Chapter 5
Streamlining Semantic Integration Systems

Yannis Kalfoglou
Ricoh Europe plc & University of Southampton, UK

Bo Hu
SAP Research CEC Belfast, UK

EXECUTIVE SUMMARY

Yannis Kalfoglou and Bo Hu argue for the use of a streamlined approach to integrate semantic integration systems. The authors elaborate on the abundance and diversity of semantic integration solutions and how this impairs strict engineering practice and ease of application. The versatile and dynamic nature of these solutions comes at a price: they are not working in sync with each other neither is it easy to align them. Rather, they work as standalone systems often leading to diverse and sometimes incompatible results. Hence the irony that we might need to address the interoperability issue of tools tackling information interoperability. Kalfoglou and Hu also report on an exemplar case from the field of ontology mapping where systems that used seemingly similar integration algorithms and data, yield different results which are arbitrary formatted and annotated making interpretation and reuse of the results difficult. This makes it difficult to apply semantic integration solutions in a principled manner. The authors argue for a holistic approach to streamline and glue together different integration systems and algorithms. This will bring uniformity of results and effective application of the semantic integration solutions. If the proposed streamlining respects design principles of the underlying systems, then the engineers will have maximum configuration power and tune the streamlined systems in order to get uniform and well understood results. The authors propose a framework for building such streamlined system based on engineering principles and an exemplar, purpose built system, CROSI Mapping System (CMS), which targets the problem of ontology mapping.

DOI: 10.4018/978-1-60566-894-9.ch005
THE NECESSITY FOR SEMANTIC INTEROPERABILITY

“We need interoperable systems.”

Time has long gone when manufacturers designed and assembled artefacts as stand-alone objects, ready to be used for whatever purpose they had been originally conceived. The necessity for more and more complex devices and the industrialisation/standardisation of manufacturing processes have led to the engineering of very specialised components that can be reused for a variety of component-based systems, which neither have been designed nor assembled by a sole manufacturer and for a unique purpose.

Analogously, in our information age, a similar phenomenon has occurred to the “manufacturing” of information technology (IT) artefacts. Originally, software applications, databases, and expert systems were all designed and constructed by a dedicated group of software or knowledge engineers who had overall control of the entire lifecycle of IT artefacts. But this time has gone too, as software engineering praxis is shifting from the implementation of custom-made, stand-alone systems to component-based software engineering (COTS, ERP, etc.). Databases are gradually deployed in distributed architectures and subsequently federated, and knowledge-based systems are built by reusing more and more previously constructed knowledge bases and inference engines. A compelling example on this front is SAP Business One™, which contains 14 core modules specialised in the immediate and long-term needs (e.g. customer relationship management, finance, purchasing, etc.) of small or medium-sized enterprises (SMEs). Individual SMEs then decide which fields of business activity they want to support and align the relatively independent modules into an integral framework. While accessing a raft of functionalities through one seemingly unified interface, the users normally are not aware of the underlying integration effort that seamlessly juxtaposes heterogeneous data from different units of an organisation and different business policies.

Moreover, the World Wide Web, and its ambitious extension the Semantic Web, has brought us an unprecedented global distribution of information in the form of hypertext documents, online databases, open-source code, terminological repositories (like for example Wiktionary), web services, blogs, etc., all of which continually challenge the traditional role of IT in our society. As a result, the distributed nature of IT systems has experienced a dramatic explosion with major IT suppliers starting to provide on demand web-based services (empowered by Service Oriented Architectures) instead of all-in-one boxed products and localised solutions (fine-tuned against the legal system, currency, and accountancy policy in each country) instead of universal solutions.

But in contrast to traditional industrial manufacturing and composition of artefacts, the composition and interaction of IT components at the level of distribution on the Web is still at its infancy, and we are just grasping the scope of this endeavour: successful IT component interoperability beyond basic syntactic communication is very hard. Unlike with industrial manufacturing, our era’s basic commodity around which all IT technology is evolving, namely information, is not yet well understood. While industrial and civil engineers know how to apply well-established mathematical models to derive an artefact’s characteristics from the physical properties of its components, software engineers and knowledge workers lack the machinery that will enable them to do the same with information assets. The problem with understanding information is that we need ways with which we can reveal, expose and communicate the meaning (semantics) of information. But this has eluded any mechanistic approach to interoperability. Putting together different databases has proved to be successful only for closed environments and under very strong assumptions; the same holds for distributed arti-