Tallinn University of Technology
Department of Informatics
Chair of Software Engineering

Adaptive Personal Training Advisor

A mini-project in the course of
“Agent-oriented modelling and multi-agent systems”

Tallinn 2016

Table of Contents

Introduction

The goal of this Project is to learn the basics of agent-oriented modelling by building a multi-agent system, that helps people in improving their health by composing a gym training plan. The person willing to train defines a goal and the system creates a personalized training plan. The key feature of the system is adaptivity in terms of training plan difficulty and gym equipment availability.

The document consists of three main parts: Requirements analysis, Design and CPN simulation. The first two parts consist mainly of the corresponding diagrams together with descriptions. The third part contains verification and validation of a simplified system simulation using Coloured Petri nets (CPN). Lastly a Conclusion is presented with the overview of the work done.

Requirements analysis

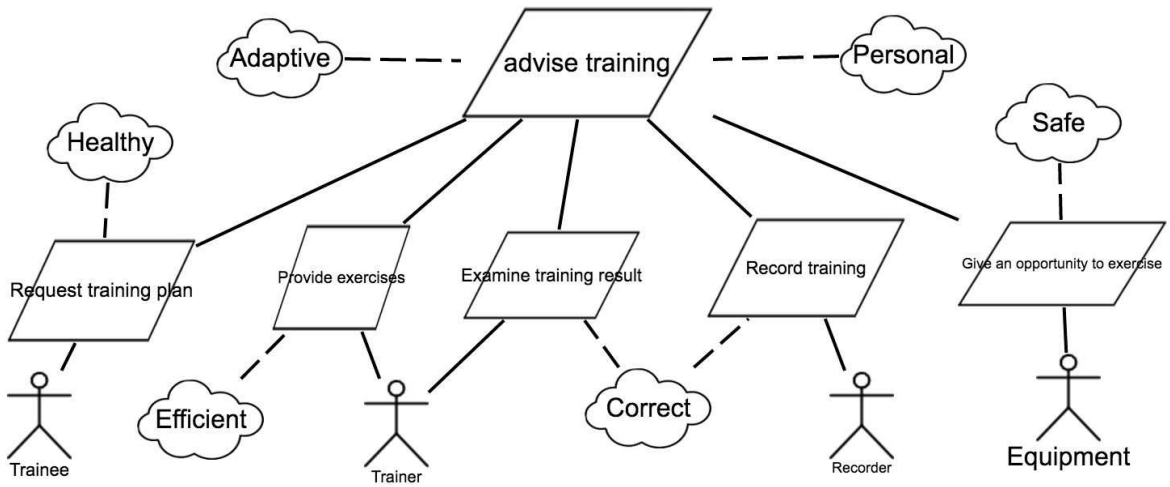
Goal model

The main goal of the system is to advise an adaptive and personalized training for the person (Trainee) willing to improve his/her health. To reach this goal a number of subgoals is created.

Trainee requests a training plan, that must be healthy (sufficiently difficult to present a challenge and be beneficial, but not too difficult to adversely affect well-being).

Training plan consists of exercises, that the Trainer must provide. Trainee has to do the exercises using the Equipment which gives safe opportunity to exercise. The Recorder must correctly record the training session, and the Trainer must analyze the recorded data to make training plan improvements if needed.

The advised training plans are adaptive (an alternative exercise is provided if the Equipment is used by someone else or unavailable at specific gym, the difficulty is adjusted if it is too easy or too hard for the Trainee) and personal (based on the age, weight, height, gender and other personal data the Trainee provides).



Role model

There are 4 roles in the system: Trainee, Trainer, Recorder and Equipment. Below are the detailed characteristics of the roles.

Role name	Trainee
Description	The person who wants to improve his/her physical health

Responsibilities	• Defines goal • Goes to training • Does exercises provided by Trainer
Constraints	Has to do exercises provided by Trainer

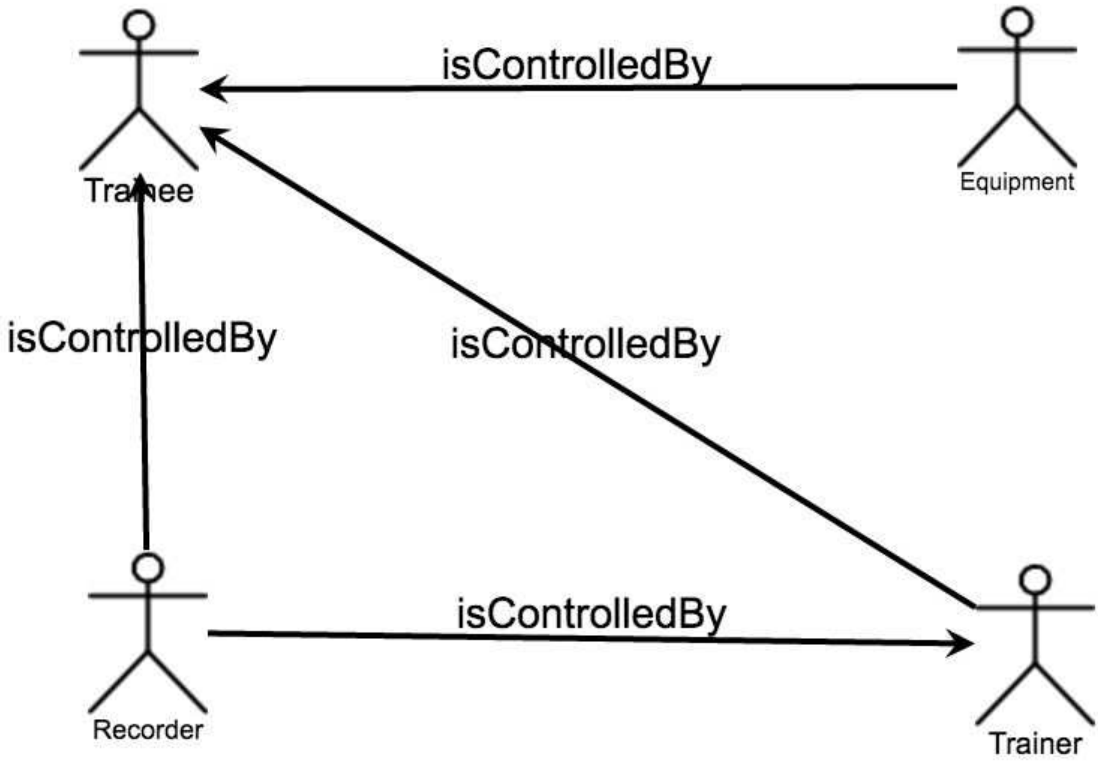
Role name	Trainer
Description	Helps Trainee to achieve his physical goal, provides training plan and exercise information
Responsibilities	• Provides Trainee a training plan • Provides information about exercise (how to do, etc.) • Provides Trainee an alternative exercise • Examine training results
Constraints	• Has to provide adaptive and personal training plan • Has to provide efficient exercises (training plan) • Has to examine training results correctly

