

traffic destined to them is a priori suspicious. The sensor spoofs services, rather than merely recording attempted connections, to determine the intention behind the traffic.

Billy Goat is built atop a security-hardened Linux machine that offers no real services beyond very restricted login. It is configured in conjunction with the network on which it runs so that traffic directed toward address subnets that are not used is routed to the Billy Goat.

The Billy Goat itself offers a virtualization infrastructure that allows individual sensors to be written as if they were running on a single host. It also provides a logging infrastructure based on a relational database facilitating correlation and analysis of the copious data produced by the large number of virtual sensors.

The spoofed services include HTTP, Microsoft RPC (remote procedure call), Microsoft SQL (database), and SMB (file sharing and printing). Vulnerabilities in these services are commonly used as vectors for worm and virus propagation.

One very important requirement of the Billy Goat is that it continue to function in times of heavy worm activity. In particular, it must retain its utility even when the performance of the network is dramatically diminished or even completely unreliable. This requirement implies a distributed system of Billy Goats on the network. Each local Billy Goat serves to inform administrators of local infection so that it can be eliminated. In order to support the distributed architecture, the Billy Goat distribution contains an automatic update mecha-

nism. This mechanism ensures that a deployed Billy Goat has all the latest sensors and signatures.

It is worth contrasting the Billy Goat Box with another security tool called a Honey Pot. A Honey Pot, as its name suggests, offers "something desirable" to lure attacks to a particular machine. Unlike a Honey Pot, the Billy Goat does not advertise any services, and does not allow any of its services to be corrupted. The Billy Goat is designed to detect large-scale automated attacks rather than trying to deceive sentient human attackers.

Please contact:

James Riordan, Diego Zamboni
IBM Zurich Research Lab/SARIT
Tel.: +41 1 724 89 81
E-mail: rij@zurich.ibm.com

Decision Support for Personnel and Resources Management in Multi-Project Environments

by Antonia Bertolino, Eda Marchetti and Raffaella Mirandola

Decision making in project management requires accurate and automated support to monitor and control the development process. Managers must make reliable schedule predictions and optimize personnel utilization. They thus must be able to dynamically evaluate whether the resources assigned to a job are sufficient and whether the organization structure is adequate to meet the scheduled deadlines. To support managers in these tasks, scientists at ISTI-CNR are studying the potential of classical performance engineering and propose the Propean (Project Performance Analysis) methodology.

Propean (Figure 1) relies on the metaphor that project teams correspond to the processing resources of classical performance models, and project activities are the tasks to be performed within established time intervals.

According to this metaphor, well known techniques from performance analysis can be usefully adapted to managers' needs, for eg assessing the time for completion of specified tasks, handling personnel multitasking over concurrent projects, distributing the workloads in development cycles, deciding about product release, and similar issues.

However, a key aspect for the effective deployment of Propean is to adopt an interface that appears familiar to managers. Hence, we adopt as the input modelling notation the UML (Unified Modelling Language) Profile for Schedulability, Performance and Time (RT-UML) recently approved as an OMG standard. In this way, managers do not need to become knowledgeable of performance notations, but can develop a model of the flow of activities and of tasks distribution using the widespread UML notations and tools. The translation into a format that can be processed by performance analysis algorithms is then performed automatically.

The SPE Approach

The Propean technology is based on the Software Performance Engineering (SPE) approach, which is a systematic, quantitative approach to software system construction in order to meet performance objectives. Its basic concept is the separation of the Software Model (SM) from its environment or Machinery Model (MM).

The SM captures the essential aspects of software behavior; it can be represented by means of Execution Graphs, in which nodes are software workload components and edges represent transfers of control. Each node is weighted by use of a demand vector that represents the

resource usage of the node. The MM models the hardware platform and is based on Queueing Network (QN) models.

The Methodology

Starting from the above metaphor we capture the aspects relative to the activity planning in the SM, and those relative to people utilization and distribution in the MM. These models are derived from Sequence Diagrams (SD) and Deployment Diagrams (DD) respectively, and are automatically combined by Propean which generates a unique QN, that can be analysed by traditional performance analysis tools.

The methodology is currently being implemented. Figure 2 illustrates the tool architecture including: a UML case tool (specifically ARGO) which processes the input diagrams and generates an XML file; a component which takes as input the XML file and produces the corresponding QN; a QN analytical solver or simulator, which derives the performance analysis results; and finally, a results converter which analyses the performance results and converts them back into UML.

