Tele-operated service robots for household and care

Michiel van Osch [1]*, Debyjoti Bera [2], Yvonne Koks [3], and Kees M. van Hee [1]

[1] Department of Mathematics & Computer Science, [2] Department of Mechanical Engineering, Technische Universiteit Eindhoven, The Netherlands
[3] ZuidZorg Home Care, The Netherlands
** Corresponding author (m.p.w.j.van.osch@tue.nl)*

Purpose Service robots are a relatively new branch of robotics after the successful industry robots and the experimental humanoids. Service robots are supposed to perform tasks that normally are done by humans in particular daily life activities. However they do not have to do it in the same way as humans and neither do they not have to look like a human. Service robots have to operate in environments meant for humans so they have to navigate in environments with unforeseen moving objects and subjects, to mention only one of the challenges. This paper is based on our experience with a tele-operated service robot, named Rose. Tele-operated means that Rose is not completely autonomous but that it is remotely controlled by a human operator. The human operator has the option to control the robot manually or provide indications to carry out complex coordinated procedures (eg. move and grab object). Tele-operated service robots have a wide range of applications, such as in the building industry to carry and place heavy objects, or in the security business. We focused on daily life household tasks. Experiments with Rose were performed in a field lab setting, for the care of elderly people. They need attention several times a day, but in total not more than two hours. This means that one operator can service several homes. This results in a five-fold productivity improvement of care takers. After an introduction on tele-operated service robots, we concisely present the system design of Rose. Then we sketch the field lab experiments and we consider the lessons learned. Based on this information we present some scenarios for the future of tele-operated service robots for caretaking and household tasks. Both the technical and the application aspects will be covered. **Method** First, a number of general purpose use cases covering daily household activities such as warming a meal, picking and placing objects, turning on the light etc. were defined. From these use cases, the system architecture of Rose was derived using a construction technique called correctness by construction. This construction technique guarantees deadlock freedom and livelock freedom. The software development was carried out using the popular Robot Operating System (ROS) framework. Four rounds of tests covering all use cases were carried out by nurses from a care organization called Zuid Zorg in Waalre, the Netherlands. In the first three sessions the cockpit was located in a room next to the robot. In the fourth session, the cockpit was located in Veldhoven, another nearby town, while the robot remained in Waalre. Results from each test round were used to improve the software of Rose. **Results & Discussion** From the results of all test sessions we were able to confirm feasibility in real-life scenarios and increased productivity by an operator while servicing multiple homes. Furthermore, consistency in performance can be achieved by exploiting the robot's autonomy.

Keywords*: service robots, home-care, correctness by construction*

INTRODUCTION

Over the last years many initiatives have emerged in developing robots for assisted living: Robots that operate in a domestic environment and perform household tasks. For instance Willow Garage has been developing the PR2 Robot (Personal Robot 2)[1] as a research platform to develop domestic applications. In 2010 this robot was successfully deployed at ten universities all over the world and several have been sold. The Fraunhofer Institute has been developing the Care-o-Bot 3[2]. IBM has been developing HERB (Home Exploring Robot Butler)[3]. Furthermore, since 2008 the Robocup at Home competition[4] is held each year to push the development of domestic robots.

Most of these research efforts focus on robots that can either deal with any situation they may encounter autonomously: for instance they can open any kind of door. However, currently they can only deal with very specific tasks in very specific situations. We believe that in order to make a robot applicable in a domestic environment, for the forthcoming years a human operator has to stay in the loop. This is because the robot will always encounter situations it cannot handle autonomously yet. For instance using well established SLAM (Simultaneous Localization and Mapping)[5] technologies does not prevent the robot from getting stuck or cornered. A human operator can help in these situations by taking manual control of the robot and help the robot to continue on its way.

In 2009 we initiated the research project TSR (Tele-Service Robot)[6] in which, together with several industrial and research partners, developed a domestic tele-operated service robot. As application domain we chose home care. Our robot had to be able to assist elderly and disabled people. A nurse can control the robot from a central location.

In line with our philosophy our initial focus is on building a manually controlled tele-operated robot. After that we add autonomy to make control easier for the operator. Our robot is named ROSE (Remotely Operated SErvice) and was developed in three iterations.

