PLOVDIV UNIVERSITY „PAISII HILENDARSKI“

FACULTY OF MATHEMATICS AND INFORMATICS

CCA Editor in a Nutshell

Table of Contents

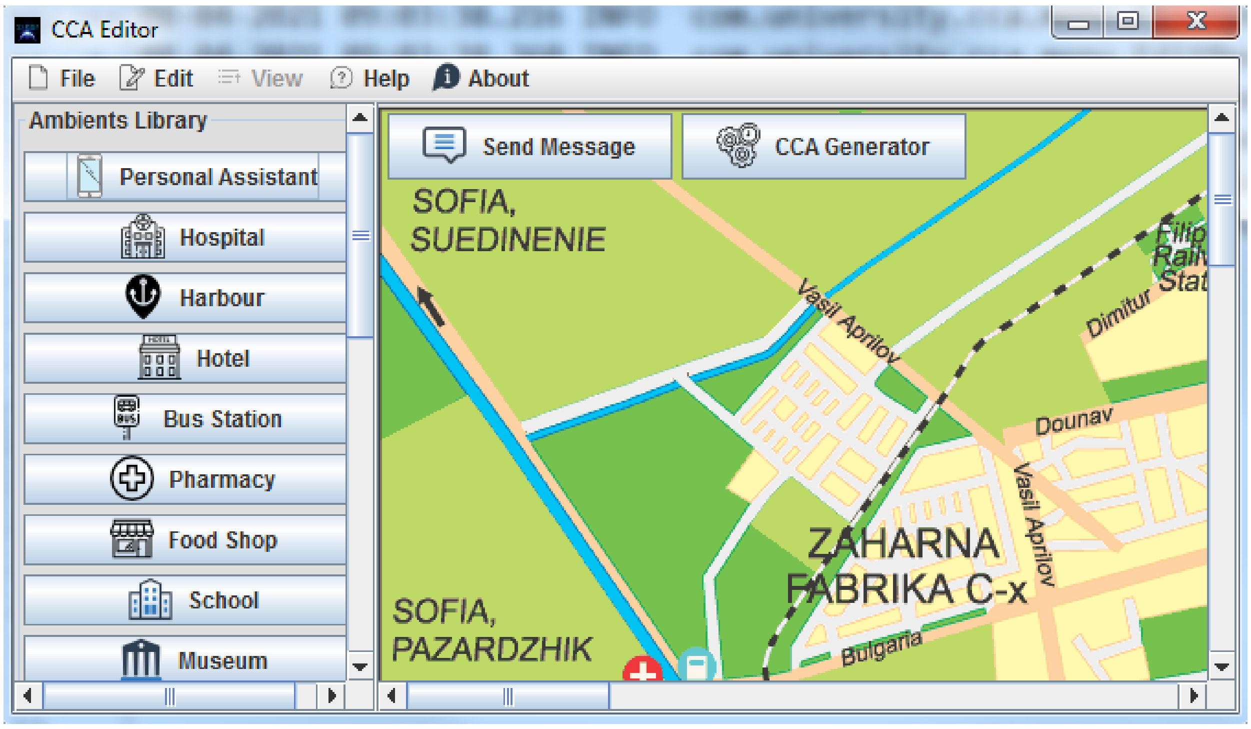
1. Main Screen 3
2. File Menu Item Screen 3
3. Edit Menu Item Screen 5
4. View Menu Item Screen 5
5. Help Menu Item Screen 5
6. Frequently Asked Questions Sub-menu Item Screen 6
7. Terms and Conditions Sub-menu Item Screen 7
8. Tips and Tricks Sub-menu Item Screen 7
9. About Menu Item Screen 8
10. About CCA Sub-menu Item Screen 9
11. About CCA Editor Sub-menu Item Screen 10
12. CCA Editor Team Sub-menu Item Screen 10
13. Useful Links Sub-menu Item Screen 11
14. Technologies Sub-menu Item Screen 12
15. Contact Us Sub-menu Item Screen 13
16. Create a Specific Ambient Screen 14
17. Send Ambient Message Screen 15
18. CCA Generator Screen 16
19. Open CCA File Screen 17
20. CCA Ambients in CSV File 18
21. CCA Ambient Messages in CSV File 18
22. Generated CCA File 19
23. CCA Generation Templates 20

References 22

1. *Main Screen*

The main screen of the CCA editor application is separated into three main sections.

- The main menu of the application, which is located on the top of the application screen (the header of the screen);
- The library with all available ambient elements that you can create, which is located on the left side of the main screen of the application;
- The main section of the screen, which include the map of the Plovdiv, Bulgaria and two buttons: Create Message and Generate CCA (will describe them into the following sections of the document). This main section is located on the right side of the main screen of the application.

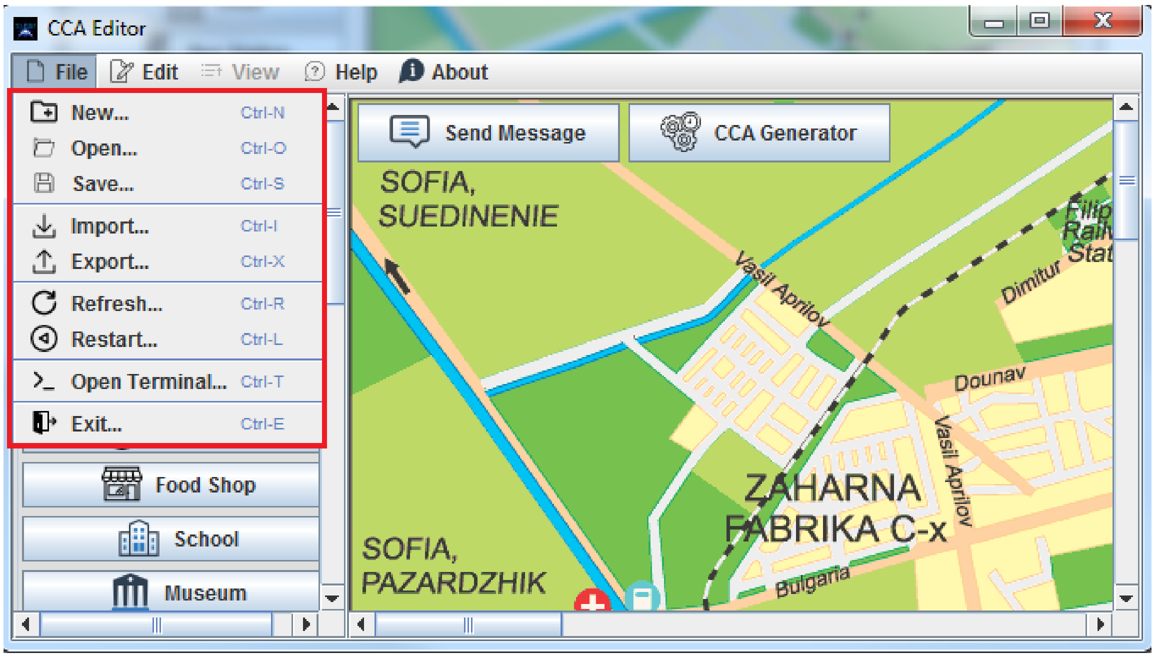


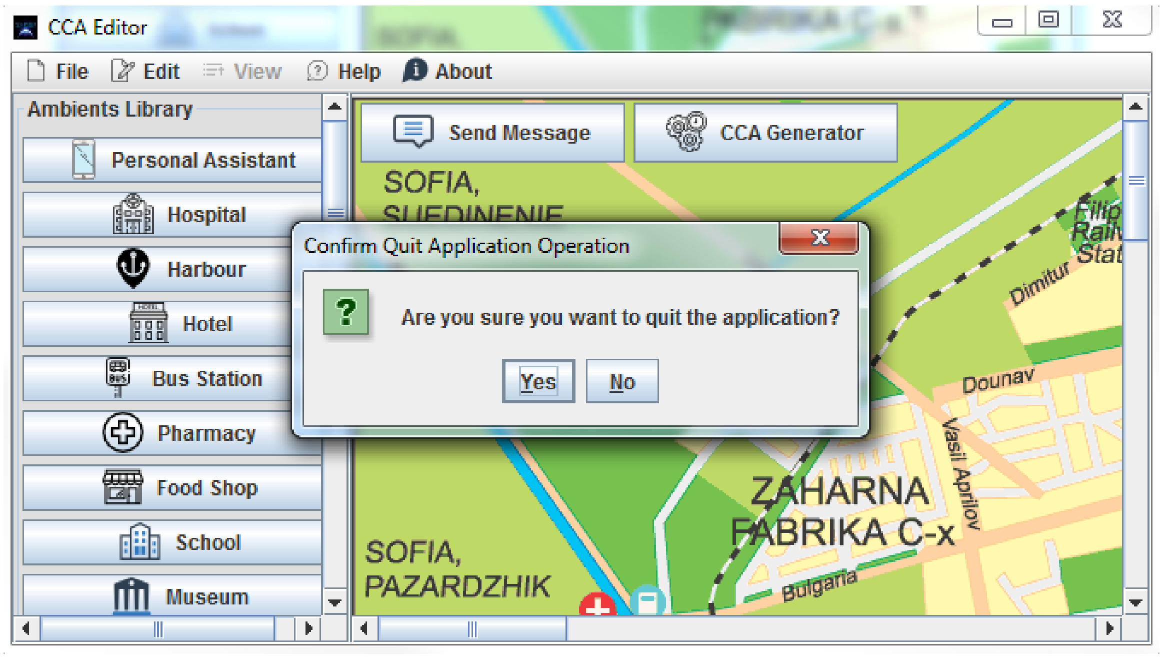
2. *File Menu Item Screen*

The file menu item section includes different sub-menu items with different options which provide a lot of, not so related to the CCA topic, functionalities.

