

## ELECTRONIC TOLL COLLECTION AT PARKING AREAS

The paper presents an object-oriented modeling of electronic toll collection in transport applications, deals with designing of a use case diagram, class diagram, sequence diagram, and state chart. The modelling of parking system activities by the UML language (Unified Modeling Language) creates suitable and effective environment for team members communication, simplifies software design process, and unifies principles and techniques used for documentation creation.

### 1. Introduction

Collection of tolls for parking place rentals is a basic function of parking systems. Experience shows that parking system installation can significantly increase the parking services revenues compared with standard forms of parking toll collection.

Exact specification of on-road vehicles and persons for whom the parking system is to be designed and classification of vehicles based on proper criteria are very important parts of a parking system design. General specification of on-road vehicles is given by the Act of the National Council on Traffic on Land Thoroughfares. Classification of on-road vehicles is performed on the base of their physical properties (size, weight, etc.), produced emissions, vehicle purpose, etc.

Based on the given classification vehicles are given rights to move within a parking system and amount of the parking toll is defined as well. Additional strict check of a vehicle weight must be ensured in the case of closed multi-storey car parks.

Tolls are collected not only by standard forms (human attendance, money and/or card automatons, etc.), but also by automatic forms requiring obligatory installation of on-board transponders and relevant technologies.

Modelling of park system operation by the UML (Unified Modeling Language) consists of creating the following types of diagrams: use case diagram, class diagram, sequence diagram and statechart diagram. Yet another types of diagrams are available but not used in our models. Examples of a use case diagram, class diagram and sequence diagram are given in the paper to model an automatically billing parking system.

The parking system consists of three basic sections that need to be seen as logical units performing specific operations. These are:

*Logical unit Input* (presented by classes - Detector, Transceiver, InfoTable, Stop)

Its task is to control input to a parking place in situations like:

- Input without support of a vehicle on-board unit (OBU). In this case a passive transducer of identification pulse is expected as the minimum on-board equipment. Reservation system and function of priority input are not available;
- Input with support of a vehicle on-board unit. This alternative can be used provided that the vehicle is equipped with a passive transponder at least. Priority input is possible based on request transmitted by the transponder;
- Input of vehicle with a defective transponder will be considered as input without support of a vehicle on-board unit;
- Input in the case of parking system failure. The parking system will be put out of service and its internal functions will be temporary taken over and performed by operative emergency service. The top control level will automatically reroute all requirements from higher levels to other parking systems. Transactions will be stored, and processed immediately after parking system recovery;
- Input with a transponder when an internal discrepancy is detected will be solved when leaving the system (at its output);
- Input rejection occurs when non-standard conditions are detected (vehicle overweight, length of vehicle exceeds defined input length, parking place is full, etc.).

*Logical unit Output* (presented by classes - Detector, Transceiver, InfoTable, PaymentAutomaton, Stop)

Its task is to perform operations that support the following situations:

- Output with a support of vehicle on-board unit, payments through transponder;
- Output without any problems, other payments form (credit card, subscriber card, coins, etc.);
- Output in the case of discrepancy in transponder-system data is considered as output without support of vehicle on-board unit;

\* Pirník Rastislav, Zahradník Jiří, Janota Aleš, Cigánek Peter

Department of control and information systems, EF, ŽU in Žilina, E-Mail: rastislav.pirnik@fel.utc.sk, jiri.zahradnik@fel.utc.sk, ales.janota@fel.utc.sk, peter.ciganek@fel.utc.sk

- Output with exit rejection (vehicle theft detection, insufficient credit, etc.);
- Non-standard output (escape, transport in other vehicle, etc.);
- Output of priority vehicles;
- Output from a parking place of an open type.

**Logical unit Kernel** (presented by classes - P&Rcontroler, Record, UserDetection, Clasificator, UserIdentification, PaymentRealization, InfoGuidelines)

Its task is to control a parking system and other partial systems of the parking place. These are:

- Free place guidance system;
- Parking place information system;
- Parking place management;
- Security systems;
- Monitoring system;
- Emergency system;
- Fire systems.

