ABSTRACT

In this thesis, the overall **architecture for a tool-environment** targeted to the building of Distributed Control Systems with Real Time constraints is proposed. This open, flexible and extensible architecture allows the direct **integration** of those **software tools** most frequently used in the involved domains (control engineering, real-time engineering, software engineering). The integration is achieved by means of the **separation** of the **semantics of the software tools** (managed by a **Tool Collaboration Engine**) and the semantics of the **domain-specific models** handled by the environment (managed by a **Model Collaboration Engine**). The **technologies to be used** in the implementation of the set of modules of the architecture are also discussed.

The environment is based on the **declarative programming** and the extensive use of ***Little Languages* expressed in XML**. These languages are easy to use and understand, they fit the needs of each specialist (not only programmers), they can be straightforwardly modified and extended (no need of new compilations) and standard APIs and analyzers can deal with their instances and perform validations. But the most interesting quality of these languages, concerning the environment, is their nature of **formal languages** (not in the mathematical sense, but regarding their ability to perform instance validation). The requirements of each domain-specific language can be expressed as formal grammars (**schemas**) and/or as rules (**schematrons**) and the translations between languages are described in **XSLT** style-sheets.

Thus, the proposed design performs the **"horizontal" integration of "vertical" domain-specific tools**. Tool integration allows domain experts to communicate and to collaborate towards the design of Distributed Control Systems with temporal constraints. The framework also allows the traceability of functional and non-functional requirements through the design phases.