Final Report

IC card management system (ICMS) feature extension

# 1.Introduction

The IC card Management System (ICMS) offers a graphical interface to manage ICCards. It is possible to create, edit, group, and visualize ICCards. The system also provides the user with access control functionality that allows each user to control his own set of ICCards. Also, it is possible to classify ICCards as representations of patterns and export as standardized XML documents.

# 2.Implementation

My task was to extend the current feature to decompose or transform ICCards. The current implementation forced the user to decompose/transform ICCards before their creation trough static questions that were hard to modify and did not applied to every scenario. Therefore, a possible improvement would be to implement it in a way that would allow existing ICCards to be decomposed or transformed. For this end it is necessary to consider existing ICCards properties to foster this process. My approach was to add a new classification to each ICCard and also consider their interaction pattern. The new IC card creation interface is presented in Figure 1.

 

**Figure 1.** Added classifications in ICCards

Now while creating new ICCards the user can also classify their type as:

* Hybrid: A ICCard that has both human and software components.
* Organic: A ICCards that has mainly human interaction and activities.
* Inorganic: A ICCards that has mainly software interaction and activities.

With this new classification the system can now be aware of possible ICCards that can be further refined since Hybrid ICCards have both human and software components and is inevitably decomposable into lower level components. Also, it is possible to define ICCards tasks as complex, meaning that it can also be decomposed into 1 or more lower level components.

The new feature is added in the Project screen as shown by Figure 2.



**Figure 2**. Decomposition/Transformation feature

This feature allows the user to choose which cards he wants to decompose or transform. Also, the system uses the ICCards pattern interaction combined with its type (i.e., Hybrid, Organic or Inorganic) to organize a list ordered by the likelihood that the cards can be considered to further decomposition or transformation. This is show in Figure 3.



**Figure 3.** ICard list ordered by likelihood of being decomposed or transformed.

As show, each ICCard property (Type, Complexity and Interaction pattern) are associated with weights that are summed and then used to order the list. An initial proposal for weights are presented below:

* Hybrid: 5
* Complex Card: 3
* Green (Myself no interaction): 5
* Yellow (Myself with interaction): 4
* Orange (Others no interaction): 3
* Red (Other with interaction): 2
* Purple (Mixed state): 1

Hybrid cards get high score since they are a composite. Also, Green ICCards get a high weight given that isolated components tend to be hidden super component that can be further subdivided. Conversely, Purple cards get a lower weight given that components that are highly interconnect tend to be defining lower level tasks.

After choosing an ICCard the user can then decompose it into one or more ICCards as presented in Figure 4.



**Figure 4.** A green IC Card being decomposed.

An ICCard can then be further transformed into one or decomposed into two or more ICCards. The system can now guide and preset the properties of decomposed ICCard. For example, now it is possible to classify the siblings of decomposed Hybrid ICCard into Organic and Inorganic automatically. The guiding procedure can easily be extended.

# 3. Code Changes

There have been a significant number of changes in the code, but they were isolated. Four new columns were created in the IC\_CARD table were created to support the card properties and two new html pages. The code is now being versioned under GIT and the details of the modifications can be found in its history.

# 4. Future work and conclusion

As presented the ICCard decomposition/transformation have been successfully implemented. The current implementation is much more extensible, and many features can be implemented, such as keeping track of ICCard of decomposition and transformation history or a better automatic guidance during the decomposition. For example, the system can dynamically present tips or questions given the target ICCard.