

## UbiCARS Model-Driven Development (MDD) Framework Evaluation

Recommender systems are software tools able to discover the necessary knowledge about users (customers) to offer them personalized recommendations. The UbiCARS MDD Framework enables developers that are not experts in recommender systems to design and develop context-aware recommender systems for e-stores and physical retail stores. The resulting recommender systems use external recommendation algorithms and engines, and take advantage of customer-product interaction data from the e-store (customer ratings on products, customers' browsing history and purchase history) and the physical store (customers' staying time in front of products). UbiCARS stands for Ubiquitous Context-Aware Recommender Systems.

The UbiCARS MDD Framework is composed of the modelling editor and the System UI. The modelling editor is used to create a recommender system application model that is then fed to the system through the System UI, in order to configure the e-store and enable it to accommodate, compute and present recommendations to customers. This configuration includes using plugins to retrieve user-product interaction data from the e-store, retrieving users' staying time in front of products in the physical store via the usage of a mobile application, as well as configuration and usage of recommendation engines. The configuration is conducted in an automated fashion.

The system supports Wordpress Woocommerce, Drupal e-Commerce and custom e-commerce platforms. For the purposes of this evaluation, a Wordpress Woocommerce e-store has been set up.

The evaluation of the UbiCARS MDD Framework aims to make you familiar with the framework, what it does, how it is used and what its results are, in order to provide your expert feedback through an online questionnaire.

There are two evaluation methods for the evaluation of the UbiCARS MDD Framework:

1. You may watch a [video](#) about how the UbiCARS MDD Framework is used and what its results are, and then complete an [online questionnaire](#). This process takes 10 minutes.

or

2. You may directly use the UbiCARS MDD Framework by interacting with the modelling editor and the System UI, observing the results (product recommendations to customers) on the e-store, and then complete the [online questionnaire](#) (the same questionnaire as above<sup>1</sup>). This process takes about 20 minutes. More information can be found in the following paragraphs.

**Important:** during this evaluation method you will need to use the System UI to configure the e-store. As configurations include alteration of the database infrastructure and data, to obtain correct results, only one participant may perform the evaluation at any given time. Therefore, to access the e-store during the evaluation, please reserve a one-hour period and please conduct the evaluation sometime within the reserved hour. Please use our reservation web app [here](#).

If you decide to participate in this evaluation, you may want to read the [Information about the Modelling Editor](#) and [Information about the Modelling Elements](#) sections of this document.

For questions or if you face any issues with this study, please contact: Christos Mettouris, University of Cyprus, [mettour@cs.ucy.ac.cy](mailto:mettour@cs.ucy.ac.cy), +357 99292254.

---

<sup>1</sup> If you decide to do both 1. and 2., please fill in the questionnaire only once in the end.

## Usage of the UbiCARS MDD Framework

SETUP (takes about 5 minutes excluding downloading time)

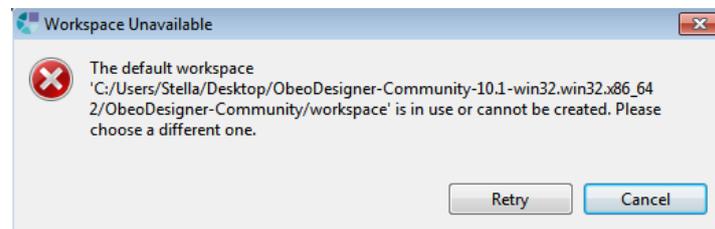
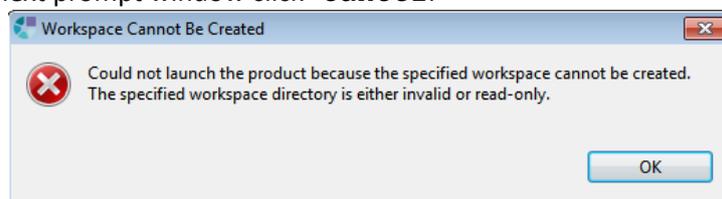
Requirements:

