CS2310 TOY Project Milestone 1

Kavya Sree Kilari (kak435@pitt.edu)

1. TOY Project Description

The purpose of the TOY Project is to promote intergenerational learning (IG) and create new possibilities for older adults and young children to learn together and benefit from each other's company. This project focuses on TOY scenario that is 'Together Young and Old'.

This handles the scenarios involving a child and a senior citizen.

The application allows enrolment of senior citizens as well as children. The senior citizen can enrol themselves whereas the children are enrolled by their parents. So, the information related to senior citizens will be stored separately and the children information will be stored along with parent contact information separately.

The primary scenario where the senior citizen takes care of the child who is left for sometime by the parent at the TOY station is handled in this application.

The supercomponent called 'Emergency' is present that will handle complex scenarios like for example when the parent does not return by the end of allotted schedule. This supercomponent will communicate with other components as shown in the *Deployment diagram* below to handle such situations.

In application perspective, there will be a UI that contains the required fields to input information related to different components involved. The child-parent information, senior citizen information, schedules available are all shown on the UI. The TOY tool can be used at the TOY Station for a parent to sign up. Similarly the parent can sign up a child.

The logic is performed on this input and the information is used by the application to handle the schedule, handle emergency situation etc. The TOY tool suggests a suitable senior citizen to take care of the child while the parent is away using the logic. The TOY project monitors the interactions between the kid and senior citizen.

The information is stored in respective database servers for different components. This data will be used to suggest available schedules a parent can choose from and leave the child at the TOY station.

The information regarding the various components involved in the application are shown in detail in the deployment diagram *Figure 1*.

Below is an example of how the scenario would be executed:

- A senior citizen signs up in the TOY application by inputting data in the UI. This is senior citizen registration. This data will be stored in senior citizen database.
- A parent signs up the child by inputting the child and parent contact information using the UI. This data will be stored in Child database.
- After registration, the TOY station shows the available schedules for different available senior citizens in the system. The parent can choose a schedule and enrol to it.
- A schedule will be created containg senior citizen-child information and the activities are monitored during the schedule.
- At the end of schedule, a parent returns and claims the child, This will end the schedule.
- If in case the parent does not return, the schedule will be handled by supercomponent by assigning a new activity by contacting the senior citizen.

CS2310 TOY Project Milestone 1

Kavya Sree Kilari (kak435@pitt.edu)

2. Deployment Diagram

Description: This project focusses on building TOY system that deals with TOY(Senior-Child) scenarios. The Deployment diagram below shows various components involved in the system.

