**PARKme System**

**Proof-of-Concept Model**



George Mason University

SYST 798, Prof. Speller

**Craig Emmerton**

**Earl Morton**

**Shaun McDonald**

**David Richards**

**Nikki Torres-Avila**

Table of Contents

1.0 OVERVIEW 3

1.1 Definition 3

1.2 System Architecture 3

1.2.1 Over Arching Architecture 3

1.2.2 Human Interface 3

1.2.3 Connectivity Network 3

1.2.4 Main System 3

1.2.5 Parking Space Sensor 3

1.3 System Perks 3

1.4 Assumptions 4

1.4.1 Scalability 4

1.4.2 Timing 4

1.5 PED Versus Electronic Sign Instantiations 4

2.0 MODEL 4

2.1 The Model 4

2.2 CPN Concept Diagram 5

2.3 System Data Flow 5

2.4 Model parameters 5

2.4.1 Car Tokens (Driver) 5

2.4.2 Parking Space Tokens (Space Update) 5

2.4.3 Parking Lot Tokens (lotList) 6

2.4.4 Display Token (Updated display Val) 6

2.4.5 Top Level Model (Used for Both PED and electronic Signs) 6

2.4.6 CPN Model Declarations 7

2.4.7 CPN Model Declarations Continued 8

2.5 CPN Model Variants 9

2.5.1 PED Screen Captures 9

2.5.1.1 PED DRIVER Module 9

2.5.1.2 PED PARKme Spaces Locator Module 10

2.5.1.3 PED Human Interface Module 11

2.5.1.4 PED PARKme Parking Lot Module 12

2.5.2 Electronic Signage Screen Captures 13

2.5.2.1 Signs DRIVER Module 13

2.5.2.2 Signs PARKme Spaces Locator Module 14

2.5.2.3 Signs Human Interface Module 15

2.5.2.4 Signs PARKme Parking Lot Module 16

1.0 OVERVIEW

1.1 Definition

The PARKme system is designed to direct a user entering a facility to the closest parking lot with available parking spaces. This could be implemented in one of two current scenarios. The first is as set of electronic signs directing them to the closest lot to there current location. The second is the user accessing data specific to them that would direct them to the parking lot nearest to a pre-defined final location at the facility.

1.2 System Architecture

1.2.1 Over Arching Architecture

The architecture will consist of many independent plug-and-play sub-architectures. These will include the Main System, Parking Space Sensors, Communication Network, and Human Interface.

1.2.2 Human Interface

The human interface is a means for the user to access the information on the availability of parking spaces at the facility.

1.2.3 Connectivity Network

The connectivity network is the means in which the system relays data between the system components and the system users.

1.2.4 Main System

This is the main control module for the system. This unit will interpret the signals received from the parking space sensors. It will use this data to generate the status of the parking lots at the facility. This information is then relayed to the human interface modules.

1.2.5 Parking Space Sensor

The sensors will transmit the availability status of individual parking spaces.

1.3 System Perks

The system will also model several additional features of the PARKme system. The ability to monitor fire lane parking and notify security will be modeled. The model will also represent a sensor failing and maintenance monitoring the system. The system will also collect statistics on parking patterns.

1.4 Assumptions

This model will make a number of assumptions to define the scope of the problem.

1.4.1 Scalability

The model will assume the system is fully scalable. Thus the model will represent two entryways into the facility. Two parking lots will be modeled with two to three parking spaces per lot. Also only two buildings located at the facility will represented.

1.4.2 Timing

The model will include time, but time will not be reflective of the true system. The time it takes for the user to walk from the parking space to there destination building is a constant. The time it takes for the user to arrive at the facility perimeter is also a constant. Time is not a factor in the sensors, main system, or connectivity network. Any delta’s in these sub-architectures is relatively small compared to human perception.

