DEMONSTRATION GUIDE

USC Center for Software Engineering
Department of Computer Science
Annual Research Review
February 8-11, 2000
# Demonstration Schedule

**Tuesday, February 9, 1999**
**Time: 12:00pm-2:00pm**

## LIST OF DEMONSTRATIONS

<table>
<thead>
<tr>
<th>Location</th>
<th>Demonstrator</th>
<th>Title</th>
<th>Machine</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAL 327</td>
<td>Alex Egyed</td>
<td>Architect’s Automated Assistant (AAA)</td>
<td></td>
</tr>
<tr>
<td>CSE Lab</td>
<td>Yu Ting Catherine Kao (?)</td>
<td>COCOMO Spreadsheet Tool</td>
<td></td>
</tr>
<tr>
<td>SAL 218</td>
<td>Jongmoon Balk</td>
<td>COCOMO II.1999.0 (COnstructive COst Model)</td>
<td></td>
</tr>
<tr>
<td>SAL 337</td>
<td>Cyrus Fakharzadeh</td>
<td>CORADMO</td>
<td></td>
</tr>
<tr>
<td>SAL 339 (?)</td>
<td>Jung-Won Park</td>
<td>Distributed Collaboration Priorities Tool (DCPT)</td>
<td></td>
</tr>
<tr>
<td>SAL 336</td>
<td>Neno Medvidovic</td>
<td>DRADEL</td>
<td></td>
</tr>
<tr>
<td>CSE Lab</td>
<td>Nikunj Mehta</td>
<td>Hispanic Digital Archive</td>
<td></td>
</tr>
<tr>
<td>SAL 320</td>
<td>JuneSup Lee</td>
<td>IRPC Tool (Incremental RPC Protocol Generator)</td>
<td></td>
</tr>
<tr>
<td>Barry's/Winsor's office/Lab (?)</td>
<td>Hoh In</td>
<td>S-COST II (Software Cost Option Strategy Tool II)</td>
<td></td>
</tr>
<tr>
<td>CSE Lab</td>
<td>Team:</td>
<td>Using WinWin to Get Agreement on Software Requirements</td>
<td></td>
</tr>
<tr>
<td>CSE Lab</td>
<td>Rajnish Lal</td>
<td>World War I Book Collection - Access Enhancement</td>
<td></td>
</tr>
</tbody>
</table>

(Please see attached floor plans for demonstration locations.)
Name: COCOMOII.2000.0 (CONstructive COst Model)

Objective: To develop software cost and schedule estimation model tuned to the life cycle practices of the 1990's and 2000's; to develop software cost database and tool support capabilities for continuous model improvement; to provide a quantitative analytic framework, and set of tools and techniques for evaluating the effects of software technology improvements on software life cycle costs and schedule.

Rationale: Tailorability of an organization process to its own process drivers; consistency of the granularity of the cost estimation with the granularity of available information; provide range estimates to the degree of definition of the estimation inputs.

Target Users: Customers, Managers, System Engineers, Software Engineers, Cost Analysts

Scope: Generation of Effort and Schedule Estimates; Calibration; Risk Assessment

Project Type: Multi-year USC-CSE research project

Developers: Principal Investigator: Dr. Ellis Horowitz; Student Programmers and Testers: Jongmoon Baik

Runs On: Windows95/98/NT

IPR Status: COCOMOII copyright owned by USC-CSE. Affiliates free to use, modify, but not restrict other affiliates' use

Technical Approach: COCOMOII follows the openness principles used in the original COCOMO. Thus, all of its relationships and algorithms will be publicly available. Also, all of interfaces are designed to be public, well-defined, and parameterized, so that complementary preprocessors, post-processors, and higher level packages, can be combined straightforwardly with COCOMOII.

Future Directions:
Integrate COTS Integration cost model(COCOTS)
Publish COCOMOII book with USC COCOMOII CD in June, 2000
Name: Constructive Productivity Model - COPROMO

Objective: Provide strategic planning decision assistant for senior management.