Role name	Recorder
Description	Records result of the training (done exercises, order of exercises, etc.)
Responsibilities	• Records done exercises (and their order) • Records Trainee's health/body data
Constraints	• Has to record data without loss • Has to record data correctly

Role name	Equipment
Description	Gym equipment
Responsibilities	• Gives an opportunity to do exercise
Constraints	Has to provide safe opportunity to do exercise

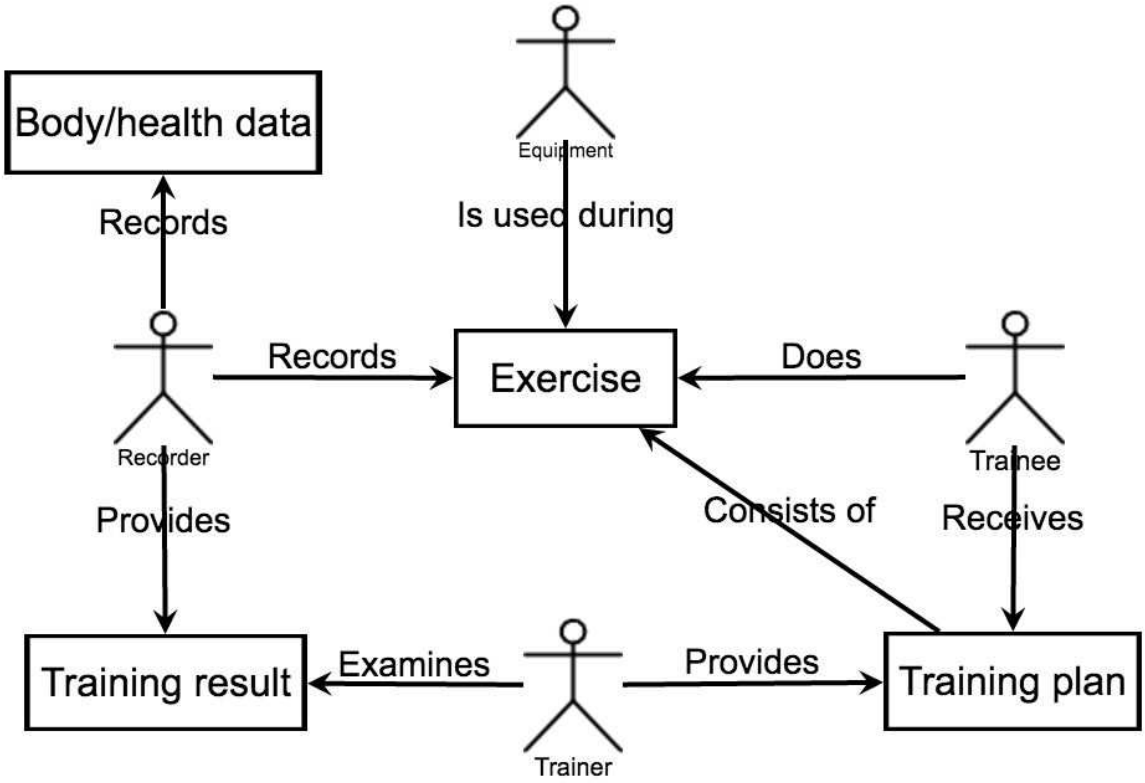
Organization model

Trainee being the single human role has the control over other non-human roles. Meaning its requests are always being complied with. Recorder is also controlled by the Trainer, as Trainer needs recorded data to improve Trainee's training plan.



Domain model

The Trainee receives a Training plan provided by the Trainer. Training plans consist of Exercises. The Trainee does the Exercises using some Equipment. During training the Recorder logs the completed Exercises, records Body/Health data and provides the Training result to the Trainer. The Trainer examines the Training result and makes a new Training plan.


Body/health data
Equipment
Records
Is used during
Records
Exercise
Does
Recorder
Trainee
Provides
Consists of
Receives
Training result
Examines
Provides
Training plan
Trainer

Design

Agent model

There are four agents in the system - Trainee human agent, Trainee software agent, Trainer agent and Equipment agent. Below are the detailed characteristics of the agents.

Agent name	Trainee human agent
Description	Human Agent. The one who wants to improve his/her physical health
Roles	Trainee
Responsibilities	• Defines goal • Goes to training • Does exercises provided by Trainer

Agent name	Trainee software agent
Description	Software Assistant of Trainee Human Agent
Roles	Trainee
Responsibilities	-

Agent name	Trainer agent
Description	Agent that is responsible for providing exercises, recording training and examining training result
Roles	• Trainer • Recorder
Responsibilities	• Provides Trainee a training plan • Provides information about exercise (how to do, etc.) • Provides Trainee an alternative exercise • Examine training results • Records done exercises (and their order) • Records Trainee's health/body data

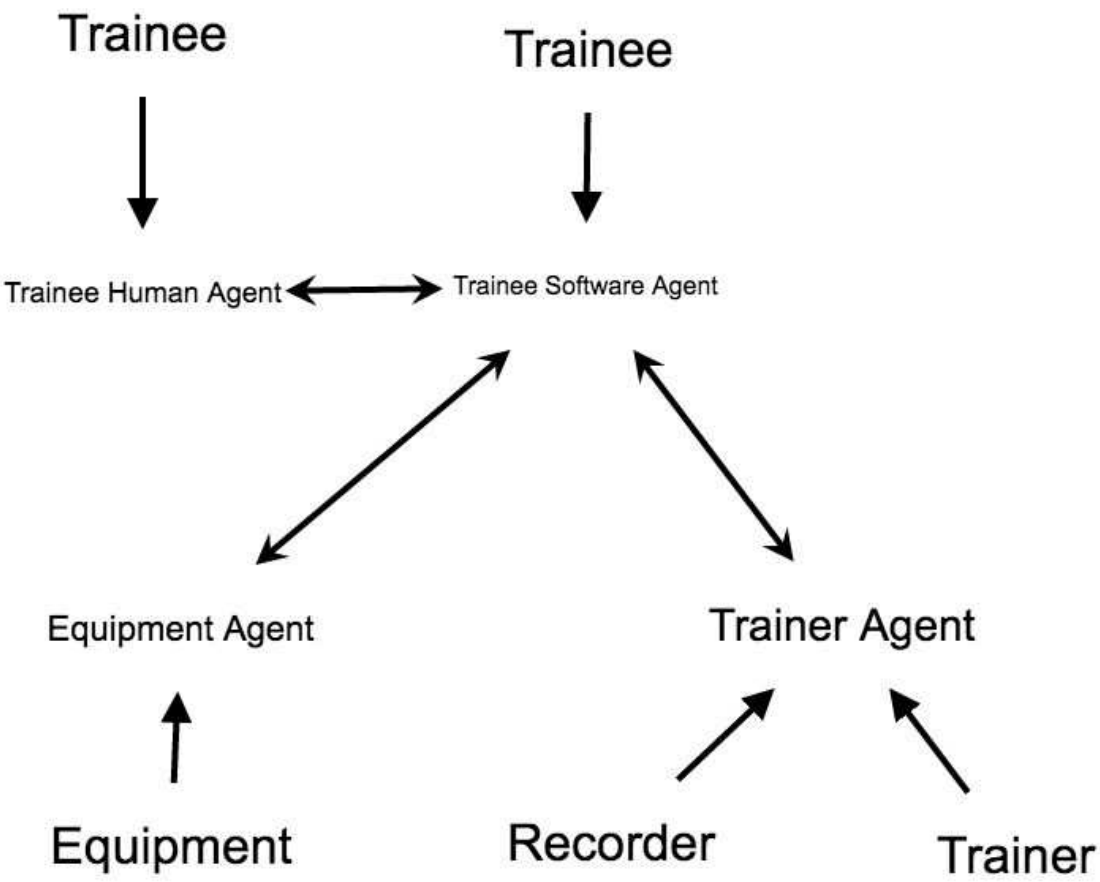
Agent name	Equipment agent
Description	Gym equipment agent
Roles	Equipment
Responsibilities	• Gives an opportunity to do exercise