- Java JRE 8 (to run ObeoDesigner)
- Win 7, Win 8 or Win 10

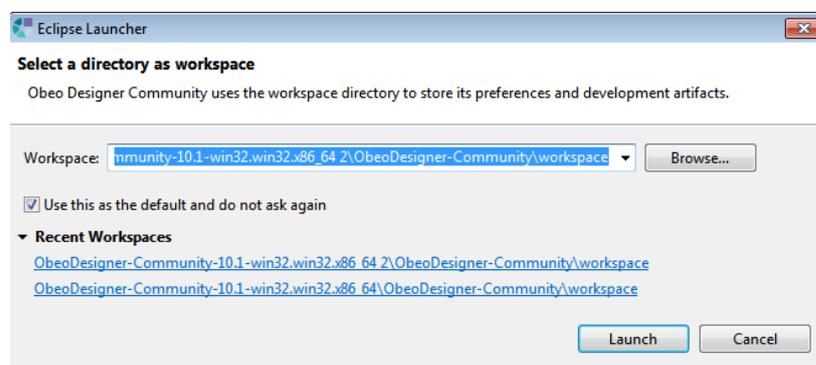
Download ObeoDesigner-Community\_UbiCARS.rar file (file size 374 MB) from this [link](#) (ignore any “network problems” and click on “download”).

After downloading ObeoDesigner-Community\_UbiCARS.rar:

- Unzip ObeoDesigner-Community\_UbiCARS.rar on your Desktop
- In folder ObeoDesigner-Community double click on obeodesigner.exe
- You will be prompted a message that the workspace directory cannot be created. Click OK and to the next prompt window click cancel.

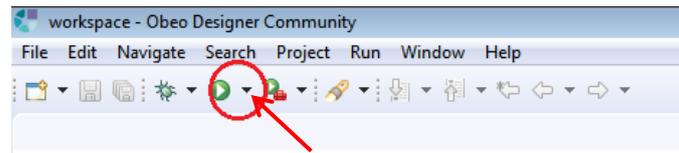


- You will be prompted to enter a working directory for workspace. Click browse, go to the Desktop folder where you have extracted ObeoDesigner-Community\_UbiCARS.rar, go to ObeoDesigner-Community folder and select the workspace folder. E.g. C:\...path to your extracted folder "ObeoDesigner-Community\_UbiCARS"\ObeoDesigner-Community\_UbiCARS\ObeoDesigner-Community\workspace

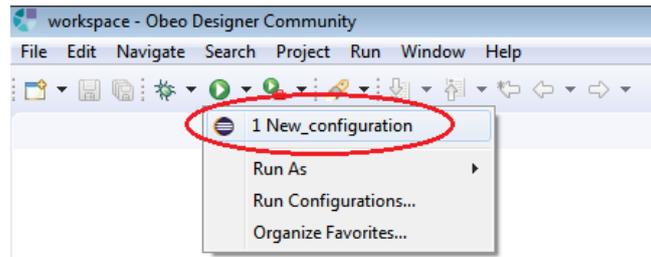


- Click Launch

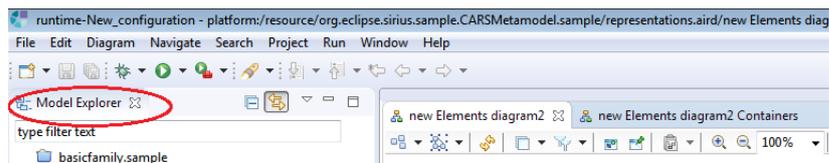
- When ObeoDesigner starts, open the RUN menu by clicking on the black arrow next to the green RUN button, as in the figure below.