The logical unit *Kernel* simultaneously covers total communication with higher levels of the parking system. The top level is responsible for information exchange between the system and other equipment of ITS subsystems.

Creating an object-oriented model of the parking system and its operation must consider such criteria that enable as many options

for future development as possible. This is the reason why the model is realized as an open system. Definition of parking system requirements enables to create a base of basic knowledge usable for logical architecture design.

The use case diagram (Fig. 1.) defines possible uses of the parking system as described above. The class diagram that describes requested functions of the parking system is presented in Fig. 2. Described in detail in literature [2].

Sequence diagrams represent particular situations as indicated and described in an introductory part of this paper. During the modelling process these basic sequence diagrams can be linked to create more complex and integrated units. This way can be used to model even complex situations occurring in individual I/O parts of the parking system. Thus combining different scenarios shown in Fig. 3. it is possible to create specific situations that occur at input or output of the parking system, and consequently to create relevant sub diagrams (Fig. 4.).

Such an approach can be used to include all situations needed to create sequence diagrams with the help of the software tool Rational ROSE.

The use of the described approach resulted in creating sequence sub-diagrams that were further used to model electronic payments for parking system services such as:

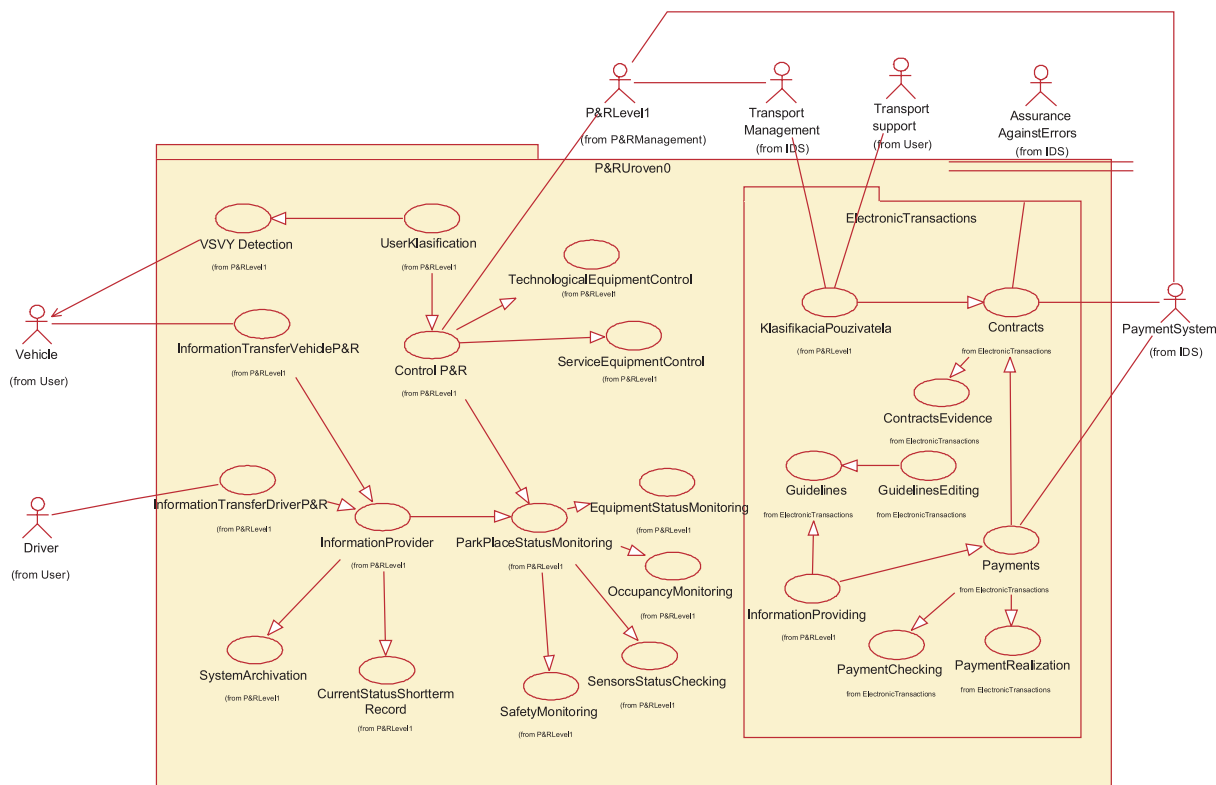


Fig. 1. Use case diagram

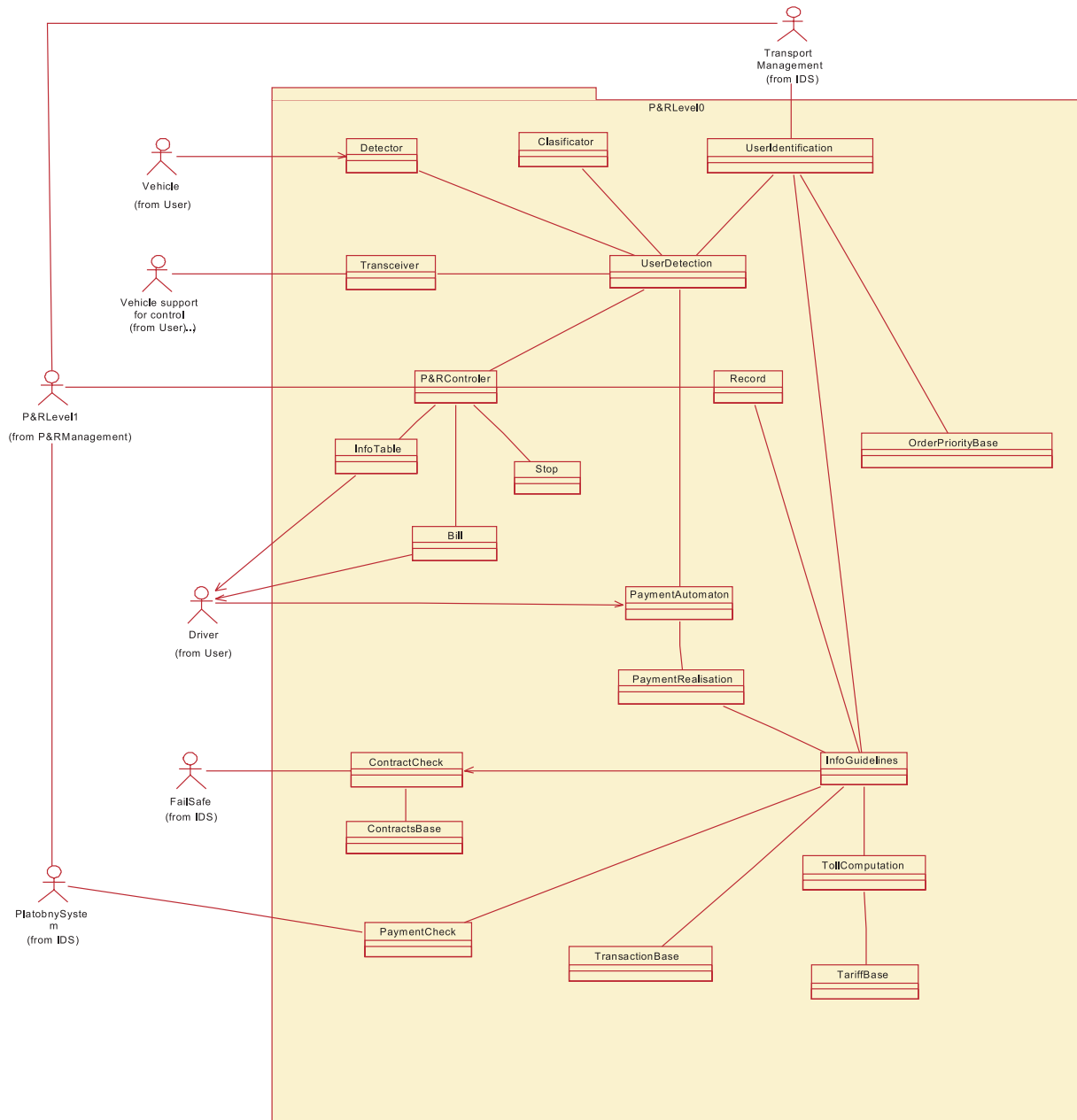


Fig. 2. Class diagram

a) A closed parking place

- Input without support of the vehicle on-board unit;
- Input with support of vehicle on-board unit;
- Input of vehicle with a defective transponder will be considered as input without support of the vehicle on-board unit;
- Input with a transponder when an internal discrepancy is detected will be solved when leaving the system (at its output);
- Input rejection occurs when non-standard conditions are detected (vehicle overweight, length of vehicle exceeds defined input length, parking place is full, etc.);

- Output without support of a vehicle on-board unit, payments through a transponder;
- Output without any problems, other payment forms (credit card, subscriber card, coins, etc.);