Acquaintance model

The Trainee role is represented by two agents - a human agent representing a real person and a software agent, which forwards the actions of the human agent to other agents and displays information from them.

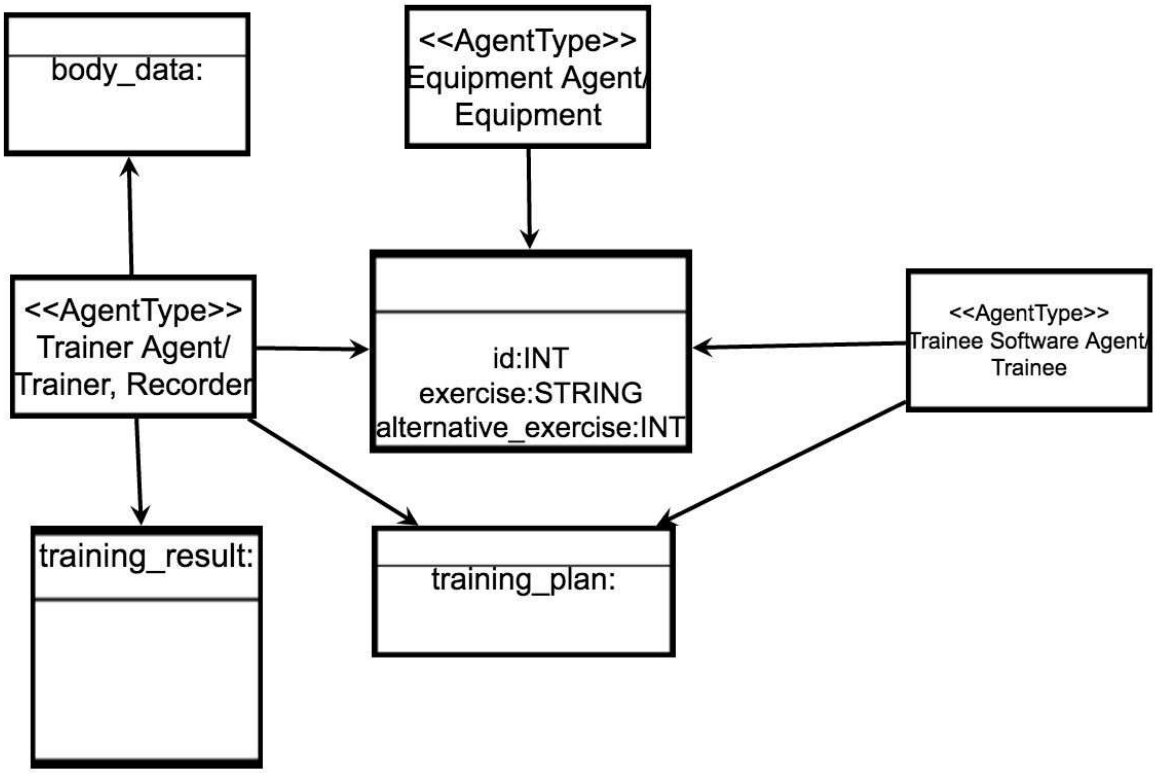
The Trainer agent has two roles - the Recorder and the Trainer and is typically comprised of a software application on a smartwatch or a smartphone.

The Equipment agent has a single role with the same name. Equipment is a single piece of gym equipment.



Knowledge model

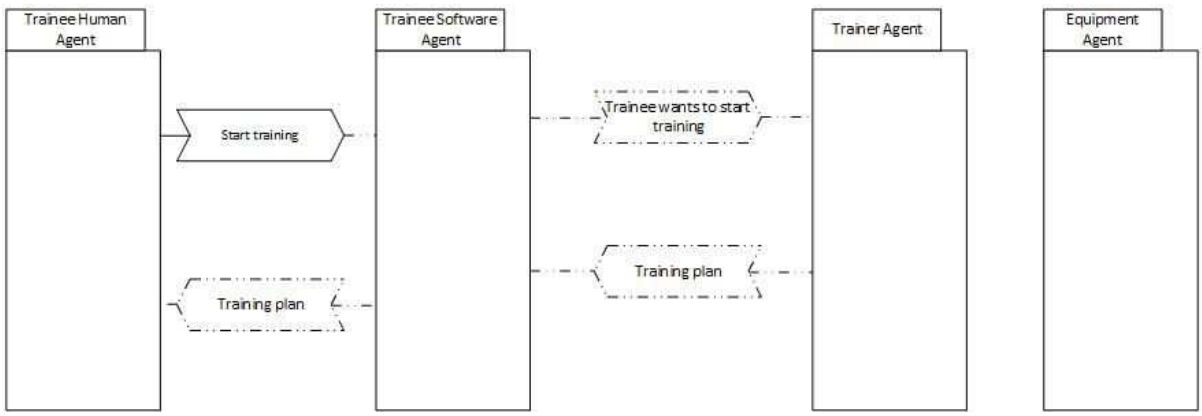
The following model contains the knowledge requirements of the agents. Only the basic attributes are shown.