- **New** – not yet implemented
- **Open** – not yet implemented

- **Save** – not yet implemented
- **Import** – not yet implemented
- **Export** – not yet implemented
- **Refresh** – refreshes the application in order to increase the performance of the application
- **Restart** – restart the application
- **Open Terminal** – opens a terminal (console), which you can use in order to execute some commands
- **Exit** – close the application permanently (confirmation dialog is implemented as well)





Confirm Quit Application Operation

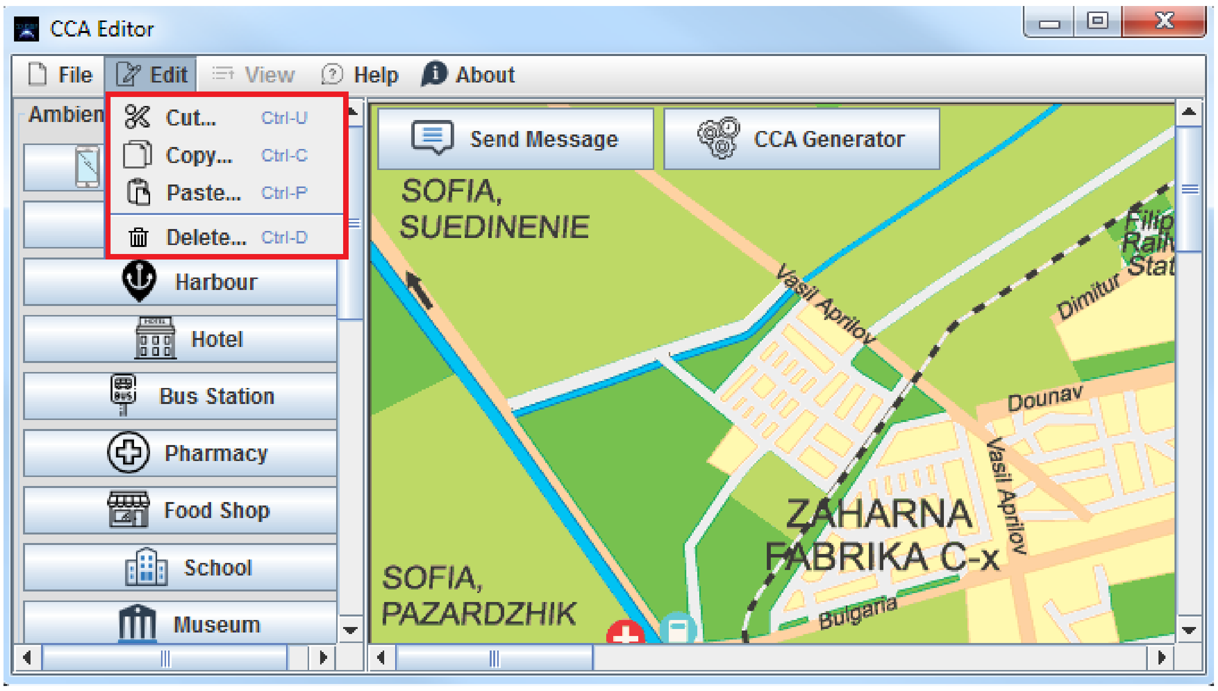
Are you sure you want to quit the application?

Yes No

3. Edit Menu Item Screen

The edit menu item section includes different sub-menu items with different options which provide a lot of, not so related to the CCA topic, functionalities, but also required for each editor.

- **Cut** – not yet implemented
- **Copy** – not yet implemented
- **Paste** – not yet implemented
- **Delete** – not yet implemented



4. View Menu Item Screen

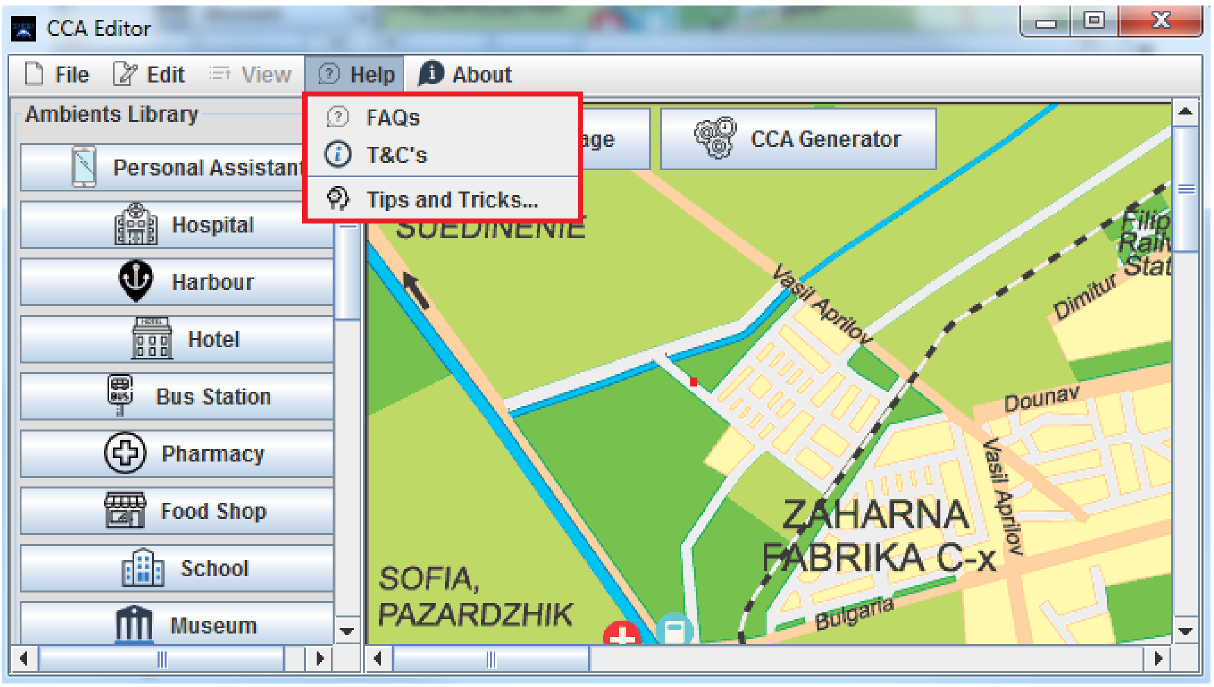
The view menu item section is not yet implemented, that why it's currently disabled. The idea here is to include different sub-menu items with different options, which provide a lot of functionalities, related to the CCA paradigm and helping to create a CCA model successfully.

5. Help Menu Item Screen

The help menu item section includes different sub-menu items with different options which provide a lot of, not so related to the CCA topic, functionalities, but also required to be able to work with the CCA Editor easily. The following sub-menu items are part of the help menu item.

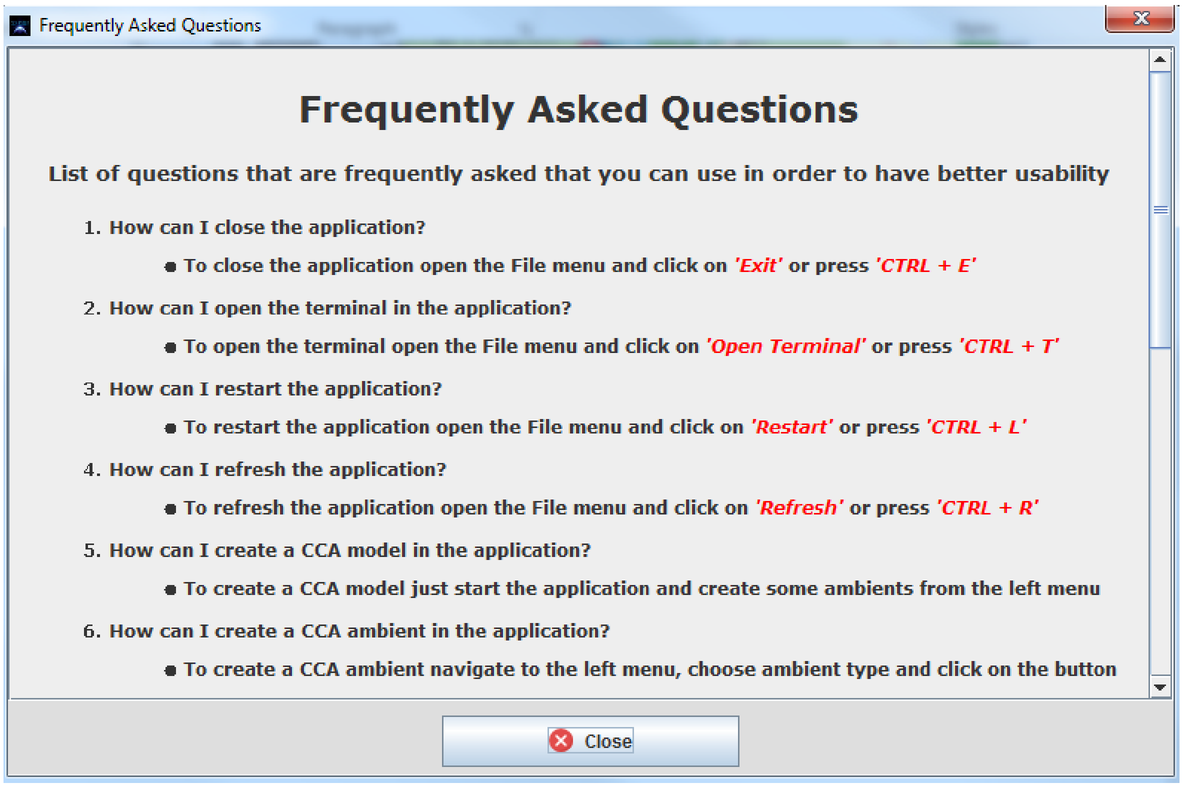
- **Frequently Asked Questions** (FAQs)

- **Terms and Conditions** (T&C's)
- **Tips and Tricks**



6. Frequently Asked Questions Sub-menu Item Screen

Frequently asked questions section contains questions that are already asked from different people during their interaction with the application, and answers of these questions.