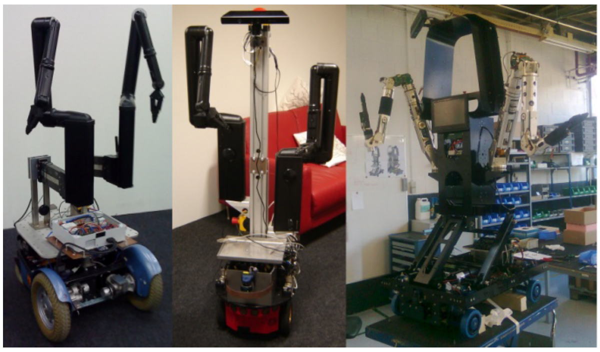

Fig.1. Rose 0, Rose 1, and Rose 2

In the first iteration we mounted two robotic arms on the base of an electric wheelchair and made this apparatus wireless controllable. The robot was built within three months and was used to establish the requirements for the later stages of the project. We concluded that the wheelchair was too massive to control efficiently (or autonomously). In the second iteration we made a robot that could be controlled efficiently from a distance using simple interfaces and we added autonomous movement to the robot platform, both by machine vision and a map. This robot was tested in an apartment for elderly. In the third iteration we designed a new robot based on our test experience from the second iteration. This robot features four individually driven and steerable wheels, a lift to reach the upper shelves of a kitchen cabinet and to pick objects from the floor, additional autonomous functionality, and other improved software.

In this paper first we describe what we mean by service robots for care and tele-operated service robots. Second we describe the use cases that our robots had to be able to achieve. These use cases were selected together with home care personnel and the future clients for our robots (elderly and disabled people) and they are a representative set of tasks that a service robot like ours has to be able to perform in the future. Third we describe the user requirements. These were also defined together with the care organization and input from the elderly. Fourth we describe the architecture of our second iteration robot and cockpit. Fifth we describe the results of the tests at the field tests with Rose 1. We round up this paper with our conclusions and future perspectives.

Service robots for Care

The population of elderly is rapidly increasing and the working population is decreasing[7]. Already there are not enough people who are willing to take care of our elderly. Elderly and physically disabled people mostly want to stay at home and keep their independence as long as possible and institutional care is expensive. Recently the European Union launched the European Innovation Partnership on Active and Healthy Ageing (EIPAHA)[8] that has the goal by 2020 to increase by 2 years the average time a person can live at home. For these reasons we believe it is inevitable that domestic robots will become a solution to help the ageing society.

Tele-Operated Service Robots

A TSR (Tele-operated Service Robot) is a robot that is controlled by a human from a distance, tele-operation, and performs tasks (services) typically in dangerous environments. Tele-operation is probably the oldest form of robotics[9]. Tele-operation enables a person, called operator, to act remotely as if the operator was on the spot, by for instance copying the manipulations of the operator at a distance. An example is the Da Vinci Robot[10] used for medical surgery .

However, in order to compete with a human care giver the TSR system should be able to perform tasks rapidly, which means that the operator should be able to give a simple command to perform a complex task. Therefore a TSR will also require autonomy in task performance. The TSR field differs fundamentally from industry robots. The field of industry robots is mature and the most well-known applications are in the automotive industry. The main difference between an industrial robot and a TSR is that industry robots are operating in a completely controlled environment that is often designed for them. To program them, only the kinematics of the system are important: the control is completely determined by the coordinates of a position of the robot or its arm. For example the robot arm moves fast (often in an optimal way) exactly to a given position. An industry robot can repeatedly perform an arbitrary sequence of complex tasks.

A TSR moves in an unknown and unadapted environment and the operator is not able to give coordinates. For example consider the movement to a door that must be opened: the operator sees the door via a camera on a screen and he has to give a command to move to the door, to grip the door handle, to move that handle downwards and to pull (or push) the door. This complex task cannot be commanded by providing coordinates. There could be obstacles in the room that were not there when a similar task was performed before, so the same set of instructions cannot be replayed, as is done with industry robots. Therefore the field of TSR differs fundamentally from the more classical field of industry robots.

