Ensuring Reproducibility through Tool-Based Simulation Process Management

(Poster Abstract)

Konrad Jünemann, Jochen Dinger, Hannes Hartenstein
Institute of Telematics & Steinbuch Centre for Computing
Karlsruhe Institute of Technology, Germany
[konrad.juenemann | jochen.dinger | hannes.hartenstein]@kit.edu

ABSTRACT

For credibility of simulation results, reproducibility of simulation runs is a *must*. However, reproducibility requires a thorough management of *all* data involved in the simulation process. The corresponding management of data can be error-prone and time consuming if performed manually. In this paper we introduce a simulation management approach that ensures reproducibility and traceability of simulation runs as well as improves efficiency of simulation processes by automation of common simulation tasks. We implemented our approach as an Eclipse plugin. We show that information gained by explicit simulation management can be used to automatically organize and archive all necessary data to reproduce a simulation. While our tool was implemented with focus on the network simulator ns-2, our concepts can be applied to other simulation environments, too.

Categories and Subject Descriptors

I.6.7 [**Simulation and Modeling**]: Simulation Support Systems

General Terms

Management, Experimentation

Keywords

Simulation Management, Simulation Workflow, Eclipse

1. INTRODUCTION

The process of simulation consists of a diverse set of tasks that range from modeling to data analysis. Administrative tasks like archiving and organization of simulation data tend to be time consuming and error-prone. We perceive these data management tasks as part of a simulation management approach, whose objective is the efficient management of the whole simulation process. This paper presents an Eclipse-based simulation management solution called KEMP that integrates widely used tools like Subversion and SSH.

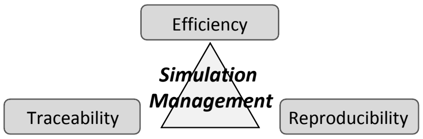

Figure 1: Key simulation management challenges

Three key challenges in the field of simulation management (see also [3]) are *reproducibility*, *traceability* and *efficiency*. These challenges also constitute our main goals and will be briefly reviewed in the following (see also Fig. 1).

Reproducibility. For credibility of results, simulation runs have to be reproducible by others. Thus ensuring reproducibility is a key requirement for simulation studies.

Traceability. To facilitate understanding of configurations and structure of older simulations is as important for simulation studies as is reproducibility. To achieve this goal of traceability, dedicated management techniques are required. They can be grouped into the aspects of organizing and documenting data as well as of providing a centralized library to archived simulation runs.

Efficiency. Automation and simplification of common simulation tasks enable users to spend more time on actual research, thus improving their overall efficiency. Also, collaboration between team members should be eased.

In [3] a general overview about simulation management tools is given. A few tools exist that support simulation management aspects. The OMNeT++ 4.0 IDE for instance provides a rich set of functionalities including file organization, but does not ensure reproducibility or support collaboration. In [1] the authors present an integrated script-based simulation tool-chain. While the tool of [1] also follows the approach of integration, it does not ensure reproducibility and is strictly bound to a specific simulator.

2. SIMULATION PROCESS WITH KEMP

Based on the workflow defined in [2], we modeled a simulation process w.r.t. the goals defined before. Figure 2 shows a simplified model of the process in Business Process Modeling Notation. In the following we will discuss each phase.

Create Project. At first a directory structure is created automatically to organize all involved data like configuration settings, documentation, results, etc. The semantics of this

SIMUTools 2010 March 15–19, Torremolinos, Malaga, Spain.

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2010.8714
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2010.8714

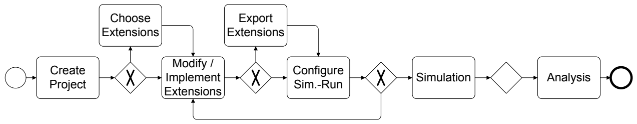

Figure 2: Simplified simulation process

structure are harnessed in most other phases. In particular, they enable KEMP (i) to differentiate between simulation runs and (ii) automatically detect and archive new, changed or deleted files. The latter prevents that the user breaks reproducibility by forgetting to archive files.

Choose Simulator Extensions. When the project has been created, the user selects a version of the desired simulator, including own or third party extensions like communication protocols. For this purpose, such extensions as well as the simulator itself are bundled to *modules* by KEMP. Modules consist of bundled files and a *descriptor* that contains meta information, including a description of the module's content and information about dependencies to other modules. Modules are managed by a so called *module library* that is accessible by a web-service interface. While the contained files are archived to a subversion repository, the descriptors are held within an internal database. The module library allows collaboratively managing and accessing modules within a team. Modules can be imported from the module library, leveraging the knowledge about the projects structure. Conflicting files will be detected automatically.