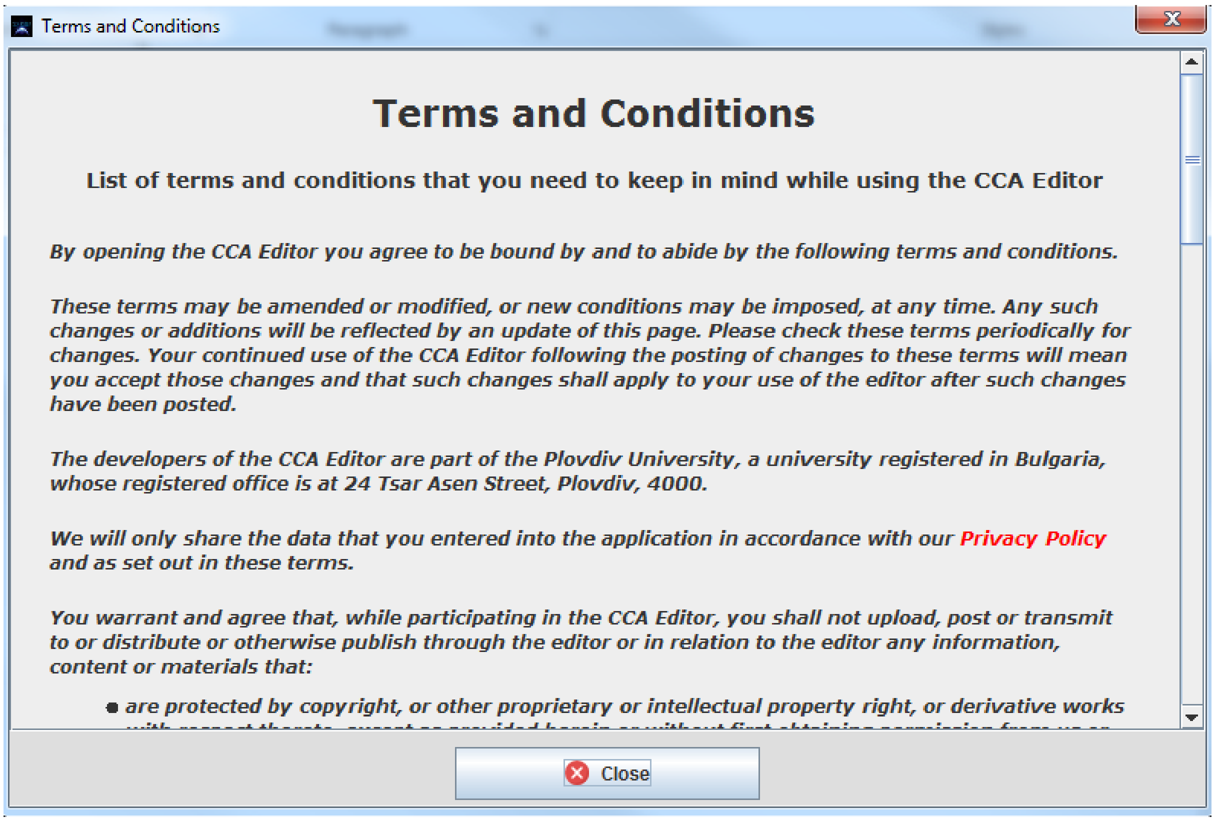
Frequently Asked Questions

List of questions that are frequently asked that you can use in order to have better usability

1. **How can I close the application?**
   - **To close the application open the File menu and click on *'Exit'* or press *'CTRL + E'***
2. **How can I open the terminal in the application?**
   - **To open the terminal open the File menu and click on *'Open Terminal'* or press *'CTRL + T'***
3. **How can I restart the application?**
   - **To restart the application open the File menu and click on *'Restart'* or press *'CTRL + L'***
4. **How can I refresh the application?**
   - **To refresh the application open the File menu and click on *'Refresh'* or press *'CTRL + R'***
5. **How can I create a CCA model in the application?**
   - **To create a CCA model just start the application and create some ambients from the left menu**
6. **How can I create a CCA ambient in the application?**
   - **To create a CCA ambient navigate to the left menu, choose ambient type and click on the button**

7. Terms and Conditions Sub-menu Item Screen

Terms and conditions section of the application includes a lot of rules that each user of the CCA Editor should keep in mind while using the application. Each user should read the terms and conditions that are provided here before start using the application and create CCA models.



Terms and Conditions

List of terms and conditions that you need to keep in mind while using the CCA Editor

By opening the CCA Editor you agree to be bound by and to abide by the following terms and conditions.

These terms may be amended or modified, or new conditions may be imposed, at any time. Any such changes or additions will be reflected by an update of this page. Please check these terms periodically for changes. Your continued use of the CCA Editor following the posting of changes to these terms will mean you accept those changes and that such changes shall apply to your use of the editor after such changes have been posted.

The developers of the CCA Editor are part of the Plovdiv University, a university registered in Bulgaria, whose registered office is at 24 Tsar Asen Street, Plovdiv, 4000.

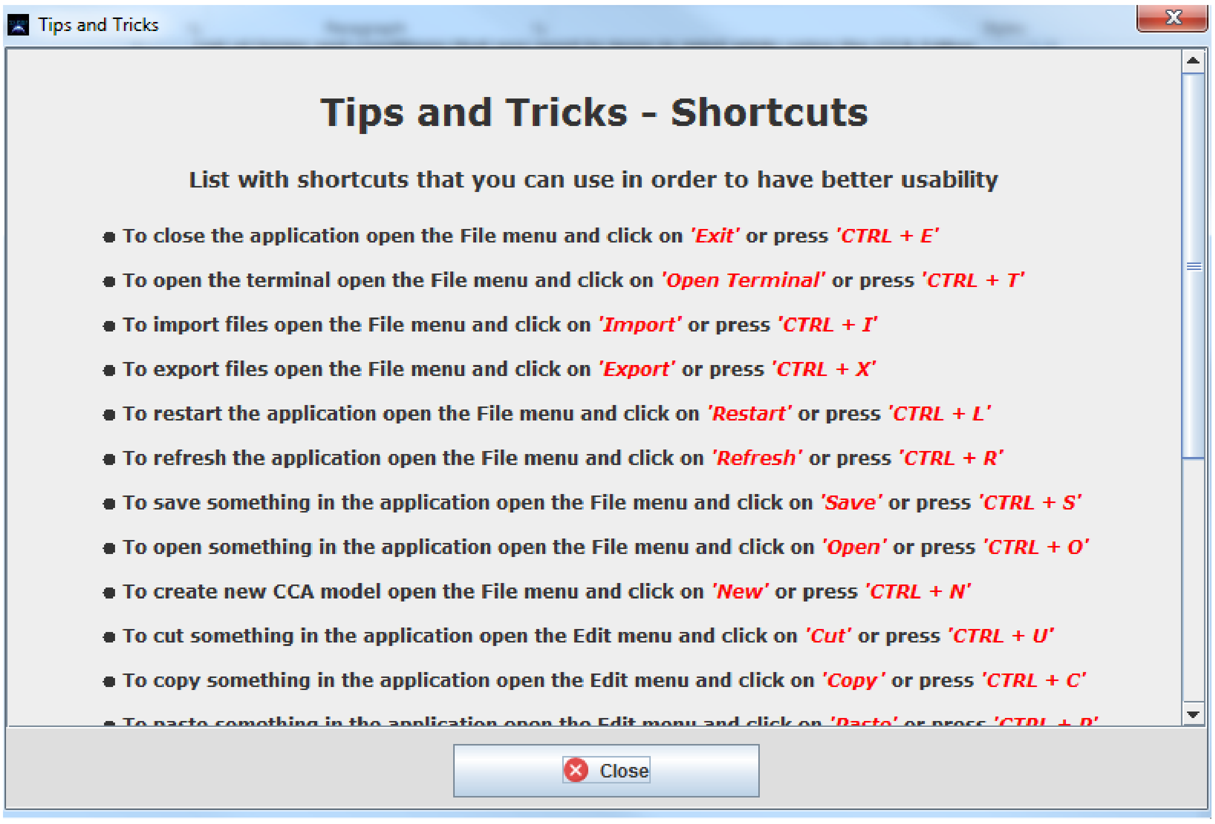
We will only share the data that you entered into the application in accordance with our Privacy Policy and as set out in these terms.

You warrant and agree that, while participating in the CCA Editor, you shall not upload, post or transmit to or distribute or otherwise publish through the editor or in relation to the editor any information, content or materials that:

- *are protected by copyright, or other proprietary or intellectual property right, or derivative works*

8. Tips and Tricks Sub-menu Item Screen

The tips and tricks section of the CCA Editor application includes some shortcuts and key combinations in order to make the interaction between the user and the application faster. Also these key combinations increase the usability of the application. You can see some of the shortcuts and key combinations on the picture below, which is a snippet from the CCA Editor application.


Tips and Tricks - Shortcuts

List with shortcuts that you can use in order to have better usability

- To close the application open the File menu and click on *'Exit'* or press *'CTRL + E'*
- To open the terminal open the File menu and click on *'Open Terminal'* or press *'CTRL + T'*
- To import files open the File menu and click on *'Import'* or press *'CTRL + I'*
- To export files open the File menu and click on *'Export'* or press *'CTRL + X'*
- To restart the application open the File menu and click on *'Restart'* or press *'CTRL + L'*
- To refresh the application open the File menu and click on *'Refresh'* or press *'CTRL + R'*
- To save something in the application open the File menu and click on *'Save'* or press *'CTRL + S'*
- To open something in the application open the File menu and click on *'Open'* or press *'CTRL + O'*
- To create new CCA model open the File menu and click on *'New'* or press *'CTRL + N'*
- To cut something in the application open the Edit menu and click on *'Cut'* or press *'CTRL + U'*
- To copy something in the application open the Edit menu and click on *'Copy'* or press *'CTRL + C'*
- To paste something in the application open the Edit menu and click on *'Paste'* or press *'CTRL + P'*

