

## MTAT.03.083 – Systems Modelling

### Regular Exam – 13 January 2015

#### Notes:

- The exam is open-book and open-laptop. Web browsing is allowed
- You are not allowed to communicate with anyone during the exam in any way (except with the lecturer).
- You may submit your exam on paper, or electronically using the “Submit” button in the course Web page.
- If you find that there is not enough information in the text below and you need to make additional assumptions, please write down your assumptions.

## Part 1. Trip Planner

### Part 1.1

The following is a map of the four Tallinn tramway routes. A similar map could be drawn for trolleybus and for bus routes.



Figure 1. Map of Tallinn's Tramway lines as of December 2014

**Task 1. [5 points]** Draw a domain diagram for tramway routes such as the one shown above. Note that every tramway stop is associated to a GPS coordinate. Also, note you are not asked a model of the map itself, but only of the routes and location of stops. The domain model should also be able to cover trolleybus and bus routes.

*Part 1.2*

Below is a sample schedule of tramway line number 4. The schedule is given starting from one of the terminals of the route, namely Vana-Lõuna. There is a column missing in the picture corresponding to the “Sunday” schedule, which is similar to Saturday but with slightly different departure times. You can view the full schedule here: <http://soiduplaan.tallinn.ee/#tram/4/a-b/en>

**4 Vana-Lõuna - Ülemiste ▼**

Stop <b>Vana-Lõuna</b> (fare zone 1) <b>4</b>			
<b>4</b> Route streets: Pärnu mnt, Narva mnt, Tartu mnt, Majaka tn, Peterburi tee			
	<b>Monday-Friday</b>	<b>Saturday</b>	
Vana-Lõuna	<b>5</b> 20 35 50	<b>5</b> 55	<b>5</b>
Vineeri	<b>6</b> 00 10 18 26 34 42 50 57	<b>6</b> 15 30 45	<b>6</b>
"Kosmos"	<b>7</b> 04 11 17 23 28 34 39 45 51 56	<b>7</b> 00 15 30 45	<b>7</b>
Vabaduse väljak	<b>8</b> 02 08 13 19 24 28 33 39 44 50 56	<b>8</b> 00 10 20 30 40 50	<b>8</b>
Viru	<b>9</b> 02 08 13 19 25 32 39 46 53	<b>9</b> 00 10 19 28 37 46 54	<b>9</b>
Hobujaama	<b>10</b> 00 07 14 21 28 35 42 49 56	<b>10</b> 02 10 18 26 34 42 49 56	<b>10</b>
Paberi	<b>11</b> 03 10 17 24 31 38 45 52 59	<b>11</b> 03 10 17 24 31 38 45 52 59	<b>11</b>
Keskurg	<b>12</b> 06 13 20 27 34 41 48 55	<b>12</b> 06 13 20 27 34 41 48 55	<b>12</b>
Autobussijaam	<b>13</b> 02 09 16 23 30 37 43 50 57	<b>13</b> 02 09 16 23 30 37 44 51 58	<b>13</b>
Lubja	<b>14</b> 03 10 16 22 28 35 41 47 53 59	<b>14</b> 05 12 19 26 33 40 47 54	<b>14</b>
Majaka	<b>15</b> 05 11 17 22 28 34 39 45 51 56	<b>15</b> 01 08 15 22 29 36 43 50 57	<b>15</b>
Sikupilli	<b>16</b> 02 08 14 19 25 31 36 42 48 53 59	<b>16</b> 04 11 18 25 32 40 48 56	<b>16</b>
Majaka põik	<b>17</b> 05 10 16 22 27 33 38 43 49 55	<b>17</b> 04 12 20 28 36 44 52	<b>17</b>
Väike-Paala	<b>18</b> 00 05 11 17 23 28 34 40 47 54	<b>18</b> 00 09 17 26 34 43 52	<b>18</b>
Ülemiste	<b>19</b> 01 08 15 22 29 37 44 52 59	<b>19</b> 01 10 19 28 37 46 55	<b>19</b>
	<b>20</b> 07 15 23 31 39 47 55	<b>20</b> 05 15 25 35 45 55	<b>20</b>
	<b>21</b> 05 15 25 35 45 55	<b>21</b> 05 15 25 40 55	<b>21</b>
	<b>22</b> 10 25 40 55	<b>22</b> 10 25 40 55	<b>22</b>
	<b>23</b> 10 30 50	<b>23</b> 10 30 50	<b>23</b>