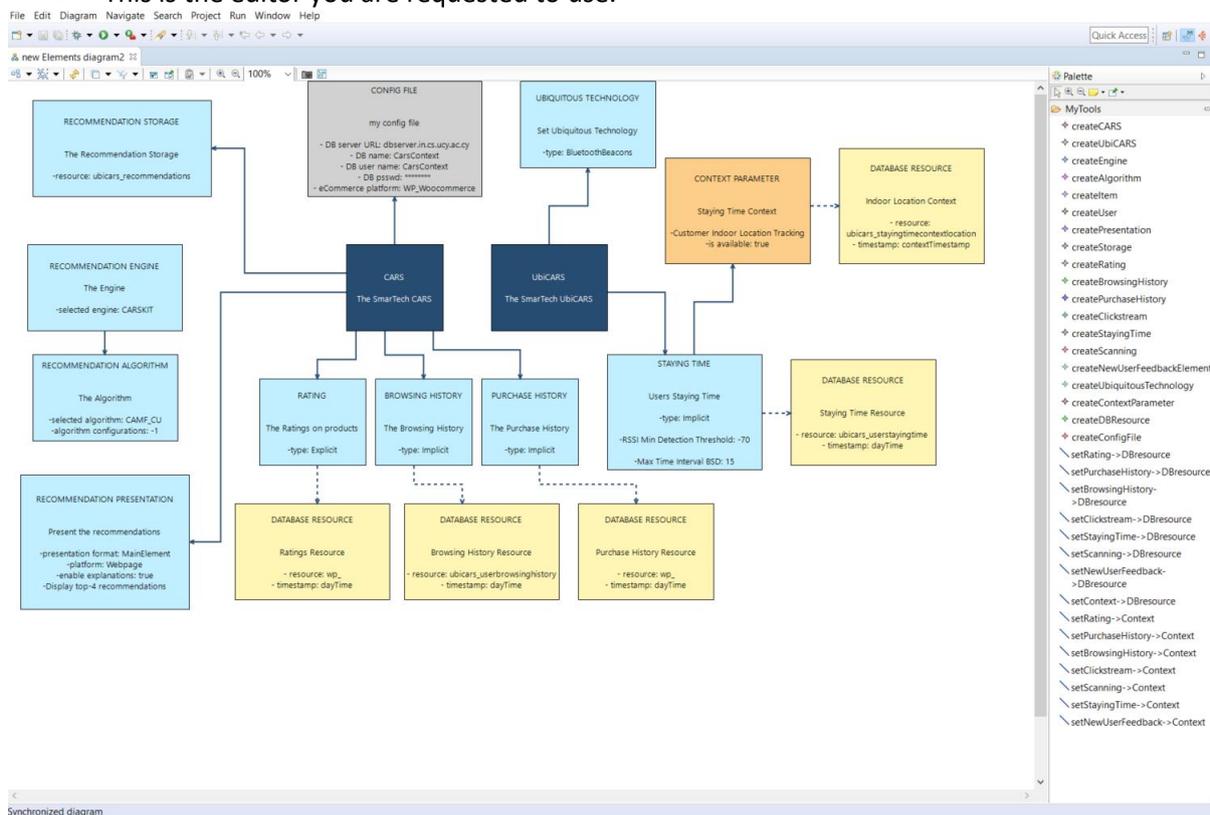
- Click on 1 New\_configuration, as shown in the figure below.



- A new runtime window will open in ObeoDesigner, as shown below. If Model Explorer appears, close it by clicking on X.



- This is the editor you are requested to use.



- The model displayed represents a recommender system for "SmarTech electronics store that has an e-commerce website and physical showrooms with electronic products. The model is provided instead of guiding participants to design it themselves in order to keep the evaluation session short. In case you would like to design it from start to end, please follow the steps described [here](#).
- Please read the [Information about the Modelling Editor](#) and [Information about the Modelling Elements](#) sections of this document as they will help you understand the modelling process. You may change the model according to the rules described there.

## E-Store and System UI Access

- When done from the editor, please access the e-store: **make sure you access the e-store in a reserved by you time period.**
- Access the e-store [here](#) and sign in using a guest user account. Instead of registering to the platform, an existing user is being used as this user already has user interaction data stored in the platform (ratings of products, browsing history of products, purchase of products, staying time in front of products), which are being used by the System UI.  
Login: simos  
Psswd: yb^ExpX)ySBjBE^IjSYh&Jlf
- The e-store does not include any recommendations at the moment (you may navigate to "Recommendations" page).
- Navigate to the [System UI page](#) and upload the model that appears in the modelling editor:  
File My.carsmetamodel located in the following directory: C:\...*path to your extracted folder "ObeoDesigner-Community\_UbiCARS"*\ObeoDesigner-Community\runtime-New\_configuration\org.eclipse.sirius.sample.CARSMetamodel.sample\
- In the model the database (DB) password has been removed for security reasons.
- Execute the steps shown below, once each (active buttons are highlighted in blue in the UI):
  - **Step 1: Configure.** The system parses the model and creates the necessary configurations on the e-store and its database.
  - **Step 2: Fill with Data** is important for computing recommendations as it fills the created DB tables with customer-product interaction data that resulted from our interaction with the e-store and physical store. This is a mandatory step as with no data, no recommendation can be computed.
  - **Step 3: Compile datasets:** the corresponding datasets for Rating, Browsing History, Purchasing History and Staying Time (as per the model) are computed
  - **Step 4: Compute Recommendations** by clicking on the "Compute Recommendations" button and then on the Compute Recommendations link of the dataset of your choice. You can view the datasets by clicking on their title.
  - **Step 5: Store Recommendations.** You may now observe the recommendations for you [on the e-store](#).
  - You may repeat steps 4-5 with different datasets.
- You may experiment by enhancing the system with more user-product interaction data (user feedback on products), e.g. by browsing more products and then repeating steps 3 to 5. Observe that the newly browsed products are now included in the Browsing History dataset in step 4. However, please note that this does not guarantee that you will receive different recommendations.

