search, 5) XML instance generation.

4.1. Generate base form

XDF initially generates a base form based on the XML schemas (mainly including the domain ontology schema but not limited to). The mechanism behind this operation is that the XML schemas are first transformed into reusable form components. Then, some basic generic form components (e.g., industry, product, and certification types) are chosen and rendered as the base form.


BasicInfo Service
name
url
▶ hasIndustryFocus (List)
▶ hasProductFocus (List)
▶ hasCertification (List)
Extended Item

BasicInfo Service
name
languageID
Description
▶ Keywords
▶ SpecilityGroup
▶ FacilityGroup
▶ hasMaterial (List)
Extended Item

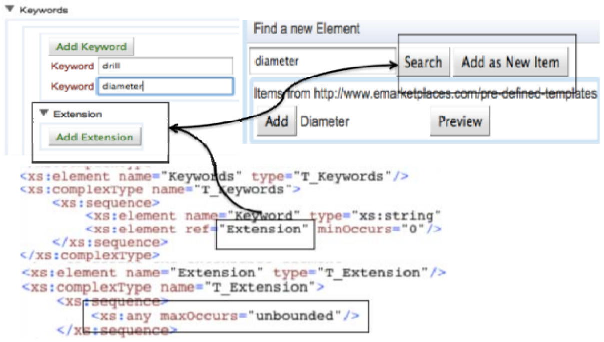
Figure 2. The base form of XDF

Figure 2 shows the screenshot of the base form generated by XDF. The base form initially includes the form components for supplier's basic information and service information.

4.2. Extend form

Users can, by clicking the "Extended Item" button at appropriate place, "extend" the base form by either creating their own form components or reusing the existing form components by search. Figure 3 illustrates an example of the form "extend" operation.

When a user clicks the "Extended Item" button in the form, XDF shows a form component search interface. The user can decide to either search the existing form components with some keywords or to create new form components for their own contents. For example, Figure 3 shows the search operation with a keyword "diameter" when extending "drill" service. The XDF searches the form component library and returns a list of form components whose names match or similar to the search keyword “diameter”. The user can preview each of the form components returned and choose one as a sub-structure of the current form.



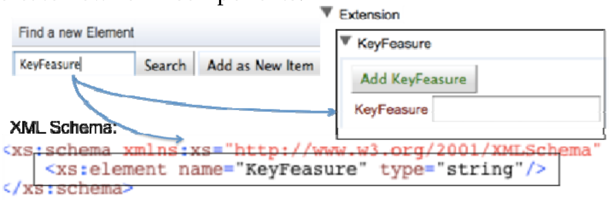
```
<xs:element name="Keywords" type="T_Keywords"/>
<xs:complexType name="T_Keywords">
    <xs:sequence>
        <xs:element name="Keyword" type="xs:string"
        <xs:element ref="Extension" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
<xs:element name="Extension" type="T_Extension"/>
<xs:complexType name="T_Extension">
    <xs:sequence>
      <xs:any maxOccurs="unbounded"/>
    </xs:sequence>
```

Figure 3. An example of form extending

Details of creating or searching the form components are described in the following subsections.

4.3. Create new form component

The form components created by users are called *user-defined form components*. These components are stored as *user-defined schema*, to be searched and reused by other users later. Figure 4 illustrates an example of how to create new form components.



```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
   <xs:element name="KeyFeasure" type="string"/>
</xs:schema>
```

Figure 4. An example of form component creation

In this example, a user inputs a keyword “KeyFeature” and clicks the “Add as New Item”. Then, XDF creates a new form component named “KeyFeature” that has a text input box, and insert it into the current position of the form. The generated form component is encoded as a *string* type element in the user-defined schema.

4.4. Search existing form component

To extend the form, users can also reuse the existing form components by search. The search method basically finds form components in the library whose names are semantically matched to the search keyword. We have investigated several search methods based on keyword, n-gram, WordNet, and synonyms.

Both keyword-based and n-gram based methods are simple and fast. However, both use string-based similarity metrics and thus cannot find matches for semantically similar words. In addition, keyword matching cannot deal with word variations and typos appearing in user’s input that n-gram based search can.

Kim [2] proposed a **WordNet-based** approach to measure the semantic similarity between two sets of words based on word-senses similarity scores drawn from WordNet [14]. This method works well for matching semantically similar concepts and synonyms. However, it is relatively slow due to the search of a large lexical database of WordNet, and it cannot deal with the problem of word variations and typos.

To overcome the shortcomings of the aforementioned methods, we propose an efficient new method which considers semantics of each input word by its synonyms and deals with word variations and typos by n-gram techniques. Specifically, this **Synonym+ngram search** goes as follows**.**

Denote the search keyword as string k and the name of the form component to be compared as string e.

1) Tokenize k and e to L(k) = {v_1, v_2…v_i…} and L(e) = {u_1, u_2…u_q…}.

