Abstract—Computer communications advances, functional similarities in related systems, and enhanced information description mechanisms have led to the widespread interconnection of existing systems to meet current and future system requirements in such diverse fields as healthcare, e-commerce, and military applications. Full realization of the potential offered by such interconnections can only be achieved if systems are fully interoperable. Interoperability among independently developed heterogeneous systems is difficult to achieve: systems often have different architectures, hardware platforms, operating systems, host languages and data models.

The Object-Oriented Method for Interoperability (OOMI) introduced by Young offers the capability for resolving modeling differences among heterogeneous systems, thereby enabling system interoperability. The OOMI resolves modeling differences by first identifying corresponding entities among systems using semantic and syntactic correlation approaches involving keyword search and neural network techniques. This paper describes how Semantic Web technologies can be used to improve the accuracy of the OOMI correlation methodology by adding Web Ontology Language (OWL) and reasoning capabilities to the OOMI model.

Keywords: Interoperability, Object-Oriented, Semantic Web, Information Exchange, XML, Web Ontology Language

1 Introduction

Until fairly recently, most software-intensive systems were developed as special-purpose, stand-alone applications. Recent advances in computer communications technology, the recognition of common areas of functionality in related systems, and an increased awareness of how enhanced information access can lead to improved capability have led to an increased interest in integrating current stand-alone systems to meet future system requirements.

One example of where independently developed systems might be interconnected in the healthcare domain would be the integration of your primary care physician’s healthcare management system with the local hospital’s Magnetic Resonance Imaging (MRI) system to provide your physician with near-real-time results of the scan he ordered on your knee. In business, connecting a company’s inventory management system with its retail outlet point-of-sale (POS) system can result in reduced inventory requirements and faster re-supply of needed items. Finally, a military application might connect a squadron’s mission planning suite with the theatre intelligence center to provide the latest location of the targeted adversary.

In addition to being able to exchange information, full system interoperability requires that systems share tasks as well as information [1, 2]. Such an interconnected collection of independently developed heterogeneous systems or components is referred to as a system federation. This differs from the concept of an integrated system, where homogeneous components are connected by a development team that has common objectives and shares a common view of the problem environment being modeled.

Attempts to interconnect independently developed systems that were never intended to interoperate are often faced with difficulty. Typically such existing systems were developed without any of the constructs normally included when forward-fitting a system to support interoperability. Any modification to existing systems to enable them to interoperate is costly and time-consuming. Therefore, methodologies that will enable systems to interoperate without requiring modification to existing software are highly desirable.

In the past, establishing communication between different software systems involved manually resolving differences for each system interface, which required time and cost to customize. In the Object-Oriented Method for Interoperability (OOMI) [3] Young’s first step in creating an interoperable federation of independently developed heterogeneous systems or components was to develop an object model of the entities involved in the interoperation between systems in the federation, termed a Federation Interoperability Object Model (FIOM) under the OOMI, shown in Fig. 1. Then, the FIOM is used by a translator at run-time to reconcile any representational differences among a component’s attributes and methods, as seen in Fig. 2. The translator serves as an intermediary between component systems. It can be implemented as part of a software wrapper enveloping each system, as shown in Fig. 2, or as a stand-alone module that lies between interconnected systems.

Correspondences are established among entities shared by systems in the federation in order to enable the resolution of any differences that might be found. The current method used to establish correspondences among shared entities uses a combined keyword and neural network search. Inaccuracies tied to each of these correlation approaches have led to the introduction of Semantic Web technologies to assist in the cor-
relation process and to improve search precision and recall. This has resulted in a modification of the methodology used for creating an FIOM for a federation of component systems and a projected improvement in the correlation process.

The remainder of the paper first provides background information on the types and categories of heterogeneities that can be found among systems, how the OOMI utilizes those categories in creating the FIOM, how the FIOM uses correspondences among information shared among systems in a federation to resolve system heterogeneities, and how the Semantic Web can be utilized to improve the ability to establish correspondences among shared information. The paper next discusses how Semantic Web technology can be integrated into the FIOM, comparing the original OOMI correlation model with the Semantic Web-OOMI correlation model. Status of the Semantic Web-OOMI correlation model implementation is provided next, followed by a summary of insights gained from adding Semantic Web capabilities to the OOMI, and finally a recommendation for future work.

