

Maintenance document

PULSU

Fall 2008
Software Engineering Project

UNIVERSITY OF HELSINKI
Department of Computer Science

Course

581260 Software Engineering Project (9 cr)

Project group

Heikki Manninen
Noora Joensuu
Sami Vuorivirta
Joel Kaasinen
Erno Liukkonen

Client

Peter von Etter
Roman Yangarber

Instructor

Peter von Etter

Homepage

<http://www.cs.helsinki.fi/group/pulsu/>

Version

Version	Date	Description
1.0	Dec 11 2008	Ready

Table of contents

1. Introduction.....	1
2. Glossary.....	1
3. Architecture.....	2
3.1 User interface	2
3.2 Subsystems.....	2
4 User interface.....	4
4.1 Unified case.....	4
4.2 Document	4
4.3 Plain case.....	5
4.4 Map	5
4.5 Help.....	5
5 Components	7
5.1 GUI.....	7
5.2 JS.....	12
5.3 Controller	14
6. Unrealized requirements	19
7. Maintenance of code	19
7.1 Adding new scenarios	19
8. Other maintenance.....	19
8.1 Database	19
9. Other	19
9.1 Used libraries and plug-in softwares.....	19

1. Introduction

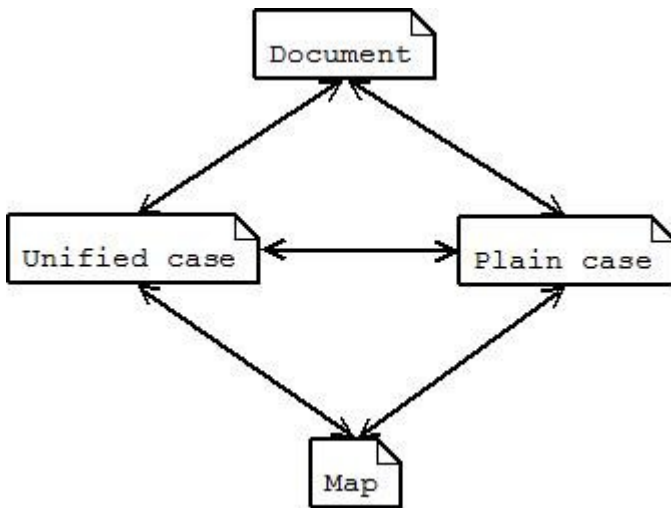
This is the maintenance document of the software engineering group PULSU that is taking part to software engineering course on fall 2008.

2. Glossary

Administrator	Systems Administrator who has enhanced search functionality in System, which helps to do work better.
Expert	A user who is an expert of scenario, for example in disease scenario expert is doctor. He or she has privileges to modify cases when logs into system.
Average user	Anyone who uses the system. He or she does not necessarily have any experience of scenario.
Domain	Tells what particular domain this user interface is supporting. Examples of domains: Medical and financial
Scenario	Tells what particular scenario this user interface is supporting. Examples of scenarios: Diseases, investments
View	Individual view shows information from certain aspect. Examples of views: Overview, table, enhanced search
Map	Map which includes marks where different diseases have been found.
Charts	Charts which show information about diseases, for example which diseases have recently been most active in documents.
Document	Document which has been found from internet. Individual document may contain many cases.
Case (IE event)	Fact, which has been collected from document by the system. Doctors use word "case" because word "event" means the whole process of a disease for them. Only on medical domain.
Unified case	Group of cases which have similar attributes (Level 1).
Group	Group of unified cases which are related to same disease in same country on period of two weeks (Level 2).
Plain cases	Cases without any grouping (Level 0).
Attribute	Disease, country, time, total, status etc. Disease: Tells the name of disease in this case. Country: Tells the name of country in this case. Time: Tells when this case happened. Total: Tells how many people have got disease. Status: Tells the current situation of disease.
Search	Basic search which all users may use.
Enhanced search	A extended search which is not necessarily available for average users. All other users may use this search functionality to define better what they want to search.
Hierarchical browsing	User may for example search only documents that are from certain continent.
Help	Help information for all users.

