TACS 2526 Assignment

First Delivery - group 08

Antonio Augusto Brito de Sousa - 202000705 - Structural Model
Lucas Borborema Nakajima - 202001337 - Structural Model
Álvaro Siqueira Galvão - 202502639 - DSML
Justino Orlando Sachilombo - 202302527 - DSML
Krsto Kustudic - 202509015 - GUI
Hugo Castro - 202403131 - GUI
Maria Leonor Monteiro Beirão - 201806798 - Review Work

1. Introduction

Problem Brief

Local communities often face difficulties exchanging goods and services efficiently. Many items that could be reused end up discarded, and people lack a simple platform to connect within their own neighborhoods, schools, or workplaces. The challenge is to design a **flexible system** that **encourages reuse**, **supports donations and trades**, and **strengthens social connections** while reducing dependency on external markets.

Application Solution Overview (Main Concepts)

Our proposed application provides a community exchange platform inspired by systems such as OLX and Freecycle, but focused on local and sustainable interactions. It allows users to:

- Publish **products or services** for sale, exchange, or donation.
- **Chat** securely with other members.
- Rate transactions through unidirectional/**bidirectional evaluations**.
- Customize visual themes, rules, and payment modes according to each community.

The architecture is **model-driven-development (MDD)**, following the **BESSER framework**, which separates the system into three core metamodels:

- **Structural Metamodel** — defines core entities (Users, Items, Offers, Transactions).
- **DSML Metamodel** — controls customizable features (Payments, Messaging, Policies).
- **GUI Metamodel** — defines how data and screens are organized and visualized.

2. Structural Metamodel

The **Structural Metamodel** defines the logical and data core of the system (based on systems like **OLX**, **Vinted**, **Facebook Marketplace**, and **Freecycle Network**). Main entities include:

- **Community** and **SubCommunity**: represent organizational boundaries.
- **User**: participants with roles (admin, moderator).
- **Item**: main unit of trade, with attributes like price, condition, and modes (Price, Exchange, Donation).

We created OCL Rules with the aim of ensuring consistency across the different layers of the application. Below are some of the OCLs:

- Prices must be non-negative.
- Donations cannot define prices.
- Ratings range from 1 to 5 stars.
- Sales must have a payment method different from “None.”
- Usernames must be unique within a community.

3. DSML Metamodel

The **DSML** enables each community to **customize** its generated application without changing the source code. It defines optional and mandatory features through a **feature model** represented in PlantUML mindmap form.

The DSML was inspired by **software product-line feature models**, applying declarative configuration principles to simplify product derivation. We adapted payment options to the **Portuguese context** (MBWay, Multibanco, …) and structured all features to support **variability and reuse** across different communities, enabling automatic generation of multiple, domain-specific versions of the application.

Main configuration areas:

- **Accounts**: local login, OAuth, phone verification, moderators.
- **Listings**: products, services, modes of exchange, expiry, variants.
- **Messaging**: chat and images in chat.
- **Ratings**: simple or bidirectional.
- **Payments**: MB Way, Multibanco, PayPal, or None (XOR).

OCL Constraints ensure valid configurations:

- At least one trade mode must be active.
- Only one payment method can be selected.
- Payments require the “Price” mode to be enabled.

Through the DSML, the same system can generate many different community-specific apps.

4. GUI Metamodel

The GUI metamodel shows the structure of the application’s user interface and defines the rules for how data will be organized and shown to the user.

It acts as an abstract framework that allows generating different GUI models within the same family of applications. They may differ in content and settings, but they all share the same logic and general interface layout.

This metamodel brings together the main ideas found in popular product and service exchange apps such as *OLX* and *Freecycle Network*. Even though these apps are not strictly local, their interface structure, including lists of items, item views, user profiles and user communication, inspired the design of our metamodel. Our main goal was to create a simple but flexible GUI metamodel that can be used to generate both web and mobile apps for different kinds of local communities such as school, work or neighborhood groups. The design follows the model-driven approach that lies at the core of the BESSER framework.

The metamodel has a clear hierarchy based on the *GUIModel* class, which includes all screens of the application. Each screen, represented by the *Screen* class, combines visual elements (*ViewElement*) and containers (*ViewContainer*). The way users move between screens is defined by *NavigationAction*, while the *Binding* class connects specific data attributes with the visual components that display them. The *DataSource* represents the available information, and the *Binding* determines how this information is visually mapped to each GUI element. This makes the connection between data and interface dynamic and consistent, turning the GUI metamodel into a universal and adaptable foundation for different domains and types of applications.

5. Conclusion

Part I, deliverables provide a complete base for generating local community exchange applications automatically using the BESSER framework. The **Structural**, **DSML**, and **GUI** metamodels together define:

- Data structure and integrity.
- Configuration flexibility.
- Interface generation logic.

6. Next Steps (Part II):

1. Create a **graphical syntax** for the DSML using PipeBlocks/Model2Block.
2. Define **model-to-model (M2M)** transformations to generate structural and GUI instances.
3. Implement **model-to-text (M2T)** generation for web and mobile versions.
4. Deploy one example app for real use within a local community.