2 Background

2.1 Heterogeneities among systems

When two systems are not designed to interoperate from the start, there can be a number of differences between them that arise when trying to include them in a system federation. There can be differences due to being developed and deployed on different hardware and software architectures, differences in the conceptual models used in their creation, differences in the structure of how data and information are organized, differences in application domain, units of measure, or data type, or differences caused by the existence of synonyms, homonyms, abbreviations or spellings used in the systems’ data models. In addition there can be differences in the attributes which are considered important to model about an entity, or in how atomic data elements are aggregated between systems.
Finally there can be disparities in the timeframe or scale used to model an entity [4] [5] [6] [7] [8].

Resolving heterogeneities among systems involves first constructing a model of the external interface of each system involved in the interoperation. Modeling differences among systems can be classified as differences in view or in representation [3]. Differences in view exist when different information is shared about an entity as a result of differences in scope, level of abstraction, or temporal validity. Fig. 3 provides an example of a difference in view where Patient Record A contains information related to the patient’s medical history, whereas Patient Record B contains medical billing information. Both model different aspects of a Patient’s Record, which is modeled as a Federation Entity (FE) in the FIOM.

Even when two systems share the same view of an entity, representational differences can also exist, such as those caused by different systems of measurement. In Fig. 4, attributes shown in italics in System A are measured using the metric system of measure whereas the same attributes in System C use the U.S. system. Differences in attribute naming, a common occurrence in computing, also results in representational differences among systems.

2.2 Introduction to the Object-Oriented Method for Interoperability (OOMI)

The OOMI [9, 3] has been developed as a means for resolving differences among software systems that were not designed to interoperate in order to enable them to exchange information and share tasks. The foundation for the OOMI is built upon the FIOM, an object model of the information shared by the external interfaces of the systems connected in a system federation.

2.2.1 Federation Interoperability Object Model (FIOM)

The FIOM consists of a number of Federation Entities (FEs), each used to represent an item shared among systems, such as a Patient Record or Lab Report. Entities will normally be shared among systems utilizing some sort of messaging standard, preferably represented using the eXtensible Markup Language (XML). These entities will subsequently be represented as Java classes in the OOMI in order to facilitate functional transformations required to resolve differences in view and representation among entities.

For each FE modeled by the systems in the FIOM, differences in view are captured by one or more Federation Entity Views (FEVs). A Federation Class Representation (FCR) provides a standard representation for each view. Creation of a standard representation for each FEV facilitates the use of a two-step process for resolving differences among software systems. In a two-step process the component representation of one system is first translated into a common standard representation before being converted into the representation used by the second system. Use of a two-step process over the more traditional direct system-to-system translation enables a reduction in the number of translations required from $n(n-1)$ to $2n$ for a federation of $n$ systems.

Associated with each FCR are one or more Component Class Representations (CCRs) which are used to provide component system specific representations of a view. Fig. 1 provides an overview of the FIOM’s top-level components.

A translation class is provided for each FCR-CCR pair to resolve differences between component system and standard representations of a view. The OOMI includes an Integrated Development Environment (OOMI IDE) which contains a Translation Generator that provides computer aid in creating the translation class in three areas. First, the IDE uses correspondences between a CCR’s and FCR’s attributes and operations identified by the user to provide a framework for translation definition. Second, it provides facilities for creation and maintenance of a library of pre-defined translation definitions for insertion into this translation framework. Finally, the OOMI IDE provides the user tools to facilitate customization of the translations used for representational difference resolution [3].

In the current OOMI, resolution of differences in view of an entity between component systems is handled by means of a transformation class used by a translator at run time. A component system’s view of an entity is represented by the FCR to which it is registered. Multiple FCRs shall be defined for an entity having different views. Each of the views for an entity are related by means of an inheritance hierarchy whereby all FCRs defined for the view extend from a common superclass. Differences in view are resolved using the transformation class through exploitation of the information contained in this FCR inheritance hierarchy using Liskov and Wing’s notion of behavioral subtyping [10]. These classes and the relationships among component systems are used by a
2.2.2 Creating a Model of System Interoperation

The first task in resolving heterogeneities among systems in the OOMI is the creation of the FIOM. In the OOMI, it is assumed that a component system’s external interface can be expressed as a series of messages by which information and task requests are exported and imported. These messages represent the entities that are to be shared among systems in the federation. In the OOMI, these messages are further presumed to be in the form of a series of XML messages, the content and composition of which are constrained by a corresponding series of XML Schema documents. Thus, communication between systems is done via XML messaging.