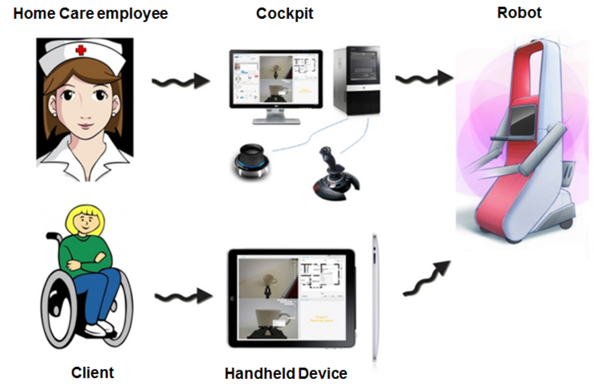

Fig.2. Tele-operation for Care

A TSR consists of a master and a slave component. The slave is in fact the robot that is executing at a distance the commands given by the master. The slave consists of three components: a mobile platform, a set of arms (one, two or even more) each one equipped with a gripper and a vision system. The master, called cockpit, is an integrated set of devices that enables the operator to control the slave. Seen from a different perspective the master is also a robot, but one with a "human in the loop". . In a basic TSR the operator has to demonstrate the actions to be executed by the service precisely, maybe at a different scale. In advanced TSR the operator has a high-level command language in which he can order a complex task for the service robot with a simple command. Such a command can be given to the slave by means of advanced input devices such as gloves, joysticks with haptic feedback or by voice recognition. An even more advanced TSR is able to learn behavior from past behavior, programming by example, and by operator training which is in fact supervised learning.

USE CASES

The most cumbersome tasks for home care employees are the frequently and simple tasks. For instance opening curtains, preparing fruit, doing dishes, posting letters. Reducing these simple tasks makes the job more attractive. Elderly and disabled most of all want to maintain their independence as much as possible. However, they become dependent on others already for small tasks.

After extensively workshops with both home care employees and elderly we selected a number of representative use cases that both would give a good cross section of the technical possibilities and the wishes of the care employees and elderly/

- Person detection: The robot moves through the apartment and the operator sees where the inhabitants of the apartment are located. This use case allows us to test whether the robot is able to navigate smoothly in a home environment.
- Turning on the light: The robot moves through the apartment and switches on a light. This use case allows us to test whether the robot is able to manipulate buttons in a home environment.
- Moving an obstacle: The robot moves through the environment and moves an obstacle aside. This use case allows us to test whether the robot is able to move objects in its environment using both arms.
- Preparing a set of breakfast items: The robot takes a set of breakfast items and brings them to the client. This use case allows us to test whether to robot is able to pick and place a set of different sized and shaped objects.
- Pouring a glass of milk: The robot pours milk from a carton into a mug. This use case allows to test whether the robot is able to handle liquids and perform more precise tasks.
- Preparing a pre-cooked meal: The robot takes a precooked meal, heats it in an ordinary microwave and brings it to the client. This use case allows us to test whether the robot is able to perform even more precise tasks and handle dials and small buttons and synchronous arm movements

To get a better idea of what the robot had to be able to do we described these use cases in detailed scenarios which were also used for scenario-based testing.

REQUIREMENTS

Based on these scenarios we specified the user requirements for the robot and cockpit.

The most important of these requirements were:

- The robot must be operable in an unadapted home environment. This requirement lead to requirements regarding size, for instance it must fit through a door, task space, for instance it must reach the floor and reach the cup boards, and manoeuvrability, for instance it must be able to move through narrow lanes.
- The robot must be operable by a home care nurse which lead to usability requirements on the cockpit side, for instance to use common interface devices and a simple graphical user interface.

- The robot must be safe for its environment and itself.
- Whenever automation is possible, the robot must perform tasks autonomously, however the human must always be able to take control of the robot.

These user requirements were then translated into mechanical, electronic, and software requirements and further developed into a system architecture.

ROSE 1 SYSTEM ARCHITECTURE

The system architecture of Rose 1 consisted of the mechanical hardware, electronics, software and cockpit.

The main components of Rose 1, which was the robot used for the tests described in this paper further on, are:

- a mobile robotics Pioneer 3DX mobile base;
- two Exact Dynamics iArms;
- a Hokuyo UBG-04LX-F0 laser scanner used for mapping the environment, positioned low at the front of the robot;
- a Videre mini stereo camera used as overview camera and for determining the distance to objects. This camera was later replaced by an Xbox Kinect camera which gave much better performance;
- two Dynamixel AX-12 actuators that formed the neck of the robot on which the overview camera was placed;
- a quad core I5 3.33Ghz computer with 8Gb RAM;
- two spy camera's positioned on the gripper of each iArm; and
- a Wifi Router for communication with the cockpit (locally) or via an access point to the internet and a cockpit on a remote location.

The mobile base was powered by a single 12 Volt acid battery while the rest of the robot was powered by two 12 Volt acid batteries, which placed together provided 24 Volt output. A dedicated power board was built to distribute the different voltages required by the individual components. The arms required 22 Volts each, the computer required 19.6 volt. The neck actuators required 9.6 Volts and the laser scanner and the Xbox Kinect required 12 Volts. The Videre camera was powered directly by the computer through firewire.

The operator cockpit consisted of an ordinary PC with a Trustmaster joystick (no force feedback) for manually moving the robot and a space navigator for manually using the arms.