Modify / Implement / Export Extensions. The tool detects any modifications on files within the project. In this case, the user is given the opportunity to export the changes as a new module to the library. The module is automatically archived and exported to the module library.

Configure Settings. During this phase, the simulation runs are being configured. To avoid dependencies of specific simulators, our tool does not require the simulator to save its configuration in a specific format. Instead, files can be marked as configuration files. The configuration files are separated from the corresponding run's results and modules.

Simulation. After the configuration is complete, simulation runs can be executed. KEMP is able to transfer all necessary files via SSH to a remote machine, start the simulation and copy the generated results back. The automatic transfer allows the tool to make sure that the run is simulated exactly as configured: files will not be accidentally mixed up or forgotten. Also, the files will automatically be archived at the time of simulation, ensuring reproducibility.

Analysis. In this phase the user analyzes the results. As we focus on simulation management, no dedicated analysis tools are offered. However, organizational benefits like clear separation between configuration and results persist.

3. EVALUATION

Figure 3 depicts the GUI of KEMP. On the left, the structure of a simulation project is shown. Under the working directory that contains the user's current work in progress, a list of already simulated scenarios is shown. Every scenario contains the modules it is defined from and a list of simulation runs. Below the scenarios, an entry offers quick access to the module library. In the following, we show how KEMP addresses its three main goals.

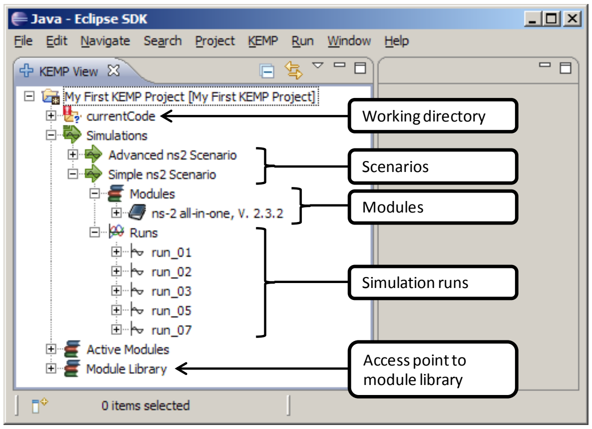

Figure 3: The graphical user interface of KEMP

Reproducibility. KEMP automatically archives all necessary files so that the simulation can be re-run later on in exactly the same configuration. This is possible because of three factors: (i) All files within the project folder are known to KEMP. (ii) When a simulation is started, all known files are synchronized with the remote machine. Hence, only known files are used for simulation. (iii) All known files are archived automatically at the beginning of a simulation.

Traceability. KEMP improves clarity by organizing all data involved in the simulation process in a hierarchical structure combined with automatic archiving. The traceability of simulation runs is further enhanced by the clear separation of modules, configuration and results.

Efficiency. To reduce the user's workload, several administrative tasks are automated. In particular, the tool automatically archives necessary files, creates the project structure, eases the retrieval of modules and handles simulation on remote machines. Furthermore, the introduction of a central module library simplifies reuse and organization of software modules. Through integration of different tools into one GUI, KEMP was also made comfortable to use.

4. CONCLUSIONS

This paper presented an integrated simulation management tool called KEMP. It ensures long time reproducibility of simulation runs, improves traceability by project organization and increases productivity by automation of common simulation tasks. Furthermore, it simplifies collaboration by bundling simulation extensions to modules and making them available to team members. Currently, we are evaluating KEMP in our group and working on the integration of cloud computing services.

5. REFERENCES

[1] T. Dreibholz, E. P. Rathgeb, and X. Zhou. Simproctc: the design and realization of a powerful tool-chain for omnet++ simulations. In *Simutools '09: Proc. of the 2nd Int'l. Conf. on Sim. Tools and Techniques*, 2009.

[2] A. M. Law. *Simulation Modeling and Analysis.* McGraw-Hill, 4th edition, 2007.

[3] L. F. Perrone, C. Cicconetti, G. Stea, and B. C. Ward. On the automation of computer network simulators. In *Simutools '09: Proc. of the 2nd Int'l. Conf. on Sim. Tools and Techniques*, 2009.

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2010.8714
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2010.8714