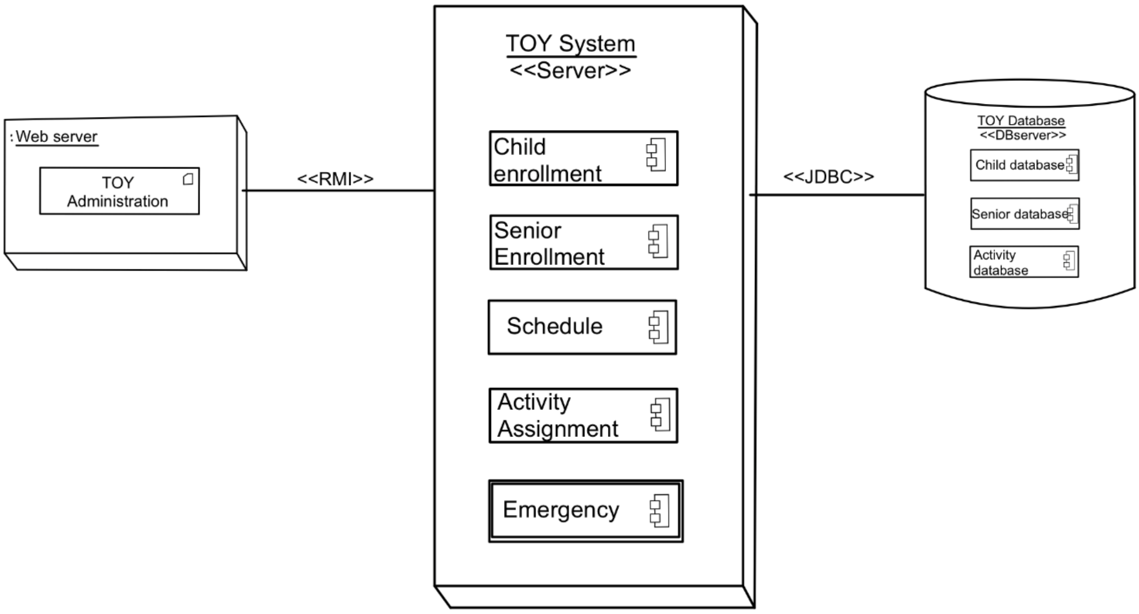

Figure 1. Deployment Diagram

TOY Administration: is a web server that contains UI with web pages that allow Senior enrolment, Child enrolment, TOY Activity assignment information.

TOY System: This is the server that is responsible for the logic behind TOY Administration. It contains multiple components each dealing with different aspects of the system and allows cross communication within the components.

Child component: Deals with enrolment of a child on a request from the parent. The logic checks the age, health condition etc of the child.

Senior component: Deals with enrolment of a senior. Collects information of the senior availability timings, health information, criminal background etc.

Schedule: Manages the logic to schedule an activity by assigning a child to the senior based on the senior's availability. Monitors the timing of the schedule.

Activity Assignment: Contains logic to assign an activity for the allotted schedule. Has logic to choose an activity from the available resources.

Emergency: Acts as a supercomponent. Handles different scenarios of emergency and acts accordingly. This collaborates between above components in TOY system to provide right solution for the situation. For example, during a schedule, if the child refuses to continue with the ongoing activity, this component receives the message and provides an alternative activity for the remaining time. This component handles the NOVEL feature of managing the activity under emergency situation.

TOY Database: Contains multiple database components each containing data related to respective components of the TOY system like child, senior, activities. The TOY system uses data from this database to perform logic for the given scenario.

3. I-Card and C-Card Diagram:

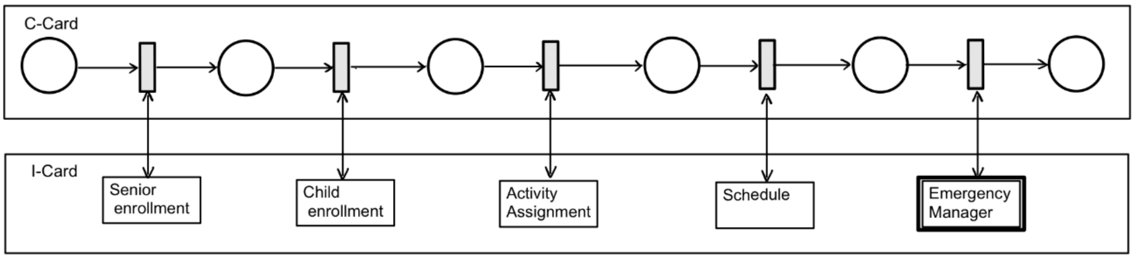

Figure 2. I-Card and C-Card Diagram

C-Card diagram shows the control part of the TOY system for the TOY scenario.

From left to right, each place represents an activity. The corresponding transition handles the input from the place.

I-Card diagram shows the classes that contain information to handle the activity at each transition. The supercomponent Emergency Manager is represented using double edged box.

The I-Card and C-Card are connected to show that they act together to represent the system.

For example, in C-Card, at first place, a senior requests for the enrolment into TOY system. The transition accepts this input and checks with Senior Enrolment class in the I-Card and performs the necessary actions.

