A Framework for Ontologies-based User Interface Integration

Heiko Paulheim and Florian Probst
SAP Research CEC Darmstadt
Bleichstrasse 8
64283 Darmstadt, Germany
{heiko.paulheim,f.probst}@sap.com

ABSTRACT

Application integration can be carried out on three different levels: the data source level, the business logic level, and the user interface level. With ontologies-based integration on the data source level dating back to the 1990s and semantic web services for integrating on the business logic level coming of age, it is time for the next logical step: employing ontologies for integration on the user interface level. Such an approach will improve both the development times and the usability of integrated applications. In this poster, we present an approach employing ontologies for integrating applications on the user interface level.

Categories and Subject Descriptors

D.2.2 [**Software Engineering**]: Design Tools and Techniques—*User Interfaces*; D.2.13 [**Reusable Software**]: Reuse models

General Terms

Design, Algorithms

Keywords

Ontologies, User Interfaces, Integration

1. INTRODUCTION

Applications are often described in three layers: data, business logic, and user interface. Consequently, application integration can be performed on each of those three levels [1].

Integrating applications on the user interface level means reusing *existing* user interfaces or parts thereof and coupling them in a way that *a user can interact with those interfaces as if they were a single application*, e.g. by using a common toolbar and allowing drag and drop from one application to the other. Current approaches such as plugin-based systems, portals, and, most recently, mashups are either very limited with respect to cross-application interaction or require deep changes of the applications in order to facilitate such interactions [1].

With semantic database integration as well as ontology-based agents and semantic web services, there have been considerable efforts to using ontologies in the integration on the database and business logic layer. In contrast to those approaches, user interface level integration has two significant advantages:

First, the development of the user interface consumes about 50% of the total efforts in developing an application [5]. Therefore, the *benefit from reusing existing user interface components* is significant.

Second, users interacting with applications integrated on the user interface level will experience a *decreased learning effort* if already familiar with the applications' original interfaces, compared to interacting with a newly developed common user interface.

Yu et al. [7] argue that user interface integration requires a description of the interfaces to be integrated that is *formal*, *human readable*, *modular*, and *simple*. Ontologies are formal, readable for people with an appropriate background, and also provide the possibility for modularization [3]. As simplicity is a rather subjective criterion, and the description language must be flexible enough to cover all possible cases of integration, hence must not be too simple, we claim that ontologies are a suitable approach.

Since, as discussed above, *interaction* plays a crucial role in integrated user interfaces as well as it imposes problems in current approaches to user interface integration, the approach shown in this poster aims at describing and enabling such *cross-application interactions* by employing ontologies.

2. PROTOTYPE

So far, a first prototype has been developed that shows how user interfaces can be integrated by using ontologies [6].

2.1 Types of Ontologies

Three types of ontologies are used (see Fig. 1):

- An *ontology of the user interfaces and interactions domain*, which defines basic categories for describing applications.
- An *ontology of the application's real world domain*, which defines the categories of real world objects of the domain that the integrated application is built for (such as banking, travel, etc.). The information objects processed by the application *represent* those real worlds objects.
- One or more *application ontologies*, which use the user interfaces and interactions ontology's basic concepts to describe the applications to be integrated, and the interactions that are possible with them, as well as the necessary parts of those UIs. The application ontologies may refer to the real world domain ontology for

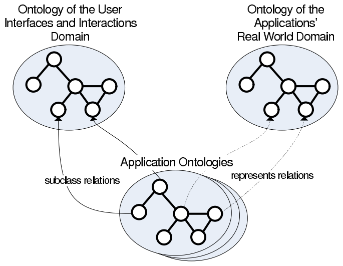
Ontology of the User Interfaces and Interactions Domain
Ontology of the Applications' Real World Domain
Application Ontologies
subclass relations
represents relations

Figure 1: Using two domain ontologies and several application ontologies for integration on the user interface level

describing the types of objects that may be processed by the integrated applications. During the integration process, one application ontology per integrated application has to be developed.

