ABout Time Linux - Configuration Manager (v0.6)

Sander Bosman sbosman@gmail.com
Nordine Omari n.omari@studen.ru.nl
Maikel Couwenberg maikelc@gmx.net
Eric D. Schabell ericS@abtlinux.org

December 7, 2006

Contents

1 Introduction

AbTLinux is a source-based Linux distribution for which new functionality is continuously being released. Mostly this includes improvements or add-ons. Functionality is released in the form of packages, which can be downloaded, configured and installed.
For most of the AbTLinux Community some of the most annoying problems are the configuration problems. Some of these problems are for example that the user is unable to find the configuration or that the installation process completely overwrites the current configuration of the user. Therefore the AbTLinux Community has asked a group of Radboud University Students to develop the requirements for a component that will solve these problems. In this paper we will present our findings.

2 Problem Statement

The configuration of packages can be rather difficult when dealing with different options for every package and different choices for each user, existing configurations, etc. Because of the diversity in backgrounds of its end-users and developers, these difficulties need to be minimized as much as possible.

3 Stakeholder Analysis

3.1 Stakeholder 1

Name:	Eric D. Schabell
Contact:	ericsabtlinux.org
Description:	Project Leader of the AbTLinux Community and initiator of the configManager project.

3.2 Stakeholder 2

Name:	AbTLinux Community
Contact:	http://www.abtlinux.org/
Description:	All the persons that contribute to the development of AbTLinux and of course the End Users

3.3 Stakeholder 3

Name:	RuDev
Contact:	sbosman@gmail.com,maikelc@gmx.net, n.omari@student.ru.nl
Description:	The development team of the requirements from the config-Manager project. The members of RuDev are: • Sander Bosman • Maikel Couwenberg • Nordine Omari

3.4 Stakeholder 4

Name:	Stijn Hoppenbrouwers
Contact:	stijnh@cs.kun.nl
Description:	Supervisor from the Radboud University which evaluate the process of designing this requirement document.

4 Mission, Vision and Values

4.1 Mission

The AbTLinux community is trying to build up a source-based Linux distribution based on a tool set that is well documented. After designing a well documented dependency engine, the community now wants to get rid of configuration trouble. In most of the current Linux distributions the configuration can give the user a lot of trouble. The biggest problems which need to be solved are the overwriting of your existing configuration files, after install not knowing where to find the configuration files and not be able to merge older and newer configuration files.

4.2 Vision

The aim of this configManager project is to develop a well documented configManager for the AbTLinux community which will solve the problems as mentioned at the Problem statement (Chapter 2).

4.3 Values

The AbTLinux community has as main value:

> With clear goals, clear documentation and clear coding practices everyone from users on down to developers can benefit from our project.

We will try to accomplish this value by designing well documented requirements for the configManager Project.

5 Statement of Work

5.1 Project Scope

This project is committed to delivering the requirements documentation for a configManager for the AbTLinux project. The documentation should be understandable for all relevant stakeholders and is to be used by the developers. Therefore it is agreed upon that the documentation includes a description of the system presented as a complete set of use cases and use case diagram. The configManager should be able to manage configurations of packages in a user-friendly manner. AbTLinux and its users will depend on the configManager for the configuration of packages when it is running. The development of the configManager is outside the scope of this project.

5.2 Objectives

- The project team will deliver the requirements documentation for a configManager for AbTLinux.
- The documentation will describe the system requirements with use cases.
- The requirements documentation forms the basis for the development of the system.
- The documentation should include all relevant system requirements and thus be formed in cooperation with all relevant stakeholders.

5.3 Application overview

The configManager is part of the distribution AbTLinux. It must be able to provide a mechanism for management of options such as:

- Install new configuration
- Manage upgrade of configuration
- Merge two configurations
- Ensure never losing an existing configuration
- Backing up configurations
- Edit configurations
- Find configurations

5.4 User demography

Developers
The developers need a clear description of the system requirements.

AbTLinux
The AbTLinux system will depend on the configManager for the management of configurations for packages in AbTLinux.

End-users
The end-users will depend on the configManager for the management of configurations for packages in AbTLinux. When participating as stakeholder in the project they will have to be able to understand the system requirements as documented.

5.5 Constraints

- The system requirements need to be described with use cases and use case diagrams.
- The requirements documentation must be completed before 8th of December 2006.
- The groupmembers wil each be available for 6 ects during this project.
- The use of the English language because the developers come from different countries.

5.6 Assumptions

The project team relies on input from all relevant stakeholders. We expect to be able to meet with the executive sponsors (Eric Schabell) and his assistant (Ilona Wilmont) for their input and feedback. Also we expect to be supported in the documentation process by Stijn Hoppenbrouwers with issues related to requirements engineering. The Configuration Management use cases are related to the AbTLinux use cases and partially dependant on them. We expect these use cases to be complete and correct.

5.7 Staffing and cost

The project will be performed by:

Name	Email	StudentNr	Availability
Sander Bosman	sbosman@gmail.com	0534323	6 ects
Maikel Couwenberg	maikelc@gmx.net	0427551	6 ects
Nordine Omari	n.omari@student.ru.nl	?	6 ects

Project costs include time effort in work by the project team.

5.8 Deliverable outlines

The documentation of the configManager for AbTLinux should include the following documents:

- Introduction
- Problem statement
- Stakeholder list/analysis
- Mission-Vision-Values
- Statement of Work
- Risk Analysis
- Goal Analysis
- Use Case Survey
- Use case Diagram(s)
- Use Cases
- Scenarios
- Domain Models
- Business Rules Catalogue
- Non-functional Requirements
- Terminological Definitions
- Executive sponsor viewpoint
- Use case tests
- Business process definitions (if necessary)
- GUI metaphors / storyboards (if necessary)

5.9 Planning

Deliverable	Deliverable Type	Façade	Filled	Focused	Made by who?
Introduction	Contextual	Preliminary version	Preliminary version	Complete	Maikel
Problem statement	Key deliverable	As good as possible	As good as possible	Complete	Maikel
Stakeholder list/analysis	Contextual	As good as possible	As good as possible	Complete	Maikel
Mission-Vision-Values	Contextual	Complete	Complete	Complete	Sander
Statement of Work	Contextual	Complete, and up-to-date	Complete, and up-to-date	Complete, and up-to-date	Sander
Risk Analysis	Contextual	Complete, and up-to-date	Complete, and up-to-date	Complete, and up-to-date	Nordine
Goal Analysis	Key deliverable	Sketch	Nearly complete	Complete	Maikel
Use Case Survey	Key deliverable	As good as possible	Nearly complete	Complete	Sander
Use case Diagram(s)	Key deliverable	As good as possible	Nearly complete	Complete	Sander
Use Cases	Key deliverable	Not yet!	"Filled" level	Complete	Sander
Scenarios	Key deliverable	Not yet!	Several for each UC	Complete ("focused" level)	Nordine
Domain Models	Key deliverable	Not yet!	Sketch	Complete	Maikel
Busines Rules Catalogue	Key deliverable	A few	Partially complete	Complete	Nordine
Non-functional Requirements	Key deliverable	Some initial notes	Partially complete	Complete	Nordine
STerminological Definitions	Key deliverable	Some initial notes	Partially complete	Complete	Sander
Executive sponsor viewpoint	Implicit deliverable	Complete	Complete	Complete	Nordine
Use case tests	Implicit deliverable	some initial notes	As good as possible	Complete	Sander
Busienss process definitions	Optional appendix	If available / relevant	If relevant	If relevant	Maikel
GUI metaphors / storyboards	Optional appendix	If relevant	If relevant	If relevant	Maikel

