
Ensuring open and scalable interoperability for smart information system

Boubou Thiam Niang^{*1}, Giacomo Kahn¹, Nawel Amokrane², Yacine Ouzrout¹,
Mustapha Derras², and Jannik Laval¹

¹DISP, EA4570, 69676 Bron, France – Univ Lyon, Univ Lumière Lyon 2, INSA Lyon, Université Claude
Bernard Lyon 1 – France

²Berger-Levrault – Berger-Levrault – France

Résumé

Information systems (IS) in modern companies must be reactive and able to communicate with third-party IS. It is therefore necessary to establish agile interoperability between information systems. Indeed, the components are designed independently and meet different technical and domain standards that are constantly evolving. Establishing and maintaining interoperability becomes a major challenge, as solutions are developed manually and are not reusable in most cases. This poster presents our work on the automatic generation of interoperability mechanisms, called connectors. Indeed, connectors share common characteristics. For instance, most components use a protocol for communication, encapsulate the information exchanged using data formats. In addition, each of these common characteristics can vary depending on the communication need. With this in mind, we have adopted for software product line engineering to help manage connector variability. Our approach identifies commonalities and variability in connectors from existing interoperability mechanisms implemented in the industry through reverse engineering. We then constructed a feature model to represent the common and variable characteristics of the connectors. These extracted features allowed us to create a connector metamodel that covers all the entities needed to create any connector. This metamodel provides an abstraction level for connectors to make them platform independent. The resulting metamodel of connectors shows that they can be considered as first-class entities, which makes it possible to automatically generate source code for these connectors or to generate a configuration to create connectors from existing subcomponents. The generation of the codes or the configuration of the connectors is done according to the configuration of a valid connector based on the feature model. An example of use case is available on git <https://cvs.disp-lab.fr/spl-interop-connector-generation-approach>.

^{*}Intervenant