- Output in the case of discrepancy in transponder-system data detected will be considered as output without support of a vehicle on-board unit;
- Output with exit rejection (vehicle theft detection, insufficient credit, etc.);
- Non-standard output (escape, transport in other vehicle, etc.);

Preferred	E-payments	Incoming	Open parking system	Input	With OBU support
With order	Other payments	Rejected	Close parking system	Output	Without OBU support
Subscribed	Improper	Retained	Multi-storey	Waiting*	Passive identification
Other		Outgoing		Arrival*	No identification
				Departure*	

\* valid for open parking places

Fig. 3. Basic sequence diagrams

Input	Preferred	With OBU support	Improper	Incoming	Close parking system
Input	Subscribed	Passive identification	Other payment type	Rejected	Close parking system
Output	Other	With OBU support	E-payments	Outgoing	Multi-storey parking place
Output	Other	Without OBU support	Other payment type	Outgoing	Close parking system
Arrival	Other	With OBU support	E-payments	Incoming	Open parking system
Departure	Preferred	With OBU support	Improper	Outgoing	Open parking system
.....					
Other					

Fig. 4. Modelling of parking system specific situations

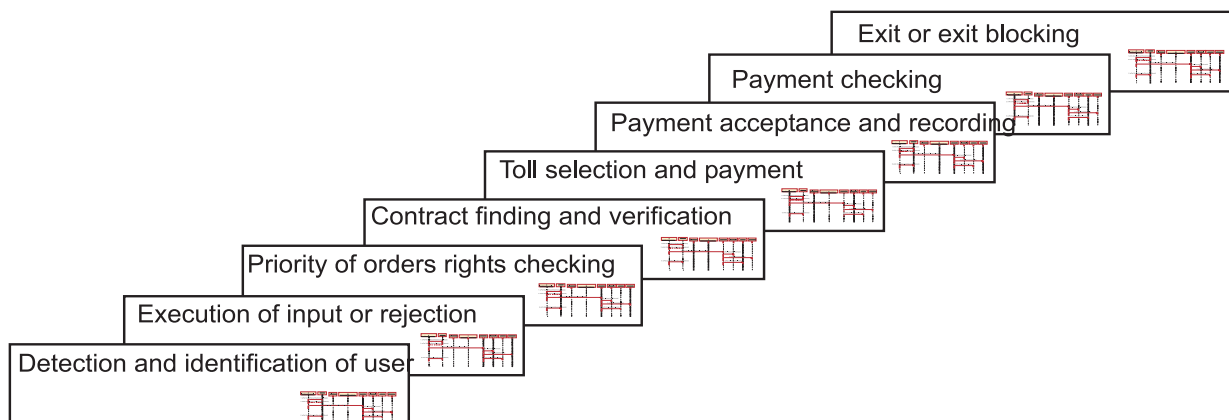


Fig. 5 Sub-diagrams modules

- Output of priority vehicles;
- Output from an open type parking place.