4. Supercomponent

Super components can act as building block in developing slow intelligence systems that is they are capable of improving their performance overtime in a changing environment. Since the super-component enumerates the different solutions by executing different algorithms, the expansion into parallel/serial Petri nets can take on different forms. We can make the execution of the algorithms all serial, all parallel, probabilistic, and so on.

A supercomponent can be used ti handle scenarios in different ways like serial, parallel, probabilistic. In a supercomponent multiple components can work together to handle that given scenario.

CS2310 TOY Project Milestone 1

Kavya Sree Kilari (kak435@pitt.edu)

Below are the diagrams showing supercomponent using I-card and scenarios handled using I-card and C-cards.

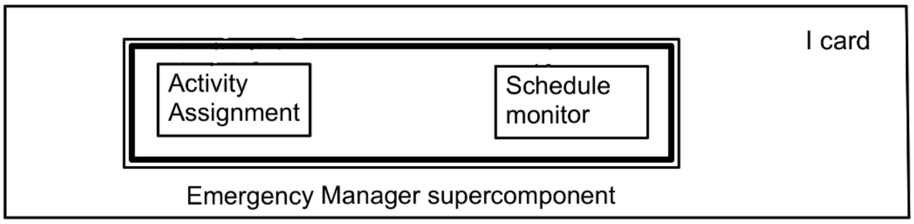

Figure 3. I Card for Emergency super component

The I card above shows the Information card diagram for Emergency super component that contains Activity assignment and Schedule monitor class components to handle different emergency scenarios in TOY project.

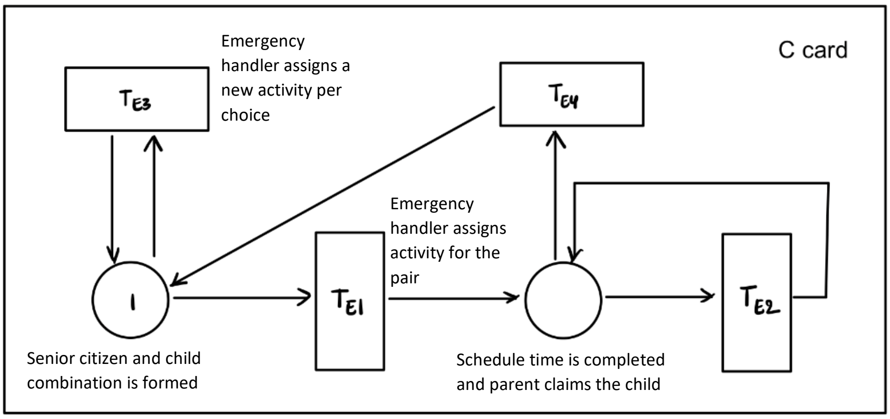
C card
Emergency handler assigns a new activity per choice
T_{E3}
T_{E4}
Emergency handler assigns activity for the pair
I
T_{E1}
T_{E2}
Senior citizen and child combination is formed
Schedule time is completed and parent claims the child

Figure 4. C card illustration for super component

The C card diagram shows control flow of an activity using super component. After the senior citizen and child information is registered, a schedule can be picked to assign activity. Transition 1 handles normal situation where the Emergency handler supercomponent assigns the activity for the pair.

Once the schedule time is completed, the parent comes back and claims the child which is shown in place 2. A parent can redo the schedule for next hour which is handled by transition 4. This again repeats transition 1 internally.

Below diagrams illustrate the communication between I card and C card where the control takes help of the information to handle different scenarios. As discussed above, the diagrams also show situations using enumeration types like parallel and serial.

