GESSI Pizza Day

The GESSI research group of the Universitat Politècnica de Catalunya (UPC) conducts research in many fields of software engineering, with particular emphasis on:

- Smart Cities and Service Oriented Computing
- Requirements Engineering
- Software Architectures
- Open Source Software
- Empirical Software Engineering

Currently, we have several works to carry on (see a description below). If you are interested to collaborate in them, just contact us. Also, feel free to propose us any other topic you may be interested in.

Smart Cities and Service Oriented Computing

Title: Developing a planning services app for smart cities

Description: The number, diversity and complexity of software services and apps (SSA) that smart cities offer to citizens is dramatically increasing. Regardless of this trend, smart cities still work in isolation and therefore the SSA that they offer to citizens cannot be used anywhere else. As a result, smart cities end up offering a similar portfolio of SSA built from scratch, and citizens need to use the different SSA for the same need when they move from one city to another.

The aim of this project is to develop an app that provides planning services to Barcelona (using open data provided by the city). This app should be easily adaptable to other smart cities.

Contact: Cristina Gómez (cristina@essi.upc.edu)

T**itle: Developing a transport services app for smart cities**

Description: The number, diversity and complexity of software services and apps (SSA) that smart cities offer to citizens is dramatically increasing. Regardless of this trend, smart cities still work in isolation and therefore the SSA that they offer to citizens cannot be used anywhere else. As a result, smart cities end up offering a similar portfolio of SSA built from scratch, and citizens need to use the different SSA for the same need when they move from one city to another.

The aim of this project is to develop an app that provides services related to transports in the city of Barcelona (using open data provided by the city). This app should be easily adaptable to other smart cities.

Contact: Cristina Gómez (cristina@essi.upc.edu)

Title: City-noisity: A location-aware app for detecting noisy spots in a large urban conurbation

Description: Noise (from traffic, from bars, from urban works) is one of the topmost disadvantages of living in a big city. Having a continuously updated noise map would be certainly helpful for city managers as a way to plan future actuations that may support healthier life of citizens.

We are currently developing an app able to show a noise map but with some limitations: bound to a particular city (Barcelona), with a particular map service (Google maps) and not real-time (i.e., only providing statistics on historical data). The purpose of the project would be the creation of a new generation app, City-noisity, able to overcome these limitations. City-noisity should be easily installed in whatever city that complies with some data standards (to be investigated as part of the project), and would continuously update the noise map as new data from noise sensors arrive. How could we live without this app?

Contact: Xavier Franch (franch@essi.upc.edu)

Title: I-will-survive: A portable app for crowd-based emergency management

Description: Emergency situations in urban spaces occur from time to time, unfortunately. From natural disasters (floods, earthquakes, typhoons, etc.) to human-rooted causes (terrorist attacks, etc.), they required a timely response that may be crucial to save lives or at least mitigate damages. With smartphones at their hands, citizens may help in reacting both by communicating quickly what happens to emergency official dealers (police, fire department, etc.) and especially by broadcasting very quickly to other citizens the occurrence and evolution of these events.

This project proposes the construction of the I-will-survive app, an application for smartphones empowering citizens with an easy-to-use tool to communicate and learn about emergency as they happen. The app should provide a crowd-based reputation mechanism to allow preventing its abuse or misuse (by communicating fake or irrelevant emergencies) and should be designed in a way that it can be eventually (in later projects) adapted to other cities.

Contact: Xavier Franch (franch@essi.upc.edu)

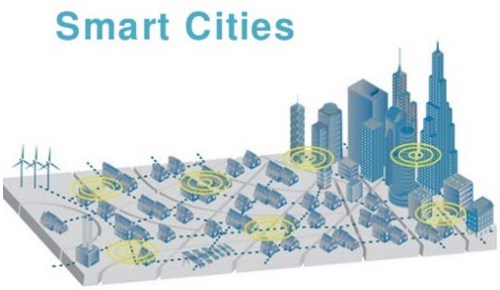
Title: Smart-City-Quality - Construction of a Quality Model for Smart City

Description: Smart City has as a goal to improve the quality of life of citizens. One way to achieve this goal is through the provision of apps and services that facilitate the citizens daily life. The potential and capabilities of modern ICT infrastructure is provoking that the number, diversity and complexity of software services and apps offered to the individual citizen is exponentially growing. In order to facilitate the citizens the selection of these services and apps it will be interesting to provide a quality model that defines the quality characteristics that can influence. Quality characteristics will include functional, non-functional and non-

technical aspects of apps and services. This project consists on constructing a quality model for Smart City services and apps. This is a project that does not consist on developing a tool, but on studying the Smart City software context and defining the quality characteristics in the quality model, along with the metrics that will allow to evaluate such characteristics of specific services.