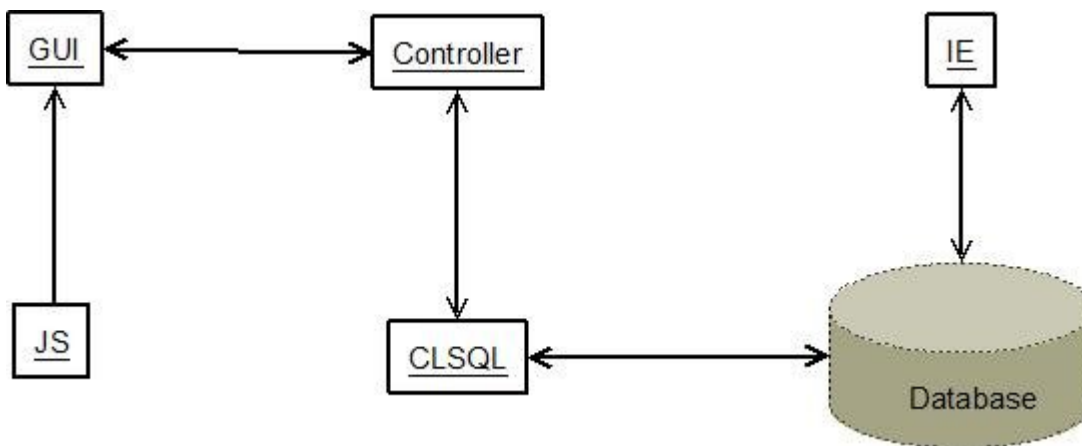
3. Architecture

3.1 User interface



In addition is Help-page. User interface more specific in chapter 4.

3.2 Subsystems



GUI defines pages for web application and components of which these are built.

JS includes javascripts.

Controller offers interface to the database for GUI.

CLSQL is library of third party which realize above-mentioned.

IE is natural language processing system which is used for extract facts from unstructured texts, and stores them in database.

Database is database where above-mentioned facts are stored.

4 User interface

4.1 Unified case

There is a list of 25 newest unified cases in the page. About every unified case is told disease and country as well as a title of the document of the newest case. By clicking the title user gets to the Document page of the case. In front of the unified cases, to which belongs more than one case, are number of cases which belong to it and Triangle button. By clicking it user brings out all cases of the unified case. By clicking it again other cases hide themselves. The list also hides itself automatic if user clicks Triangle button of some other unified case. On top of the page is Info button which brings out and hides sentences from where triggers of cases were found.

In front of verified cases are checked boxes and in front of unverified cases empty boxes. If all cases of unified case are verified then there is also checked box in front of unified case. Whereas if one or more cases of unified case are unverified then in front of unified case is empty box.

On top of page is simple searchbar to where user can write search parameter to find, for example, cases which deal with certain disease. When needed he or she can also clicks Advanced search link and make searches more specific by using filters. User can use up to five filters simultaneously in which case he or she can do OR-searches. This means that user can search unified cases which fulfill at least all conditions of one filter, but not necessarily all filters. Besides searchbar and filters user can choose whether all unified cases are shown or just verified or unverified.

On top and bottom of page is pager. If user clicks, for example, number 2, he or she gets next 25 newest unified case. Numbering keep continued until 30. All numbers aren't shown simultaneously but shown numbers depend on what is number of current page.

Requirements	Implementation
T1	List of unified cases.
T2	Title of the document of the case and phrase where triggers of the case were found.
U1	Searchbar and filters.

4.2 Document

In the middle of page is the document from where the case was found. This case is bold on the document text and its triggers are colored.

All cases of the unified case or the document are listed to left side of the page, and in front of these are checkboxes. Above these cases is a link to cases of the unified case or document depending on which ones are currently shown. By clicking it the list of cases of the unified case changes to list of document cases and vice versa. By clicking cases user can easily read all cases which belong to the unified case or the document. There is also the currently shown case but instead of link it's just bold.

On the right side of page are attributes of the case. If user thinks values of attributes are correct he or she can verify the case by clicking Accept case button. If they aren't, user can first correct them and then click above-mentioned button.

If user wants to modify some value of attribute he or she clicks Pen button and the fixed text field of the value of the attribute changes to dropdown and the user can see the listed suitable values to this attribute found from the document. If neither of these values doesn't please user he or she can click the Pen button again and the dropdown changes to text field where he or she can write some other value. By clicking button once again field changes back to fixed text field. Below the attributes, user can see how many values of attributes are changed and to how many cases these changes will affect.

User can also remove case clicking Reject case button or create a new case clicking New case button when values of attributes changes to dropdowns or text field depending on type of the attribute. Lastly he or she clicks Make new case button when the new case is saved to the database.