Here below, we outline the basic steps of the Propean methodology:

1. Analysis: definition of project activities (manager)
2. Modelling: definition of the SDs and the DD (manager)
3. Model annotation: specification of parameters and values (manager)
4. SPE models generation: derivation of the SM and MM models (automatic)

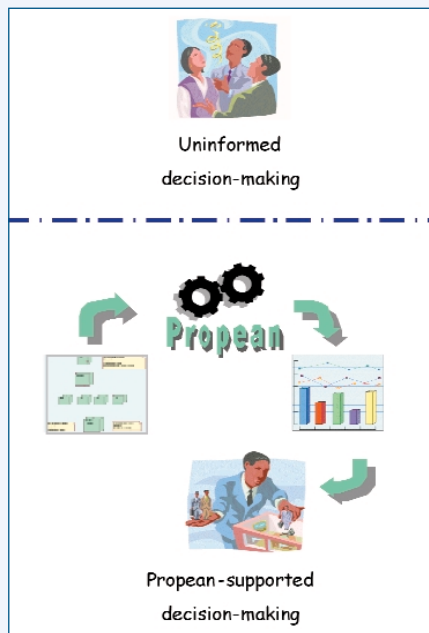


Figure 1: Different approaches to decision-making.

5. Model evaluation: resolution of the QN and derivation of the relevant predictions (automatic)
6. Analysis of results: evaluation of the results obtained (manager).

Applications

One of the advantages of using performance techniques for project management is their natural capability to handle multiple projects and their mutual involvement in schedule and resources usage. In fact, Propean results particularly useful in environments where multiple projects compete for common resources, which is for instance the case in most software development environments.

Propean is the result of a collaboration between ISTI-CNR in Pisa, the University of Rome Tor Vergata and Ericsson Lab Italy (ERI) in Rome. It has been applied with encouraging results to some industrial case studies. Here we mention two of them: one is the widespread Rational Unified Process (RUP) customised for a specific ERI project. Considering RUP itself as a product, Propean provided answers to the following questions: How long will RUP take to process a certain project? How will RUP utilize the available resources? How is the RUP schedule affected by the concurrent processing of several projects? The application of Propean to RUP was divided into two steps: the first, called RUP modelling, consisted of the description of the functionality and architecture of the RUP product by means of UML diagrams annotated according to the RT-UML profile. The second step, called RUP customisation, represented the core of Propean application and included the refinement of the UML diagrams according to personnel availability and process requirements.

In a second case study, Propean was used to support the decision to release a software product, after testing is completed with no trouble report left open. Propean can assist managers in the early prediction of the expected time until release, considering the actual resource availability (with personnel multitasking). Alternatively, if the release time is fixed with the customer, Propean can help decide the most adequate personnel configuration to fulfil the time constraints.

Propean can provide useful support to decision makers in project management. The tool can provide valuable input for analysis, but human expertise remains essential to model the processes, to tune them with the correct values and to interpret the results obtained.

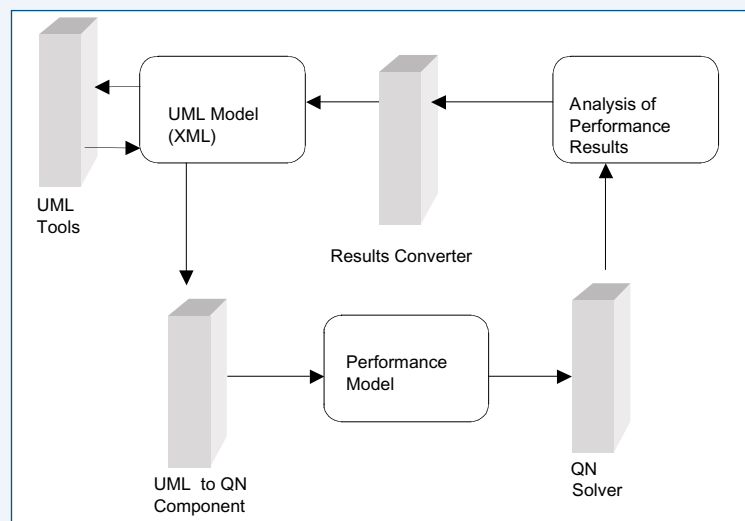


Figure 2: Propean Tool Architecture.

Links:
<http://www.isti.cnr.it/ResearchUnits/Labs/se-lab/index.html>
<http://www.omg.org/technology/documents/formal/schedulability.htm>
<http://www.perfeng.com/>

Please contact:
 Eda Marchetti, ISTI-CNR
 Tel: +39 050 315 3467
 E-mail: eda.marchetti@isti.cnr.it