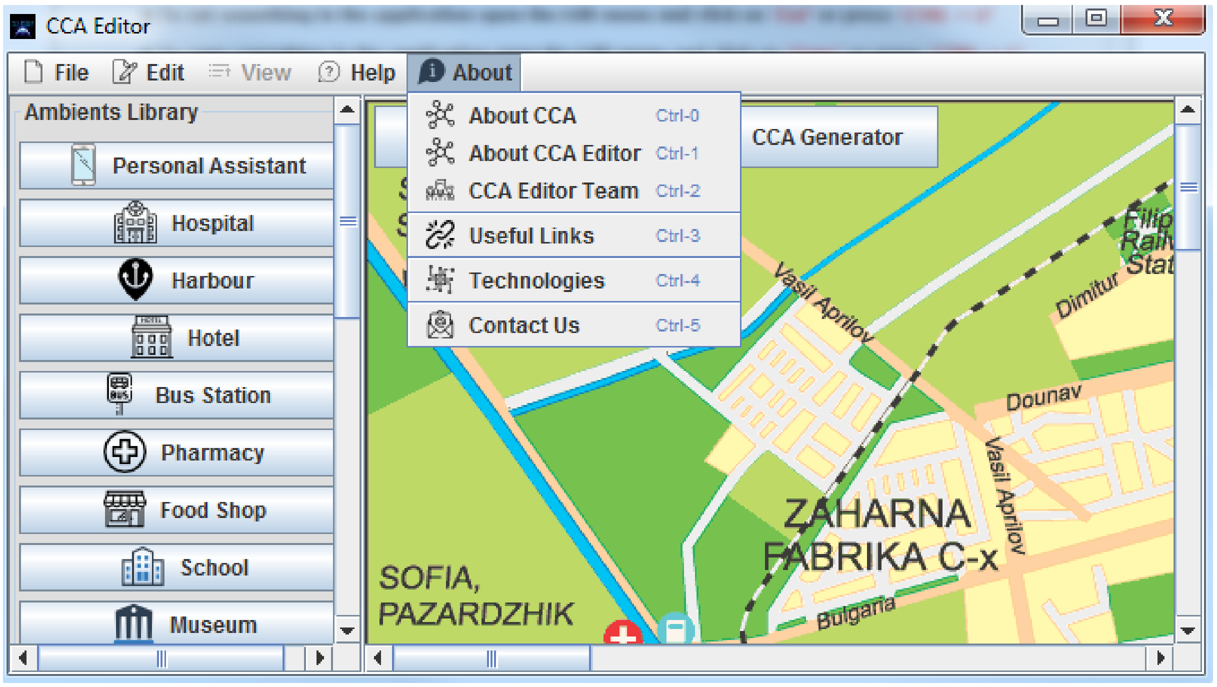
9. *About Menu Item Screen*

The about menu item section includes different sub-menu items with different options which provide a lot of, not so related to the CCA topic, functionalities, but also required to be able to work with the CCA Editor easily. The following sub-menu items are part of the about menu item.

- About CCA
- About CCA Editor
- CCA Editor Team
- Useful Links
- Technologies
- Contact us form

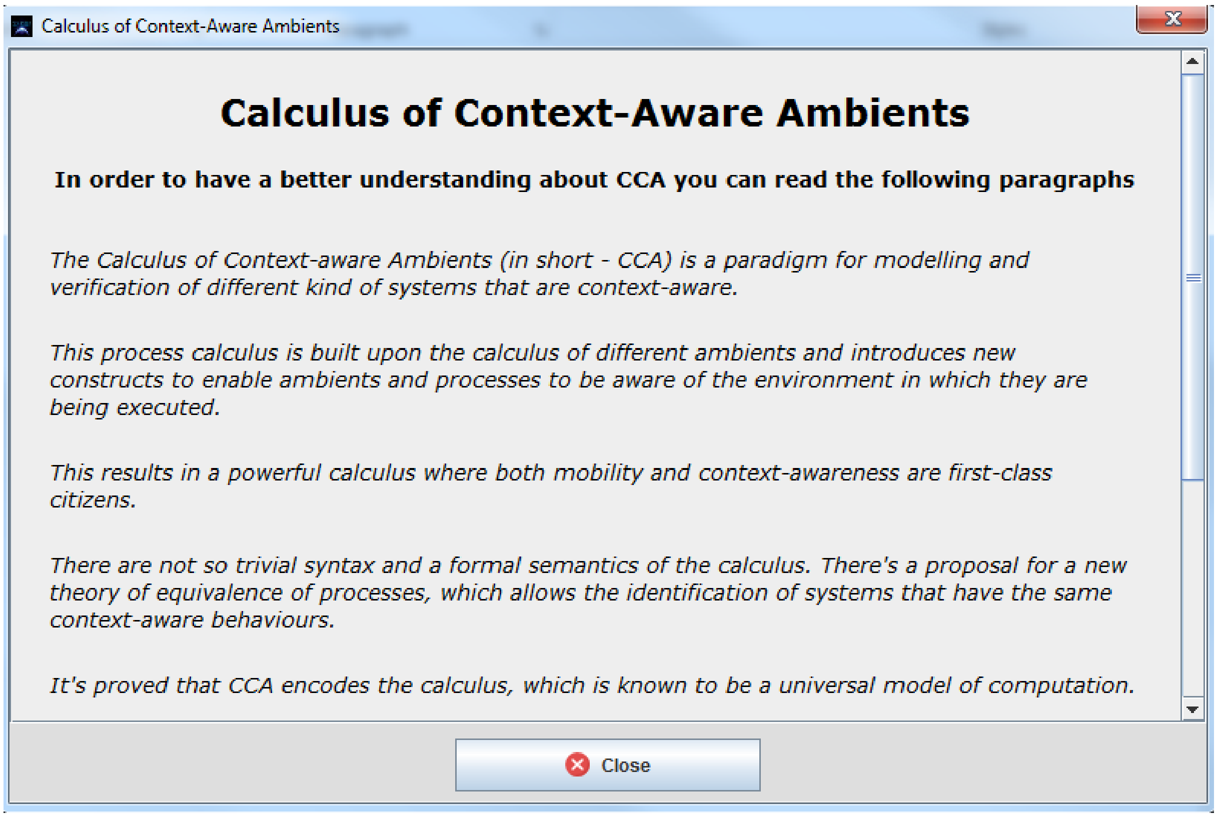
Each sub-menu item of the about menu option will be fully described into the following sections of the current document.

You can see how the about menu item and its sub-menus look like on the picture provided below.



10. About CCA Sub-menu Item Screen

The about CCA sub-menu item section includes basic information about the CCA paradigm and architecture. Also has an information about the ccaPL language, which is used to develop a CCA model. You can see how this section looks like into the CCA Editor application on the picture below.



Calculus of Context-Aware Ambients

Calculus of Context-Aware Ambients

In order to have a better understanding about CCA you can read the following paragraphs

The Calculus of Context-aware Ambients (in short - CCA) is a paradigm for modelling and verification of different kind of systems that are context-aware.

This process calculus is built upon the calculus of different ambients and introduces new constructs to enable ambients and processes to be aware of the environment in which they are being executed.

This results in a powerful calculus where both mobility and context-awareness are first-class citizens.

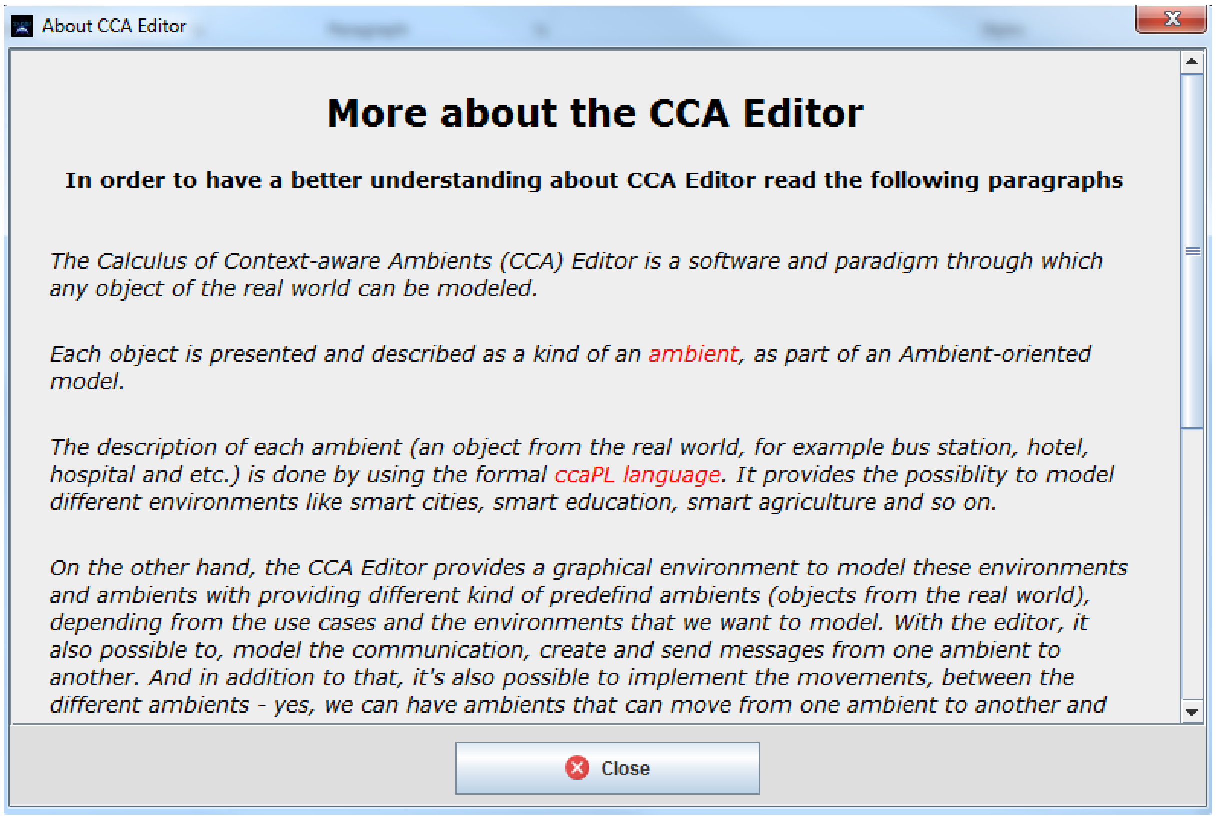
There are not so trivial syntax and a formal semantics of the calculus. There's a proposal for a new theory of equivalence of processes, which allows the identification of systems that have the same context-aware behaviours.

It's proved that CCA encodes the calculus, which is known to be a universal model of computation.

11. About CCA Editor Sub-menu Item Screen

The about CCA editor sub-menu item section includes basic information about the CCA Editor and how we can use it in order to be able to create a CCA model successfully. Also has an information about the ccaPL language, which is used to develop a CCA model.

You can see how this section looks like into the CCA Editor application on the picture below.



More about the CCA Editor

In order to have a better understanding about CCA Editor read the following paragraphs

The Calculus of Context-aware Ambients (CCA) Editor is a software and paradigm through which any object of the real world can be modeled.

