CONSOLIDATING SEMANTIC INTEROPERABILITY IN SOFTWARE ARCHITECTURES

Access-and-play semantic interoperability in contemporary architectural paradigms

Paul Brandt,
Eindhoven University of Technology; Netherlands Organization of Applied Scientific Research TNO, Den Haag, The Netherlands,

Eric Grandry,
Luxembourg Institute of Science and Technology, Esch-sur-Alzette, Luxembourg,

Johan Lukkien,
Eindhoven University of Technology, Eindhoven, The Netherlands,

Twan Basten,
Eindhoven University of Technology, Eindhoven, The Netherlands,

Abstract

Background: *Access-and-Play SIOp is the next glass ceiling in ICT. We can think of two approaches to break through the ceiling, i.e., using either strong AI or weak AI. Strong AI is not yet available, while weak AI, despite its current applications in Semantic Web or ontologies, has not yet been embedded in contemporary software architectural paradigms. Current approaches towards SIOp can be considered accepted folklore.*

Objective: *The objective of this study is to identify and define the fundamental guidance towards access-and-play semantic interoperability in contemporary architectural paradigms.*

Method: *Our approach is based on the discipline of semiotics. After identifying semiotic shortcomings in MDA and view-based architectural paradigms and their subsequent definition as missing concerns, we develop the necessary guiding architectural principles. We finally investigate their consolidation in view-based and model-driven architectural paradigms. [We evaluate these principles by designing a reference architecture and proof its use in SIOp between two software agents.]*

Results: *The semiotic approach/discipline demonstrates/proves semantics to be the result of a reciprocity between data and the software code that operates on them. The major shortcomings in architectural paradigms to account for semantic interoperability are their negligence of semiotic fundamentals and, particularly, the absence of an explicit ontological commitment that stands at the root of semantics. Therefore, the concern about a semantic loose coupling should be added to the architectural paradigms. The supporting principles are (i) semantic transparency, (ii) semantic separation of concerns, and (iii) explicit computational semantics. In view-based architectures their consolidation implies a new semantic view, while the MDA paradigm requires an ontological commitment on M3. Both paradigms need to include an alignment-based semantic mediation capability.*

Conclusions: *Access-and-play SIOp can be achieved when considering semiotic fundamentals and adding loosely coupled semantics to contemporary architectural paradigms.*

Chapter 1

Introduction

Never before, data were so ubiquitous, and managed access to external data was so easy. Because current ICT is unable to *use* all that same external, non-native data as access-and-play service, agility in business collaboration is hampered in all domains. For instance, consider two nations each of which registers a company, as follows:

brandtp, 2018-04-20 This is a typical TOOP example that unfortunately expresses a more structural than clear semantic issues. Improve on it?

Nation A:

```
RegisteredOrganisation ---hasCompanyCode--> CompanyCode ---hasNotation--> "DX-2004/84492"
```

Nation B:

```
MaatschappelijkeActiviteit ---hasKvkNummer--> "DX-2004/84492"
```

This example shows that an identifier for a company, `DX-2004/84492`, is considered by nation B a simple label, pointed by relationship `hasKvkNummer` and belonging to the class `MaatschappelijkeActiviteit`. The class denotes the company, which can be considered equivalent to the class `RegisteredOrganisation` from nation A. However, nation A considers the company identifier a class in its own right, `CompanyCode`, with a particular `hasNotation` relation to its value. In this simple example we already see two characteristics that hamper interoperability between the nations. Firstly, use of different terms for equivalent things, and secondly, application of different structures to express equivalent things. When the values had denoted a magnitude as opposed to the current identifier, e.g., a temperature, a third complicating characteristic would be the use of different measures of unit. Many such *semantic interoperability* (SIOp) issues exist, and a short overview is given in Section ##.

So, lack of automated SIOp represents the next glass ceiling for ICT. The most disconcerting consequences are time-to-deliver, flat interoperability failures, and even seemingly correct but quite invalid data analysis probably leading to devastating system behaviour.

Current SIOp implementations are essentially based on the (time-consuming) process of establishing a convention on the semantics of the terms that are exchanged during collaboration. Such conventions can be considered a semantic monolith, which makes dealing with data outside the monolith impossible, unless again a time consuming (months) semantic adoption process is applied. Moreover, these semantic conventions consider semantic heterogeneity as a bug instead of a feature necessary to achieve semantic accuracy. But still, this conventions-based approach towards SIOp is accepted folklore in ICT. In view of the large uptake of the Internet, the Internet of Things (IoT), cloud computing and big data, and in view of economical pressure to intensify enterprise collaboration, we consider this approach "too little, too late".

Business agility will emerge once you remove the semantic monolith. Then, you only have to access-and-play, namely (i) achieve SIOp in due time (ii) with all external data. In this way, software systems, applications or software components – henceforth collectively denoted as software agents – can quickly realise collaboration, also with data not anticipated for during software design, at any point in their life cycle. Metaphorically speaking, we consider access-and-play SIOp as a *bridge* overarching a (semantic) gap: with *bridgeheads* on each side of the gap, with a *spanning* resting on them to structurally support the bridge and its traffic, and with a *roadway* enabling the crossing of the traffic. Finally, architectural *principles* provide the necessary guidance to the architect for the various design decisions that effectively result in a particular bridge over a particular (semantic) gap. Our contributions to consolidating semantic interoperability in software architectures are fourfold, and represented as architectural principles and concerns, as follows:

- *Principles*: We base SIOp on establishing loose-coupling at the semantic level by introducing principles on semantic separation of concerns and semantic transparency (Section ##), and show how these principles can be operationalised;
- *Semantic concerns (bridgehead)*: Based on the discipline of semiotics, we provide for a fundamental notion on software semantics and include this as a separate artifact contributing to the software agent. We show how such explicit notion on semantics fits with other architectural components (Section ##);
- *Alignment concerns (spanning)*: Since "strong AI" does not yet exist, SIOp remains in demand of human intervention in order to reconcile the semantic differences between collaborating software agents. We reduce the necessary human intervention to complement weak AI to a task that suffices in the context of SIOp, viz. authoring semantic alignments only (Section ##);
- *Mediation concerns (roadway)*: We provide for a prototypical implementation of a mediator as the necessary component to automatically translate data when transferred between the collaborating software agents (Section ##);

Based on these contributions we [argue/defend] that a definitive SIOp should aim at an access-and-play potential as upfront "design to collaborate".

References