If user wants to modify many cases simultaneously he or she can marks checkboxes of these cases from left side of the page. Then changing value of attribute will affect to all these cases.

Requirements	Implementation
T3	Text of the document and Document cases list.
T4	Unified case list.
T5	Attributes of the case.
T6	Modifying values of attributes of the case by using dropdowns and text fields.
T7	Creating a new case.
T8	Removing of the case.
U4	Checkboxes of cases and modifying values of attributes.
U5	No implementation.
U6	No implementation.

4.3 Plain case

There is a list of 25 newest cases without any grouping in the page. Otherwise the page is similar to Unified cases.

Requirements	Implementation
T2	Title of the document of the case and phrase where triggers of the case were found.
U1	Searchbar and filters.

4.4 Map

There is a map in the page. The map is implemented by Google Maps.

Requirements	Implementation
U1	Searchbar and filters.
U2	Map.

4.5 Help

There is more information about functions of this web application in the page.

5 Components

5.1 GUI

Component	Description of component	Service	Type of service	Description of service	Parameters	Description of parameters	Return value
document-view3	Creates Document view. Requires docustyle.css	render-case-attributes	html	prints div which contains attribute table of the case	row	attributes of the case as list	lista casen triggereit�
					group	id of the unified case	
					page	page number of Unified case or Plain case -page where to the document page is come from	
					back	defines whether to Document page is come from Unified case or Plain case page	
		search-offsets	fun	searches offsets of triggers from the document	words	triggers of the case as list	offsets as list
					docno	document id	
		updatedb	page	sends parameters to the database	docno	document id	
					event_id	case id	
					doc_selected_cases	list of selected cases	
					submitter	string which defines actions for the database so tells which button user has pressed	
					group_number	id of the unified case	

					trigger	text of the trigger	
					page	page number of Unified case or Plain case -page where to the document page is come from	
					back	defines whether to Document page is come from Unified case or Plain case page	
		render-main-content	html	prints div which contains modified text of the document	string	text of the document	
					row	attributes of the case as list	
					rows	other cases of the document as list	
		setselected	page	adds parameters to url and redirect to the page	selected_cases	list of selected cases	
					docno	document id	
					event_id	case id	
					group_number	unified casen id	
					back	defines whether to Document page is come from Unified case or Plain case page	
					page	page number of Unified case or Plain case -page where to the document page is come from	
		document-view	page	prints Document view	n	document id	
					c	case id	

					group_number	id of the unified case			
					selected_cases	list of selected cases			
		render-left-container	html	prints checkbox list of cases	current-n	document id			
					current-c	case id			
					row	attributes of the case as list			
newest	Prints Unified case view and Plain case view. Requires indexstyle.css	newest-unified-cases	page	prints Unified case view (Level 1)	number-of-cases	number of unified cases			
					25				
					page 0	page number			
					search nil	search parameters as list			
							verify true	defines whether verified unified cases are shown or not	
		newest-cases	page	prints Plain case view (Level 0)	number-of-cases	number of cases			
					25				
					page 0	page number			
					search nil	search parameters as list			
							verify	defines whether verified unified cases are shown or not	
		render-mainview	html	print html-page whose content depends on parameters	number-of-cases	number of cases or unified cases			
					page	page number			
view	defines whether page is Unified case or Plain case								

					filter	search parameters as list	
					verify	defines whether verified cases or unified cases are shown or not	
		render-group	html	prints div which contains one title of the unified case and opening list of other cases of the unified case	unified-cases	cases of the unified case as list	
	view				defines whether page is Unified case or Plain case		
	filter				search parameters as list		
	verify				defines whether verified cases or unified cases are shown or not		
		render-cases	html	prints one case	case	attributes of the case as list	
	view				defines whether page is Unified case or Plain case		
	group nil (optional)				unified casen id		
map	Prints Map view	mapview	page	prints Map view	-		
		write-map-kml	fun	creates kml file	-		
		scp-file	fun	copies kml file to Melkki	filename	name of the kml file	