**Figure 2. Schedule of Tramway line #4 starting from the Vana-Lõuna stop**

**Task 2. [8 points]** Design a new domain model by adding all information required to generate the schedule of a given tramway line starting from a given stop (you can use the model from the previous question as a basis, but this should be submitted as a separate model). Please assume that the tramway takes a given amount of minutes (e.g. 1, 2 or 3 minutes) to go from a stop to the next one. The number of minutes it takes from the tramway to go from one stop to another is the same regardless of the week, but differs depending on the pair of stops being considered, e.g. between Lubja and Majaka it takes 1 minute, whereas between Majaka and Sikupilli it takes 2 minutes. Note that the domain model produced for this task does not need to represent the concept of “schedule” itself, but only the information required to generate a schedule.

### Part 1.3

A trip planner is an application that allows a user to enter a departure point, a destination point, and a time of departure, and returns a plan for going from the departure to the destination via public transport. The departure and destination points may be an address, a name of a stop or a place of interest (e.g. National museum). If the string entered for the departure (or destination) is incorrect, the application returns the most similar points it knows about (e.g. similar addresses), and allows the user to select one of the proposed alternatives or to type in a new departure or destination address, until the departure and destination points are recognized. If both the departure and destination points are recognized, the trip planner returns a plan.

The plan returned by the trip planner consists of *travel legs*. Some travel legs involve walking from one point to another (e.g. from an address to a given stop or vice-versa), while other legs involve taking a tramway or bus from one stop to another. As an example, the trip planner of the city of Berlin is available here:

- <http://fahrinfo.bvg.de/Fahrinfo/bin/query.bin/en>

For example, the following is a plan produced by the Berlin travel planner. In this exam, we do not consider anything associated with “fares” (cost of tickets).

**BVG** Time: 14:30  Departure  Arrival Date: Sa, 27.12.14

#### Results

Station / Stop	Time	Duration	Conn.	with	Fare*
15738 Zeuthen, Nürnberger Str. 8	14:39 dep				3,20 €
U Karl-Marx-Str. (Berlin)	15:40 arr	1:01	1		2,30 €
14:39 dep	<b>15738 Zeuthen, Nürnberger Str. 8</b>				
Footpath	ca. 1,5 km				
15:11 arr	<b>S Eichwalde</b>				
15:11 dep Platform 4					
S46	<b>Direction S Westend (Berlin)</b> barrierefrei, Fahrradmitnahme möglich operates in 20 -minute intervals.				
15:34 arr Platform 1	<b>S+U Neukölln (Berlin)</b>				
15:34 dep					
Footpath	Ausgang zur U-Bahn				
15:37 arr	<b>S+U Neukölln (Berlin) [U7]</b>				
15:39 dep					
U7	<b>Direction S+U Rathaus Spandau (Berlin)</b> barrierefrei, Fahrradmitnahme möglich operates in 5 -minute intervals.				
15:40 arr	<b>U Karl-Marx-Str. (Berlin)</b>				

Figure 3. Plan produced by the Berlin trip planner from Nürnberger Str. 8 to Karl-Marx-Platz

**Task 3. [12 points]** Design a new domain model for a trip planner (you can use the model from the previous question as a basis, but this should be submitted as a separate model). The model should capture all the information required to generate a plan, and it should capture the generated plan itself. You may assume that for every address or place of interest, the corresponding GPS coordinates are available. You may also assume that given two GPS coordinates, we can calculate the distance and the walking time between these two coordinates.