Rationale: Existing parametric models, COCOMO-II and CORADMO, are used as evaluation tools for projected costs of development of a prototypical application from the domain of interest.

Target Users: Senior management and staff, planners and SEPGs.

Scope: A broad application of the COCOMO related models to assess impacts of driver changes into the future in order to make trade-offs of investment strategies in technology/process.

Project Type: multi-year USC-CSE research project, initially supported by and implemented for KRSA Life Cycle Evaluation, AFRL.

Developers: A. Winsor Brown and Dr. Barry Boehm.

Runs On: Excel Office 97 version; version 4(?) or higher upon request and at own risk (multiple worksheet; no macros).

IPR Status: Copyright USC-CSE.

Technical Approach: The model is based on the use of COCOMO II\(^1\) and CORADMO\(^2\) as valuation mechanisms. The implementation approach uses a representative application from the domain of concern to the senior management, and the identification of technology drivers and time frames. One version of the tool, a Technology Impact Analyzer, has been implemented and used in the evaluation the Knowledge Based Software Assistant.

The COPROMO implementation approach is to identify an application, time frames and specific technologies that are expected to impact productivity for the prototypical application over the time frames selected. The prototypical application should be one that is representative of the domain of concern of the senior management. The time frames should be long enough to have the selected technology mature and come into use, spanning at least eight to fifteen years. The specific technologies should be identifiable and have relatively clearly defined, even if still evolving, content. One of the technologies should be the commercial and milieu specific (e.g. DoD) technologies that will evolve independently of the specific technologies.

---

\(^1\) Constructive Cost Model, version II, 1998 calibration.

\(^2\) Constructive RAD-schedule Model, a currently uncalibrated extension to COCOMO II.
The parametric drivers of the valuation model include COCOMO II's effort scale factors and multipliers and which cover process, product, platform, personnel and project; and CORADMO's schedule and effort multipliers. Each of the drivers' values are then gathered for the current baseline or assessed into the future using engineering judgement based on the assumed impacts of the selected, specific technologies.

All of the information on the drivers, their evolution over time, and the rationales are then input into a spreadsheet tool. The tool, called a Technology Impact Analyzer, consists of multiple, parallel COCOMO II and CORADMO parametric model executions. The tool graphically displays each of the drivers' values over time to allow reasoning and discourse about their values and evolution over time. The tool also provides fields for the capture of the rationales for each of the drivers' values and evolution on the same page as the tabular and graphic display of values. Finally, the tool displays a comprehensive set of graphs showing the impact of the selected technologies over time for the issues of concern: effort, schedule and recommended head-count.

Future Directions: COPROMO 0.3 use and feedback. Generation of more example implementations to get more experience. Short range: re-packaging as an auto-adapting spreadsheet for variable years and kinds of improvements. Long range: possible extensions into investment cost projections/calculations and ROI.
Name: COPSEMO

Objective: The intent of the CoPSEMO.xls is to distribute the schedule (months, M) and adjusted effort (person-months, PM) based on user selected percentage allocations to the various MBASE/RUP phases. The results of the selected distributions are made available to other tools, and a person-power loading chart (per stage) is shown.

Rationale: Originally designed as a pre-processor for the COCOMO RAD model, it is an extension of the COCOMO II model, providing for the extrapolation of the COCOMO II schedule and effort to the phases it does not cover (Inception and Transition). Other COCOMO extensions, like COCOTs, also calculate schedule and/or effort

Target Users: Managers, Project Planners, and Software Engineers

Scope: COPSEMO.xls takes its data from the exported output of a COCOMOII.exe run via an intermediate tool, CoCoMoII.xls that also provides useful results itself

Project Type: Multi-year USC-CSE research project

Developer: A. Winsor Brown

Runs On: Microsoft Windows 95/98 and NT 4.0 with Excel 97

IPR Status: Copyright USC-CSE

Technical Approach: Using Excel control objects and links to other spreadsheets, users can supply directly or indirectly as much of the information that COPSEMO needs to do its job. All other areas of the worksheet, except those designated for user input, can not be modified by the user. Users do NOT need to be proficient with the use of Microsoft Excel.