The first step in creating an FIOM in the current OOMI IDE is to take each XML Schema document used to represent a message exported or imported by a particular component system and convert it to an equivalent Java class, referred to in the FIOM as a Component Class Representation (CCR). This CCR is added to a CCR Library to be used by the translator during run-time resolution of system differences.

Either as each CCR is added to the FIOM, or after all CCRs have been created, an FCR is selected which holds the same view as the CCR. In order for two systems to have the same view of an entity, “there must be no difference in scope, level of abstraction, or temporal validity between the two systems’ models of the entity … this means that at some level of aggregation each attribute set and each operation set of each system must be in one-to-one correspondence” [3]. Furthermore, “corresponding operations must be behaviorally equivalent for two systems to have the same view of a real-world entity” [3]. This FCR may already exist in the FIOM or if not, it must be created. Creating an FCR can follow either a step-by-step process whereby the FCR is defined and attributes and operations added, or by following a similar procedure used to create a CCR from an XML Schema document. The current implementation of the OOMI IDE provides only the latter method for FCR creation.

2.2.3 Original OOMI Correlation Model

When adding a new component system CCR to a federation, effort must be expended to determine if there are existing Federation Entities and FEV’s in the FIOM to which the new system’s entities correspond. If corresponding FE’s and FEV’s are found, translation and transformation classes can be created to resolve any differences in representation or view, respectively, between the component system and standard representations of those entities. If such FE’s and FEV’s cannot be found, the FIOM is expanded to include FEV’s and / or FE’s to which the new CCR can be added.

Determining whether an FCR already exists in the FIOM which corresponds to a CCR being added can be done by manually browsing the FIOM or by using a more automated approach. The Component Model Correlator is responsible for locating corresponding FE’s and FEV’s in the FIOM if they exist. The original Correlator used in the OOMI utilized a combined keyword and neural network-based approach for establishing correspondences among CCRs and FCRs in the FIOM. While providing a foundation for demonstrating the correlation process in the OOMI, this approach is believed to not provide sufficient values for precision and recall in the correlation process. Therefore, an investigation was conducted to determine what added promise the Semantic Web might offer to the correlation problem.

Fig. 5 summarizes how the components in the FIOM are used in resolving differences in representation and view between a source and destination system. First, an XML Schema, to which a source system XML document conforms, is used to create a CCR class using XML data binding [11]. Then, the source system XML document is converted to a CCR object which is an instance of the CCR class created initially using the same data binding process. The CCR object is converted to an FCR object using the CCR-FCR translation class created by the OOMI IDE Translation Generator described previously. The FCR object, in turn, is an instance of an FCR class created by the OOMI IDE. The source system FCR object is converted to a destination system FCR object using Liskov and Wing’s notion of behavioral subtyping [10].

2.3 Introduction to Semantic Web technologies

As discussed in the previous section, the process of constructing an FIOM includes creating a standard representation, called an FCR, for every view of an entity that is present in the object model. The terminology used to represent the information in the FCR should be based on terms which reflect the federation-sanctioned representation of an entity’s state and behavior [3].

An ontology defines the terms used to represent an area of knowledge. There have been many languages developed to de-
scribe the contents of an ontology, the most predominant of which is the Web Ontology Language (OWL) which was selected by the World Wide Web Consortium (W3C) as its standard ontology language [12]. OWL is a Resource Description Framework (RDF) language developed by the W3C for defining classes and properties. It can also be used for enabling more powerful reasoning and inference over relationships between classes and properties as well as between objects and classes. OWL is based on XML-authoring tools, used mainly to express the needs of computer applications to deal with knowledge and information presented in the internet [13].