Interaction models

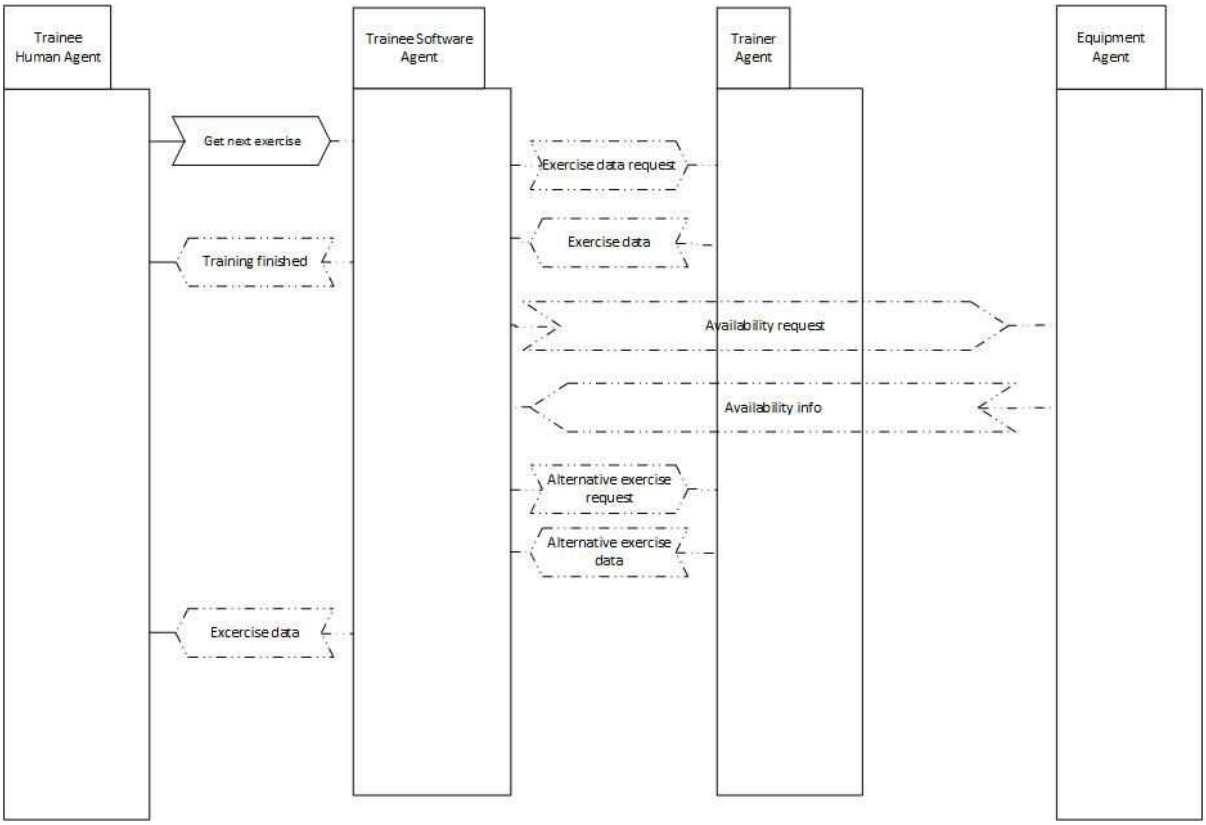
The basic interactions between human and software agents are shown below. Three interaction models were chosen, that capture the key functionality of the project - adaptive training.

Beginning of training



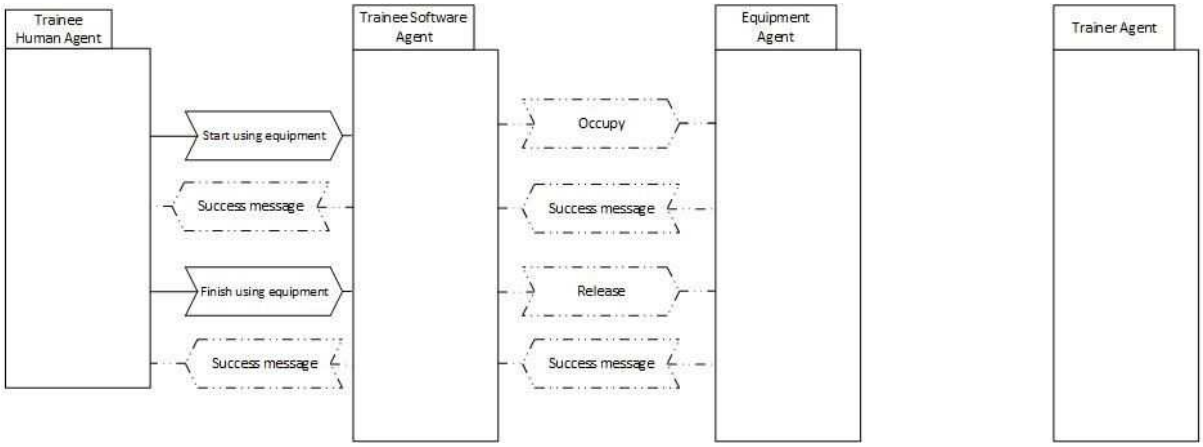
This interaction model shows beginning of the training. After Trainee reached the gym, he tells the Trainer (some wearable device or smartphone) that he wants to start training. All this is done through some UI (Trainee Software Agent). In response Trainer sends an overview of today's training to the Trainee, what is also displayed through some UI.

Exercise selection



Given model is a sequel of previous one (Beginning of training). After Trainee receives today's plan, he requests information about first exercise from Trainer. This information contains different types of media that describe how and using what gym equipment the exercise has to be done. Before Trainee can start exercising, he must to make sure that necessary gym equipment is available. If it is not available, then Trainee (Software Agent) asks for alternative exercise from the Trainer. Then the process is repeated until there is no more exercises left to do.

Exercise

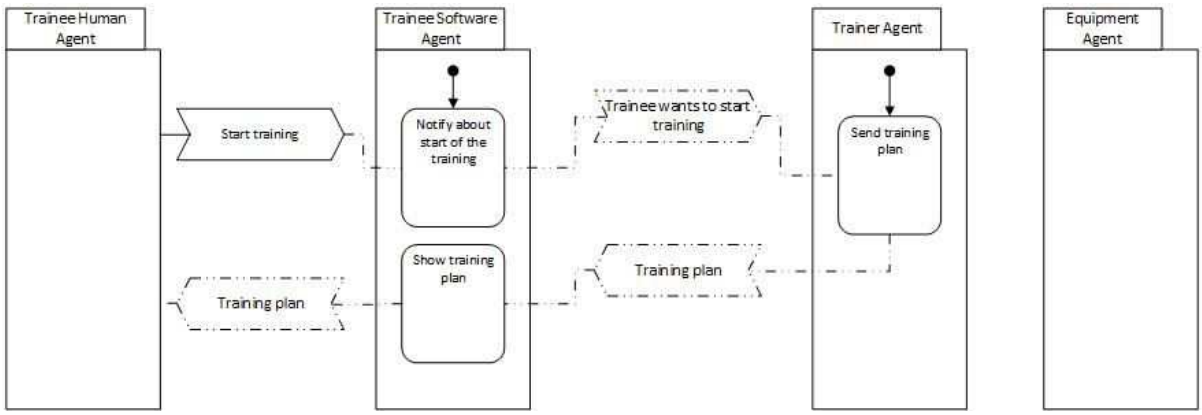


After Trainee receives all needed information, he can finally start exercising. Trainee occupies the gym equipment, does the exercise and, finally, releases the equipment, so others could also use it.