Future Directions: Short range (expected in 2000), to permit the distribution of effort to the various activities that make up the MBASE/RUP development lifecycle model. Long range, to have a Windows/Java program that performs the same tasks as this spreadsheet with the added functionality of the USC-COCOMO.exe.
Name: CORADMO

Objective: The intent of the CoRADMo.xls is to calculate (predict) the schedule (months, M), personnel (P), and adjusted effort (person-months, PM) based on the distribution of effort and schedule to the various MBASE/RUP phases, and the impacts of selected schedule driver ratings on the M, P, and PM of each stage.

Rationale: The COCOMO RAD model is an extension of the COCOMO II model, which focuses on the cost and schedule impacts of developing software using rapid application development techniques. RAD is taken to mean an application of any of a number of techniques or strategies to reduce software development cycle time.

Target Users: Managers, Project Planners, and Software Engineers

Scope: CoRADMo.xls takes its data from the exported output of a COCOMOII.exe run via an intermediate tool, CoCoMolII.xls that also provides useful results itself.

Project Type: Multi-year USC-CSE research project

Developers: A. Winsor Brown and Cyrus Fakharzadeh

Runs On: Microsoft Windows 95/98 and NT 4.0 with Excel 97

IPR Status: Copyright USC-CSE.

Technical Approach: CORADMO relies on COPSEMO to do the distribution of effort and schedule to the MBASE/RUP phases and then applies the CORADMO drivers to adjust effort and schedule parametrically. Using Excel control objects and links to other spreadsheets, users supply the RAD driver rating levels which impact effort, schedule and, indirectly, personnel. All other areas of the worksheet, except those designated for user input, can not be modified by the user. Users need NOT be proficient with the use of Microsoft Excel.

This model is not yet calibrated but relies on expert judgement (through a wideband delphi) for its initial driver value to schedule and effort driver rating.

Future Directions: Short range: investigate new driver rating values based on data from a COCOMOII Affiliate. Long range: To eventually have a Windows/Java program that performs the same tasks as this spreadsheet with the added functionality of the USC-COCOMO.exe.
Name: UML/Analyzer

Objective: A system for defining and analyzing the conceptual integrity of UML models.

Rationale: Software development is about modeling a real problem, solving the model problem, and interpreting the model solution in the real world. In doing so, a major emphasis is placed on mismatch identification and reconciliation within and among system views (such as diagrams). UML/Analyzer describes and identifies causes of architectural and design mismatches across UML views as well as outside views represented in UML (e.g., C2 style architectures).

Target Users: The typical users are software designer and architects (for constructing and analyzing development models) and reusers (for generating model abstractions out of reverse engineered source code).

Scope: UML/Analyzer provides architecture modeling, analysis, implementation, and evolution support at system specification time for both forward and reverse engineering.

Project Type: The project was supported by DARPA’s EDCS, AFRL/IFTD, and USC-CSE Affiliates.

Developers: Primary developer is Alexander Egyed. The tool is integrated with Rational Rose developed by Rational Corporation and we reused code from the Argo/UML project developed by the University of California, Irvine.

Runs On: Java

IPR Status: Copyright owned by the University of Southern California.

Technical Approach: UML/Analyzer supports the definition of mismatch rules and model constraints. It also defines what information can be exchanged and how it can be exchanged. With that, architects can identify and resolve inconsistencies between views automatically. The fundamental principles underneath UML/Analyzer are: 1) Mapping which identifies related pieces of information and thereby describes what information is overlapping and can be exchanged; 2) Transformation which extracts and manipulates model elements of views in such a manner that they can be interpreted and used by other views (how to exchange information); and 3) Differentiation which traverses the model to identify (potential) mismatches within its elements. Mismatch identification rules can frequently be complemented by mismatch resolution rules.