2) For each word v_i in L(k), a set of synonyms are drawn from WordNet, denoted as S_i = {s_{i1}, s_{i2}…, s_{ij}…}.

3) For each word u_q in L(e), find the synonym s_{ij} in all synonym sets S_i with the highest n-gram similarity score between u_q and s_{ij}, recorded as Score(u_q). The set of scores for all words in L(e) form ScoreSet = {Score (u_q), q = 1, 2,…, |L(e)|}. The average scores in ScoreSet is the similarity, subject to a penalty for length difference between L(k) and L(e). The similarity is computed as:

$$Sim(e,k)=\begin{cases}\beta\times\dfrac{1}{|L(e)|}\sum_{q=1}^{|L(e)|}Score(u_q) & \text{if } |L(k)|\geq|L(e)| \\ \beta\times\dfrac{1}{|L(k)|}\sum_{q=1}^{|L(k)|}Score(u_q) & \text{otherwise}\end{cases} \qquad (1)$$

where $\beta=1-\dfrac{abs(|L(k)|-|L(e)|)}{|L(k)|+|L(e)|}$ is the length penalty.

We have conducted experiments with limited scope to assess the performance of these four search methods using eight sample form components from the form component library to generate search keys. We consider these samples as the expected search results. Then, we generate the arbitrary search keywords based on the names in the samples. There are two versions of search keywords: a *synonymous version* and a *typo version*.

Using these two versions of keywords, we use each of the four different methods to search the form component library of 902 components. Each method returns the top 10 form components with the highest similarity scores. If the sample form component exists in a search result, we call it a hit. Hit rate is the ratio of the number of hits to the number of search queries (i.e., 8). We compare the average hit rates of search for the eight samples. Table I shows the comparison results of the four search methods.

For search results of the synonymous version of search keys, WordNet-based and synonym+ngram methods have the best hit rates (100%). This is to be expected because they both utilize semantics of word synonyms. On the other hand, the search results of the typo version show that n-gram based and synonym+ngram methods are better (87.5% hit rate) than others (12.5% hit rate). This is because WordNet-based and keyword-based search cannot deal with the similarity of mistyped words, whereas n-gram based method can compute the similarity between characters of words by edit distance.

Table 1. Comparison of four search methods

Synonymous version of search keyword		
Matching Method	Hit Rate	Average Speed (ms)
keyword-based	0%	44
n-gram based	25%	45
wordNet-based	100%	20216
Synonym+ngram	100%	2620
Typo version of search keyword		
Matching Method	Hit Rate	Average Speed (ms)
keyword-based	12.5%	25
n-gram based	87.5%	43
wordNet-based	12.5%	4345
Synonym+ngram	87.5%	2016

Overall, the synonym+ngram search method has the best performance in terms of hit rate and average computation speed.

4.6. Generate XML instance

Data inputted on XDF can be automatically transformed into XML instances. Since the base form is generated from XML schemas, it maintains the structure and constraints of the XML schemas. Therefore, the XML instances generated from user-inputted data adhere to XML schema automatically.

5. Experiment on supplier discovery

We randomly chose 30 suppliers from Thomasnet, and collected their capability data in the form of textual description. An artificially made customer requirements was created. A human expert from DSN Innovations Corp provided a similarity ranking of the 30 suppliers for the requirement.

Using XDF, we encoded the textual descriptions of the requirement and capabilities of the 30 suppliers and then generated their XML instances. XML instances are analyzed by a semantic-based XML instance matching algorithm proposed by Kim [2] to discovery suppliers.

We compare results of this XML-based supplier discovery approach to two other approaches: keyword-based and ontology-based search. For keyword-based search, we used Google custom search engine and built a sample web site that contains 30 web pages, each of which includes the textual capability description for one supplier.

For ontology-based approach, we used a small manufacturing ontology and the related match-making algorithm proposed by Ameri and Dutta [15].

A total of 16 key words were extracted from the customer requirement as the input to the Keyword-based search. Ontology-based search uses only 11 of the 16 key words (the other 5 are not defined in the ontology). For a fair comparison, we ran two experiments for XML-based approach, one creating requirement XDF form using all 16 key words, the other using only the 11 key words defined in the ontology. The results are called Full-XML and Partial-XML, respectively.

The result of each of the four experiments is a ranked list of the 30 suppliers based on their similarity scores. These lists are compared with the ranked list from the domain expert. The comparison was measured by Recall metrics and normalized nDCG (normalized Discounted Cumulative Gain) [16] with n = 30.

Table 2 shows the performances of the four search results. The measures are normalized to the 0-1 range. The recall value 0.667 on the upper-left corner is read as: among the top 3 suppliers found by the Keyword method, 2 of them are also in the top 3 from the human expert's list.