CS2310 TOY Project Milestone 1

Kavya Sree Kilari (kak435@pitt.edu)

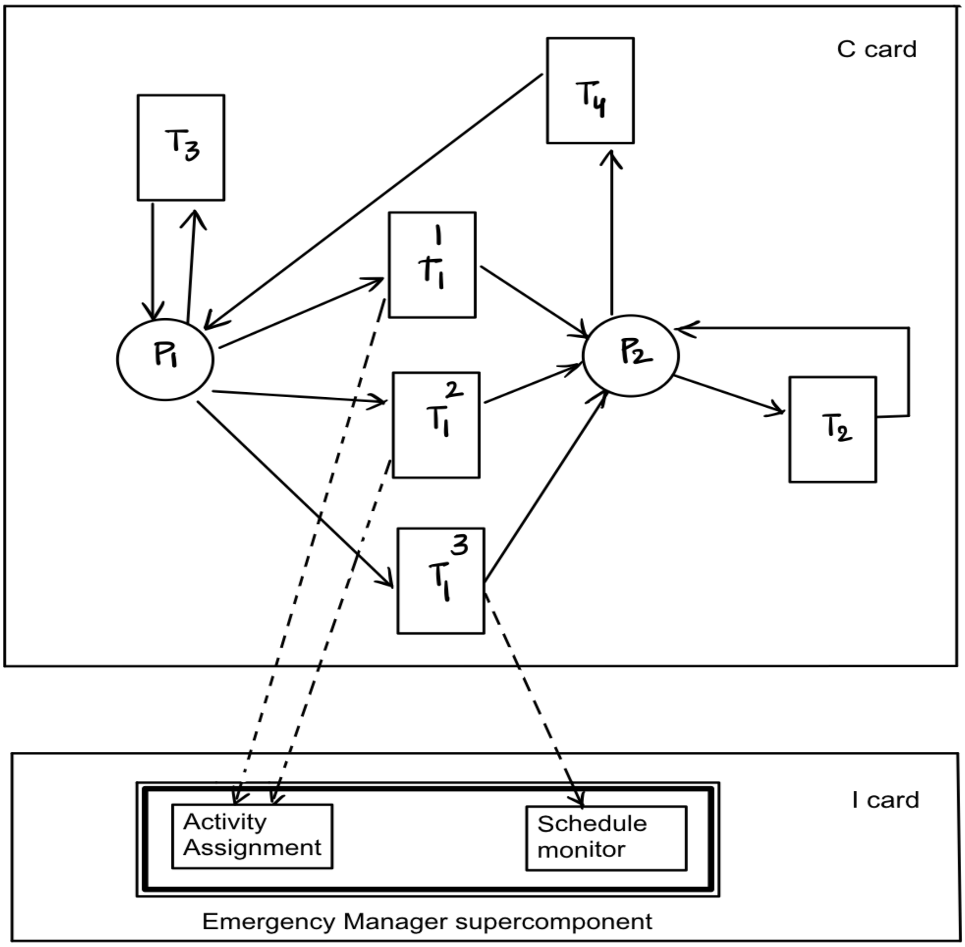

Figure 5. I card C card super component with parallel enumeration

Parallel Enumeration can be handled by super component by expanding the petri net. Following the scenario described in the C card diagram in Figure 4, a super component can assign multiple activities for multiple senior citizen-child pairs parallelly using the activity assignment class in the Emergency supercomponent. Parallelly, it can handle input from transition T4 i.e., redo the schedule of activity by updating the schedule monitor which is shown using Transition T1 marked as 3.

Serial enumeration can be handled by super component by expanding the petri net. Referring to scenario in Figure 4, the Emergency super component can handle a scenario that happens in sequence of steps one after the other. As shown in the below diagram, Place P1 is when the senior-child pair is created. Transition T1 1 with the help of Activity assignment class assigns the activity for the pair. The schedule for this pair is ended whish is shown by P2. Transition T1 2 handled the end of schedule by checking with Schedule monitor class. This is when the parent does not visit the TOY station to claim the child shown as P3. The transition T1 3 with help of activity assignment creates new activity for the next schedule. P4 is when the parent finally returns to claim the child OR wants to extend the child time in the station by creating new schedule. Now that the same senior has already taken care of the child for 2 hours, a new senior-child pair will be created. This is one by Transition T4.