headerfooter	Creates header and footer components. Requires headerstyle.css	header	html	prints header div			
		header-link	html	creates link and adds parameters to url of the link	name	name of the link	
					link	url of the link	
footer	html	prints footer div	-				
searchbar	Creates searchbar and filter component. Requires searchstyle.css	searchbar	html	prints searchbar div	-		
		setsearch	page	adds parameters to url and redirect to the page	search_values	search parameters as list	
					verify	defines whether verified cases or unified cases are shown or not	
help	Creates Help view. Requires indexstyle.css and helpstyle.css	help-page	page	prints Help view	-		
ui	Equipment for views	make-pager	html	prints list of page numbers	current-page	current page number	
					page-param	parameter of current page in url	
					number-of-pages	max number of pages	
		unique-id	fun	generate unique id	prefix "id" (optional)	-	
		make-trigger-link	fun	gets location of trigger of the case and change the trigger to link and calls make-highlight-pair function	row	attributes of the case as list	
		make-strong	fun	gets location of trigger of the case and bold the trigger	row	attributes of the case as list	

		highlight-document	fun	modify text of the document by using two function above	text	text of the document	
					row	attributes of the case as list	
					rows	other cases of the document as list	
		render-attrs	html	gets attributes of the case and calls render-attrs function repeatedly until whole list is gone through	row	attributes of the case as list	
		render-attr	html	prints table row in which are name of attribute in left cell and its value in right cell	attr	name of the attribute	
					row	attributes of the case as list	

5.2 JS

Component	Description of component	Service	Type of service	Description of service	Parameters
docuscript	Functionality of checkboxes and attribute form in Document view	getAttributes	gets values of attributes of the case from attribute form	-	
		selectAll	adds all cases to selected_cases field and sends cases-form forward	-	
		selectNone	removes all cases from selected_cases field and sends cases-form forward	-	
		selectInvert	clears selected_cases field and adds to it cases which weren't there a moment ago and sends cases-form forward	-	
		checkChecked	adds the case whose checkbox called checkbox	checkbox id	

			this function to selected_cases field or removes it from there if it was already there, and sends cases-form forward		
		markChecked	marks checkboxes of cases which were in selected_cases field	-	
		checkAttributes	check how many values of attributes are changed	-	
		checkAttributeValues	checks if values of attributes are acceptable	-	
		setSubmitter	sets name of the button which called this function to submitter field	submitter	
		submit	checks which button called this function and sends attributes-form forward	submitter	button name
showhide	Opening case list in Unified case view. Requires Prototype JavaScript framework.	switchid	brings out or hides cases of the unified case	id	id of the unified case
switch_input	Functionality of Pen button in attribute form. Requires Prototype JavaScript framework.	switch_input	checks which Pen button were pushed and which is state of the attribute field and modifies it	name	name of the attribute
		toText	changes place of value of the attribute to text field	place	name of the attribute
				id	id of the attribute
		toPlain	changes place of value of the attribute to fixed text field	place	name of the attribute
				id	id of the attribute
toDrop	changes place of value of the attribute to dropdown if it is possible for the attribute	id	id of the attribute		
				safe	table from where is checked if it is possible to change the attribute to dropdown

		toNew	modifies attribute form suitable for creating new case	-	
complete	Lists of diseases and countries.	-			
search	Adding and removing searchbar and filters. Requires Prototype JavaScript framework and scripts of script.aculo.us	newSearch	prints filter	scenario	määrittää tämän hetkisen skenaarion
		removeSearch	removes filter	divId	div:in id, jossa poistettava filteri on
		removeSimple	removes searchbar	-	
		addSimple	prints searchbar	-	
		checkArray	checks status of filters	option	määrittää tarkistettavan asian
		getSearchValues	gets values from filters	-	
		resetSearchValues	clears search parameters and sends search_form –form forward	-	

5.3 Controller

Component	Description of component	Service	Type of service	Description of service	Parameters	Description of parameters	Return value
controller	Offers interface to the database for GUI.	connect-to-db	fun	creates connection to the database	host "127.0.0.1" (optional)	server	
		get-event	fun	gets a single case	docno	document id	casen attribuutit listana
					event_id	case id	
		set-event	fun	sets values of attributes of the case to the database	docno	document id	
					event_id	case id	
					row	attributes of the case as list	

		add-event	fun	adds a new case to the document with attributes from row	docno	document id	
					row	attributes of the case as list	
		get-max-event	fun	gets the largest case id of the document	docno	document id	
		reunify-event	fun	finds an another unified case for case and creates a new one if suitable weren't found	docno	document id	
					event_id	attributes of the case as list	
		get-newest-events	fun	gets newest cases	skip 0 (key)	skips skip newest one	cases as list
					n 0 (key)	max. n cases	
					filter (key)	search parameters as list	
		get-document-events	fun	returns all cases of the document	docno	document id	cases as list
		reject-event	fun	moves the case from plus-table to plusdeleted-table	docno	document id	
					event_id	case id	
		get-unified-case	fun	gets cases which belong to the unified case identified by group_number	group_number	id of the unified case	cases as list
show_accepted t (key)	show also verified cases						
filter (key)	search parameters as list						
get-newest-unified-cases	fun	gets newest unified cases	skip 0 (key)	skips skip newest one	unified cases as list		
			n 0 (key)	max. n unified cases			