Behaviour models

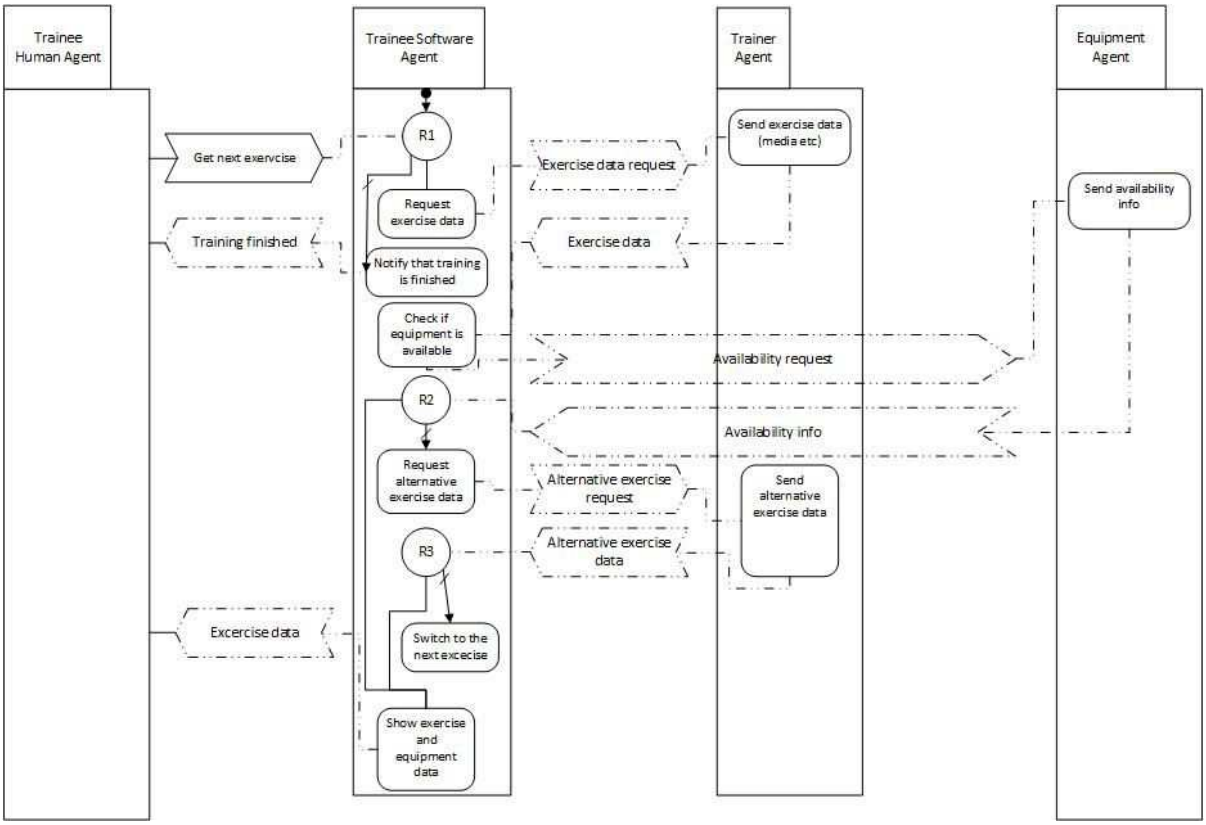
Following are the behavior models for the corresponding interaction models together with rule descriptions. Description of the models can be found in previous section (Interaction models).

Beginning of training



Given model shows beginning of the training. Equipment Agent does not participate in this phase.

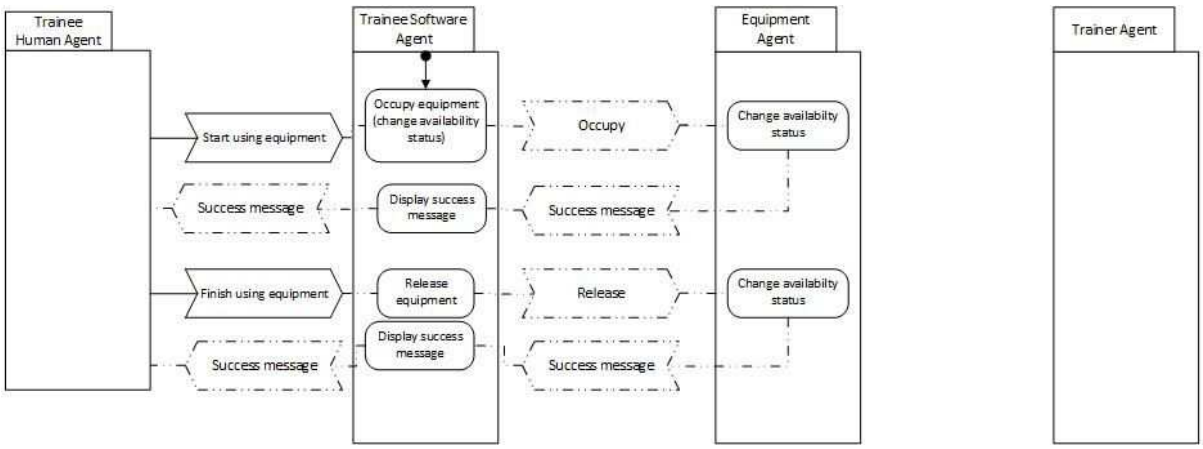
Exercise selection



Given model show exercise selection logic.