The results show that Full-XML performed better than keyword-based and ontology-based approaches. This is because Full-XML utilizes information encoded in XDF that is not available to ontology approach and structure information that is not available to keyword approach. The ontology approach even performed no better than keyword approach. However, the performance of ontology approach was achieved by utilizing smaller amount of information and we expect performance improvement when current ontology is extended. The major deficiency of ontology approach is that it takes significant amount of effort that is necessary to extend the ontology and specify instances based on the ontology. On the contrary, the performance of XML-based approach can be improved with lesser effort with the help of XDF.

Table 2. Performances of search engines

	Keyword	Ontology	Partial-XML	Full-XML
Top 3	0.667	0.667	0.333	1.000
Top 6	0.500	0.500	0.500	0.500
Top 10	0.600	0.500	0.600	0.600
nDCG	0.837	0.902	0.848	0.920

6. Conclusions and future works

In this paper, we presented an innovative form architecture called eXtensible Dynamic Form (XDF) to facilitate the process of collecting supplier profiles in supplier discovery. XDF allows users to extend the base form by creating their own form components. Thus, it helps to better capture users' domain-specific information. The form component search function based on synonyms helps users extend the base Form by reusing existing form components. We compared three supplier searching methods including Keyword-based search, Ontology-based search and XML-based search with XML instances generated from XDF. The experimental results demonstrate that XDF is valuable for facilitating the supplier discovery process and in turn improving the search accuracy.

There are several paths we would like to explore in the future. First, although synonym+ngram search employed by XDF is faster than WordNet-based search, its efficiency will still suffer when the number of words inputted by the user is large. More efficient algorithms need to be developed and integrated into the synonym+ngram search method. Second, it is necessary to investigate how to maintain and utilize *user-defined schemas* to improve the reusability of XDF. The schema merging algorithms and social network techniques can be considered.

7. Acknowledgements

This work was supported in part by NIST award 60NANB6D6206.

8. References

[1] C.S. .Li, Y.C.Chang, and J.R.Smith, "An e-marketplace infrastructure for information", *Intelligent Multimedia, Video and Speech Processing*, 2001, pp.182-185.

[2] J. Kim, Y. Peng, N. Ivezic, and J. Shin, "An Optimization Approach for Semantic-based XML Schema Matching", *International Journal of Trade, Economics, and Finance*, vol. 2, no. 1.2011.

[3] B. Rensmann, H. Weigand, Z. Zhao, V. Dignum, F. Dignum, and M. Hiel, "Assessing the value of mediators in collaborate business networks," *Establishing the Foundation of Collaborative Networks*, Springer, 2007, pp. 155-162.

[4] S. Colucci, T.D. Noia, E.D. Sciascio, F.M. Donini, and M. Mongiello, "Concept abduction and contraction forsemantic-based discovery of matches and negotiation spaces in ane-marketplace", *Electronic Commerce Research and Applications*, vol. 4, Summer 2005, pp. 345-361.

[5] M. Grieger, "Electronic marketplaces: A literaturereview and a callforsupplychain management research", *European Journal of Operational Research*, vol. 144, 2003, pp. 280–294.

[6] B. Kulvatunyou, H. Cho, and Y.J. Son, "A semantic web service framework to support intelligent distributed manufacturing", *International Journal of Knowledge-based and Intelligent Engineering Systems*, vol. 9, 2005, pp. 107–127.

[7] J. Jang, B. Jeong, B. Kulvatunyou, J. Chang, and H. Cho, "Discovering and integrating distributed manufacturing services with semantic manufacturing capability profiles", *International Journal of Computer IntegratedManufacturing*, vol. 21, no. 6, 2008, pp. 631–646.

[8] E. Rahmandand, and P.A.Bernstein, "A survey of approaches to automatic schema matching," *VLDB Journal*, vol. 10, no. 4, 2001, pp. 334-350.

[9] P. Shvaiko and J. Euzenat, "A survey of schema-based matching approaches", *Journal on Data Semantics IV,* LNCS 3730, 2005, pp. 146-171.

[10] E. Lai, "Mapping between HTML form and XML data", 2006. Available athttp://www.datamech.com/XMLForm/.

[11] O.Chipara and A.Slominski, "Xydra – An automatic form generator for web services", Extreme Computing Lab.Available at http://www.extreme.indiana.edu/xgws/xydra/.

[12] F. Ameri and D. Dutta, "An upper ontology for manufacturing service description", *ASME 2006 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Philadelphia, Sep. 10-13, 2006.

[13] The Open Application Group, "Open Application Group Integration Specification", version 8.0, 2002.

[14] WordNet, Available at http://wordnet.princeton.edu/.

[15] F. Ameri and D. Dutta, "A matchmaking methodology for supply chain deployment in distributed manufacturing environments", *Journal of Computing and Information Science in Engineering*, vol. 8, no. 1, 011002, 2008.

[16] K. Jarvelin and J. Kekalainen: "Cumulated gain-based evaluation of IR techniques". *ACM Transactions on Information Systems* vol. 20, no. 4, 422–446, 2002.