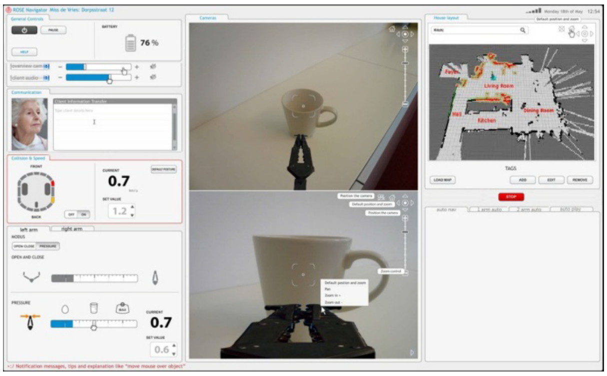

Fig.3. Cockpit User Interface

The graphical user interface showed the image of the overview camera (on the top) and the image of the active arm camera below. Operators could select different objects on the overview image by drawing a rectangle on top of them. On the right side the user interface showed the 2dmap of the environment, continuously updated by the laser scanner and containing a footprint of the robot for navigation. On the left side of the screen, space was reserved for client information and robot status.

Operators were able to move the robot autonomously to a location via the mouse by either clicking a location on the map, drawing a rectangle on the image of the overview camera, or by selecting a preprogrammed location. Operators were able to grab specific objects by drawing a rectangle on the image of the overview camera, over the specific object. They were able to put down the object by drawing a rectangle on the image of the overview camera, over a surface (e.g. a table).

Both Rose and cockpit used the Robot Operating System (ROS)[11] as high level software stack. ROS was developed by Willow Garage as an open-source software framework for robot software. They use ROS for their PR2 robot and over the last couple of years many researchers have been contributing to developing new software for ROS. ROS is a meta operating system in which different blocks of robot software (called nodes) communicate with each other via a publish-subscribe mechanism or via a goal-feedback-result mechanism. The main advantage of ROS is that many software parts are reusable and configurable to any robot. The disadvantage of ROS is performance. Stereo image processing took 85% of one CPU core in our case. Path planning took 45% of one CPU core. For Rose 1 we reused several parts of existing ROS software like the Rosaria platform interface, the navigation stack, laser, Kinect and space navigator nodes. However we also developed several new nodes like our cockpit, neck control, iARM interfaces, tracking the arm gripper with the overview camera and nodes for autonomous pick and place tasks.

In order to guarantee proper communication between ROS nodes we developed a software construction framework in which we modeled the communication between nodes using the formal modeling language Petri Nets[12]. With our framework we can guarantee that a software component inside a network of software components (like ROS) can always finish executing its task[14]. This correctness criterion is achieved by construction and we can simulate and analyze the behavior of the software with software tools like CPN-tools[13]. Specifically we modeled and validated the publish-subscribe mechanism and goal-feedback-result mechanism of ROS[15].

EXPERIMENTS

During the spring of 2011 Rose 1 was extensively tested in an apartment in Waalre, The Netherlands. The apartment was unoccupied but fully decorated as an elderly apartment.

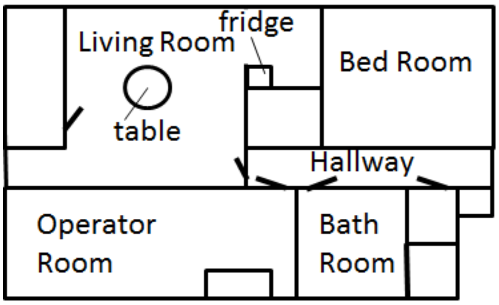

Fig.4. Floor plan of the Elderly Apartment

Over the course of 4 iterations, 3 home care employees tested Rose 1. These employees were regular computer users. Each test lasted a couple of hours. Session 1 and Session 3 were preceded by a training session of one morning each.

In the first three sessions the cockpit was located in a room next to the robot. The home care employees did not see the robot, however they did hear the robot. The cockpit and robot communicated with each other through a Wifi network. During the fourth session the cockpit was located in Veldhoven while the robot remained in Waalre. A dedicated optic fiber connection was used to guarantee sufficient bandwidth. From within the apartment Wifi was used to connect with the Robot.

In the first session primarily manual control was tested and we tried to perform the 6 preselected use cases. In the second session we introduced arm control by joystick and autonomy (preliminary) and tracking the gripper with the overview camera. Testing focused on gaining experience in grabbing (different objects) and moving around. For the third sessions the usability of autonomy functions got improved and we introduced warnings by audio. Again testing focused on grabbing and placing. For the fourth session no new functionality was introduced.