Figure 1: Detailed Planning

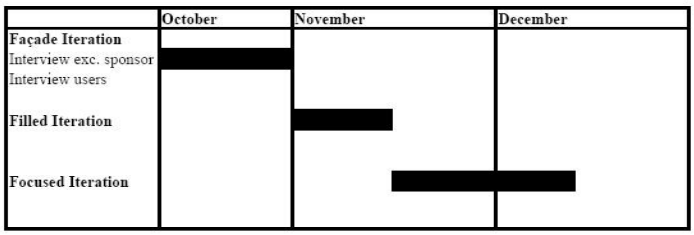


Figure 2: Grant Chart

Expected duration
The project has the following fixed deadlines:

- Phase I (Facade): Thursday 26th of October, 9.00 hrs.
  In the Facade phase the requirements are outlined in use cases without much details.
- Phase II-a (Filled): Friday 24th of November, 18.00 hrs.
- Phase II (Filled and Focused): Thursday 8th of December, 9.00 hrs.
  In the Filled phase the use cases are extended and filled.
  At the Focused phase essential use cases are separated from the nice-to-have and completed.

All required documentation will be expected one week in advance of the deadline among the project team.

A detailed worksheet of which member worked at which task can be found at:

`http://wiki.abtlinux.org/wikka.php?wakka=RuDev`

6 Risk Analysis

Nr	Category	Risk	Days Lost if it occurs	Likelihood It will happen	Risk Rating
1	User Time	The user and the Project manager are involved in this project. So they have to be available to be interviewed at all times to be able to make decisions. If the project members have no time to do that, the project will have Delay	7	55%	3.85
2	Project Member Time	By Illness or the absence of one of the project members, the project will be delayed	3	70%	2.1
3	User Contribution	The Users group is very large and divided. The project will be delayed if the point of view is different. so what is efficient and what can make the system valid will be quit complex	7	30%	2.1
4	The realization of the system	When the list of the wishes is long, the requirement list will give a problem. The conflict of ideas and misunderstanding will make the realization of this project impossible	7	65%	4.55
5	Project Member Time	When one of the group members is out of reach. It will be difficult to get the work done before the given deadline	7	25%	2.0

7 Goal Analysis

7.1 Goals Model

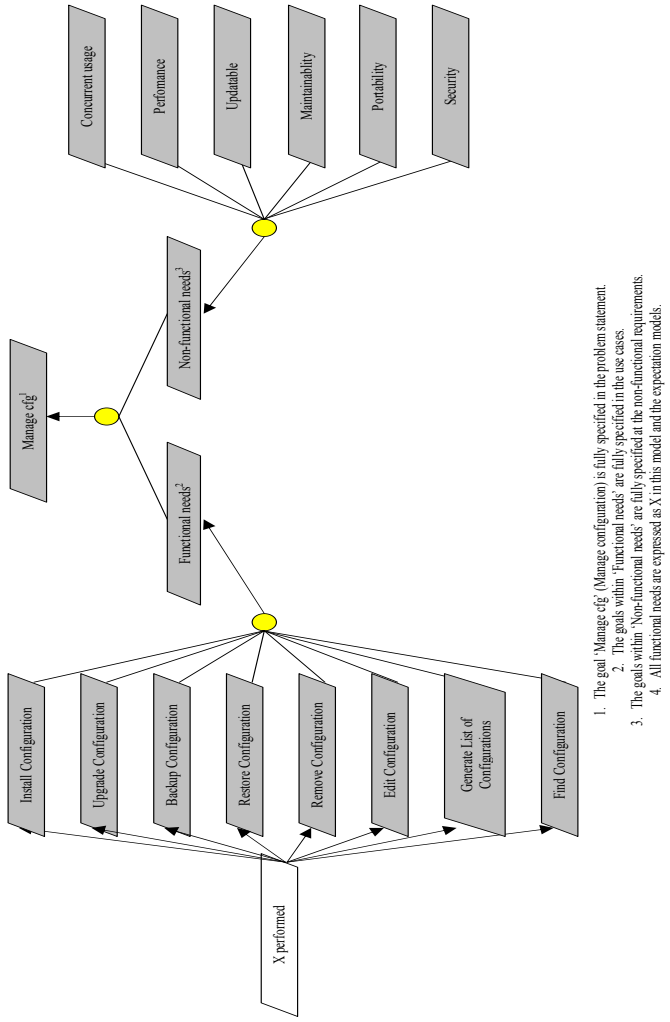
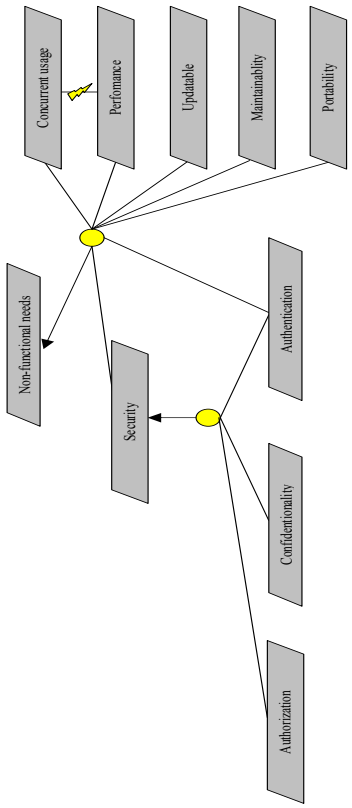


Figure 3: Goals Model

7.2 Non-functional Goals Model



1. The goal 'Manage cfg' (Manage configuration) is fully specified in the problem statement.
2. The goals within 'Functional needs' are fully specified in the use cases.
3. The goals within 'Non-functional needs' are fully specified at Non-functional requirements.
4. This functional need is also expressed as X in this model and the expectation models.