1.5 PED Versus Electronic Sign Instantiations

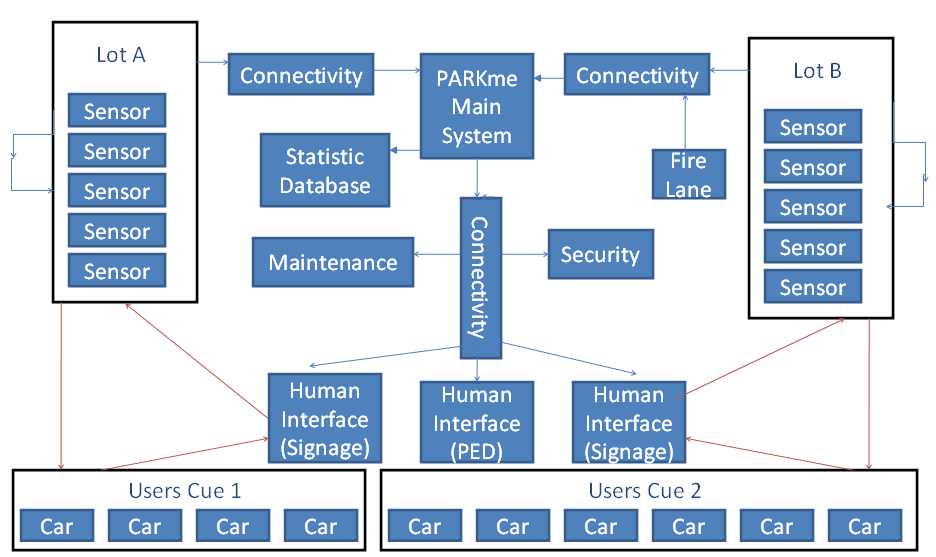
Two variants of the model will be created. The Human Interface sub-architecture will be modeled as both electronic signs and as PED’s.

2.0 MODEL

2.1 The Model

The models are implemented as a colored Petri-Net using the CPN tools software. The models are fully executable from within the CPN tools software. Both variants of the models are included as CPN files.

2.2 CPN Concept Diagram



2.3 System Data Flow

The model will include two types of tokens. One data flow path will represent the users and how they progress through the system. The other data flow path will represent the internal system data and how it is passed through the system.

2.4 Model parameters

2.4.1 Car Tokens (Driver)

Each car is represented as an integer. This integer number denotes the users ID. For the PED variant this ID is used to access a database and determine the users preferred parking location.

2.4.2 Parking Space Tokens (Space Update)

Each parking space is represented as an integer and a Boolean. The integer is the space ID and the Boolean is its status, either occupied or empty.

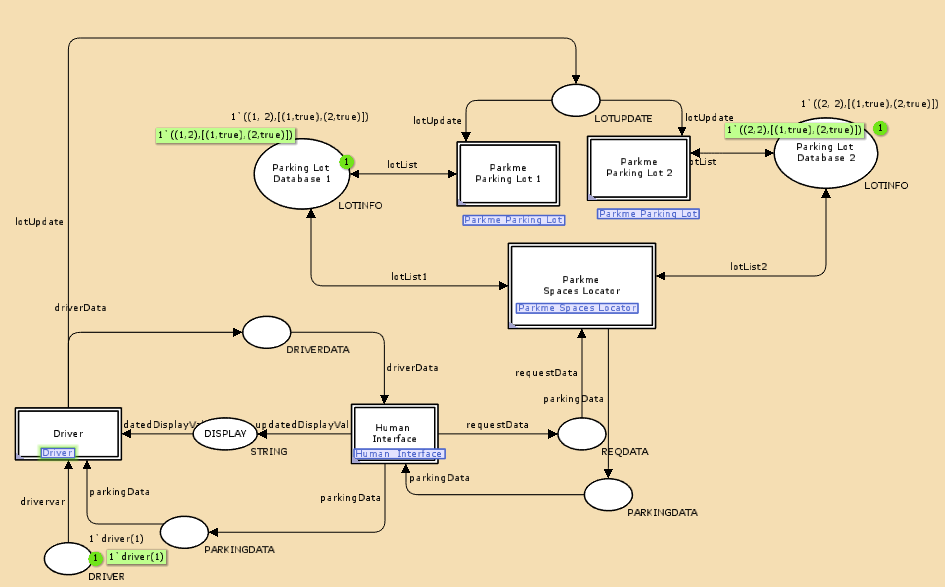
2.4.3 Parking Lot Tokens (lotList)

Each parking lot is represented as two integers and this is combined with the parking space data. The two integers are the parking lot id and the number of free parking spaces contained within it.