Future Directions: Stronger automated support for all three activities (Mapping, Transformation, and Differentiation) outlined above. Further integration with Medvidovic’s SAAGE environment to enable continuous round-trip engineering between UML and ADLs (and C2 in particular)
Name: EasyWinWin Collaborative Requirements Negotiation

Objective: EasyWinWin is an innovative new implementation of the WinWin approach that aims at enhancing the directness, extent, and frequency of stakeholder interaction.

Rationale: EasyWinWin is based on the COTS groupware product GroupSystems and combines group productivity tools (electronic brainstorming, categorizing, voting, etc.) to improve the involvement of stakeholders and facilitate interactions in the Requirements Engineering process.

Target Users: Initially all participants in the front end of the software process: system engineers, representatives of users, customers, developers, maintainers.

Scope: EasyWinWin supports brainstorming of WinConditions, categorization and prioritization of requirements, development and refinement of domain taxonomies, shared definition of terms, negotiations and conflict resolution following the WinWin negotiation model, and the joint development of use cases. An interface to the Rational Rose™ CASE tool is provided to support repository-based integration of negotiation results allowing further analyses and reports.

Project Type: This work is supported by the Austrian Science Fund (Erwin Schrödinger Grant 1999/J 1764 "Collaborative Requirements Negotiation Aids") and DARPA and Air Force Research Labs under Contract F30602-1994-C-0195.

Developers: Primary developer is Paul Grünbacher. The collaborative reasoning tools have been developed by GROUPSYSTEMS.COM.

Runs On: Clients run on Windows 95, Windows 98, Windows NT 4.0 or in a Browser (Internet Explorer 4.01 SP2, Netscape 4.5). The server is either provided by an ASP or installed on the organization's infrastructure (Windows NT 4.0).

IPR Status: WinWin copyright owned by USC-CSE, GroupSystems is a Trademark by GROUPSYSTEMS.COM

Technical Approach: GroupSystems provides a set of group support tools facilitating collaborative activities like idea generation, idea categorization and prioritization: Electronic Brainstorming, Categorizer, Alternative Analysis, Topic Commenter, GroupOutliner. The EasyWinWin Integrator links GroupSystems negotiations and Rational Rose UML artifacts using XML and COM.

Future Directions: EasyWinWin Process Guide (will be published by GroupSystems.com); EasyWinWin WebViewer

WWW: http://sunset.usc.edu/EasyWinWin/ and http://www.groupsystems.com
Name: Refining COCOMO II Drivers

Objective: To reduce the subjectivity in selecting a rating for COCOMO Scale and Cost Drivers.

Rationale: A common criticism of the COCOMO model is the subjectivity with which some of the Driver ratings are selected. In addition to size, COCOMO II uses five Scale Drivers and seventeen Cost Drivers as inputs. This work attempts to refine some of the Driver definitions to reduce or eliminate subjectivity.

Target Users: Customers, Managers, System Engineers, Software Engineers, Cost Analysts

Scope: Generation of additional Driver rating selection criteria that is consistent with the current model.

Project Type: Six month USC-CSE Affiliate research project

Developers: Dr. Brad Clark will lead the effort with participation of USC-CSE Affiliates and COCOMO tool vendors.

IPR Status: COCOMOII copyright owned by USC-CSE. Affiliates free to use, modify, but not restrict other affiliates’ use

Technical Approach: The approach is to draw on the knowledge and experience of Affiliates in developing selection criteria that is pertinent and useful. The investigation will proceed using email and draft documents. The results will be periodically reviewed at COCOMO Research meetings. The steps in this investigation are:

1. Group the Drivers in subjectivity categories.
2. Develop a Whitebox analysis that considers a driver’s influence on effort by lifecycle phase and activity.
3. Develop selection criteria based on the Whitebox analysis.
4. Pilot the enhanced selection criteria.

Future Directions:

The results of this investigation will be presented at the Fall 2000 COCOMO/Software Cost Modeling Conference.