## Information about the Modelling Editor

- The available tools are placed in the toolbox named “MyTools” situated to the right-hand side of the editor.
- To add an Element in the editor canvas: click on the respective item in the toolbox and then click in the white space in the editor (canvas).
- There are 4 types of Elements (when placed on canvas):
  - Dark Blue: CARS and UbiCARS as the starting points
  - Light Blue: main elements of a model
  - Yellow: Database Resource elements
  - Orange: Context elements
- For the Blue elements only (dark and light): you cannot add more than one element of the same type, e.g. two CARS or two Recommendation Engine elements.
- To edit the value of an element’s parameter go to the “Properties” window. You may need to scroll to see all the parameters.
  - Legal characters are only letters and numbers
  - For Names: Names can be any words. Spaces between words are allowed, but they are not allowed before the start or after the end of names
  - For parameters other than Names: spaces between words are not allowed, as well as before the start or after the end of their values
- If “Properties” window is not visible, you can access it by clicking on Eclipse Runtime main window: “window”, “Show View”, “Properties”.
- To link an element ElementA to another element ElementB: click on the respective edge in the toolbox that names the two types of elements as follows: setElementA->ElementB. Then click first on ElementA and then on ElementB. The other way around is not possible.
- Elements whose relationship with other elements is a one-to-one relationship are already linked together during creation. You only need to specifically link together:
  - Main Elements with Database Resource elements
  - Main Elements with Context Elements
  - Context Elements with Database Resource elements
- To delete an element or an edge, click on the element or edge and press delete on the keyboard.
- When using Wordpress as the platform of use in config file: for the elements Rating and Purchase History we do not need to create new tables, as the pre-existing Wordpress e-store already includes the infrastructure for this functionality. Therefore, any database Resources linked to these two elements need to have Resource Value empty or “wp\_” (from Wordpress database tables prefix).

## Information about the Modelling Elements

A product recommender system consists of a CARS system and a UbiCARS app. The CARS element represents the e-store (e-commerce scenario) which defines the user Ratings on products, the PurchaseHistory and the BrowsingHistory as user feedback elements on products. A CARS can instantiate zero or one of each of these elements, while a number of custom

`NewUserFeedback` elements can be defined by the user (not included in the current model, can be added through the toolbox). In the physical store (ubiquitous scenario) a `UbiCARS` app may use the `StayingTime` element representing the Staying Time in front of products or the `Scanning` element representing the scanning of NFC tags of products. In this evaluation the `StayingTime` element is used. As with the e-commerce scenario, a `UbiCARS` element can have number of custom explicit or implicit `NewUserFeedback` elements to be defined by the user if needed.

Assume the "SmarTech electronics store that has an e-commerce website and a number of physical showrooms with electronic products. The `UbiCARS` framework requires a `UbiCARS` app for usage in the physical showrooms, as well as a `CARS` system functioning on the e-commerce website, aiming to enhance the accuracy and availability of recommendations.

Each of the aforementioned elements uses a `DatabaseResource` and may use a `ContextParameter`. A `DatabaseResource` specifies information about the database resource where the respective information will be stored and retrieved from. Such information may constitute the table name and a timestamp that may be used as contextual information about time. A `ContextParameter` captures the context of the respective user while interacting with products. It is specified by the developer. For example, customer indoor location can be defined as context. Besides a name, context is assigned with a Boolean `isAvailable` parameter to denote whether the context sensing mechanism to sense the respective context is available or it needs to be developed. Furthermore, a `ContextParameter` uses a `DatabaseResource` to denote database related information where the corresponding context will be (or is already) stored.

A `RecommendationEngine` is used to compute the recommendations. While `RecommendationStorage` defines the place where computed recommendations are stored, `RecommendationPresentation` denotes the platform of usage for presenting the recommendations to end-users. `topN` denotes the number of recommendations to be presented (e.g. top-5).

By using the discussed model the system will be configured such that, after customers repeatedly visit both the SmarTech physical store and e-store, they will be tracked by the system in terms of the following: product ratings, browsing history and online purchase history from the e-store, and Staying time in seconds from the physical store. From each of the abovementioned, the system is able to compile datasets to be included as input in the recommendation process.