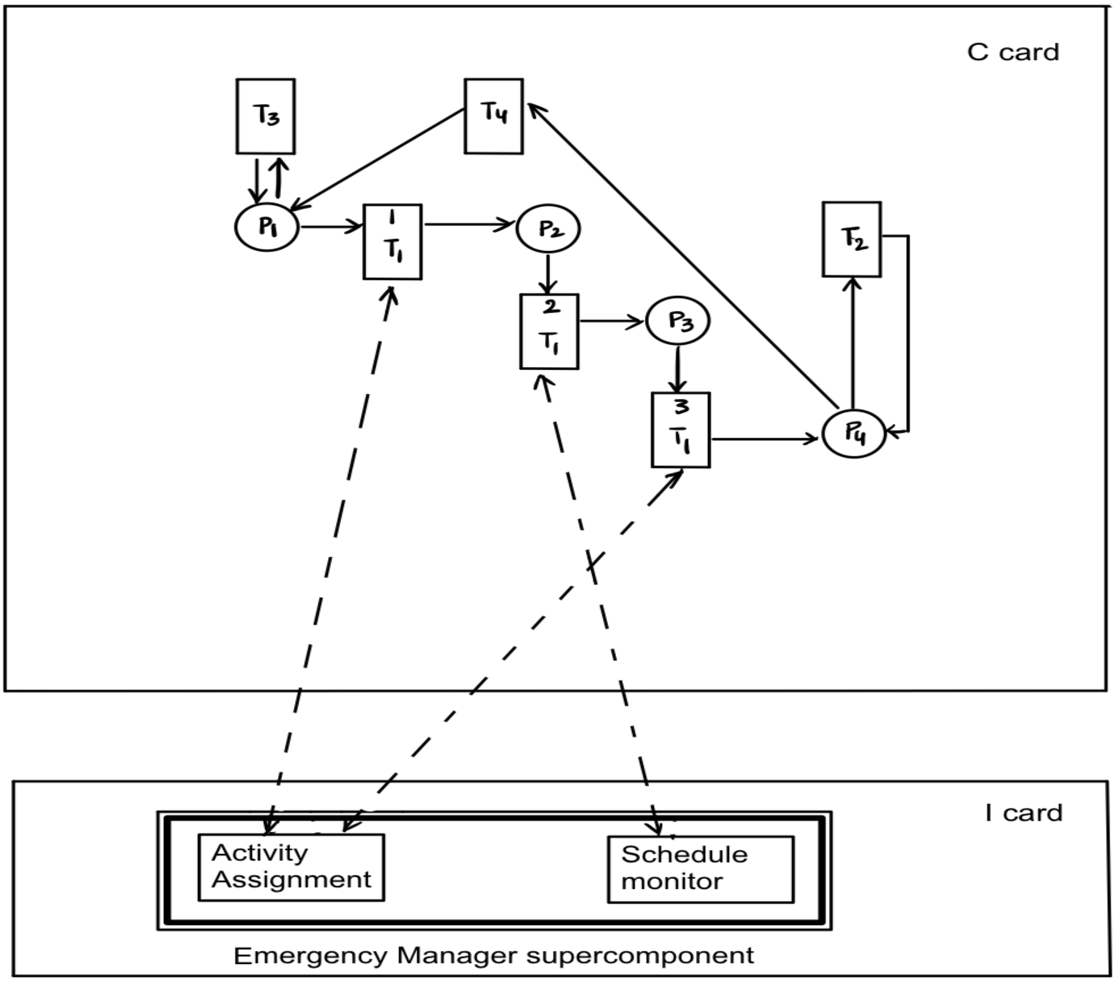

Figure 6. I card C card super component with serial enumeration

The working part of the project will be implemented as part of the project Milestone 2. The web server for UI, logical for the super component, and the database servers will be part of this implementation of this TOY project.