Figure 4: Non-functional Goals Model

7.3 Expectations Model: Performed

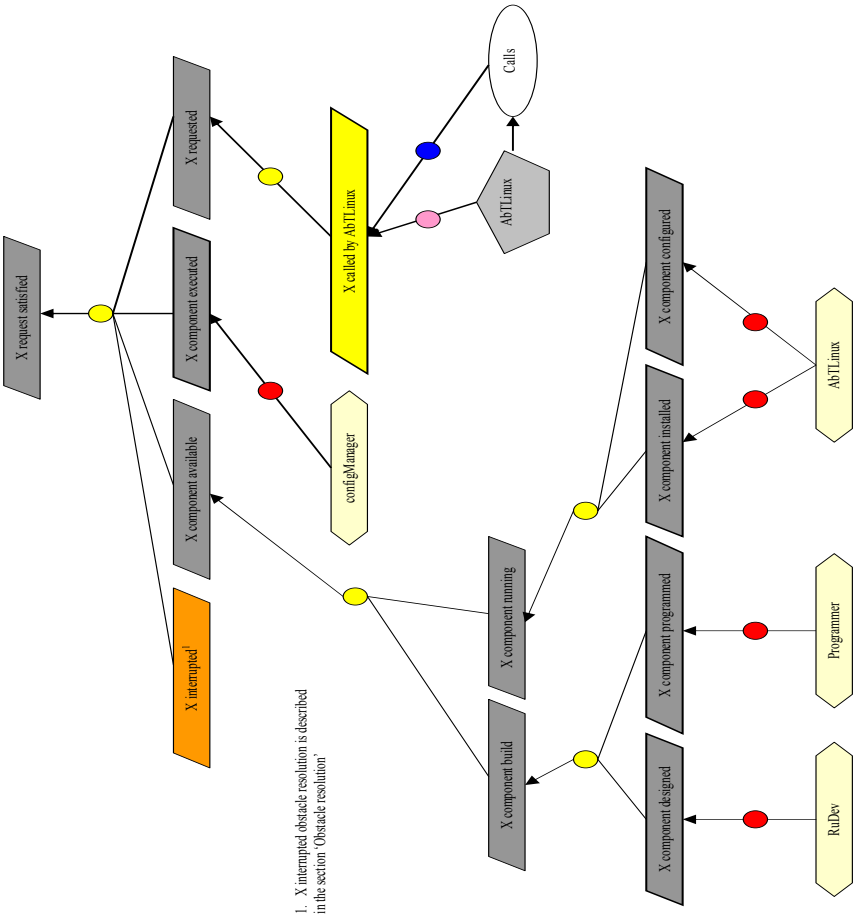


Figure 5: Expectations Model: Performed

7.4 Expectations Model: Handled

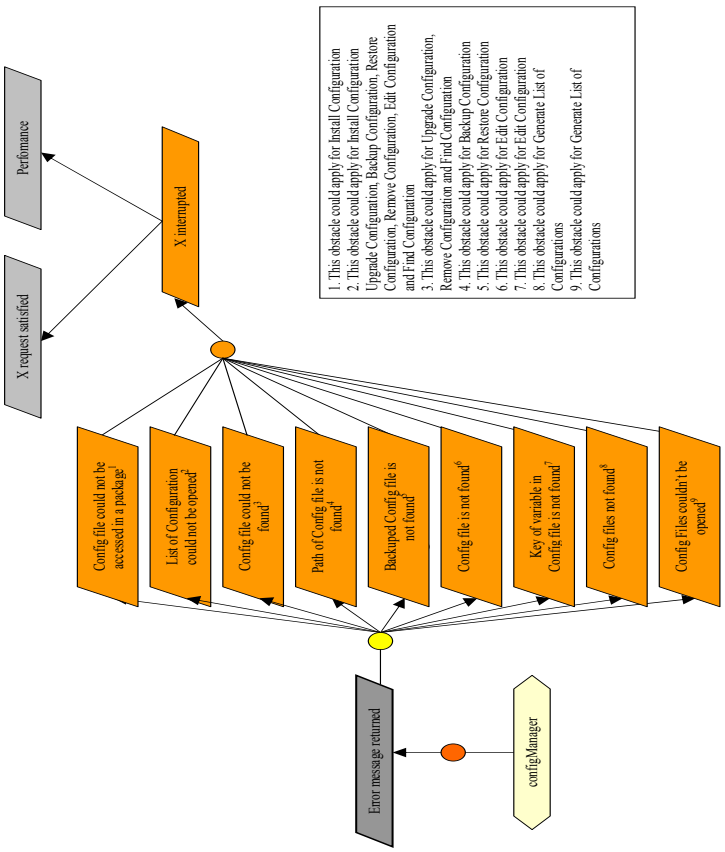


Figure 6: Expectations Model: Handled

7.5 Expectations Model: Actor View

AbTLinux

Expectations:

- X called by AbTLinux
- X component installed
- X component configured

configManager

Requirements:

- Error message returned
- X component executed

Programmer

Expectations:

- X component programmed

RuDev

Expectations:

- X component designed

8 Use Case Survey

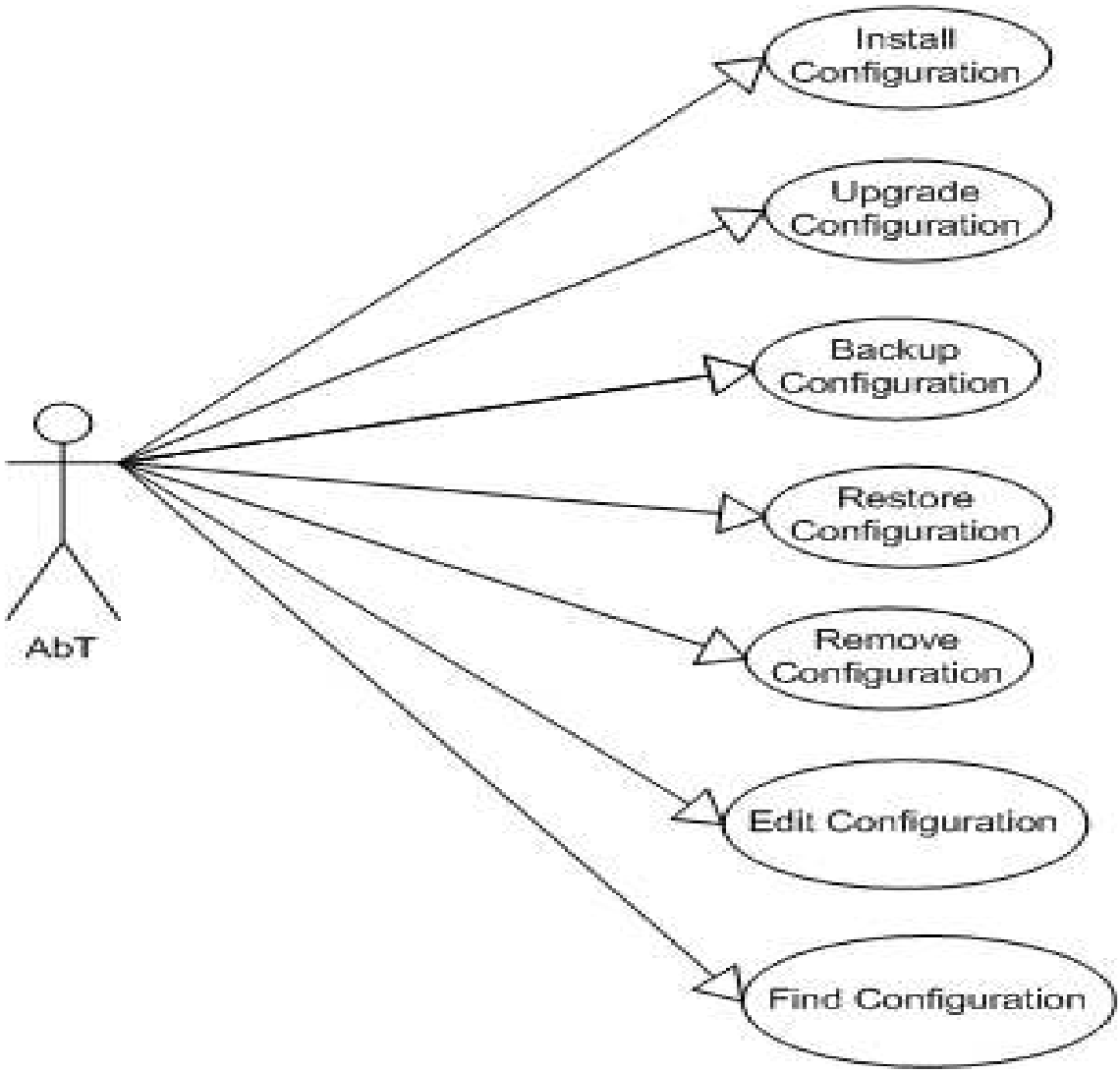


Figure 7: Use Case Diagram

8.1 Use Case Survey 1

Use Case Survey Name:	Install Configuration
Use Case Survey Number:	001
Initiating actor:	AbT
Description:	A package containing a new configuration will be installed.
Completeness:	Complete for Focused Stage
Maturity:	Mature for Focused Stage
Dependency:	None
Source:	Presentation by Eric Schabell

8.2 Use Case Survey 2

Use Case Survey Name:	Upgrade Configuration
Use Case Survey Number:	002
Initiating actor:	AbT
Description:	Upgrade the current configuration to the new standard
Completeness:	Complete for Focused Stage
Maturity:	Mature for Focused Stage
Dependency:	None
Source:	Presentation by Eric Schabell

8.3 Use Case Survey 3

Use Case Survey Name:	Backup Configuration
Use Case Survey Number:	003
Initiating actor:	AbT
Description:	Backup the current configuration to a safe location
Completeness:	Complete for Focused Stage
Maturity:	Mature for Focused Stage
Dependency:	None
Source:	Presentation by Eric Schabell

8.4 Use Case Survey 4

Use Case Survey Name:	Restore Configuration
Use Case Survey Number:	004
Initiating actor:	AbT
Description:	Restore a backup version of a configuration
Completeness:	Complete for Focused Stage
Maturity:	Mature for Focused Stage
Dependency:	None
Source:	Presentation by Eric Schabell

8.5 Use Case Survey 5

Use Case Survey Name:	Remove Configuration
Use Case Survey Number:	005
Initiating actor:	AbT
Description:	Delete a existing configuration from the system
Completeness:	Complete for Focused Stage
Maturity:	Mature for Focused Stage
Dependency:	None
Source:	Presentation by Eric Schabell

8.6 Use Case Survey 6

Use Case Survey Name:	Edit Configuration
Use Case Survey Number:	006
Initiating actor:	AbT
Description:	Edit variables in a current configuration
Completeness:	Complete for Focused Stage
Maturity:	Mature for Focused Stage
Dependency:	None
Source:	Presentation by Eric Schabell

8.7 Use Case Survey 7

Use Case Survey Name:	Generate List of Configurations
Use Case Survey Number:	007
Initiating actor:	AbT
Description:	Generate a list containing every configuration from the system
Completeness:	Complete for Focused Stage
Maturity:	Mature for Focused Stage
Dependency:	None
Source:	Presentation by Eric Schabell

8.8 Use Case Survey 8

Use Case Survey Name:	Find Configuration
Use Case Survey Number:	008
Initiating actor:	AbT
Description:	Search the entire system for a configuration identified by identifiers given by AbT
Completeness:	Complete for Focused Stage
Maturity:	Mature for Focused Stage
Dependency:	None
Source:	Discovered from a feedback session with Eric Schabell

9 Use Cases

9.1 Use Case 1

Use Case Name:	**Install Configuration**
Authors:	RuDev
Dates:	04-12-2006
Iteration:	Focused
Description:	A package containing a new configuration will be installed.
Actors:	AbT
Preconditions:	configManager has access to the package configuration file(s). configManager has generated a list of all configuration files
Triggers:	AbT submits a request for installing a configuration from a package.
Basic Course of Events:	1. AbT submits a request for installing a configuration from a package 2. Open list of all configuration files 3. Search the list for the path of the last version of the configuration. 4. Send request to AbT to backup and remove the old configuration at the path given at step 3 5. Send request to AbT to store the new configuration at the path given at step 3
Alternative Path:	At Step 3: If no last version is found of the configfile a. Continue with step 5 with a path given from AbT
Exceptions:	If configuration file(s) could not be accesed in a package a. Return ERROR List of Configuration could not be opened b. Return ERROR
Assumptions:	1. AbT logs every action. 2. The package to install contains configuration file(s)
Postconditions:	1. Installed configuration from the package 2. The old configuration is backuped
Related business rules:	-

9.2 Use Case 2

Use Case Name:	**Upgrade Configuration**
Authors:	RuDev
Dates:	04-12-2006
Iteration:	Focused
Description:	Upgrade the current configuration to the new standard
Actors:	AbT
Preconditions:	a. configManager has access to the package new configuration file(s). b. configManager has generatad a list of all configuration files
Triggers:	AbT submits a requests to upgrade the current configuration with a new one from the package.
Basic Course of Events:	1. AbT submits a requests to upgrade the current configuration with a new one from the package. 2. Open list of all configfiles. 3. Search in the list for the path of the current version of the configuration. 4. Open the current configuration from the path given at step 3 5. Compare the configuration variables from the current file given at step 4 with the variables from the new configuration file 6. Backup the current configuration 7. Return to AbT a request for updating the configuration with the differences found at step 5 8. Return to AbT a request for updating version number.
Alternative Path:	At Step 5: The configurations are equal a. Skip steps 6,7,8
Exceptions:	Current configuration could not be found a. Return ERROR List of Configuration could not be opened b. Return ERROR

Assumptions:	1. AbT logs every action. 2. The package for the upgrade process contains configuration file(s) 3. configManager has access to read configuration file(s)
Postconditions:	1. Current configuration is upgraded with the variables from the upgrade package. 2. The old configuration is backuped
Related business rules:	-