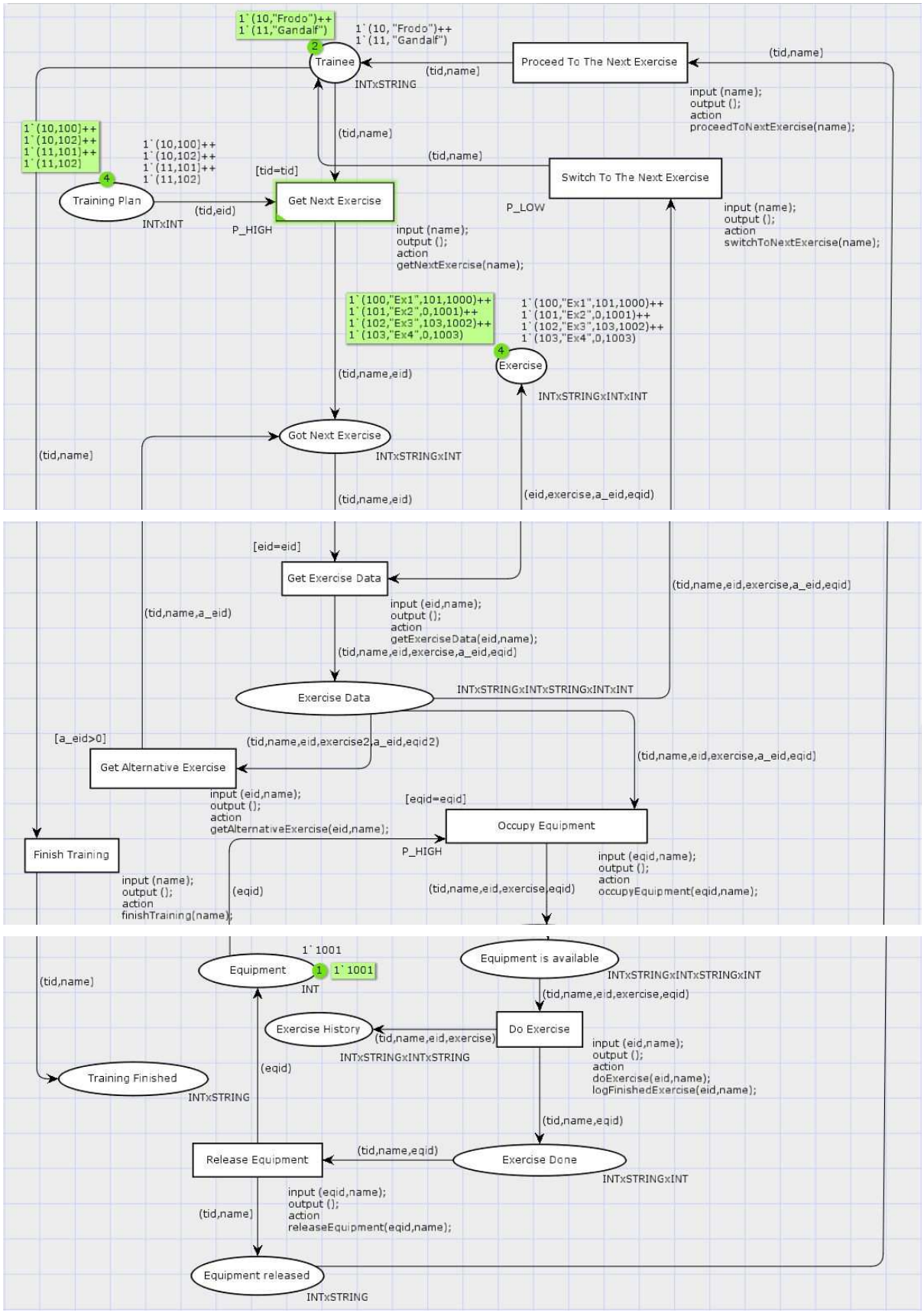
Rules:
R1 - Check if there are exercises left in today's training plan.
R2 - If gym equipment is taken, ask for alternative exercise from Trainer.
R3 - If gym equipment for alternative exercise is also taken, switch to the next exercise.

Exercise



Given model shows training process. Trainer Agent does not participate in this phase.