					show-accepted t (key)	defines whether verified cases is shown or not	
					filter (key)	search parameters as list	
		verify-unified-case	fun	verifies the unified case	group_number	id of the unified case	
					string	verifier	
files	Process texts of documents and auxil and paf files	get-document-text	fun	gets text of the document	docno	document id	text of the document
		get-document-title	fun	gets title of the document	docno	document id	title of the document
		get-trigger-spans	fun	gets triggers of the case	row	attributes of the case as list	locations of triggers as list
		get-trigger-offsets	fun	gets location of the trigger	row	attributes of the case as list	locations of the trigger as list
		get-snippet	fun	returns snippet of the case	row	attributes of the case as list	snippet
		get-document-auxil	fun	forms filepath to auxil file of the document and calls parse-auxil-file function	docno	document id	
		get-document-paf	fun	forms filepath to paf file of the document and calls parse-paf-file function	docno	document id	
scenario	Toteuttaa skenaariogeneerisyden	get-attrs	fun	gets attributes associated	-		names of

with current scenario

attributes as

						list
attr-options	fun	gets options for value of attribute	attr	attribute name	options as list	
			row	attributes of the case as list		
attr-value	fun	gets current value of the attribute	attr	attribute name	value of the attribute	
			row	attributes of the case as list		
start-index	fun	gets starting index (offset in text) of attribute	attr	attribute name	starting index of the attribute	
			row	attributes of the case as list		
end-index	fun	gets ending index (offset in text) of attribute	attr	attribute name	ending index of the attribute	
			row	attributes of the case as list		
attr-id	fun	gets short textual id for the attribute	attr	attribute name	attribute id	
attr-do-offsets	fun	inserts the offset fields for the attributes of row into row, using text as document text	row	attributes of the case as list		
			text	document text		

		get-unified-case-title	fun	generates a title for a list of cases	rows	cases of the unified case as list	title as list
		get-document-diseases	fun	gets diseases which appear in the document	row	attributes of the case as list	diseases as list
		get-document-countries	fun	gets countries which appear in the document	row	attributes of the case as list	countries as list
		get-document-locations	fun	gets locations which appear in the document	row	attributes of the case as list	locations as list
		get-document-times	fun	gets times which appear in the document	row	attributes of the case as list	times as list
		get-document-starts	fun	gets starting times which appear in the document	row	attributes of the case as list	starting times as list
		get-document-ends	fun	gets ending times which appear in the document	row	attributes of the case as list	ending times as list

6. Unrealized requirements

Code	Requirement	Reason for unrealizaton (besides lack of time)
U3	User can use charts to help to get better picture of current situation.	Low priority, map was more important.
U5	On Document view Experts can verify that the unified case has been grouped correctly.	Unclarity in method of implementation.
U6	On Document view Expert can remove the case from the unified case.	Unclarity in method of implementation.
U8	Average user can inform other users about error in a case.	Functions of average user had low priority in project.

7. Maintenance of code

7.1 Adding new scenarios

In a nutshell, to add a new scenario, you should

1. Push a list of attributes for that scenario into `*attr-alist*` (identified with a symbol) (see `scenario.lisp`)
2. Set the `*scenario*` parameter to the symbol chosen for the scenario
3. Modify `reunify-event` in `controller.lisp`
4. Check the sql statements in `controller.lisp`

8. Other maintenance

8.1 Database

The controller component needs the database to have

- A table with the name specified in `*database-table*` in `controller.lisp`. This table needs to have columns for all the attributes specified for the current scenario (see 7.2)
- A table identical to the previous one, but with `deleted` appended to it's name
- A table `*unified*` with contents as specified in the file `unified.sql`

9. Other

9.1 Used libraries and plug-in softwares

Common Lisp libraries:

- `cl-who` html library
- hunchentoot web server
- libraries provided by the PULS project

Prototype JavaScript framework, version 1.6.0.3

script.aculo.us v1.8.2 (controls.js, dragdrop.js, effects.js, scriptaculous.js, slider.js, sound.js, unittest.js)