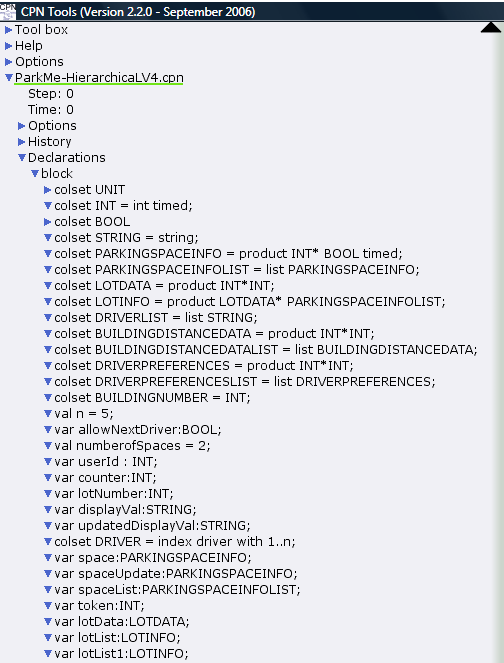
2.4.4 Display Token (Updated display Val)

This is modeled as a string. For the PED variant this is the message that is sent to the users PED.

2.4.5 Top Level Model (Used for Both PED and electronic Signs)



2.4.6 CPN Model Declarations



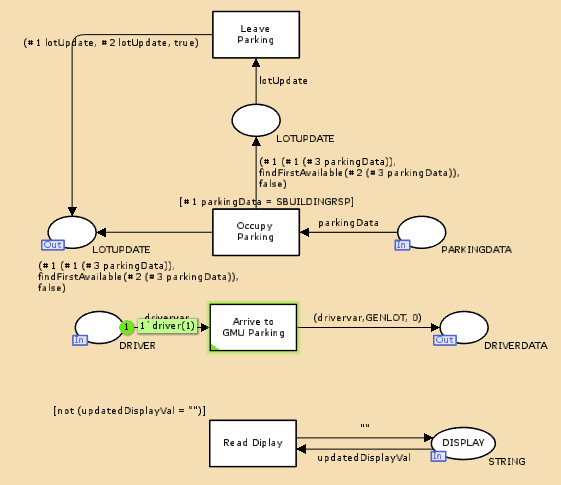