9.3 Use Case 3

Use Case Name:	**Backup Configuration**
Authors:	RuDev
Dates:	04-12-2006
Iteration:	Focused
Description:	Backup the current configuration to a safe location
Actors:	AbT
Preconditions:	Configuration to backup exists configManager has generated a list of all configuration files
Triggers:	AbT submits a request for backup a configuration with given identifiers.
Basic Course of Events:	1. AbT submits a request for backup a configuration with given identifiers. 2. Open list of all configuration files 3. Search the path of the configuration with given identifiers 4. Send request to AbT to store the configuration file, from the path given at step 3, to the backup location 5. Send request to add the current timestamp to the backup.
Alternative Path:	None
Exceptions:	Path of Configuration File is not found a. Return ERROR List of Configuration could not be opened b. Return ERROR
Assumptions:	1. AbT logs every action. 2. AbT gives valid identifiers for the configuration file.
Postconditions:	1. Backup made of Configuration file
Related business rules:	-

9.4 Use Case 4

Use Case Name:	**Restore Configuration**
Authors:	RuDev
Dates:	04-12-2006
Iteration:	Focused
Description:	Restore a backup version of your configuration.
Actors:	AbT
Preconditions:	Configuration to restore exists and is specified by identifiers and a timestamp
Triggers:	AbT submits a request for restore a configuration with given identifiers and timestamp
Basic Course of Events:	1. AbT submits a request for restore a configuration with given identifiers and timestamp 2. Open list of all configuration files 3. Search the list for the backup paths of the configurations with the given identifiers 4. Determine path of the configuration with the newest timestamp before the given timestamp. 5. Send request to AbT to move the configuration specified at step 4 from the backup location to the current location.
Alternative Path:	
Exceptions:	The backuped configuration is not found a. Return ERROR List of Configuration could not be opened b. Return ERROR
Assumptions:	1. AbT logs every action. 2. The request from AbT contains valid identifiers.
Postconditions:	1. Configuration is restored
Related business rules:	-

9.5 Use Case 5

Use Case Name:	**Remove Configuration**
Authors:	RuDev
Dates:	04-12-2006
Iteration:	Focused
Description:	Delete a existing configuration from the system
Actors:	AbT
Preconditions:	configManager has generated a list of all configuration files
Triggers:	AbT submits a request for deleting a configuration with the given identifiers
Basic Course of Events:	1. AbT submits a request for deleting a configuration with the given identifiers 2. Open list of all configuration files 3. Search the list for the path of the configuration containing the given identifiers. 4. Send request to AbT to delete the configuration from the path given at step 3
Alternative Path:	None
Exceptions:	Configuration is not found a. Return ERROR. List of Configuration could not be opened b. Return ERROR
Assumptions:	1. AbT logs every action. 2. The request from AbT contains valid identifiers. 3. AbT takes care of any backup processes if necessary
Postconditions:	1. Configuration is removed
Related business rules:	-

9.6 Use Case 6

Use Case Name:	**Edit Configuration**
Authors:	RuDev
Dates:	04-12-2006
Iteration:	Focused
Description:	Edit variables in a current configuration
Actors:	AbT
Preconditions:	Configuration to be edited is specified Variables to be edited are specified configManager has generated a list of all configuration files
Triggers:	AbT submits a request for editing a value within a specified configuration
Basic Course of Events:	1. AbT submits a request for editing a value within a specified configuration 2. Open list of all configuration files 3. Search the path of the configuration with given identifiers 4. Open the configuration from the path given at step 3 5. Search for the variables keys 6. Return request to AbT with the information from step 5 to edit the variables 7. Return request to Abt to store the configuration.
Alternative Path:	None
Exceptions:	Configuration not found a. Return ERROR Key of a variable in configuration is not found. b. Return ERROR List of Configuration could not be opened c. Return ERROR

Assumptions:	1. Keys and values are valid specified. 2. AbT logs every action. 3. AbT has access to filesystem to read and write the configuration 4. AbT takes care of any backup processes if necessary
Postconditions:	1. Configuration edited
Related business rules:	-

9.7 Use Case 7

Use Case Name:	**Generate List of Configurations**
Authors:	RuDev
Dates:	04-12-2006
Iteration:	Focused
Description:	Generate a list containing every configuration from the system
Actors:	AbT
Preconditions:	configManager has access to read the entire file system
Triggers:	AbT submits a request to generate a list of all configurations.
Basic Course of Events:	1. AbT submits a request to generate a list of all configurations. 2. Search the entire file system for configuration files 3. Collect name, timestamp, version and path information of every configuration 4. Group the information to a list 5. Return the list to AbT
Alternative Path:	None
Exceptions:	Configurations not found a. Return ERROR Configuration can not be opened b. Return ERROR
Assumptions:	1. AbT logs every action. 2. The type: 'configuration files' can be found by a identifier 3. Name, timestamp, version and path information can be accessed by configManager
Postconditions:	1. A list with all configurations is returned.
Related business rules:	-

9.8 Use Case 8

Use Case Name:	**Find Configuration**
Authors:	RuDev
Dates:	04-12-2006
Iteration:	Focused
Description:	Search the entire system for a configuration identified by identifiers given by AbT
Actors:	AbT
Preconditions:	configManager has generatad a list of all configuration files
Triggers:	AbT submits a request for searching a configuration containing the given identifiers
Basic Course of Events:	1. AbT submits a request for searching a configuration containing the given identifiers. 2. Open list of all configuration files. 3. Search in the list for the configuration with the given identifiers. 4. Return AbT the path of the configuration file.
Alternative Path:	None
Exceptions:	List of Configuration could not be opened a. Return ERROR Configuration is not found a. Return ERROR
Assumptions:	1. AbT logs every action. 2. The request from AbT contains valid identifiers.
Postconditions:	1. The path of the configuration matching the identifiers is returned.
Related business rules:	-

10 Scenarios

10.1 Scenario Use Case 1

Use Case Name:	**Install Configuration**
Use Case Steps:	1. configManager retrieves a request for installing the configuration of foo1 v.3.2. 2. configManager opens list of configurations 3. configManager searches the path path1 of last version of foo1 v3.1 4. configManager sends request to AbT to backup the old configuration foo1 v.3.1 with the path path1 5. configManager sends request to AbT to store the new configuration foo1 v.3.2 at path path1
Alternative Path:	1. configManager retrieves a request for installing the configuration of foo1 v.3.2. 2. configManager opens list of configurations 3. configManager cant find last version of foo1 v.3.2 4. configManager sends request to AbT to store the new configuration foo1 v.3.2 at path given from AbT