CPN simulation

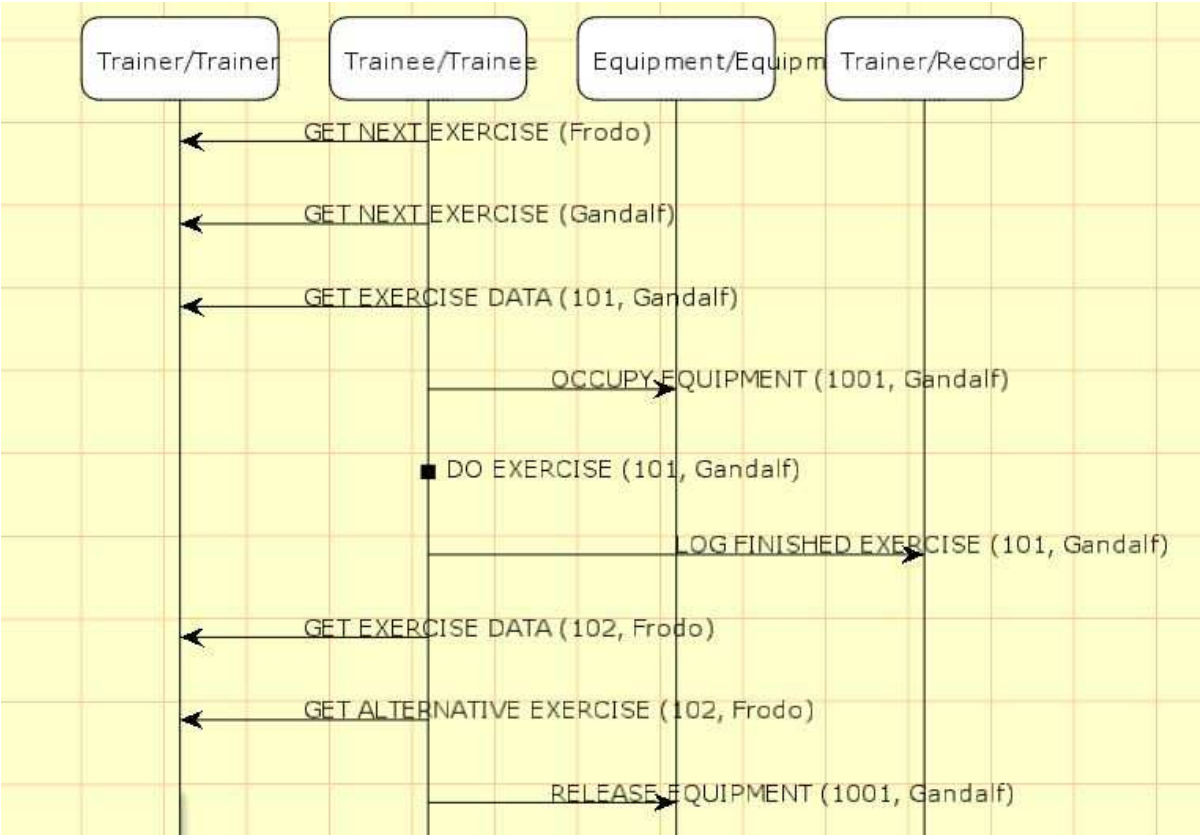


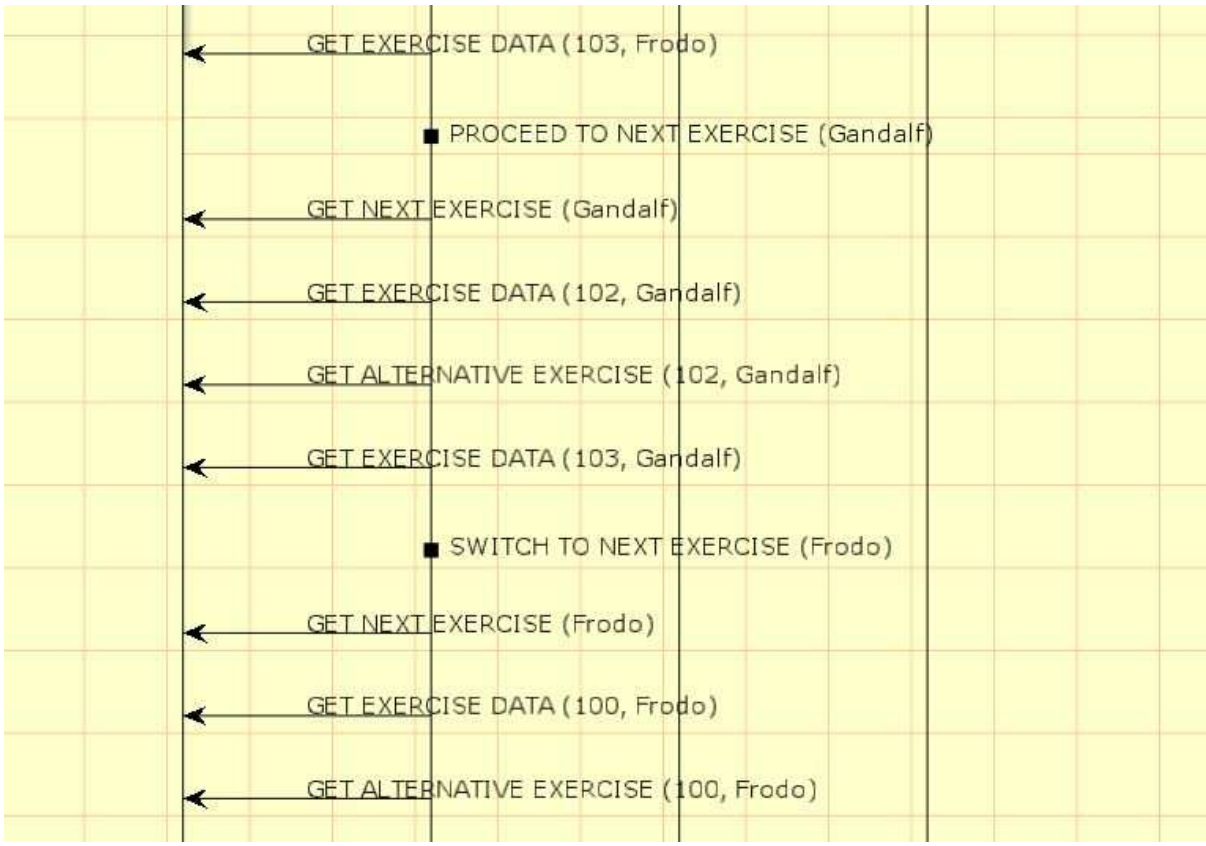
Validation

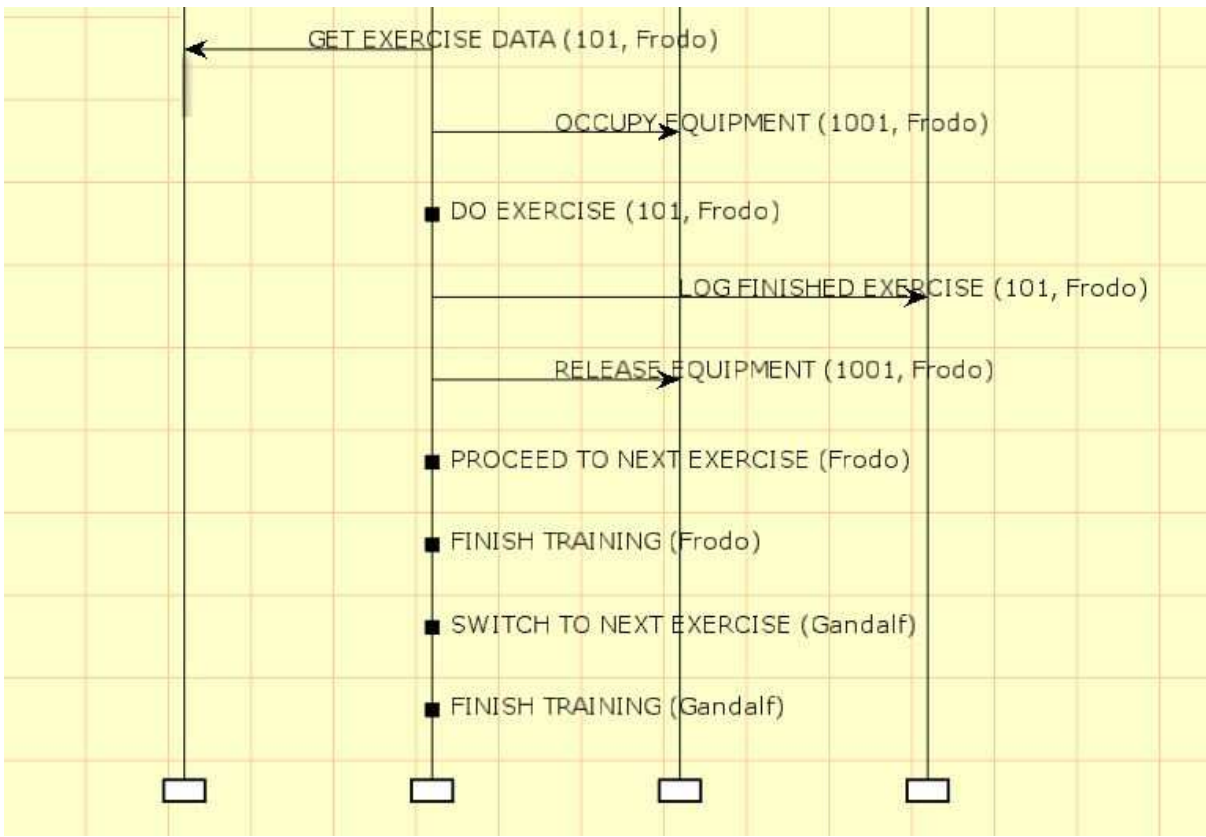
For the mini project 3 different scenarios were chosen that describe situations that can happen during the training. Scenarios are following:

1. Trainee wants to start exercise that is provided by Trainer using specified equipment. This Equipment is available at the gym and is not held by other trainees at the moment. Trainee successfully performs the exercise.
2. Trainee wants to start exercise that is provided by Trainer using specified equipment, but he can't because there is no available equipment. Trainee asks for alternative exercise. The Equipment is available and Trainee successfully performs alternative exercise.
3. Trainee wants to perform exercise, but Equipment used in this and alternative exercise are taken and Trainee has to switch to the next exercise.