Soon after the W3C released the OWL language, it has “rapidly become a de facto standard for ontology development in general” [14]. Since OWL became a standard for creating ontologies it started influencing additional research such as investigation of reasoning techniques for ontology designs [14]. This research led to the creation of numerous reasoners for the OWL language, referred to as OWL reasoners. Some of the most popular open-source reasoners are Pellet [15], FaCT++ [16], and HermiT [17].

3 Adding Semantic Web technology to the OOMI

3.1 Overview

Since an FCR is defined to embody the standard representation of an entity, the information or terminology used to model an FCR should be taken from an ontology of federation-sanctioned terms and components. As the first step in adding Semantic Web technology to the OOMI, a Federation Ontology Library is added to the FIOM. The Federation Ontology Library defines the ontology classes, properties and individuals needed to specify the standard representation of the entities involved in system interoperation.

Recalling from earlier, even though two component systems share the same view of an entity, they may not represent that view in the same manner. A CCR is used to capture the unique way each component system may choose to represent its view of an entity. The next step in constructing the FIOM is to determine the view each component system represents. This can be accomplished by determining the FCR to which each CCR corresponds.

Locating an FCR which corresponds with a particular CCR is treated as a class correlation problem in the OOMI. CCR-FCR correlation in the original OOMI is handled using keyword and neural network techniques to provide semantic and syntactic matching between the classes used to represent a CCR and FCR. In subsequent sections we describe how the use of ontologies and reasoners can be used to improve upon the results returned by the original OOMI correlation process.

3.2 Semantic Web-OOMI Correlation Model

In the Semantic Web-OOMI Correlation Model, correlation is conducted between OWL classes using reasoning capabilities provided by an OWL reasoner such as Pellet [15], FaCT++ [16], or HermiT [17]. First, a Federation Ontology Library is constructed for the federation from federation-sanctioned OWL classes, properties and individuals. This library can either be imported from an existing ontology or ontologies or created using tools included with the OOMI Integrated Development Environment (IDE).

In the Semantic Web model it is assumed that the external interfaces of component systems are available as XML Schema Documents (XSD). Then, in order to utilize OWL to establish a correspondence between a component system and standard representation of an entity, each XSD which represents an entity from the component system external interface is converted into an OWL Ontology class, termed a CCR OWL class in Fig. 6. Finally, the Federation Ontology Library is queried using an OWL reasoner to find a class in the library corresponding to the CCR OWL class. If such a corresponding class is found, termed an FCR OWL class, an interoperability engineer creates an object model of that class with selected properties by converting the candidate FCR OWL class into a corresponding FCR class.

There may be some situations where the interoperability engineer cannot find a matching class in the current ontology library. Then, with permission of the ontology library custodian, the interoperability engineer can use the OOMI IDE’s OWL editor capabilities to add a new class to the ontology library. The search for an FCR OWL class which corresponds to the desired CCR OWL class is regenerated, this time with success in finding a match almost certain. This FCR OWL class is then converted to a corresponding FCR which is used to create the desired CCR-FCR translation. This paper is mainly focused on providing the capabilities of creating and managing the OWL ontology and converting classes and their properties from an OWL ontology into object models so that it would be easy for an interoperability engineer to create corresponding FCRs.

The process to search for a corresponding OWL class in the Federation Ontology library as well as the creation of a CCR object model from an XSD representation of a system’s external interface is left as future work. Creating the translation class to resolve the representational differences between the component CCR and standard FCR has been done previously by Young [3].

4 Implementation status

At its core, this work defines a new correlation methodology for the OOMI as depicted in Fig. 6. The original OOMI relied on a keyword search and neural network-based approach for locating existing standard representations of a component system’s entities which are shared via its external interface in the FIOM. The new methodology incorporates Semantic Web technology to assist in the correlation process used to locate standard entity representations.

The foundation for the Semantic Web technology used in redefining the FIOM creation process is the creation of a Federation Ontology Library containing OWL classes, properties and individuals used to define the standard representation of entities to be shared by components in the
federation. Included with the addition of a new Federation Ontology Library are tools which enable an interoperability engineer to 1) import an existing library or libraries from other systems, federations or domains; 2) create a new library, adding classes, properties, and individuals to define the entities shared among components in the federation, and 3) modify an existing library by adding, deleting or changing existing items in the library.