Contact: Carme Quer (cquer@essi.upc.edu)

Title: Self-adaptive entities through smart monitoring of contexts

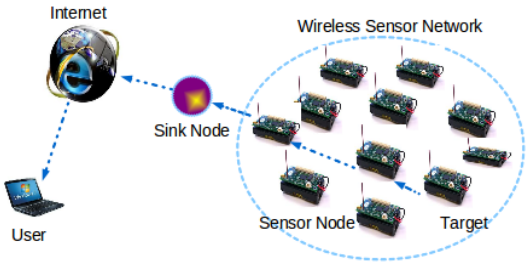


Description: *To d*evelop a monitoring and adaptation framework that through physical and logical monitors acquires the surrounding context of entities and triggering reasoning based on a context model to identify simple or complex context that could affect relevant entities also identified in an interaction process. Finally, based on the information retrieved, reasoned and analyzed the framework should be capable to identify or create an adaptation strategy that could be adopted for certain entities (e.g. adapting or changing their behavior or provide palliative options); *Use case scenario.* In a restaurant service if context information is not provided, the restaurant would typically show the menu of the day. Instead, if context information is provided, the restaurant could eventually provide different suggestions depending on who is walking by. If parents with children walk by, the restaurant would show the children's menu; if a couple is looking at it in the evening, it would show the menu for a candle light dinner; and if it is hot and sunny in the afternoon, the restaurant would advertise the selection of ice cream;

Technologies: Ontologies, sensors, smartphones, i-City infrastructure, open data, big data, open infrastructure, internet of things, pervasive and ubiquitous computing.

Contact: Oscar Cabrera (ocabrera@essi.upc.edu)

Title: A new paradigm for Web service selection from quality and context perspectives



Description: Nowadays a large percentage of services are provided on the Web commonly called web services (WS). Due the proliferation of this type of services it is relevant to provide a comprehensive selection process including the common compliance of non-functional requirements (assessed by analyzing the Quality of Service (QoS)) and compliance regarding context information surrounding potential users (e.g. environment, profiles, time, location, etc.); *Proposal.* Extend the architecture of an already

developed framework to rank services according to their QoS, to support context acquisition and reasoning capabilities (by using sensors and a context model respectively); *Result.* The extended version of the framework must be able to select services according to the context information surrounding potential users and stakeholders; *Resources.* The framework can be accessed in the following URL:

http://gessi-dev.lsi.upc.edu/wessqos/FindServQoS.htm

Technologies: Web services, ontologies, monitoring, context acquisition, context reasoning, open infrastructures, open data.

Contact: Oscar Cabrera (ocabrera@essi.upc.edu)

Title: A benchmark for comparing web adaptation technologies

Description: In service oriented computing, to achieve a given functionality (e.g., to buy products from an online store), some web services are composed in a given way to form a workflow of services whose overall quality fulfils the customer requirements. In the last years, we have witnessed the birth of several solutions to the following problem: given a workflow of services, how to detect and react to changes in the quality of service of these services? This is an important question, since any change (e.g., slower response time) may cause the violation of suh requirements. The purpose of this project is to build a benchmark that allows to compare these different existing (and upcoming) technologies to evaluate their possible adoption in real contexts.

Contact: Xavier Franch (franch@essi.upc.edu), Jordi Marco (jmarco@lsi.upc.edu)

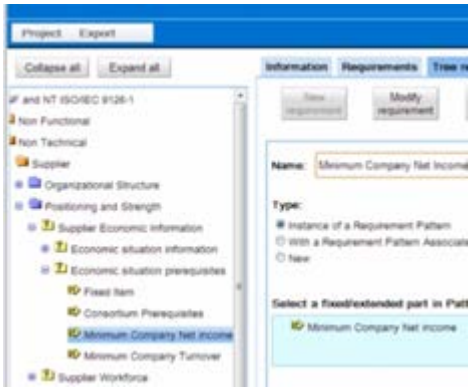
Title: Study of the available standards for smart cities

Description: There is an increasing number of standards for Smart Cities. For this project the plan is to design a map of standards for Smart Cities, the ones related to traffic, transportation, smart houses, health, etc. Once we have a clear map of the available standards the objective will be to perform an analysis and store the extracted information in a database. Finally, with all the information gathered, the final objective will be to share this information in an attractive web interface allowing to search by several criteria to the contents of the database.

