Department of Software Engineering
Building 70-1690
Phone: (585) 475-5461

Senior Project Proposal

UNDERGRADUATE

Project Title:		An Open Source, Open Architecture Collaborative Environment for Engineering Design and Product Development
Organization:		Mechanical Engineering, RIT
Contact:		Dr. Edward Hensel, PE, Professor and Department Head
	Email:	echeme@rit.edu
	Phone:	585-475-7684 (Direct Line)

Background Information

Dr. Hensel's interest is in set based concurrent engineering, and using the principles of lean systems to develop an open source, open architecture, internet based engineering design environment to allow teams of engineers and managers from a various disciplines and geographical locations to design, develop, and launch new products. This design environment consists of two primary aspects. The first is a document control and information flow management system, called EDGE. EDGE deals primarily with managing groups and information with a design group. EDGE is built around several open-source packages, and provides an access-controlled SVN project repository for every project team along with a basic user interface. EDGE is in beta testing, and is used by hundreds of people annually. The second aspect is the underlying relational database system, called FACETS, which will incorporate a wide range of design tools common to many industries. The tools associated with FACETS are currently being used only by a small research group, and are not yet ready for beta testing in EDGE. Our goal is to move the FACETS toolkits to a beta testing stage. This student team will provide incremental capabilities to EDGE, but the primary focus of this senior design project will be on the underlying FACETS system. The distinctions between EDGE and FACETS will be discussed with the SE design team early in the project.

Project Description

The FACETS system should be transparent to users. Engineers on a product development team should be able to use tools that they are familiar with, and to view the design process in a manner that they are familiar with, without forcing the engineer to learn someone else's toolkit, or someone else's preferred methodology. FACETS is focused on how knowledge should propagate between subsystems of the same product, across technology fields sharing common issues across product lines, and managing institutional knowledge from one generation of a product release to the next.

Engineering analysis lies at the heart of what engineers do every day. Most engineers resonate with the idea of a problem solving method, and agree that some method is needed, even if their own method has evolved over time to fit the needs of their career. It is with this common framework that we build the FACETS method. FACETS is used to generically describe any product development process that an engineer may use, or that may be prescribed for use by a company. We think of FACETS as becoming the "universal translator" between any product development system used by a particular company. The entire FACETS product development process is built around **first,** the concept that engineers are

fundamentally experts at solving problems, and **second,** that there are twelve broad categories of problems that must be considered in bringing a new product to the market place. The formal problem solving method underlying FACETs is a six stage process: F - Formulate, A – Assume, C – Chart, E – Execute, T – Test, s - the plural. The FACETS bring together information common to any new product development, regardless of the process used to generate the knowledge and information. The multi-faceted approach to product development allows us to perform activities on more than one facet at a time, but it also leads us in the direction of where the primary focus of the team should be at various stages in the process. We can think of each facet as being a toolbox, containing a number of tools. However, most of the tools in the toolbox have a common interface regarding how they are used – the FACETS interface. The twelve FACETS used to translate between the wide variety of design processes are:

Early Design FACETS	**Late Design FACETS**
Needs Assessment	Engineering Models
Concept Development	DFx: Detailed Design
Feasibility Assessment	Production Planning and Tooling Design
Engineering Analysis	Pilot Production
Tradeoff Analysis	Transition to Commercial Production
Preliminary Design Synthesis	Product Stewardship

The interface to the FACETS design environment is called EDGE – The Engineering Design Guide and Environment. EDGE is organized around projects. Each project may represent a design team, product development effort, subsystem design team, or an exploratory team investigating a new design concept. The system is intended to support teams of as few as three users, with no fundamental limit on the size of membership. Information may be shared and cross-linked between projects, within administrator imposed security constraints.

Technical Constraints & Assumptions

There has been a significant investment in the EDGE environment, and I prefer that the students build on top of the EDGE environment, rather than starting from scratch. On the other hand, prior work on FACETS has been only prototypical, and the team should consider previous FACETS only as illustrative of the tasks to be completed. The FACETS code should be built from scratch, but preferably as an extension of the EDGE environment.

Project Scope

EDGE

Improvements to EDGE are undoubtedly necessary, and we need to work on them to have a successful product. I will provide you with a laundry list of EDGE improvements – tackle what you can, and leave the rest for the next project. If we improve the EDGE interface, then we are more likely to have a long-term viable open-source product.

FACETS

The Early Design FACETS are to be demonstrated in this project. An empty toolbox will be established for all twelve FACETS, but at least two tools will be implemented within each toolbox for these Early Design FACETS. The user interface for each tool in each toolbox will be built around the FACETS problem solving process, and data will be stored in a relational database as an output from each tool. The team will demonstrate how User A and User B can both conduct Needs

Assessment, each using their favorite tool. However, regardless of which tool User A employs, s/he will be able to access, understand, and modify the Needs Assessment information provided by User B, seamlessly. Furthermore, User A and User B may use any number of tools within each toolbox. When data flows from one FACET (*e.g.* Needs Assessment) to another FACET (*e.g.* Concept Development) the system will provide data consistency and integrity checking. This goes beyond simply the right data types, etc, but actually requires that the software properly flows the right kind of information between FACETs. The system must "know" when a modification in one FACET requires the design engineer to reassess an outcome or decision in another FACET, and all of these changes have to be done under version control and engineering revision (change order) control.

During 2008-2 and 2008-3, a previous SE FACETS team developed a Needs Assessment Toolbox, and populated it with an Affinity Diagramming tool and an Objective Tree Tool.

During 2009-2 and 2009-3, the next SE FACETS team is asked to develop a Function Tree Tool for addition to the Needs Assessment Toolbox, and to develop a House of Quality (House 1) Tool for addition to the Needs Assessment Toolbox. The Function Tree Tool should be a straightforward extension to the previous Objective Tree Tool, and represents a good, relatively easy, starting point for new work. The House of Quality Tool will require substantially more development effort.

Department of Software Engineering Required Deliverables

1. Project website holding all work products and project artifacts maintained in the project account on the se.rit.edu web server.
2. Project plan, schedule and process methodology definition prepared by the end of week 3 of the first term.
3. Tracking report for at least two product/process metrics appropriate to the project and development methodology. Tracking reports updated on the project website at least every two weeks.
4. Interim status and final project presentations
5. Project poster and presentation at "SE Senior Project Day"
6. Project technical report

Sponsor and Project Specific Deliverables

1. The team should export their entire work product to an SVN repository on a KGCOE server. This export should occur at least once at the conclusion of each quarter.
2. All UML models, etc., developed during the course of the project are just as critical as the end-product. The sponsor requests scanned copies of student notes and work in progress, such as logbooks.
3. All work completed under this project is expected to be released as an open-source, open-architecture software package. We anticipate that copyright will be retained by RIT, solely for the purposed of insuring compliance with open source philosophy.
4. The revised system (EDGE 2.0, FACETS1.1) can be released to a pilot group of SD students at the beginning of second term. There must be a convenient means for porting all extant data to the new system.