Each object is presented and described as a kind of an ambient, as part of an Ambient-oriented model.

The description of each ambient (an object from the real world, for example bus station, hotel, hospital and etc.) is done by using the formal ccaPL language. It provides the possiblity to model different environments like smart cities, smart education, smart agriculture and so on.

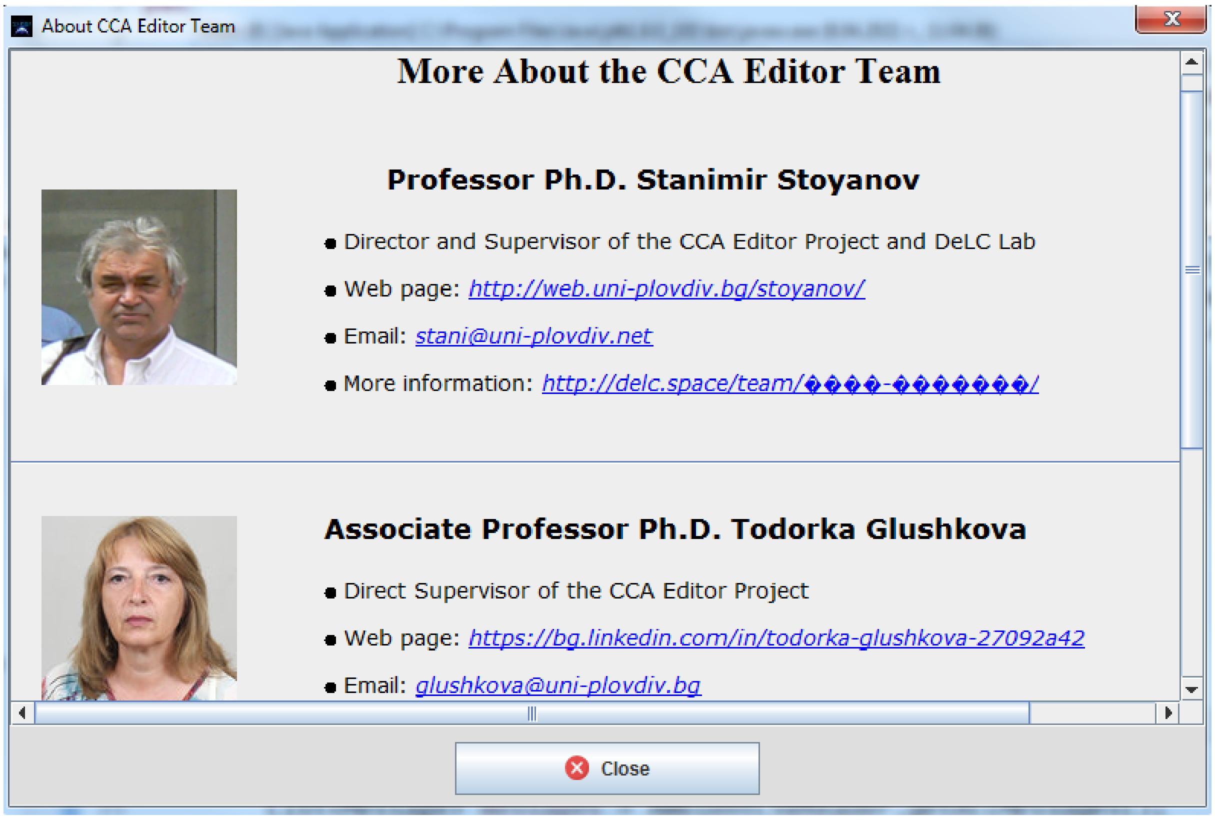
On the other hand, the CCA Editor provides a graphical environment to model these environments and ambients with providing different kind of predefind ambients (objects from the real world), depending from the use cases and the environments that we want to model. With the editor, it also possible to, model the communication, create and send messages from one ambient to another. And in addition to that, it's also possible to implement the movements, between the different ambients - yes, we can have ambients that can move from one ambient to another and

12. CCA Editor Team Sub-menu Item Screen

The about CCA editor team sub-menu item section includes an information about the CCA editor team members.

- *Professor Ph.D. Stanimir Stoyanov*
- *Associate Professor Ph.D. Todorka Glushkova*
- *Ph.D. Candidate Konstantin Rusev*

You can see how this section looks like into the CCA Editor application on the picture below.



More About the CCA Editor Team

Professor Ph.D. Stanimir Stoyanov

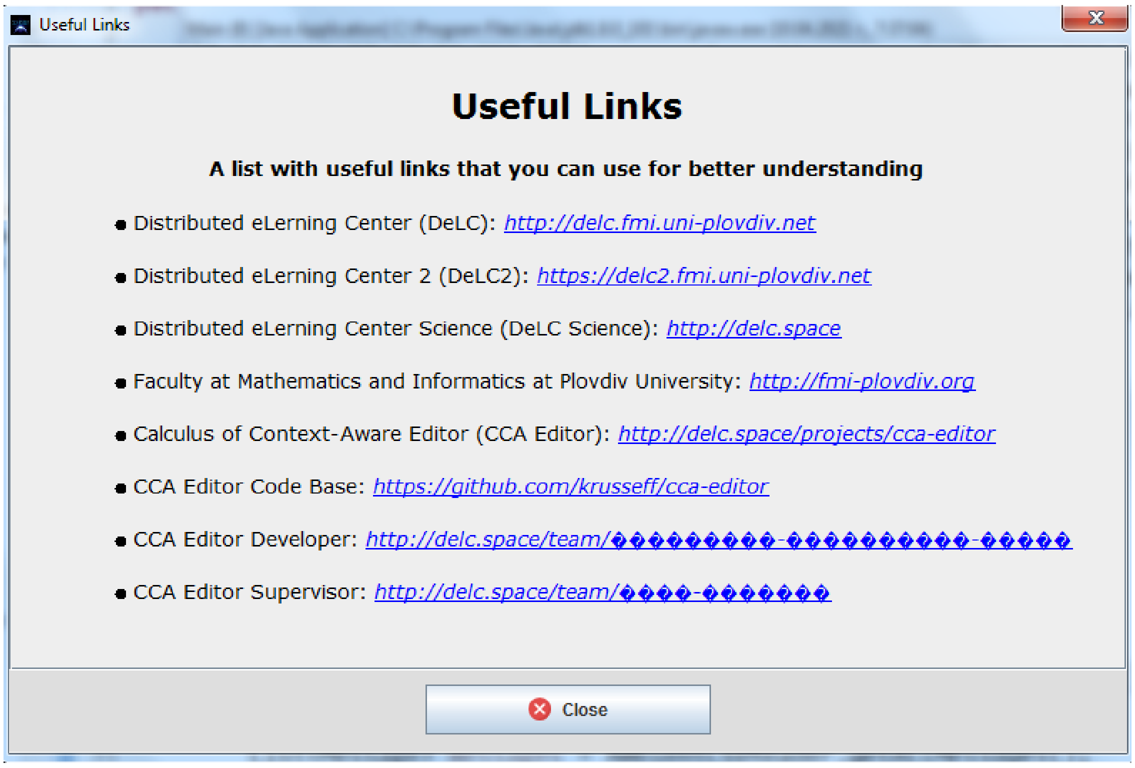
- Director and Supervisor of the CCA Editor Project and DeLC Lab
- Web page: http://web.uni-plovdiv.bg/stoyanov/
- Email: stani@uni-plovdiv.net
- More information: http://delc.space/team/����-�������/

Associate Professor Ph.D. Todorka Glushkova

- Direct Supervisor of the CCA Editor Project
- Web page: https://bg.linkedin.com/in/todorka-glushkova-27092a42
- Email: glushkova@uni-plovdiv.bg

13. Useful Links Sub-menu Item Screen

This section of the CCA Editor application provides an information for some useful links that may be helpful during the interaction with the application.



Useful Links

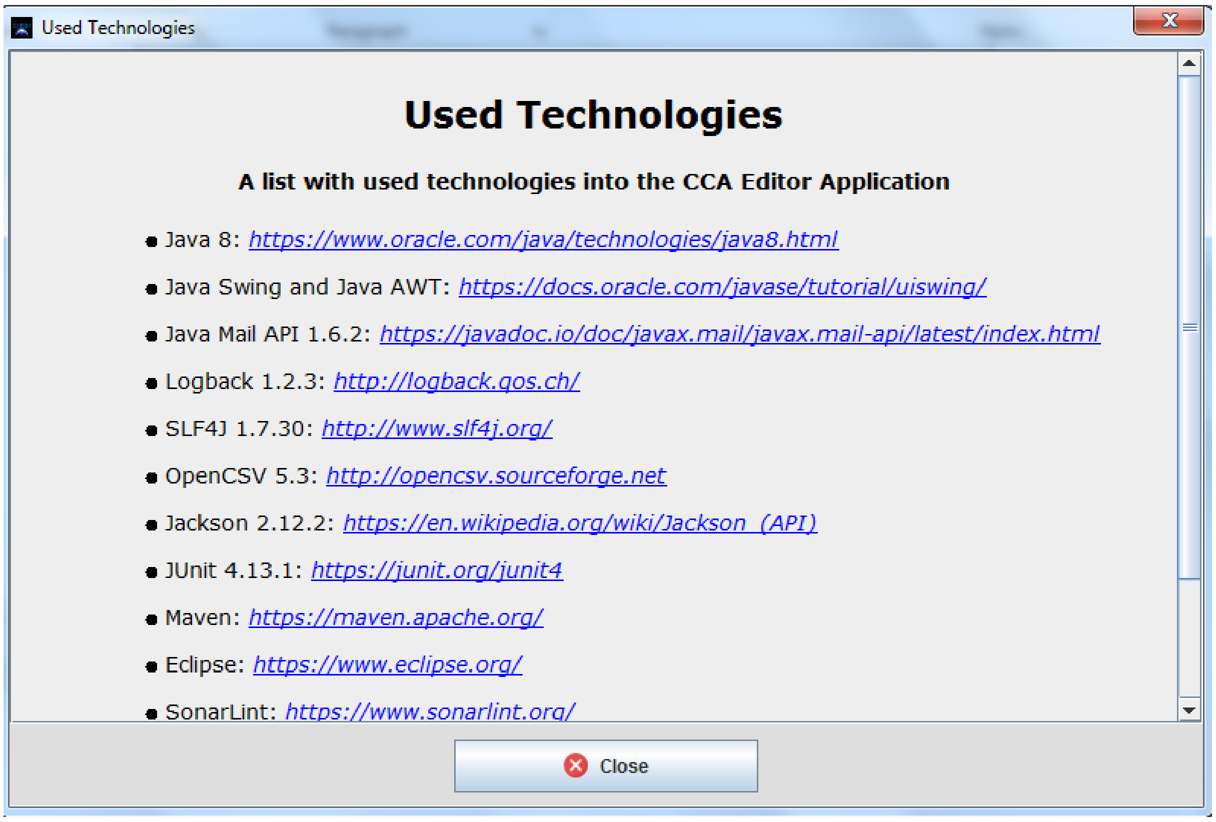
A list with useful links that you can use for better understanding