Contact: David Ameller (dameller@essi.upc.edu)

Requirements Engineering

Title: Importing and Exporting Req-IF formatted requirements from PABRE-Proj



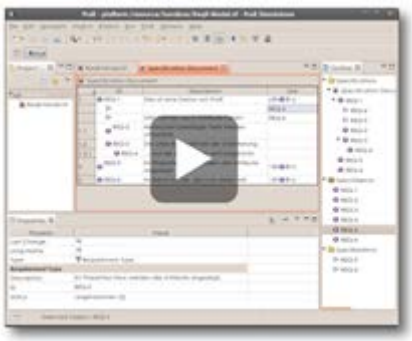
Description: PABRE-Proj is a tool that helps Requirement Analysts during the elicitation and specification of requirements. This tool is based in the use of Software Requirement Patterns (SRP), which are collected and stored in an SRP Database. ReqIF, which stands for Requirements Interchange Format, is a format for exchanging requirements, including their associated metadata and traceability information. It is an open, international standard by the Object Management Group (OMG). This work consists on studying the Req-IF standard, the current use of this standard in commercial Requirement Management Tools and implement the functionalities of importing and exporting requirements using this standard from PABRE-Proj.

Technologies: Google Web Toolkit (GWT), Java, Hibernate, Derby, MySQL

Title: Development of a Google Chrome App for PABRE

Description: PABRE is a system that helps Requirement Analysts during the elicitation and specification of requirements. PABRE system is based on the definition and use of Software Requirement Patterns (SRP), which are collected and stored in an SRP Database. This work consists on implementing a first version of the PABRE System as a Google Chrome App. These apps are like desktop software programs that people install on their computer . The main difference is that people use apps directly within their browser (https://developer.chrome.com/apps/about_apps).

Technologies: HTML5, JavaScript, CSS, Databases

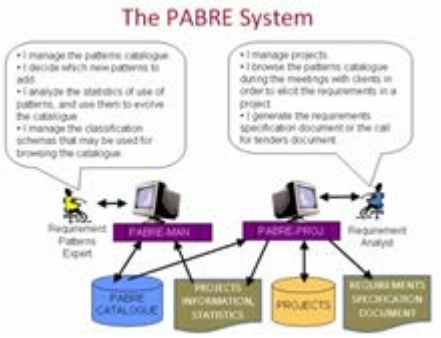


Title: Development of an Eclipse plug-in to allow the definition and use of PABRE SRP.

Description: The first step of this project is to study Eclipse, and the plug-in Requirements Management for Eclipse (https://www.eclipse.org/rmf/) to know the alternatives of extension of Eclipse to allow the definition of requirements from PABRE SRP. Once chosen the best alternative, the project will consists on implementing a first version of PABRE-SRP plug-in for Eclipse.

Technologies: Eclipse, Eclipse-RMT, Java, Derby

Title: Implementing order and constraints in applying PABRE SRP



Description: PABRE is a system that helps Requirement Analysts during the elicitation and specification of requirements. PABRE system is based on the definition and use of Software Requirement Patterns (SRP), which are collected and stored in an SRP Database. Currently it is not possible to define through PABRE an order among the SRP parts, neither constraints on their use in projects. This work consists on evolving the database schema for allowing the definition of this elements and also implementing the functionalities for their definition.

Technologies: Java, Hibernate, Derby, MySQL, JUnit.

Contact for Requirement Engineering Projects: Carme Quer (cquer@essi.upc.edu)

PABRE Web Page http://www.upc.edu/gessi/PABRE/index.html

Software Architecture

Title: A tool that implements REARM, a Reuse-Based Economic Model for Software Reference Architectures.