2.4.7 CPN Model Declarations Continued



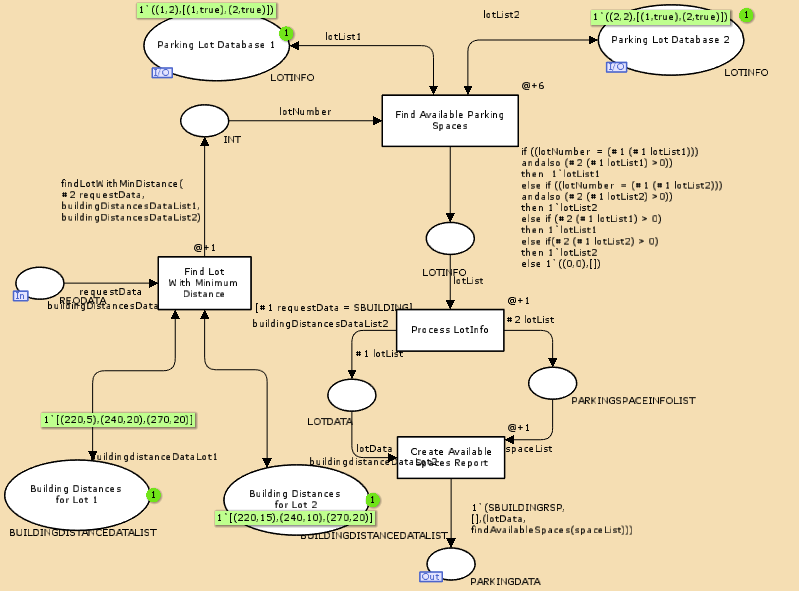
2.5 CPN Model Variants

2.5.1 PED Screen Captures

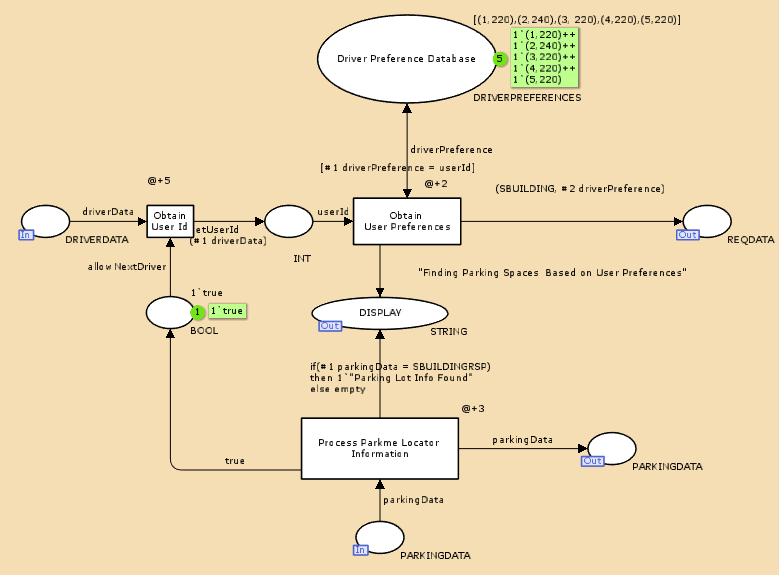
2.5.1.1 PED DRIVER Module



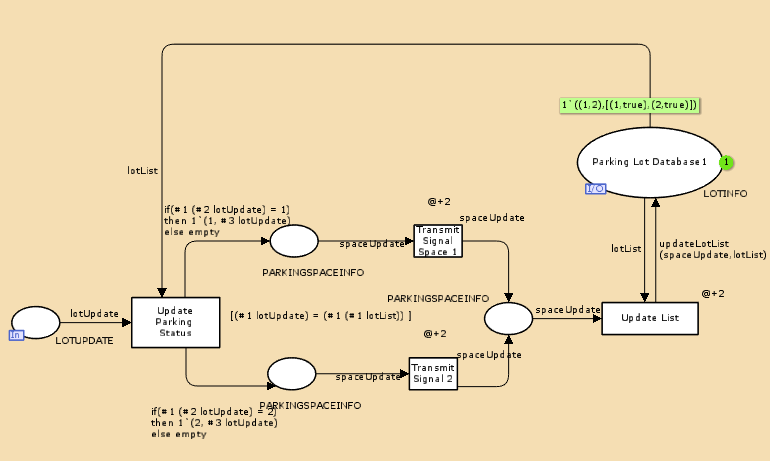
2.5.1.2 PED PARKme Spaces Locator Module