As depicted in Fig. 6 and discussed previously, an entity imported to or exported from a component system is represented by an XML Schema document. This document is to be converted into an equivalent CCR OWL Class which is used to locate an equivalent standard representation of the entity, termed an FCR OWL class, in the Federation Ontology Library using an OWL-based reasoner. While not currently developed for the OOMI, such a reasoner should be able to be constructed using currently available technology such as that provided by Pellet [15], FaCT++ [16], or HermiT [17]. This portion of the new correlation methodology has not been implemented and is left for future work.

If a corresponding FCR OWL class is found in the library, it is converted into an equivalent FCR and used to create the CCR-FCR translation class used to resolve representational differences between the component and standard versions of the entity. The ability to convert an FCR OWL class into an equivalent FCR has been partially completed; it lacks only the user interface necessary to select which elements from the FCR OWL class are to be used in creating the FCR. The functionality required to create a CCR-FCR translation class from a matching CCR and FCR had been completed as part of the original OOMI IDE development effort.

Should an existing FCR OWL class not be found which matches the CCR OWL class of interest, a new FCR OWL class can be created using the existing Federation Ontology Library tools discussed previously. Repeating the process for locating a matching standard representation for a given component representation should result in this newly created FCR OWL class being found, with the CCR-FCR translation process proceeding as previously described.

5 Conclusion and recommendation for future work

Transition of the correlation process for matching component and standard representations of entities involved in the interoperation among systems from a keyword and neural network based approach to one using Semantic Web technologies promises to increase the precision and recall results of such search efforts. In addition, introducing the ontology-based approach for maintaining the standard representation of such entities provides an accepted list of terms from which to define the vocabulary for a federation,
Ontologies and standards-based approaches for achieving interoperability enable syntactic and semantic differences among systems to be addressed by dictating a common structure and meaning for entities in a domain. An ontology defines the set of terms, entities, objects, classes, and their relationships, providing formal definitions and axioms that constrain the interpretation of these elements [18]. Adopting an ontology as a standard provides a clear, unambiguous structure and meaning to the elements in the domain to which the standard applies.

While an ontology and standards-based approach to interoperability has many benefits, one of the primary limitations to such an approach is the number of standards that have been defined. In the Health Information Technology (HIT) domain, Eichelberg, Aden, Riesmeier, Dogac, and Laleci review seven competing standards used in creating electronic health records (EHRs) with no clear winners found among them [19]. In fact, Wendt points out that Microsoft is involved in more than 100 standards-based organizations worldwide [20]. Another difficulty with standards-based approaches lies in the conformance of systems to the standards-when systems deviate from a standard they erode its value [21]. A final limitation with standards is their resistance to change. As standards are generally overseen by some organization to ensure their applicability, the time required for adapting an existing standard to changing technology or circumstances can be excessive.

The OOMI-Semantic Web approach provides the benefits of such an ontology-based methodology for maintaining the standard representation of such entities while minimizing its liabilities. By providing the capability to custom define the ontology to be used for a specified federation, the OOMI-Semantic Web method eliminates the problem of competing or conflicting standards by creating a specific ontology for the federation in question. While this ontology may be derived from an existing ontology or ontologies, it is specific to the federation being defined. This approach also virtually eliminates non-conformance among federation systems by providing the capability to customize the ontology to meet the demands of its component systems and is more flexible than traditional standards to changes in technology or requirements.

This paper lays the foundation for adapting the OOMI to utilize Semantic Web technology in constructing the FIOM. While this work goes a long way in taking advantage of the benefits a Semantic Web based approach brings, there is still work remaining to realize the full advantage of this capability. Such future work includes: 1) implementing an XML Schema to CCR Ontology Class converter; 2) implementing and integrating an OWL class reasoner to match component and standard ontology representations of an entity; 3) generating an FCR OWL class from the Ontology Library class matching the selected component representation of an entity, and 4) converting the generated FCR OWL class to an equivalent FCR. Full integration of Semantic Web technology into the OOMI will enable further exploration of areas where the OOMI approach may be adopted to improve system interoperability.

6 References