- Distributed eLerning Center (DeLC): http://delc.fmi.uni-plovdiv.net
- Distributed eLerning Center 2 (DeLC2): https://delc2.fmi.uni-plovdiv.net
- Distributed eLerning Center Science (DeLC Science): http://delc.space
- Faculty at Mathematics and Informatics at Plovdiv University: http://fmi-plovdiv.org
- Calculus of Context-Aware Editor (CCA Editor): http://delc.space/projects/cca-editor
- CCA Editor Code Base: https://github.com/krusseff/cca-editor
- CCA Editor Developer: http://delc.space/team/��������-���������-�����
- CCA Editor Supervisor: http://delc.space/team/����-�������

14. Technologies Sub-menu Item Screen

This section of the CCA Editor application provides an information in more details about the technologies that are used to develop the CCA Editor application. The main technologies are:

- Java 8
- Java Swing and Java AWT – used for the user interface
- Java Mail API – used to develop the contact us form described below
- Logback and SLF4J – used to build application logs
- OpenCSV and Jackson – used to read/write data from/to files in a specific format (CSV or JSON)
- Maven – used as a build tool of the application
- SonarLint – used to scan for the source code quality (duplicate or dead code, some optimizations and etc.)
- CCA and ccaPL – used to be able to build the CCA file with the specific syntax, which is strongly required
- JUnit – used to implement unit and integration tests of the application
- Eclipse – used as an integrated development environment
- etc.



Used Technologies

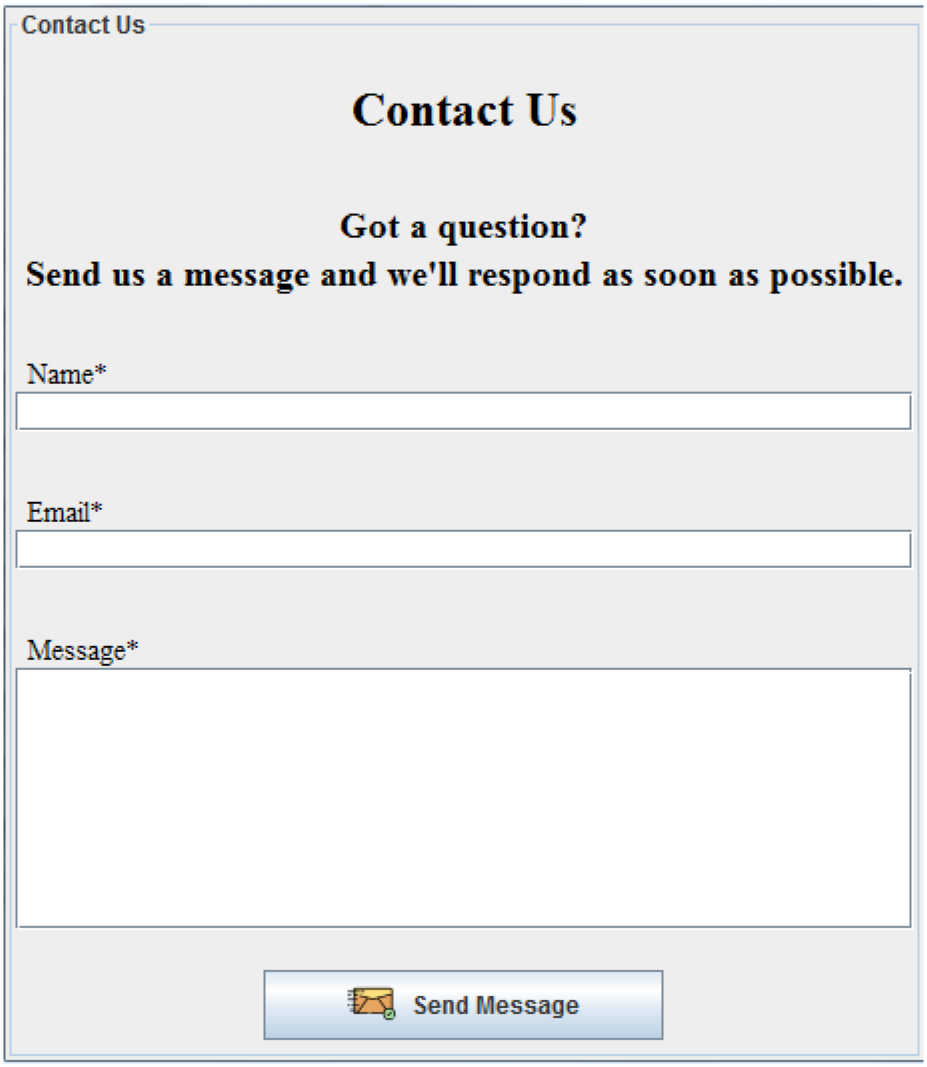
A list with used technologies into the CCA Editor Application

- Java 8: https://www.oracle.com/java/technologies/java8.html
- Java Swing and Java AWT: https://docs.oracle.com/javase/tutorial/uiswing/
- Java Mail API 1.6.2: https://javadoc.io/doc/javax.mail/javax.mail-api/latest/index.html
- Logback 1.2.3: http://logback.qos.ch/
- SLF4J 1.7.30: http://www.slf4j.org/
- OpenCSV 5.3: http://opencsv.sourceforge.net
- Jackson 2.12.2: https://en.wikipedia.org/wiki/Jackson_(API)
- JUnit 4.13.1: https://junit.org/junit4
- Maven: https://maven.apache.org/
- Eclipse: https://www.eclipse.org/
- SonarLint: https://www.sonarlint.org/

15. Contact Us Sub-menu Item Screen

This section of the screen provides the functionality to the regular user to send a message to the owners and administrators of the CCA Editor application. Need to provide a valid name and email address as long as the message to be able to contact the user afterwards. In which scenario you can use this form to contact the administrators?

- You have a question about the application or how to use it?
- You need some help with some of the functionalities?
- You have an issue using the application or unexpected error occurred while using the application?


Contact Us

Contact Us

Got a question?

Send us a message and we'll respond as soon as possible.

Name*

Email*

Message*

16. Create a Specific Ambient Screen

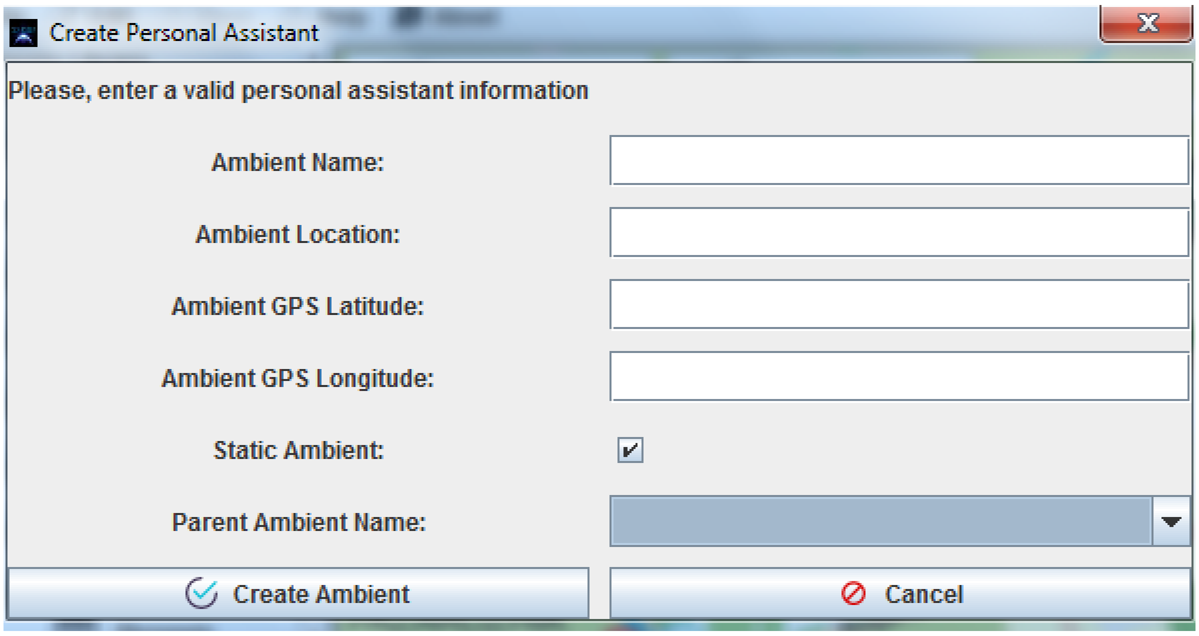
Before create an exact ambient you need to select a specific type of ambient from the ambient library section of the application, which is on the left side of the screen.

After that you need to enter a valid information (*validation is implemented for each field, so it should not be able to create an ambient with invalid input data*) for:

- **ambient name** – unique ambient name should be provided
- **ambient location** – valid ambient location should be provided
- **GPS Latitude** – valid coordinates should be provided
- **GPS Longitude** – valid coordinates should be provided
- **static ambient** – checkbox with only two possible values (yes/no)
- **parent name ambient** – need to select an already existing ambient from a drop-down list that should a parent of the currently created ambient

After that two buttons are implemented in order to be able to exit this screen without creating an ambient or to create an ambient if the data that is entered is still valid.

Confirmation dialogs are implemented for each operation (exit or create ambient) and an error dialog appears if the data, which is provided is invalid.