Description: The tool will support the evaluation of the economic impact of decisions in software reference architectures projects. The tool is expected to ease the communication among architects and management and implements a pragmatic preliminary economic model to perform cost-benefit analysis on the adoption of software reference architectures, for optimizing architectural decision-making. The tool is based on existing value-based metrics and economics-driven models used in other areas. More info: https://dl.dropboxusercontent.com/u/18332491/REARM%20tool.pdf

Technologies: To be chosen by the student. It needs to be integrated with http://www.sonarqube.org

Contact: Silverio Martínez (smartinez@essi.upc.edu)

Title: Analysis of requirements, quality attributes and architectural decisions of real software reference architectures in the software industry

Description: This survey is being done in a multinational IT consulting firm, in which their project managers, software architects and application developers have been interviewed about their daily work in software reference architecture projects. Software reference architectures allow reusing architectural knowledge and software components for the design of concrete software architectures of systems. Therefore, they are becoming widely adopted in the software industry. Nine software reference architectures, which are used by big

organizations (i.e., banks, public services, industry enterprises and insurances companies), are being studied. The work consists of processing face-to-face interviews and online questionnaires that have been already made to the involved stakeholders. This data analysis aims to understand and evaluate requirements elicitation, architecturally-significant quality attributes, architectural knowledge and decisions in software reference architecture projects. This study would deep on requirements and architectural decisions that software architects deal with daily. See an example of this type of work at: https://prezi.com/ptzmdex_r_tl/artifacts-of-software-reference-architectures/.

Technologies: Qualitative analysis programs (Microsoft Office, NVivo QSR, Atlas.ti)

Contact: Silverio Martínez (smartinez@essi.upc.edu)

T**itle: Acquiring and sharing architectural knowledge**

Description: This project has 4 parts: 1. Research: search in the web/books/etc information about architectural knowledge used when building software architectures (architectural styles, components, technologies, etc). 2. Formalization: unify the acquired knowledge using an already existing tool, ArchiTech (http://www.upc.edu/gessi/architech/index.html). This part includes learning to use the tool and the related terminology. 3. Managing: the objective of this part is to migrate the ArchiTech tool to a web interface. 4. Sharing: create a web based interface to show the architectural knowledge, and allow navigating through it.

Technologies: XMI, Web

Contact: David Ameller (dameller@essi.upc.edu)

Open Source Software

Managing Risk and Cost in Open Source Software Adoption (**RISCOSS**) is a European funded project finishing at Nov. 2015 (www.riscoss.eu). It is related to Open Source Software (OSS) adoption. OSS has become a strategic asset for a number of reasons, such as its short time-to-market software service and product delivery, reduced development and maintenance costs, and its customization capabilities. OSS technologies are currently embedded in almost all commercial software.

In spite of the increasing strategic importance of OSS technologies, IT companies and organizations face numerous difficulties and challenges when making the strategic move to integrate in their processes the open source way of working. This project's goals are: Strategic modelling and analysis of OSS-based ecosystems; Risk management of OSS projects; Business models and services for OSS solutions; Deployment of a software engineering platform for supporting decision-making.

The following specific projects for students have been defined in the context of the RISCOSS European project.

Title: Modelling and applying OSS adoption strategies

Description: Open Source Software (OSS) has become a driver for business in various sectors, estimates exist that in 2016, a 95% of all commercial software packages will include OSS components. OSS adoption impacts in fact far beyond technology, because it requires a change

in the organizational culture and reshaping IT decision-makers mindset. The way in which OSS affects and shapes business models is becoming object of increasing attention, and as a result, several OSS business strategies have been identified so far.

In order to apply an OSS adoption strategy to an organization a business expert should analyze:

(a) Coverage of the Organizational model by the strategy: it is covered if every business goal of the organization is supported by the adoption strategy

(b) Coverage of the OSS adoption strategy requirements: it is covered if the organization fulfills all the requirements of the strategy