Sequence diagram




GET EXERCISE DATA (103, Frodo)
PROCEED TO NEXT EXERCISE (Gandalf)
GET NEXT EXERCISE (Gandalf)
GET EXERCISE DATA (102, Gandalf)
GET ALTERNATIVE EXERCISE (102, Gandalf)
GET EXERCISE DATA (103, Gandalf)
SWITCH TO NEXT EXERCISE (Frodo)
GET NEXT EXERCISE (Frodo)
GET EXERCISE DATA (100, Frodo)
GET ALTERNATIVE EXERCISE (100, Frodo)


GET EXERCISE DATA (101, Frodo)
OCCUPY EQUIPMENT (1001, Frodo)
DO EXERCISE (101, Frodo)
LOG FINISHED EXERCISE (101, Frodo)
RELEASE EQUIPMENT (1001, Frodo)
PROCEED TO NEXT EXERCISE (Frodo)
FINISH TRAINING (Frodo)
SWITCH TO NEXT EXERCISE (Gandalf)
FINISH TRAINING (Gandalf)

Verification

Applying CPN Tools on our mini project we obtain the following result from Report:

Home Markings: [274],
Dead Marking: [274],
Dead Transition Instances: None
Live Transition Instances: None
Fairness Properties: No infinite occurrence sequences

The part of the state space report shown above says that there is one home and one dead marking. These markings have the same node number 274, which means that this node is both a home and a dead marking. One dead marking means that the CPN model is partially correct and if execution terminates then we have the correct result. Furthermore, because node 274 is also a home marking, it is always possible to terminate the protocol with the correct result. Also we can see that there are no infinite occurrence sequences. [4]

Report

```
CPN Tools state space report for:
/cygdrive/C/Users/Sergei/Desktop/cpn_07052016.cpn
Report generated: Sat May  7 22:25:53 2016


 Statistics
------------------------------------------------------------------------

  State Space
     Nodes:  274
     Arcs:   400
     Secs:   13
     Status: Full

  Scc Graph
     Nodes:  274
     Arcs:   400
     Secs:   0


 Boundedness Properties
------------------------------------------------------------------------

  Best Integer Bounds
                            Upper      Lower
     New_Page'Equipment 1    1          0
     New_Page'Equipment_is_available 1
                             1          0
     New_Page'Equipment_released 1
                             2          0
     New_Page'Exercise 1     4          4
     New_Page'Exercise_Data 1
                             2          0
     New_Page'Exercise_Done 1
                             1          0
```

```
     New_Page'Exercise_History 1
                              2          0
     New_Page'Got_Next_Exercise 1
                              2          0
     New_Page'Trainee 1       2          0
     New_Page'Training_Finished 1
                              2          0
     New_Page'Training_Plan 1
                              4          0

  Best Upper Multi-set Bounds
     New_Page'Equipment 1
                           1`1001
     New_Page'Equipment_is_available 1
                           1`(10,"Frodo",101,"Ex2",1001)++
1`(11,"Gandalf",101,"Ex2",1001)
     New_Page'Equipment_released 1
                           1`(10,"Frodo")++
1`(11,"Gandalf")
     New_Page'Exercise 1 1`(100,"Ex1",101,1000)++
1`(101,"Ex2",0,1001)++
1`(102,"Ex3",103,1002)++
1`(103,"Ex4",0,1003)
     New_Page'Exercise_Data 1
                           1`(10,"Frodo",100,"Ex1",101,1000)++
1`(10,"Frodo",101,"Ex2",0,1001)++
1`(10,"Frodo",102,"Ex3",103,1002)++
1`(10,"Frodo",103,"Ex4",0,1003)++
1`(11,"Gandalf",101,"Ex2",0,1001)++
1`(11,"Gandalf",102,"Ex3",103,1002)++
1`(11,"Gandalf",103,"Ex4",0,1003)
     New_Page'Exercise_Done 1
                           1`(10,"Frodo",1001)++
1`(11,"Gandalf",1001)
     New_Page'Exercise_History 1
                           1`(10,"Frodo",101,"Ex2")++
1`(11,"Gandalf",101,"Ex2")
     New_Page'Got_Next_Exercise 1
                           1`(10,"Frodo",100)++
1`(10,"Frodo",101)++
1`(10,"Frodo",102)++
1`(10,"Frodo",103)++
1`(11,"Gandalf",101)++
1`(11,"Gandalf",102)++
1`(11,"Gandalf",103)
     New_Page'Trainee 1  1`(10,"Frodo")++
1`(11,"Gandalf")
     New_Page'Training_Finished 1
                           1`(10,"Frodo")++
1`(11,"Gandalf")
     New_Page'Training_Plan 1
                           1`(10,100)++
1`(10,102)++
1`(11,101)++
1`(11,102)

  Best Lower Multi-set Bounds
     New_Page'Equipment 1
                           empty
     New_Page'Equipment_is_available 1
                           empty
     New_Page'Equipment_released 1
```

```
                        empty
     New_Page'Exercise 1 1`(100,"Ex1",101,1000)++
1`(101,"Ex2",0,1001)++
1`(102,"Ex3",103,1002)++
1`(103,"Ex4",0,1003)
     New_Page'Exercise_Data 1
                        empty
     New_Page'Exercise_Done 1
                        empty
     New_Page'Exercise_History 1
                        empty
     New_Page'Got_Next_Exercise 1
                        empty
     New_Page'Trainee 1  empty
     New_Page'Training_Finished 1
                        empty
     New_Page'Training_Plan 1
                        empty


 Home Properties
--------------------------------------------------------------------------------

  Home Markings
     [274]


 Liveness Properties
--------------------------------------------------------------------------------

  Dead Markings
     [274]

  Dead Transition Instances
     None

  Live Transition Instances
     None


 Fairness Properties
--------------------------------------------------------------------------------
     No infinite occurrence sequences.
```

Conclusion

During the project, all of the project requirements were achieved. Different level analysis and design models were constructed, core functionality CPN simulation was created. During the implementation of the project team members were introduced to web-based agent-oriented modelling tool and CPN Tools software package.

CPN Tools has a nice idea of model validation and verification, but is severely lacking a modern user interface and a decent documentation. The software package is not widely used, thus it is impossible to compensate the lack of proper documentation with online research on resources such as stackoverflow.com.

Most of the time using CPN Tools was spent on trying to find out how to implement the simplest programming constructs (e.g. loops and branches) for the most basic business logic, and refactoring (simplifying) the AOM model when the CPN created was becoming too complex. CPN Tools software package aims to be cross-platform but is not working properly on anything other than Microsoft Windows. Out of three team members only one uses Windows as the main OS.

Overall the team members find the usage of CPN Tools package counter-productive and suggest finding at least an alternative editor for CPN files. Also more complex CPN examples should be provided on course home page.

Nevertheless the members are pleased with the project and the experience gained from making it.

References

1. Laboratory of Socio-Technical Systems. "Agent-Oriented Modelling and Multiagent Systems (2016)". [WWW] http://maurus.ttu.ee/sts/?page_id=2222 (08.05.2016)
2. CPN Tools. “Documentation”. [WWW] http://cpntools.org/documentation/start (08.05.2016)
3. Sterling, Leon S., and Kuldar Taveter. The Art of Agent-Oriented Modeling (2009). The MIT Press.
4. Kurt Jensen, Lars M. Kristensen. Coloured Petri Nets: Modelling and Validation of Concurrent Systems (2009). Springer.
5. Bogdan Aman, Gabriel Ciobanu. Mobility in Process Calculi and Natural Computing (2011). Springer.