Create Personal Assistant

Please, enter a valid personal assistant information

Ambient Name:	
Ambient Location:	
Ambient GPS Latitude:	
Ambient GPS Longitude:	
Static Ambient:	✓
Parent Ambient Name:	

17. Send Ambient Message Screen

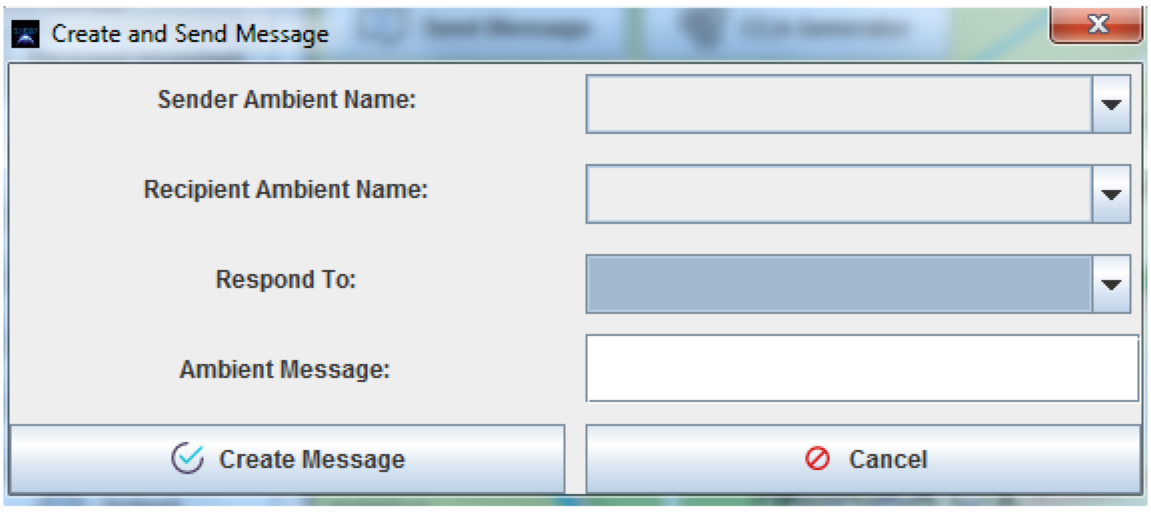
When all of the needed ambients are created using the functionality to create an ambient, as a next step we need to create some messages between them in order to be able to build a valid CCA model.

Create and send ambient message screen allows to create and store an ambient message between two ambient instances. In order to be able to do it successfully, you need to provide a valid information for the following fields:

- **Sender ambient name** – need to select an ambient that already exists from a drop-down list with all ambients that are successfully created. Required field.
- **Recipient ambient name** – need to select an ambient that already exists from a drop-down list with all ambients that are successfully created. Required field.
- **Respond to** – need to select an already existing ambient message that you want to respond to. Not required field, you can leave it as it is.
- **Ambient Message** – need to type the exact ambient message that you what to create and send between the ambients that are already selected.

After that two buttons are implemented in order to be able to exit this screen without creating an ambient message or to create an ambient message if the data that is entered is still valid.

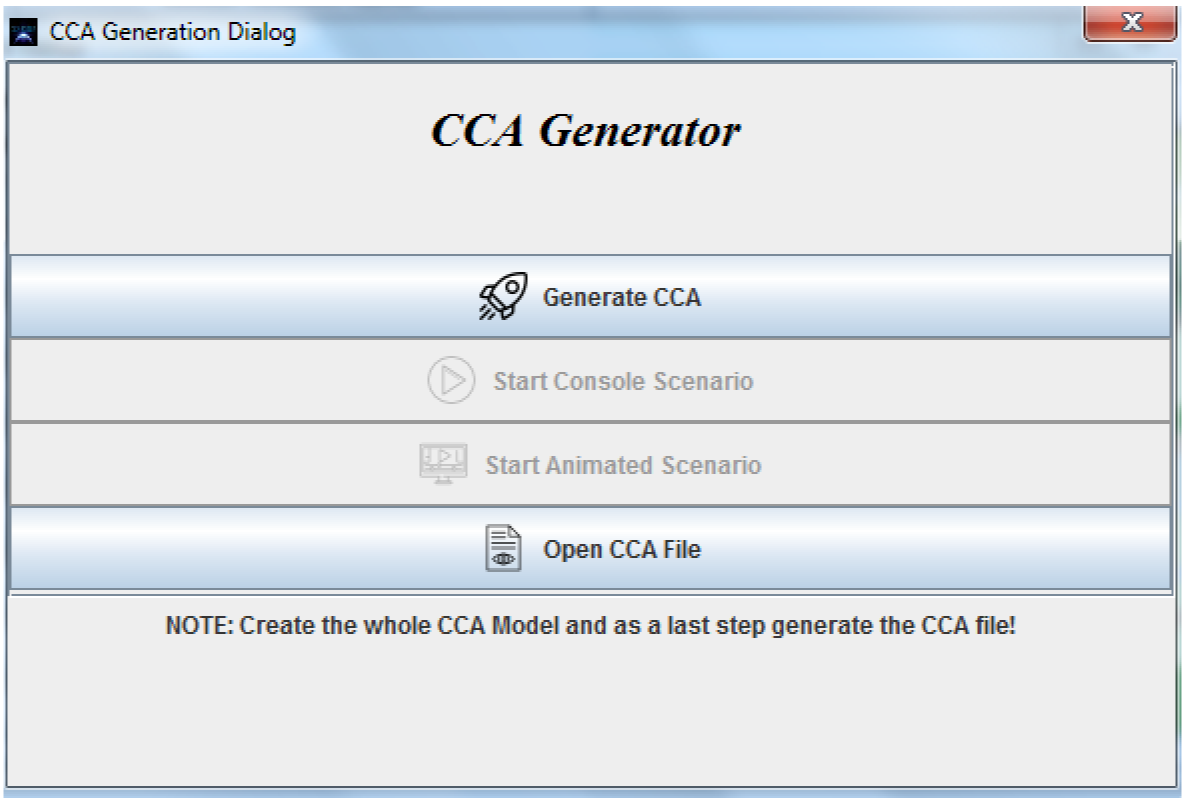
Confirmation dialogs are implemented for each operation (exit or create ambient message) and an error dialog appears if the data, which is provided is invalid.



18. CCA Generator Screen

The CCA Generator screen provides the following options, but you need to create all needed ambients and ambient messages before start using this section of the CCA Editor application (*this is mentioned as a note on the screen as well*).

- **Generate CCA** – this button triggers the generation of the CCA file with the appropriate syntax using the ambients and ambient messages that are already created in the application.
- **Start Console Scenario** – this functionality is not yet implemented, that's why the button is in disabled state, but the main goal here is to be able to start the CCA scenario into the console, using the already generated CCA file from the CCA Editor application and the CCA parser.
- **Start Animated Scenario** – this functionality is not yet implemented, that's why the button is in disabled state, but the main goal here is to be able to start the CCA scenario at animated mode, using the already generated CCA file from the CCA Editor application and the CCA animator (*we have an existing version, but we need to update and integrate it with the CCA Editor application*).
- **Open CCA File** – this button opens another screen with an option to view and edit an already generated CCA file. More information is provided into the next section of the document.



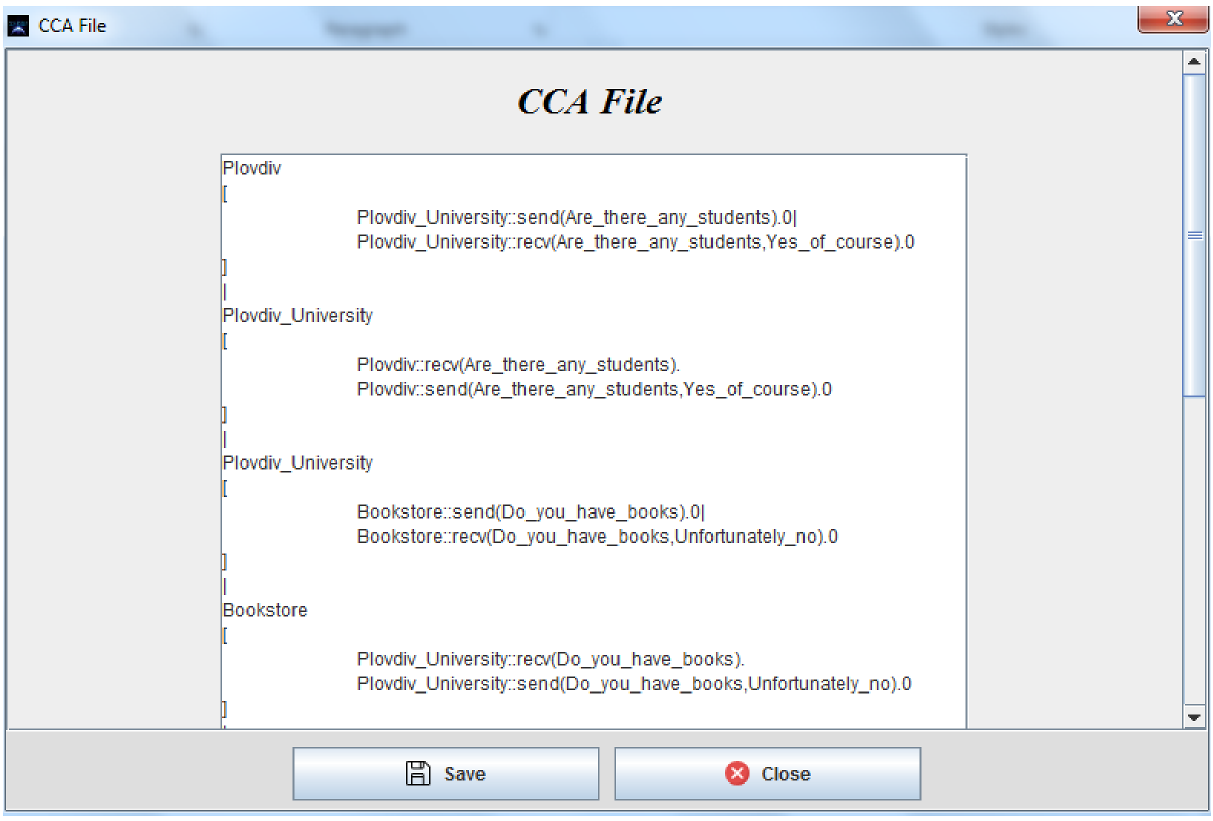
NOTE: Create the whole CCA Model and as a last step generate the CCA file!