As examples in Fig. 5 there are shown modules of sequence sub-diagrams describing operation of a closed parking place - e-

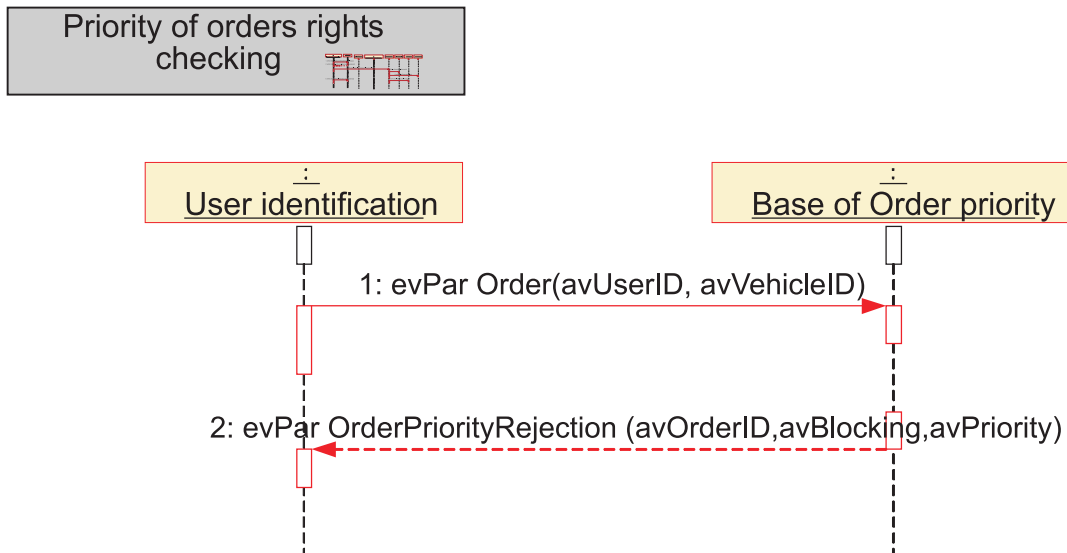


Fig. 6. Sequence diagram for Priority of Orders rights checking

payment, input and output with support of a vehicle on-board unit, payment through a transponder.

b) An open parking place

This type of a parking place does not allow for the use of some of standard elements for checking correctness and validity of payments for provided services. Checking activity must be performed with a help of an “operator”.

- Input without any problems, other payments form (credit card, subscriber card, coins, etc.);
- Output without support of a vehicle on-board unit, payments through a transponder;
- Input with a transponder when an internal discrepancy in transponder-system data is detected will be considered as input without support of a vehicle on-board unit;
- Output with exit rejection in situations being considered non-standard and solved by a human “operator” (vehicle theft detection, insufficient credit, etc.);
- Priority vehicles staying;
- Others...

A concrete sequence diagram (presents and specifies one line of Fig. 4.) example of a sub-diagram module from Fig. 5. is a sequence diagram called “Priority of orders rights checking” presented in Fig. 6.

Unlike an open parking place where the payment must be realized by deposit type of payment at the closed parking place the payment is realized at the exit of the system.

**Conclusion**

Object-oriented modelling of electronic toll collection creates a suitable and effective environment for communication of team members who work over the project, simplifies software development process, supports specifications that are independent from specific programming languages and development processes (specifications are readable and comprehensible for all subjects involved in the process of evaluation and application acceptance). It unifies principles and techniques used to create documentation and simplify maintenance of a model during project development.

A designed and modelled parking system integrates different forms of electronic payments at present mostly realized on the base of service subscription. An electronic system realising electronic toll collection evaluates and on the base of selected criteria makes modification of operation conditions possible. Thus effective control and maintenance of a parking system can be reached while a unified form of toll collection increases quality and convenience of services towards a user.

**References:**

[1] ARLOW J., NEUSTADT I.: *UML and unified process of application development*, Computer Press Brno, 2003  
 [2] VTP: *Technologies and services of intelligent transport in SR conditions (in Slovak)*, Žilina, 2005  
 [3] PIRNÍK, R.; HUDEC, R.; NAGY, P.: *Modelling of detection systems in PARK application by the using of tool RATIONAL ROSE*, In: NOVÉ SMERY V SPRACOVANÍ SIGNÁLOV VII., Tatranské Zruby, pp. 176–180, 13.–14. 5. 2004, ISBN 80-8040-232-9.