10.2 Scenario Use Case 2

Use Case Name:	Upgrade Configuration
Use Case Steps:	1. configManager retrieves a request for upgrading the current configuration of foo1 v.3.1 with a the new configuration foo1 v.3.2 2. configManager opens list of configurations 3. configManager searches the path path1 of current version of foo1 v.3.1 4. configManager opens configuration foo1 v.3.1 at path path1 5. configManager compares variables from current configuration foo1 v.3.1 with the new configuration foo1 v.3.2 6. configManager makes backup of current configuration foo1 v.3.1 with path path 1 7. configManager sends request to AbT to add the variables var12, var13 of the new configuration foo1 v.3.2 to the current configuration foo1 v.3.1 and delete the variables var18, var19 from the current configuration foo1 v.3.1 8. configManager sends request to AbT to update the version number of foo1 v.3.1 to version number foo1 v.3.2
Alternative Path:	1. configManager retrieves a request for upgrading the current configuration of foo1 v.3.1 with a the new configuration foo1 v.3.2 2. configManager opens list of configurations 3. configManager searches the path path1 of current version of foo1 v.3.1 4. configManager opens configuration foo1 v.3.1 at path path1 5. current configuration foo1 v.3.1 is equal with the new configuration foo1 v.3.2

10.3 Scenario Use Case 3

Use Case Name:	**Backup Configuration**
Use Case Steps:	1. configManager retrieves a request for making backup of configuration foo1 v.3.2. 2. configManager opens list of configurations 3. configManager searches the path path1 of foo1 v3.2. 4. configManager sends request to AbT to backup the configuration foo1 v.3.2 with the path path1 to the backup location 5. configManager sends request to AbT to add current timestamp to foo1 v.3.2 at backup location
Alternative Path:	1. configManager retrieves a request for making backup of configuration foo1 v.3.2 and foo2 v.1.2. 2. configManager opens list of configurations 3. configManager searches the path path1 of foo1 v3.2 and path path2 of foo2 v.1.2 4. configManager sends request to AbT to backup the configuration foo1 v.3.2 with the path path1 and foo2 v.1.2 with path path2 to the backup location 5. configManager sends request to AbT to add current timestamp to foo1 v.3.2 and foo2 v.1.2 at backup location

10.4 Scenario Use Case 4

Use Case Name:	**Restore Configuration**
Use Case Steps:	1. configManager retrieves a request for restoring the configuration of foo1 v3.2 before timestamp 1165543200 2. configManager opens list of configurations 3. configManager searches the paths path1, path2, path3 of configuration foo1 4. configManager retrieves path path3 of configuration foo1 based on latest version before timestamp 1165543200 5. configManager sends request to AbT to move configuration foo1 with path path3 from the backup location to the current location
Alternative Path:	

10.5 Scenario Use Case 5

Use Case Name:	**Remove Configuration**
Use Case Steps:	1. configManager retrieves a request for deleting the configuration of foo1 v3.2 2. configManager opens list of configurations 3. configManager searches the path path1 of of foo1 v.3.2 4. configManager sends request to AbT to remove configuration foo1 v.3.2 from path path1
Alternative Path:	1. configManager retrieves a request for deleting the configuration of foo1 v3.2 2. configManager opens list of configurations 3. configManager searches the path path1 and path path2 of of foo1 v.3.2 4. configManager sends request to AbT to remove configuration foo1 v.3.2 from path path1 and path path2

10.6 Scenario Use Case 6

Use Case Name:	Edit Configuration
Use Case Steps:	1. configManager retrieves a request for editing the variable var15 of the configuration foo1 v.3.2 2. configManager opens list of configurations 3. configManager searches the path path1 of configuration of foo1 v.3.2 4. configManager opens configuration foo1 v.3.2 at path path1 5. configManager searches for the key key15 of variable var15 6. configManager sends request to AbT to edit variable var15 for configuration foo1 v.3.2 at path path1 7. configManager sends request to AbT to store the configuration foo1 v.3.2
Alternative Path:	1. configManager retrieves a request for editing the variable var15, variable var 18 and variable var 19 of the configuration foo1 v.3.2 2. configManager opens list of configurations 3. configManager searches the path path1 of configuration of foo1 v.3.2 4. configManager opens configuration foo1 v.3.2 at path path1 5. configManager searches for the key key15 of variable var15, the key key18 of variable var18 and the key key19 of variable var19 6. configManager sends request to AbT to edit variables var15, var18 and var19 for configuration foo1 v.3.2 at path path1 7. configManager sends request to AbT to store the configuration foo1 v.3.2

10.7 Scenario Use Case 7

Use Case Name:	Generate List of Configurations
Use Case Steps:	1. configManager retrieves a request to generate list lst1 of all configurations 2. configManager searches file system for configurations 3. configManager adds name foo1, timestamp t1, version v.3.1, path path1 and name foo1, timestamp t2, version v.3.2, path path2 to list lst1 4. configManager groups name foo1, timestamp t1, version v.3.1, path path1 and name foo1, timestamp t2, version v.3.2, path path2 in list lst1 5. configManager returns list lst1 to AbT
Alternative Path:	1. configManager retrieves a request to generate list lst1 of all configurations 2. configManager searches file system for configurations 3. configManager adds name foo1, timestamp t1, version v.3.1, path path1, name foo1, timestamp t2, version v.3.2, path path2, name foo2, timestamp t3, version v.1.2 and path path3 to list lst1 4. configManager groups name foo1, timestamp t1, version v.3.1, path path1, name foo1, timestamp t2, version v.3.2, path path2, name foo2, timestamp t3, version v.1.2 and path path3 in list lst1 5. configManager returns list lst1 to AbT

10.8 Scenario Use Case 8

Use Case Name:	**Find Configuration**
Use Case Steps:	1. configManager retrieves a request for searching configuration foo1 v.3.2 2. configManager opens list lst1 of configurations 3. configManager searches for foo1 v.3.2 in list lst1 4. configManager returns path path1 of foo1 v.3.2
Alternative Path:	1. configManager retrieves a request for searching configuration foo2 v.1.2 2. configManager opens list lst2 of configurations 3. configManager searches for foo2 v.1.2 in list lst1 4. configManager returns path path2 of foo2 v.1.2

