



**Applications in Practical High-End Computing - Group Project**

Assignment - "Workflow"

**Requirements**

Supervisor: Dr Stuart Barnes

Authors: Mateusz Gołąb

Csaba Kerti

Jakub Kiełbasa

Zsolt Kollarits

Course: CSTE / SETC

**Table of contents**

[1. Summary 3](#_Toc318139736)

[2. System's Dictionary 4](#_Toc318139737)

[a) Actors 4](#_Toc318139738)

[a) Other entities 4](#_Toc318139739)

[Access layer 4](#_Toc318139740)

[Control layer 5](#_Toc318139741)

[Calculation layer 5](#_Toc318139742)

[3. User Requirements 6](#_Toc318139743)

[a) Functional requirements 6](#_Toc318139744)

[b) Non-functional requirements 6](#_Toc318139745)

[4. System Requirements 7](#_Toc318139746)

[a) Use case diagram 7](#_Toc318139747)

[b) User's case list 8](#_Toc318139748)

[c) Scientist's case list 8](#_Toc318139749)

[d) Administrator's case list 8](#_Toc318139750)

[5. Appendix A: Used Tools 10](#_Toc318139751)

[a) Enterprise Architect 9.2 ver. Trial Ultimate 10](#_Toc318139752)

[b) Google Code 10](#_Toc318139753)

1. Summary

Task blabla

1. System's Dictionary

In order to avoid as many misunderstandings as possible, authors decided to create dictionary of most terms being used during project.

1. Actors

**User** - person using system.

**Scientist** - User which can run simulation through terminal.

**Administrator** - User responsible for configuration of the workflow.

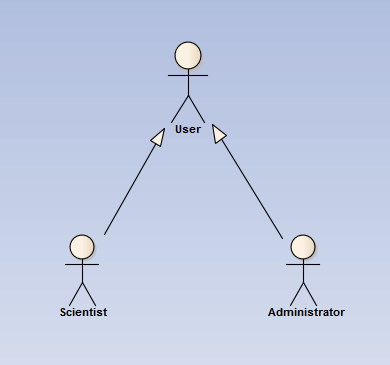


Diagram showing dependencies between actors.

1. Other entities

**System** - all resources and programs used in order to realize project aims. Organised, logically in three layers: access, controlling and calculating one.

### Access layer

**Client** - program which gives Scientist possibility to remotely run simulation or Administrator to change settings of Workflow.

**Terminal** - place where Client runs.

### Control layer

**Workflow** - main program controlling system performance. Starting all managers and therefore, indirectly modules.

**Workflow Server** - logical machine where Workflow runs.

**Workflow Sequence** - ordered, connected list of programs performing actual work of system. Sequence of steps necessary to finish task. Set up by Administrator.

**Workflow Manager** - program which executes and controls Workflow Sequence. Responsible for sending commands and parameters to particular Modules.

**Module** - single unit-program within Workflow Sequence added by Administrator.

**Recovery Manager** - program which performs all tasks in order to assure safe recovery in case of crush during execution of Workflow Manager.

### Calculation layer

**Calculation Server** - logical place where modules' calculations are being performed.

1. User Requirements
2. Functional requirements

FU1 User can add/remove arbitrary number of modules into workflow.

FU2 User can run simulation with uploaded parameters.

FU3 Recovery system: possibility of restarting workflow (from the last stable/good point) when system crushes.

* 1. FU3.1 Monitoring of errors: Users can see the exact location of failures.
  2. FU3.2 Flexible recovery policy (depending on expected time of execution we decide to store data before/after module or after each iteration)

FU4 Many users have possibility to connect to system simultaneously. But there is only one running program at time (users requests' go to queue - serial workflow).

1. Non-functional requirements

NF1 Reliability/Validation: take care of input/output formats of specific modules.

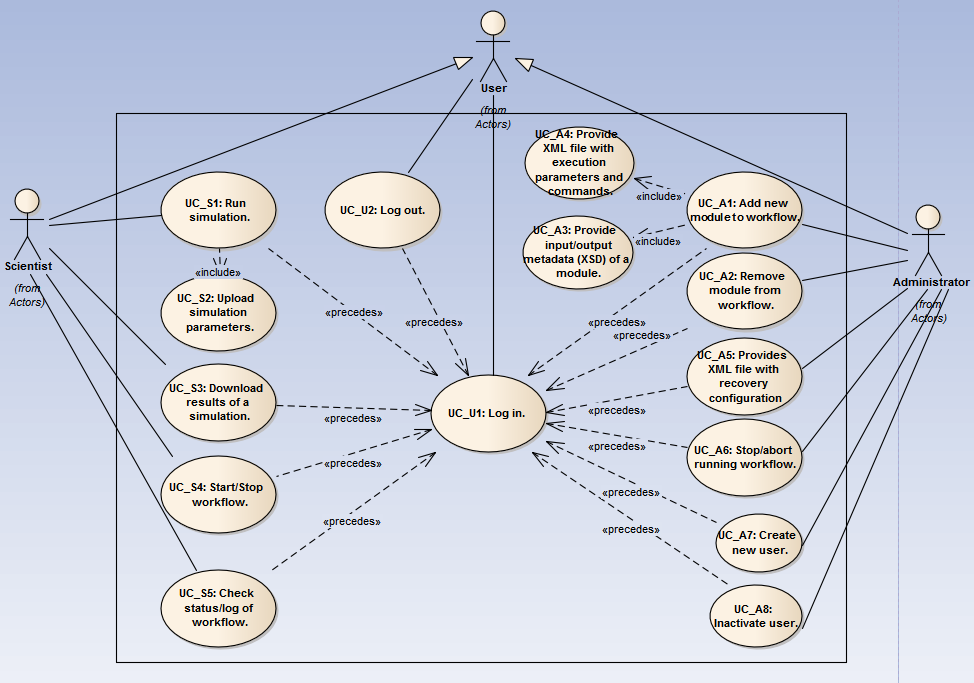
NF2 Use universal/independent communication standards between modules

NF3 Data flow should be based on XML files.

NF4 Modules can run on many different remote systems using different platforms but hey have to be already installed.

NF5 Network connection between terminal, workload server and remote systems are established.

1. System Requirements
2. Use case diagram



Diagrams shows dependencies between actors and use cases.

1. User's case list

* **UC\_U1: Log in.**

|  |  |
| --- | --- |
| **Description:** | Cziki cziki |
| **Requirements:** | **-** |
| **Constraints:** | **-** |
| **Basic Path:** | "Successful log in."  1. User connects through client to system.  2. User types credentials on log in screen.  3. User gain access to system resources. |

* **UC\_U2: Log out.**

|  |  |
| --- | --- |
| **Description:** | Cziki cziki |
| **Requirements:** | **-** |
| **Constraints:** | **-** |
| **Basic Path:** | "Successful log out."  1. User logs out from system. |

1. Scientist's case list

* UC\_S1: Run simulation.
* UC\_S2: Upload simulation parameters.
* UC\_S3: Download results of a simulation.
* UC\_S4: Start/Stop workflow.
* UC\_S5: Check status/log of workflow.

1. Administrator's case list

* UC\_A1: Add new module to workflow.
* UC\_A2: Remove module from workflow.
* UC\_A3: Provide input/output metadata (XSD) of a module.
* UC\_A4: Provide XML file with execution parameters and commands.
* UC\_A5: Provides XML file with recovery configuration
* UC\_A6: Stop/abort running workflow.
* UC\_A7: Create new user.
* UC\_A8: Inactivate user.

1. Appendix A: Used Tools
2. Enterprise Architect 9.2 ver. Trial Ultimate
3. Google Code