

I am currently working in the context of a 4-year research project called ARRIBA that has as goal to offer a range of more sophisticated lightweight methods and tools to support the integration of software entities that have not necessarily been designed to coexist. A first phase in achieving this goal is understanding these systems.

This project is supported by a user committee of Belgian enterprises and their input is a range of case studies, mainly large legacy systems in COBOL, but also in C.

Within this domain, we are currently exploring the use of aspect technology for legacy systems evolution support. Aspect technology aims at creating a better separation of concerns for software systems that cannot be fully decomposed using the decomposition mechanism a language offers. Legacy software is an excellent candidate for applying this technology since they (have) evolve(d) continuously, and thus have even a bigger need for a clean separation of concerns.

By now, we already established a mapping of COBOL code to an XML representation of that code, and we are able to go back from that XML representation to COBOL. And we are using a declarative language as a medium to express aspects and pointcuts to instrument the XML representation of the COBOL code.

Although my main focus right now is on reengineering large non-OO legacy systems, many reengineering techniques can be applied on different systems, hence my interest in this WOOR workshop.