**Task 4. [4 points]** Write the main use case of the travel planner, namely “Create travel plan”.

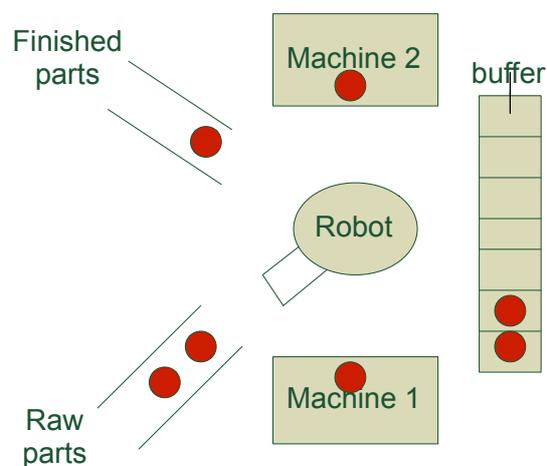
**Task 5. [5 points]** Draw a sequence diagram showing how the classes in the domain model of Task 3 interact in order to calculate a route.

**Task 7. [4 points]** Refine the domain model of Task 3 into an application model.

## Part 2. Petri net

We consider a segment of a factory with two conveyor belts, two machines, one robot and one buffer (see Figure 3). Raw parts arrive through a first conveyor belt, called the *raw line*. The robot moves each part from the *raw line* into machine M1, then into the buffer, then into machine M2 and finally to the second conveyor belt (called the *finished line*). If machine M2 is free, the robot is also allowed to move a part directly from M1 to M2.

M1 can hold at most one part at a time, and the same applies for M2. The robot can only move one part a time. The buffer can hold at most 7 parts. The conveyor belts can hold any number of parts. The following figure shows the case where the raw line holds 2 parts, M1 and M2 hold one part each, the buffer holds two parts, and the finished line holds one part.



**Figure 2:** Factory Line Structure

This system (in its initial state) is modelled as a Petri net in Figure 3.

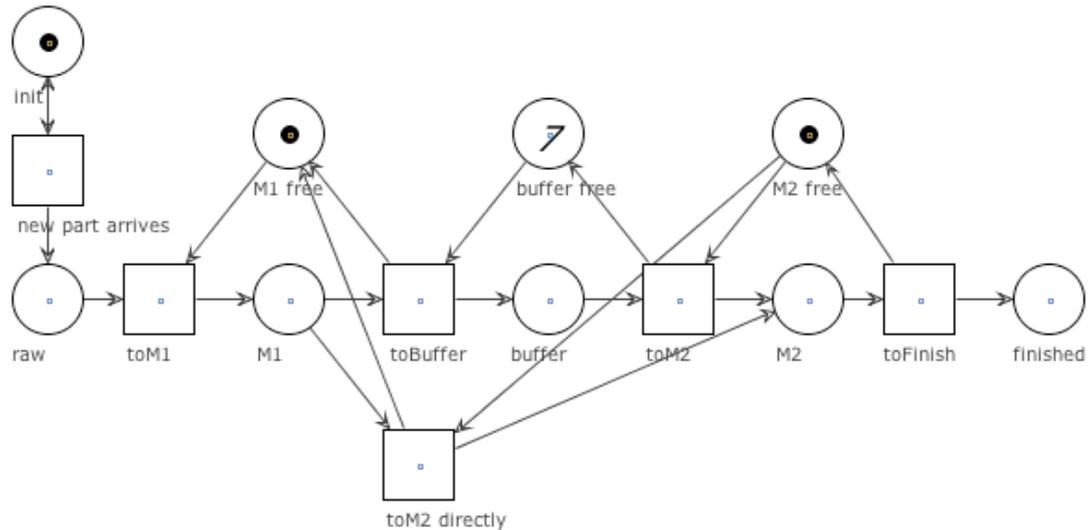


Figure 4. Petri net of the factory line

**Task 8. [12 points]** Modify the Petri net in Figure 4 so that machine 1 and machine 2 can break down and be repaired. If a machine breaks down, a part cannot be moved to it until the machine is repaired. If, when a machine breaks down, a part is in it, the part is moved to a third machine (machine 3) for recycling. From each part moved to machine 3, a new raw part is obtained. When machine 1 or machine 2 break more than twice the entire machine must be replaced and the entire system must be restarted. In this case, before restarting the system must be emptied and all the parts inside the system recycled.