2.5.1.3 PED Human Interface Module

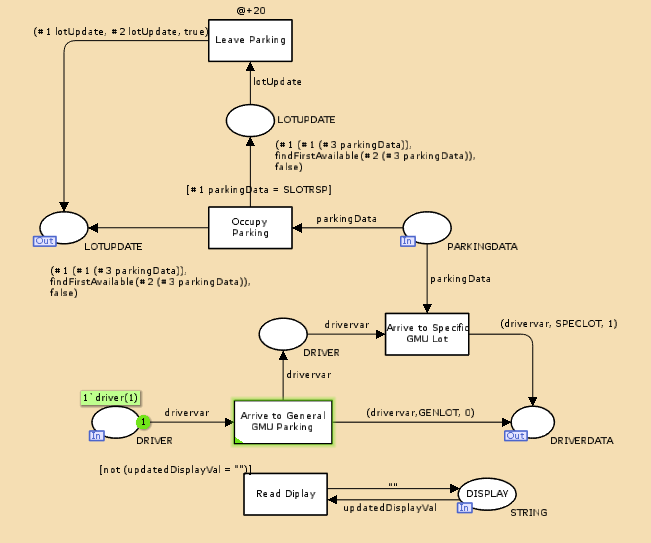


2.5.1.4 PED PARKme Parking Lot Module

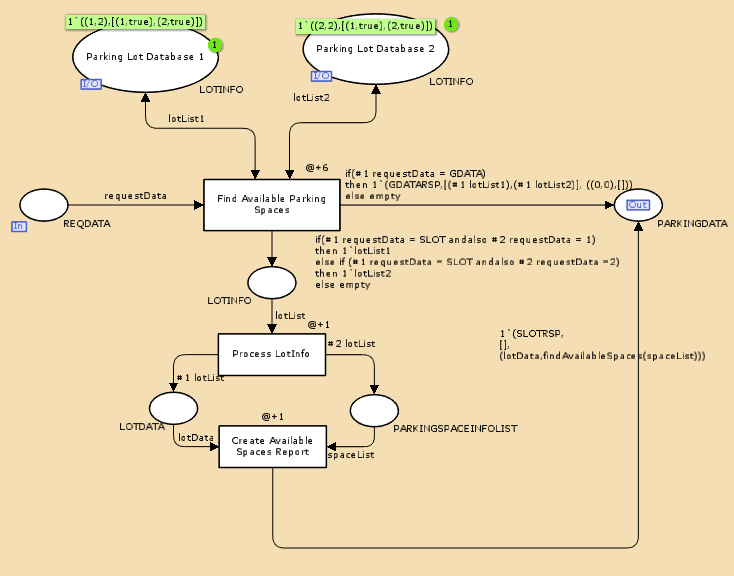


2.5.2 Electronic Signage Screen Captures

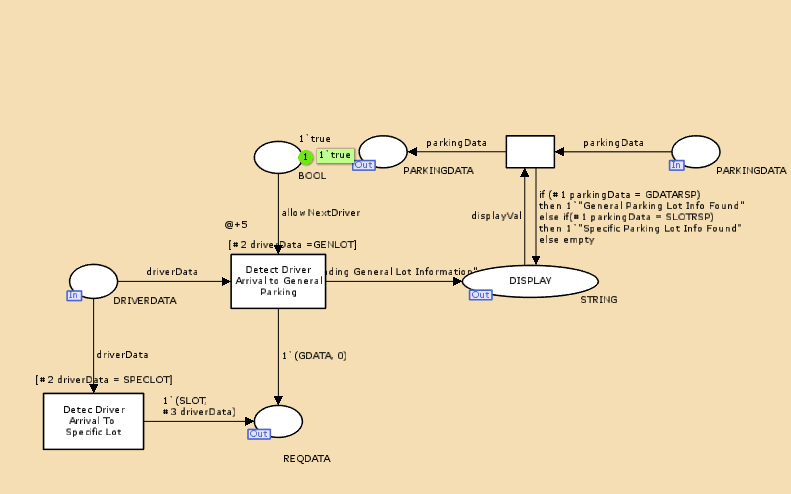
2.5.2.1 Signs DRIVER Module



2.5.2.2 Signs PARKme Spaces Locator Module



2.5.2.3 Signs Human Interface Module



2.5.2.4 Signs PARKme Parking Lot Module

