UNIVERSITÀ DEGLI STUDI DI SALERNO

Department of Computer Science

PhD in Computer Science

XIII Ciclo - Nuova Serie

PhD Thesis in Computer Science

Teamwork Collaboration around Simulation Data in an Industrial Context

Author

Donato Pirozzi

PhD Program Chair
Prof. Giuseppe Persiano

Supervisor
Prof. Vittorio Scarano

Academic Year 2013-2014

Abstract

Nowadays even more small, medium and large enterprises are world-wide and compete on a global market. In order to face the new challenges, industries have multiple co-located and geographically dispersed teams that work across time, space, and organisational boundaries. A virtual team or a dispersed team is a group of geographically, organisationally and/or time dispersed knowledge workers who coordinate their work using electronic technologies to accomplish a common goal. The advent of Internet and Computer Supported Cooperative Work (CSCW) technologies can reduce the distances between these teams and are used to support the collaboration among them. The topic of this thesis concerns the engineering dispersed teams and their collaboration within enterprises. In this context, the contributions of this thesis are the following: I was able to (1) identify the key collaborative requirements analysing a real use case of two engineering dispersed teams within Fiat Chrysler Automobiles; (2) address each of them with an integrated, extensible and modular architecture; (3) implement a working industrial prototype called Floasys to collect, centralise, search, and share simulations as well as automate repetitive, error-prone and time-consuming tasks like the document generation; (4) design a tool called ExploraTool to visually explore a repository of simulations provided by Floasys, and (5) identify the possible extensions of this work to other contexts (like aeronautic, rail and naval sectors).

The first research aim of this work is the analysis of the key collaborative requirements within a real industrial use case of geographically dispersed teams. In order to gather these requirements, I worked closely with two geographically separated engineering teams in Fiat Chrysler Automobiles (FCA): one team located in Pomigliano D'Arco (Italy) and the other one in Torino (Italy). Both teams use computer numerical Computational Fluid Dynamic (CFD) simulations to design vehicle products simulating physical phenomenons, such as vehicle aerodynamic and its drag coefficient, or the internal flow for the passengers thermal comfort. The applied methodology to collect the collaborative and engineering requirements is based on an extensive literature review, on site directly observations, stakeholders' interviews and an user survey. The identified key collaborative requirements as actions to perform to improve the collaboration among dispersed teams are: centralise simulation data, provide metadata over simulation data, provide search facility, simulation data versioning, and data sharing. Engineers, in the analysed field, use multiple simulator software, so in order to centralise simulation data, it is fundamental to collect data from multiple heterogeneous sources avoiding the Ven-

dor Lock-In Anti-Pattern. In according to the gathered collaborative and engineering requirements, a working real prototype called Floasys has been developed. The Floasys target customers are industries who use CFD simulators to design their products. Floasys collects, centralises, and stores simulation data in open format (e.g., XML). Floasys provides additional services over collected data like a simulator independent SearchTool, that is very useful to get simulations performed by different engineers, and compare the found simulations performances over multiple design revisions. In addition, the system allows the data sharing through URLs exchange.

From architectural point of view, Floasys meets the extensibility and modularity Non-Functional requirements to be tailored to the customers needs, accommodate future requirements and be used in other departments. In order to meet the extensibility and modularity Non-Functional Requirements, the architecture is based on the concept of plug-in. In this way, each module of the architecture is a plug-in that can be replaced with another equivalent implementation. The Floasys uses a three layers approach. It integrates on the bottom layer the data source wrappers (e.g., simulators software that generates simulation data, or test-beds that generates experimental data). The middle layer abstracts and manages the data source heterogeneities. On the top layer there are the tools that provide collaborative and engineering functionalities to users.

Although this research activity concerns an automotive use case, issues faced within this sector seem to be very common issues also in other sectors as highlighted in the existing literature, especially for the list of gathered requirements. Therefore, many of the considerations and design decisions described through this work could be used also for other type of simulations and experiments in other contexts (i.e., aeronautic, rail and naval sectors).

Collecting a huge amount of data from multiple sources and centralising them in open format, introduces the need to have an overview, and at same time, explore and query the dataset to get the desired data. Therefore, another contribution of this work is an interactive tool to visualise, explore and query simulations repositories called briefly EsploraTool that is a Floasys tool. Although the tool idea was born in the context of simulation and experimental data, it is enough general to be used with any dataset so that it has been tested with the amazon.co.uk clothing catalogue to demonstrate its generality. The tool is based on the Euler Venn diagrams. The tool represents the data items in a hierarchical way. The user can explore the dataset through drill-down and roll-up operations to get more or less dataset details. Going down through the hierarchy the user is filtering items within the dataset and making a graphic query. Using the experimental data, the idea is that engineers explore the available experiments and get two or more experiments to compare together. After the tool design and implementation stages, now the tool is under testing with real users to perform a User Experience and Performance test as well as Usability test. In addition, the tool its self can be the test-bed for other features, such as, the opportunity to merge two ellipses performing a logic AND operation between two properties of the dataset.