This categorization follows Guarino's classification [4] – here, two kinds of domain ontologies are used. While the ontology of the user interfaces and interactions domain is a part of the integration framework, the real world domain ontology (or ontologies in case of modular domain ontologies or cross-domain applications) and application ontologies are dynamically added for each integrated application. There is no direct connection from the user interfaces and interactions domain ontology to the real world domain ontology, thus, the framework is domain independent.

2.2 Framework

Our framework prototype is based on Java and the OntoBroker reasoner (see Fig. 2). Integrated applications are described by an application ontology and communicate via events which are annotated using that ontology and processed by a reasoning component. When one application sends an event, the reasoner reads the event, queries the application ontologies to determine those applications which declare to react to that sort of event, and notifies the respective application. Thus, no application has to directly react to other applications' events; *cross-application interactions* are facilitated only by processing the event types defined in its the respective application ontologies.

To allow mediation between different data models, an *object ontology mapping registry* is introduced, which stores annotations of its data model. Classes and properties may be annotated with concepts from the domain ontology. When the reasoner receives an object from an application or vice versa, the receiver consults the registry to analyse that object and convert it into a representation which can be processed by the reasoner.

3. ACHIEVEMENTS

In this poster, we present the idea of using ontologies for integrating applications on the user interface level. In our framework, applications are described by application ontologies, making use of two or more shared domain ontologies.

A first prototype shows that the approach is feasible. It has been successfully used in the SoKNOS project [2], where an integrated emergency management software has been built, consisting of twelve integrated single applications. A more detailed evaluation by examining integration times and interviewing developers is planned.

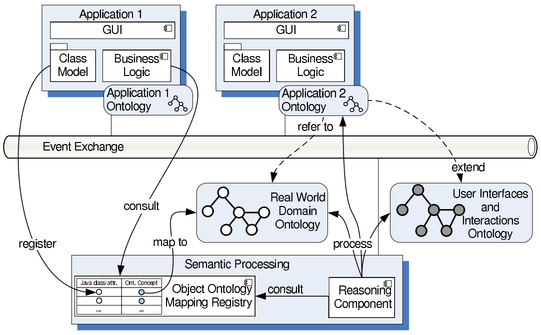

Figure 2: Overview on the prototype's framework architecture. Applications are described by ontologies. A reasoning component evaluates those ontologies to facilitate integration at run time.

In summary, we believe that such a framework for integrating applications on the user interface level is a useful complement to existing integration efforts on the data and business logic level.

Acknowledgements

The work presented in this paper has been partly funded by the German Federal Ministry of Education and Research under grant no. 01ISO7009.

4. REFERENCES

[1] F. Daniel, J. Yu, B. Benatallah, F. Casati, M. Matera, and R. Saint-Paul. Understanding UI Integration: A Survey of Problems, Technologies, and Opportunities. *IEEE Internet Computing*, 11(3):59–66, 2007.

[2] S. Doeweling, F. Probst, T. Ziegert, and K. Manske. SoKNOS - An Interactive Visual Emergency Management Framework. In R. D. Amicis, R. Stojanovic, and G. Conti, editors, *GeoSpatial Visual Analytics*, NATO Science for Peace and Security Series C: Environmental Security, pages 251–262. Springer, 2009.

[3] T. R. Gruber. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. volume 43, pages 907–928, Duluth, MN, USA, 1995. Academic Press, Inc.

[4] N. Guarino, editor. *Formal Ontology and Information Systems*. IOS Press, 1998.

[5] B. A. Myers and M. B. Rosson. Survey on user interface programming. In *CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 195–202, New York, NY, USA, 1992. ACM.

[6] H. Paulheim. Ontology-based Modularization of User Interfaces. In G. Calvary, T. C. N. Graham, and P. Gray, editors, *Proceedings of The ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, pages 23–28. ACM, 2009.

[7] J. Yu, B. Benatallah, R. Saint-Paul, F. Casati, F. Daniel, and M. Matera. A framework for rapid integration of presentation components. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 923–932, New York, NY, USA, 2007. ACM.