

# **An Evaluation Framework to Drive Future Evolution of a Research Prototype**

*A developer-focussed evaluation technique*

David Nutter, Cornelia Boldyreff & Stephen Rank

<http://www.lincoln.ac.uk/faculties/computing/index.html>

# Introduction

OSCAR was developed to store artefacts produced by other components of the GENESIS environment and developers working within it. GENESIS was evaluated by the consortium's industrial partners. This evaluation was not comprehensive for OSCAR.

- Different evaluation requirements
- OSCAR “hidden” within GENESIS
- Limited checklist based evaluation. OSCAR has many requirements not explicitly mentioned by partners  
For example, one checklist item was “Users can define processes with nesting”. Very high level, and no mention of OSCAR

# Evaluation Purpose

The purpose of this developer focussed evaluation are somewhat different from the user-focussed evaluations.

- Establish a baseline against which modifications may be compared
- Confirm suitability as a development environment for software engineers
- Demonstrate improved quality and maintainability attributes
- Possible for individual developers to perform an ongoing evaluation without large user samples.

# Industrial Partner Results

The industrial partners were pleased with the GENESIS system as it matched their stated requirements in all but a few points. The aspects evaluated included:

**Process Management Requirements** Except for the lack of Risk Management, this fulfilled the user requirements

**Resource Management Requirements** Satisfied the user requirements

**Accessibility Requirements** Allows remote and local working, thus satisfies the user requirements

**Metrics Requirements** No user expressed a requirement for this functionality, thus, like OSCAR it was never evaluated.

# Evaluation Design

The evaluation covers the following aspects, focussing on assessment by (semi-)automated techniques (metrics, best practice etc.)

**Quality** Important for future modifications

**Maintainability** Documentation, test-cases etc promotes developer comprehension and thus allows easy perfective maintenance. Related to quality.

**Usability** Focus on the stand-alone client

**Scalability** An aspect of usability. Demonstrate that OSCAR is ready for large-scale operational evaluation.

**Novel Features** OSCAR is a research system. It is necessary to demonstrate that it offers functionality not available in other systems.

# Aspects(1) - Quality

The quantifiable quality aspects of the system:

**Comprehension** Assessed with metrics. The more problems, the harder the system will be to maintain.

**Existing problems** Defect rates, missing requirements etc.

**Process** Ownership issues, defect rate, outdated documentation etc.

**Usability** Visual clutter, inconsistency etc.

Scored from zero. No scale, closest to zero the better.

# Aspects(2) - Maintainability

*Emphasis* indicates a qualitative attribute.

- Defect density
- Complexity
- Call depth
- Modularity
- Module size & function duplication
- *Ownership issues*
- Test cases and documentation

# Aspects(3) - User Interface

An extension of the qualitative evaluation undertaken during GENESIS

- Consistency
- Clarity
- Succinctness (clicks-per-op)
- *Extensibility*
- *Visual clutter*
- Component nesting depth
- *Integration with other applications*



# The GQM Model

Some details about the OSCAR GQM model:

- Focusses on improvement, instead of measurement like standard GQM.
- Extendable later to encompass any operational evaluation
- Model output used for planning maintenance etc
- Another level of review that can find potential problems

The model is relatively small:

- 5 goals
- 17 questions

# Future Work

Due to space constraints this paper did not contain a results set. In future:

- A “typical” results set will be published
- OSCAR developers will perform this evaluation regularly to assess their performance
- The model will be refined as new goals are elicited.

Additionally an operational evaluation involving users will be conducted:

- Users are necessary at some point!
- Actually check that our synthetic model reflects developer reality
- Check for changing requirements

# Conclusions

- Extensive evaluation of OSCAR was not performed within GENESIS
- Ongoing evaluation is necessary for successful evolution
- However, this is resource intensive!

The proposed evaluation is a lightweight process that can be conducted by developers alone:

- Check for recurring problems
- Less overhead than traditional evaluation techniques, but less thorough.

Consequently, a user focussed evaluation is necessary at some stage in development.