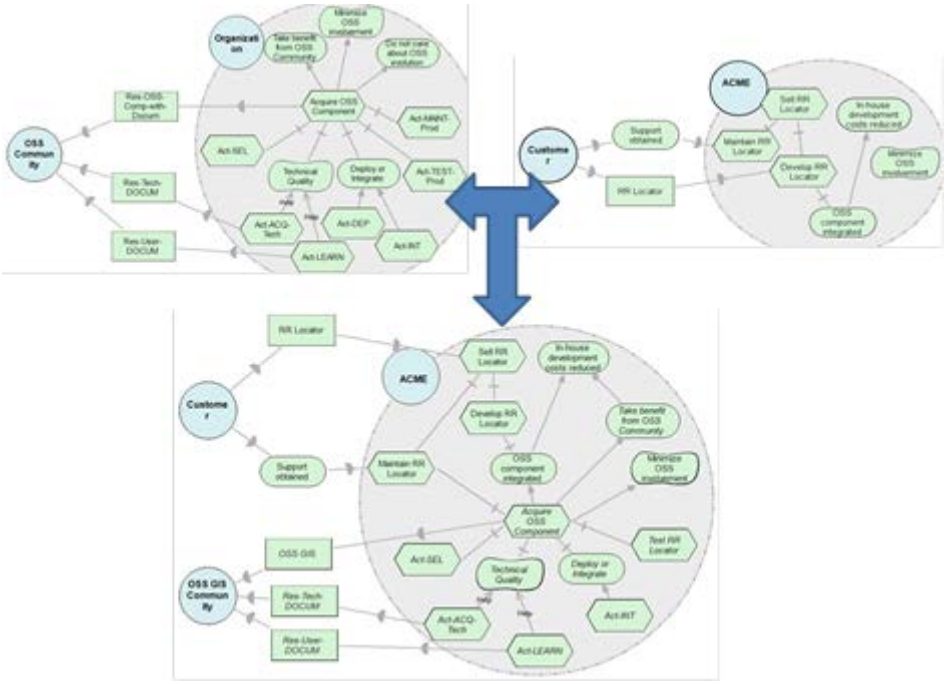
The project has the goal of developing a support tool providing the following functionalities:

1- Embed knowledge of several predefined OSS adoption strategies

2- Receive as input an organizational model of a company

3- Support business expert to perform the analysis of coverages (a) and (b) in order to decide which OSS adoption strategy is more suited to the company

4-Support business expert to integrate the OSS adoption strategy model and the organizational model



Technologies: Java, Eclipse

Contact: Lidia Lopez (llopez@essi.upc.edu) and Dolors Costal (dolors@essi.upc.edu)

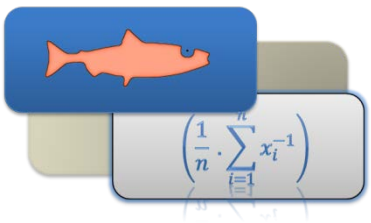
Title: Refactoring of the ccistarml library

Description: *i*StarML is an XML compliant format to represent *i** diagrams. Different methodologies have been created based on *i** concepts and modelling techniques. In particular the *i** framework has been exploited in different areas such as organizational modelling, business process reengineering and requirements engineering. Moreover, several proposals have been made to incorporate *i** modelling concepts to deal with software systems requirements representation and design. In RISCOSS project, i* is used to model OSS ecosystems, so we need a complete library to manage i* diagrams.

Technologies: Java, Eclipse

Contact: Lidia Lopez (llopez@essi.upc.edu) and Dolors Costal (dolors@essi.upc.edu)

Title: Monitoring Open Source Software Health



Description: Open Source Software (OSS) development is characterized by a distributed and collaborative work by OSS communities. These communities interact and cooperate within a scope known as OSS ecosystem. The health of OSS ecosystems has a significant impact on the quality of the OSS products. The goal of this project is to develop a framework to assess the OSS ecosystem's health. To attain this goal the applicant will: (1) investigate on the characteristics that define the OSS ecosystem's health and approaches to decompose these characteristics into finer-granular measurable metrics; and (2) develop a platform to automatically monitor and evaluate the characteristics of OSS ecosystems using concepts and techniques derived from Service Oriented Computing (e.g. Service Level Agreements for OSS). The platform will extend SALMonADA, a framework developed by the GESSI research group in conjunction with Universidad de Sevilla to assess Service Level Agreements for web services. (http://gessi.lsi.upc.edu/salmon).

Technologies: Eclipse, Java, web services.