19. Open CCA File Screen

The open CCA file screen of the CCA Editor application allows you to **review** and **edit** an already existing and generated CCA file. If the CCA file does not exist, descriptive message will be displayed on the screen with an information that the CCA file does not exist or an unexpected error occurred while using the application.

As long as the text area, where you can review and edit the CCA file, two buttons are implemented in order to be able to exit/close this screen without updating the CCA file or to update it with an additional ambients and messages between them. Keep in mind that the syntax is really specific and you should follow it in order to be able to start the CCA scenario afterwards.

Confirmation dialogs are implemented for each operation (exit/close or save the CCA file) and an error dialog appears if the data, which is provided is invalid.



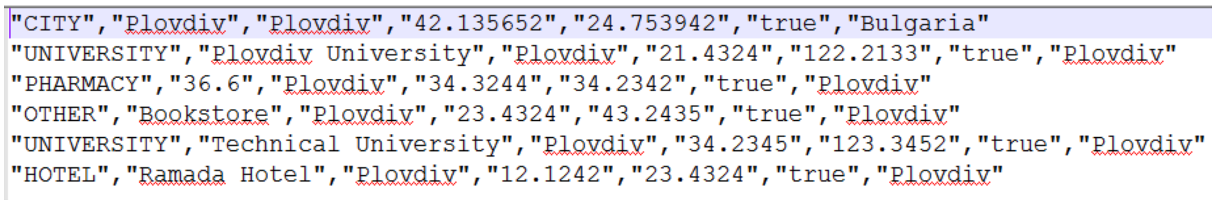
```
Plovdiv
[
                Plovdiv_University::send(Are_there_any_students).0|
                Plovdiv_University::recv(Are_there_any_students,Yes_of_course).0
]
|
Plovdiv_University
[
                Plovdiv::recv(Are_there_any_students).
                Plovdiv::send(Are_there_any_students,Yes_of_course).0
]
|
Plovdiv_University
[
                Bookstore::send(Do_you_have_books).0|
                Bookstore::recv(Do_you_have_books,Unfortunately_no).0
]
|
Bookstore
[
                Plovdiv_University::recv(Do_you_have_books).
                Plovdiv_University::send(Do_you_have_books,Unfortunately_no).0
]
```

20. CCA Ambients in CSV File

Immediately after an ambient is created successfully, it is stored into a file with a CSV (comma-separated value) format with the following information. The file is with the name: *ambients.csv*

It is need to store each ambient into this file, because of the how the generate CCA functionality is implemented. This will be described into the following sections of the document.

- **Ambient Type** – this is the first column of the file, for example "UNIVERSITY"
- **Ambient Name** – this is the second column of the file, for example "Plovdiv University"
- **Ambient Location** – this is the third column of the file, for example "Plovdiv"
- **Ambient GPS Latitude** – this is the fourth column of the file, for example "21.4324"
- **Ambient GPS Longitude** – this is the fifth column of the file, for example "122.2133"
- **Static Ambient** – this is the sixth column of the file, for example "true"
- **Parent Ambient Name** – this is the seventh column of the file, for example "Plovdiv"



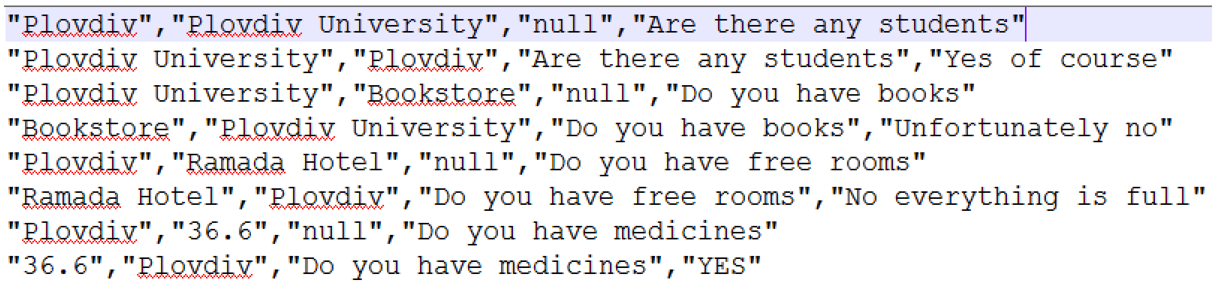
```
"CITY","Plovdiv","Plovdiv","42.135652","24.753942","true","Bulgaria"
"UNIVERSITY","Plovdiv University","Plovdiv","21.4324","122.2133","true","Plovdiv"
"PHARMACY","36.6","Plovdiv","34.3244","34.2342","true","Plovdiv"
"OTHER","Bookstore","Plovdiv","23.4324","43.2435","true","Plovdiv"
"UNIVERSITY","Technical University","Plovdiv","34.2345","123.3452","true","Plovdiv"
"HOTEL","Ramada Hotel","Plovdiv","12.1242","23.4324","true","Plovdiv"
```

21. CCA Ambient Messages in CSV File

Immediately after an ambient message is created successfully, it is stored into a file with a CSV (comma-separated value) format with the following information. The file is with the name: *messages.csv*

It is need to store each ambient message into this file, because of the how the generate CCA functionality is currently implemented. This will be described into the following sections of the document.

- **Ambient Sender** – this is the first column of the file, for example "Plovdiv University".
- **Ambient Recipient** – this is the second column of the file, for example "Plovdiv".
- **Respond To Message** – this is the third column of the file, for example "Are there any students". If the value is "null", no respond to message is provided, when the ambient message is created into the CCA Editor.
- **Ambient Message** – this is the fourth column of the file, for example "Yes of course". This is the column, where the exact ambient message is stored.



```
"Plovdiv","Plovdiv University","null","Are there any students"
"Plovdiv University","Plovdiv","Are there any students","Yes of course"
"Plovdiv University","Bookstore","null","Do you have books"
"Bookstore","Plovdiv University","Do you have books","Unfortunately no"
"Plovdiv","Ramada Hotel","null","Do you have free rooms"
"Ramada Hotel","Plovdiv","Do you have free rooms","No everything is full"
"Plovdiv","36.6","null","Do you have medicines"
"36.6","Plovdiv","Do you have medicines","YES"
```

22. Generated CCA File

Immediately after the "*Generate CCA*" button into the CCA Editor application is clicked, a file with a specific syntax (*ccaPL language*) is created with a CCA format with the following information. The file is with the name: *ambients.cca*

In order to be able for the CCA Editor application to generate valid and executable CCA file, you need to follow some steps:

1. Create all ambients that you want using the user interface.
2. Create all ambient messages that you want using the user interface.
3. Based on the first two steps, two files will be created and will be used afterwards.
4. Navigate to the "Generate CCA" button and click on it. With this action you will trigger the generation of the CCA file. The exact generation of the CCA file uses the already created files with all ambients and ambient messages to generate a valid CCA file with the appropriate syntax. As long as these files, another predefined files with CCA templates are used in order to be able to generate the CCA file.

```
Plovdiv
[
    Plovdiv_University::send(Are_there_any_students).0|
    Plovdiv_University::recv(Are_there_any_students,Yes_of_course).0
]
|
Plovdiv_University
[
    Plovdiv::recv(Are_there_any_students).
    Plovdiv::send(Are_there_any_students,Yes_of_course).0
]
|
Plovdiv_University
[
    Bookstore::send(Do_you_have_books).0|
    Bookstore::recv(Do_you_have_books,Unfortunately_no).0
]
|
Bookstore
[
    Plovdiv_University::recv(Do_you_have_books).
    Plovdiv_University::send(Do_you_have_books,Unfortunately_no).0
]
```

23. CCA Generation Templates

To be able for the CCA Editor to generate the CCA file successfully, different templates with the specific ccaPL syntax are implemented and used for the different cases. For example, at the moment there are two templates:

- **Send-receive CCA template** – this template is used, when an ambient sends a message to the other ambient and waits for a response from it. The following placeholders are implemented and they will be replaced during the process of generating the CCA model
    - **AMBIENT_SENDER** – this will be replaced with the ambient name that sends the message
    - **AMBIENT_RECEIPIENT** – this will be replaced with the ambient name that receives the message
    - **SEND_PARAMETERS** – this will be replaced with the parameters that will be send from one ambient to another (the exact ambient message)
    - **RECEIVE_PARAMETERS** – this will be replaced with the parameters that will be received as a response to the sender.

```
AMBIENT_SENDER
[
    AMBIENT_RECEIPIENT::send(SEND_PARAMETERS).0|
    AMBIENT_RECEIPIENT::recv(RECEIVE_PARAMETERS).0
]
```

- **Receive send CCA template –** this template is used, when an ambient receives a message from the other ambient and sends a response for it. The following placeholders are implemented and they will be replaced during the process of generating the CCA model
    - **AMBIENT_SENDER** – this will be replaced with the ambient name that sends the message as a response
    - **AMBIENT_RECEIPIENT** – this will be replaced with the ambient name that sends the message and waiting for a response
    - **SEND_PARAMETERS** – this will be replaced with the parameters that will be received from the ambient sender
    - **RECEIVE_PARAMETERS** – this will be replaced with the parameters that will be send from the ambient recipient to the ambient sender as a response (the exact ambient message).

```
AMBIENT_SENDER
[
    AMBIENT_RECEIPIENT::recv(SEND_PARAMETERS).
    AMBIENT_RECEIPIENT::send(RECEIVE_PARAMETERS).0
]
```

References

1. Siewe, F., Zedan, H., Cau, A., The Calculus of Context-aware Ambients, Journal of Computerand System Sciences, 77, 597–620 (2011).
2. Al-Sammarraie, M. H., Policy-based Approach For Context-aware Systems, Software Technology Research Laboratory, De Montfort University, Leicester, UK (2011).
3. Stoyanov, S., Context-Aware and Adaptable eLearning Systems, Internal Report, Software Technology Research Laboratory, De Montfort University, Leicester (2012).