11 Domain Models

11.1 Domain Model: Use Case 1

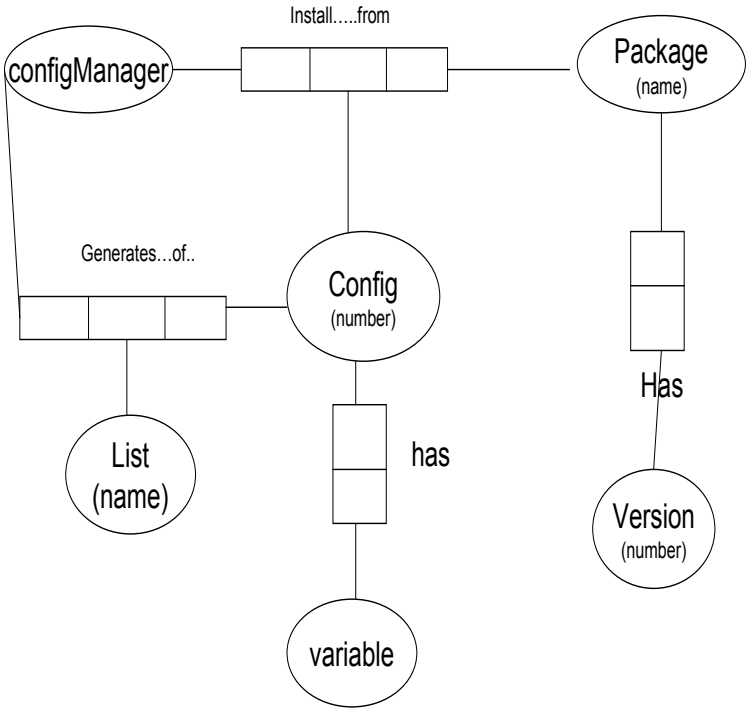


Figure 8: Use Case 1

11.2 Domain Model: Use Case 2

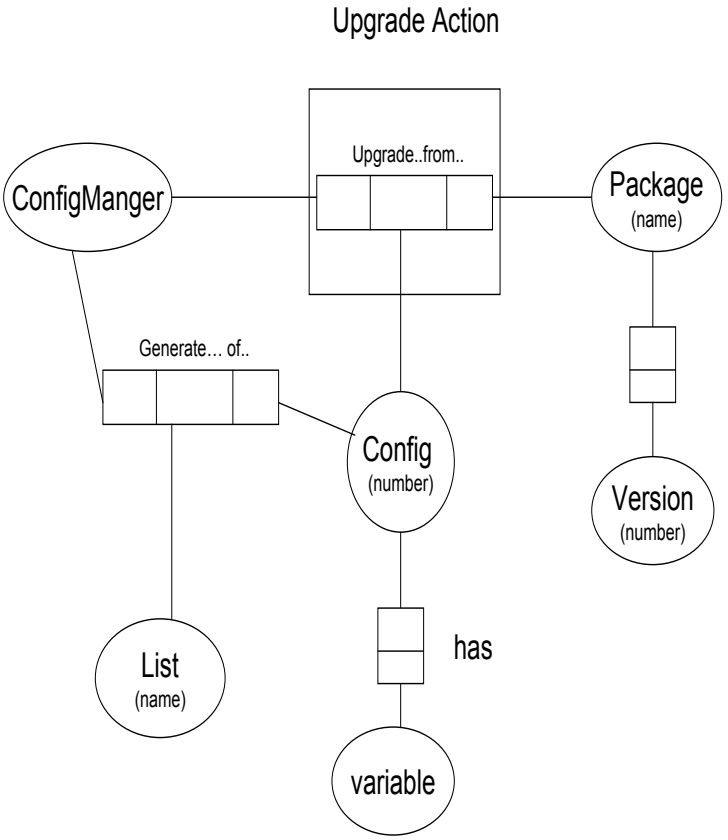


Figure 9: Use Case 2

11.3 Domain Model: Use Case 3

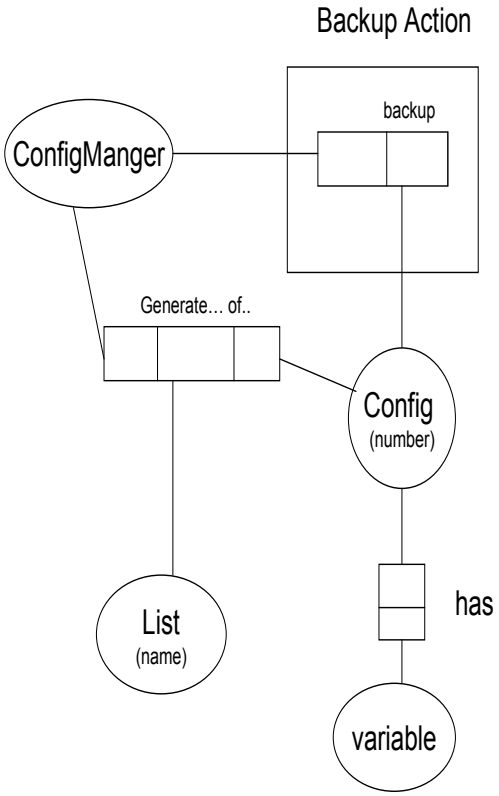


Figure 10: Use Case 3

11.4 Domain Model: Use Case 4

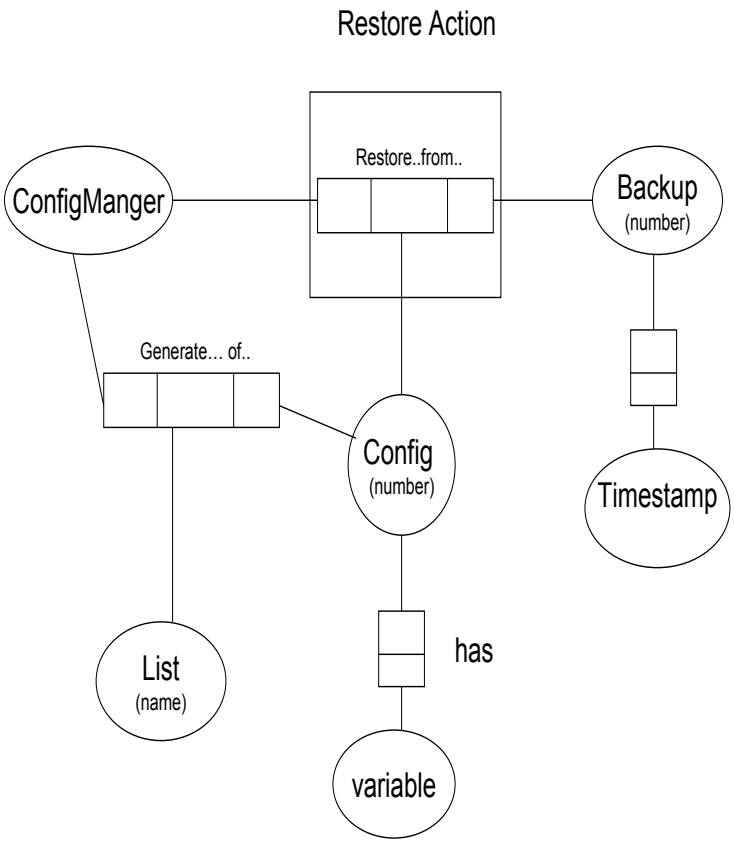


Figure 11: Use Case 4

11.5 Domain Model: Use Case 5

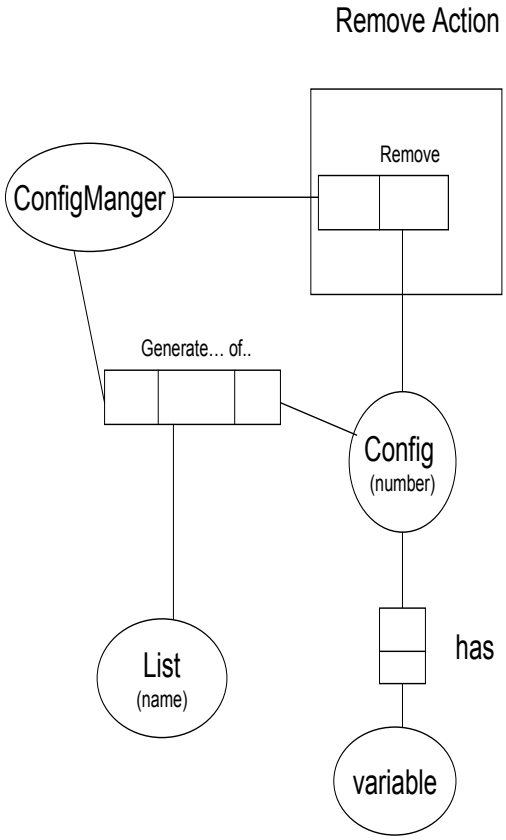


Figure 12: Use Case 5

11.6 Domain Model: Use Case 6

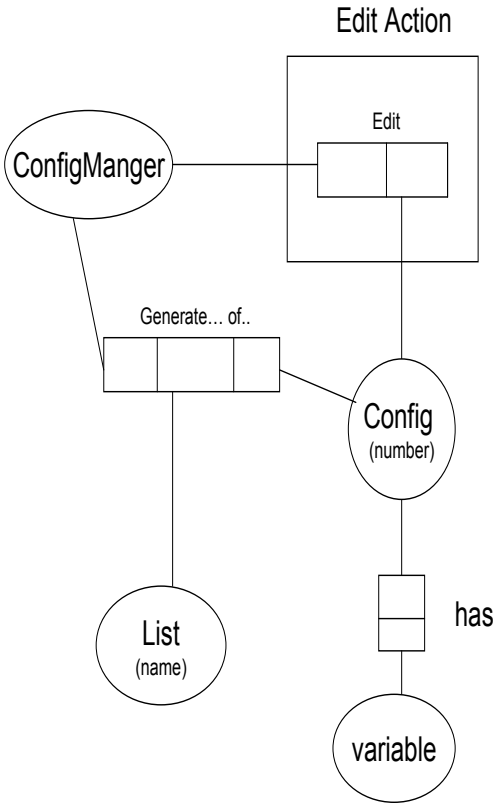


Figure 13: Use Case 6

11.7 Domain Model: Use Case 7

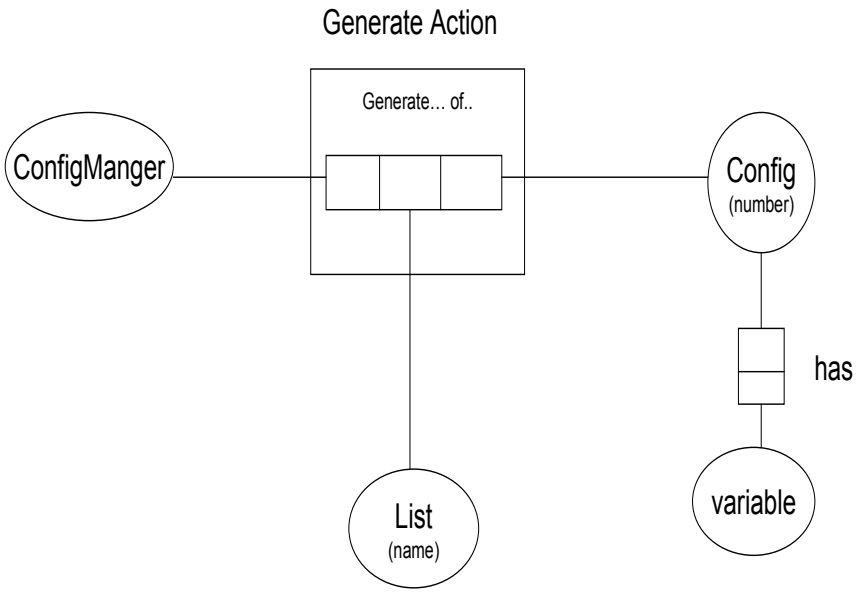


Figure 14: Use Case 7

11.8 Domain Model: Use Case 8

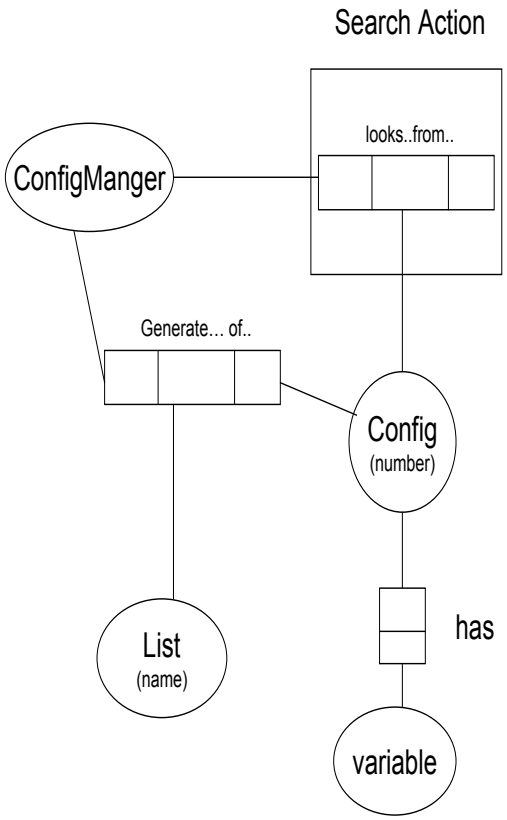


Figure 15: Use Case 8

12 Business Rules

Rule id.	Description	Category	Static/Dynamic	Source
1	The Project Manager describe the problem and maybe a list of needs and present it to the users and the supervisor	Structural Facts	Static	Presentation
2	The members community of ABT-linux have right to use and go into the system.	Action Triggering	Static	Management policy

13 Non-functional Requirements

- IT must be possible for the users to use the system at the same time
- The system must fulfil in 1-10 sec the desires of the end-users
- The system must able to up to date sometimes
- Ability for the end-users to change different aspects of the system easily
- The system must be easy to use in different or in the most necessary platforms
- Security requirement to ensure confidentiality and authentication
- Security requirement to ensure that users can access only to some functions with in the system

14 Terminological Definitions

Term	Explanation
Abt	The name of the software package management system in this project.
AbTLinux	The name of the software package management system in this project.
Requirement	A characteristic of a system in order to be acceptable to the acquirer.
configManager	The configuration engine, part of the tool set used by abt to control the configuration management.
Configuration	An arrangement of functional unit(s) containing information about their nature, number, and chief characteristics.
Variable	A key-value combination within a configuration defining the characteristics of the configuration.
Timestamp	Is a unix timestamp, the number of seconds since 00:00:00 UTC on January 1, 1970.
Configuration File	A single file, containing variables, which is part of a configuration .
Package	A complete set of data used in the Software Package Management System (SPMS) to manage all aspects of using this single piece of software.
Path	The fully specified name of a configuration, including the position of the configuration in the file system's directory structure.