Contact: Marc Oriol (moriol@essi.upc.edu)

Title: Open source software ecosystem quality indicators modelling

Description: The traditional view of software product development is that a software product is managed entirely within a company. This view is challenged by one recent developments. It is the transition from software product to software ecosystems (SECO), The name is obviously deduced from the archetype natural ecosystem. This transition takes place when a product line company makes its platform available to developers outside the company or when the company want to include open source software components in its software products. A SECO includes internal developers, strategic partners with long-term relationships, undirected external developers, and independent solution providers. Software ecosystems, It is clear that the term was coined to reflect the organization of software vendors, third-party developers, suppliers, and users. The use of the term software ecosystem in the open source domain is defined as a collection of software projects which are developed and evolve together in the same environment (e.g. Eclipse, Linux, Android, etc.) Software ecosystems represent a new software development approach that has two basic principles: transparency, a pillar in open source development, and modular system design. Nowadays we need to more systematically and formally SECO models in order to help in the reasoning about the quality of the relationships between SECO actors. The goals of this project are: (1) To conduct a literature study about software ecosystem modeling. (2) To develop an application that extend the GESSI Tools: iStarML (Is a xml-based format for representing i* models) and the jUCMNav Eclipse Plugin (iStarML Import/Export plugin for jUCMNav) for modelling OSS ecosystem relationships quality.

Conceptual Technologies : i*, XML, systematic literature review, software ecosystem.

Technical technologies: Java, Eclipse

Contact: Oscar Hernan (ohernan@essi.upc.edu)

Title: Mining OSS-ecosystem communication channels.

Description: In an open source software ecosystem a good communications is an essential success factor. A better understanding of how members of an open source community

interact, communicate and collaborate allow us to understand the OSS-ecosystem model. An OSS community use different communication channels like : mailing list, wikis, social media sites, web pages, etc. The goal of this project is developed a software component for identify the OSS-ecosystem actors and the relationships between them using: natural language processing, text mining and social network analysis (SNA) techniques.

Conceptual Technologies: Natural language processing, text mining and social network analysis.

Technical technologies: Java, Eclipse, R, Genie.

Contact: Oscar Hernan (ohernan@essi.upc.edu)

Empirical Software Engineering

Software engineering research needs to be performed in an experimental context that allows to observe and experiment with the technologies in use, understand their weaknesses and strengths, tailor the technologies for the goals and characteristics of particular projects, and package them together with empirically gained experience to enhance their reuse potential in future projects. With this goal in mind, we are fostering the application of empirical approaches as a transversal topic in our research. Thus, all our research lines are developed under the umbrella of the empirical software engineering paradigm. We have applied several empirical strategies such as Systematic Literature Reviews, case studies, surveys, experiments and action-research approaches.

We currently have the following projects:

Title: Tool-support for gathering publications in Systematic Literature Reviews (SLR)

Description: To conduct any type of literature study in an accurate and objective manner, it is necessary to use a precise and rigorous methodology. For such a purpose, Kitchenham and Charters proposed a list of guidelines in [1]. These guidelines have been derived from other existing studies used by medical researchers and adapted to reflect the specific problems of software engineering research. Since their inception, these guidelines have been widely used by software engineering researchers and when applied properly, they drastically reduce the risk of bias and incompleteness in the review results. The goal of this project is to implement a tool to automate the search of publications in a Systematic Literature Review. To this aim, the tool will automate several activities, i.e. search on different databases and individual journals/conferences, integration of the results, deletion of repeated entries, etc.

Technologies: Eclipse, Java, web.

Contact: Marc Oriol (moriol@essi.upc.edu)

Title: SOA for a collaborative tool supporting Systematic Literature Reviews (SLR)

Description: To conduct any type of literature study in an accurate and objective manner, it is necessary to use a precise and rigorous methodology. For such a purpose, Kitchenham and Charters proposed a list of guidelines in [1]. These guidelines have been derived from other existing studies used by medical researchers and adapted to reflect the specific problems of software engineering research. Since their inception, these guidelines have been widely used

by software engineering researchers and when applied properly, they drastically reduce the risk of bias and incompleteness in the review results. The aim of this project is automating the management of the steps defined in the process corresponding to the filtering of the publications to be included in the SLR.

Technologies: Java, Eclipse, webservices, mysql

Contact: Lidia Lopez (llopez@essi.upc.edu)

[1] B. Kitchenham and S. Charters, “Guidelines for performing systematic literature reviews in software engineering, Version 2.3,” 2007.