After each session each operator had to fill in a questionnaire and grade different usability aspects of Rose 1 and they had to give comments on each of these aspects. During the second through fourth session a time trial was conducted in which the operators had to go to the kitchen and bring an object to the living room table.

The following use cases were performed successfully by each operator: turning on the light; finding a person; moving an object (chair) and parts of the other use cases were performed successfully: opening the fridge; opening the microwave; pouring content from one cup to another (not milk but M&M candy); grabbing a number of breakfast items: jar, milk carton, cup, cutlery (from within the cup), butter, coffee. We were able to swing open a door that was already partially opened. We were not able to use a door handle safely. The reason was that the gripper would slip from the handle due to the forces involved.

We were able to touch the floor but not enough to pick up an object. Besides the camera view was insufficient to pick up an object from the floor. We were able to grab a package of coffee from the first shelf of the upper cupboard in the kitchen. We were able to reach the second shelf but were not able to pick up an object from that shelf.

In later sessions we manipulated different objects at different locations in the house: for instance we picked up a TV remote control from a chair and handed it to a person. We picked up candle holders, candles, small coffee milk cups, and bottles.

TIME TRIALS

During Session 2 through Session 4 time trials were conducted. Figure 5 shows the results of the time trials during the second test session. The task consisted of: (1) driving from the hallway to the kitchen; (2) pick up an object from the fridge; (3) drive to the living room table (4) and put down the object. For Operator 1, three time trials were recorded. In the first trial (Man) she used the joystick and space navigator to perform the tasks. For the second and third trial (Auto 1 and Auto 2) she used autonomous commands for picking, placing, and moving. Operator 2 performed two time trials (Auto 1 and Auto 2) The times are indicated in minutes.seconds. Over the course of the trials the end time (ET) got better and better because operators got more practiced and they had easier means (autonomous grab and put) at their disposal.

	Operator 1			Operator 2	
Task	Man	Auto 1	Auto 2	Auto 1	Auto 2
1	2.25	2.18	2.58	1.58	0.45
2	5.15	0.57	0.26	0.37	0.40
3	2.40	2.15	0.36	0.25	1.10
4	0.47	0.58	0.57	0.55	0.38
ET	**11.07**	**7.28**	**5.07**	**3.55**	**3.13**

Fig.5. Time Trials During the Second Test Session

Using autonomous commands greatly reduced the time needed for the task. Autonomously the robot was able to drive faster to the fridge and the table and it increased the operability of the robot.

OPERATOR FEEDBACK

From the first session onward, the operators were very satisfied with manually moving the robot through the house. The 2dmap with the robot footprint were already sufficient to move between door posts. In the first session the operators were not very satisfied with manually operating the arms with the Space Navigator. However in the later sessions the operators gained experience with this device and the Space Navigator was preferred over using the Joystick for manual control. Autonomy simplified control of the robot. However, it also took time to learn to efficiently use the autonomous functions.

The biggest problem with perception was to accurately perceive depth. By using a camera on the gripper it was possible to exactly position the gripper in front of an object. However the distance to the object could not be perceived accurately. Both the overview camera and the arm camera needed to be used to get an indication of the gripper position with respect to the object. Other perception issues are that sometimes the arm moved in front of the overview camera and when an object had been picked up, the object blocks the view of the arm camera.

Warnings (about possible dangerous situations) were sometimes confusing and contradicted with the information on screen. Audio Warnings had greater were noticed better than textual warnings and some operators preferred to hear the environment sounds while others did not miss them while operating the robot from Veldhoven. There were also problems with the WiFi network. Grabbing objects manually proved difficult, and whether Rose is able to grab and object depended heavily on the shape of the object and the location of the object.

After each session the operators also graded a number of statements on a scale of 0 to 10 (0 meaning fully disagree, 10 meaning fully agree). The operators graded most aspects of the robot with 'sufficient' (Grades 6 to 8). The Operator satisfaction confirmed the remarks described above. The number of operators (3) was too small to draw any further conclusions based on these grades.

CONCLUSIONS AND PERSPECTIVES

Taking the robot out of the lab environment and into the home environment with potential operators greatly helped in getting more insight in what the robot could do and what it should do. It gave us a lot of feedback for our next robot. The operators liked our robot and were enthusiastic about it, and saw a future in home care for the robot. Operating the robot (especially with the map) felt like ‘playing a video game’. The operators got used to a device that the ordinarily do not use, a Space Navigator. Perceiving depth information via the cockpit was difficult and it seemed force feedback is required for improving the manual control of the robot, for instance to feel when an object hits the table.

In the spring of 2012 we are testing Rose 2 in Waalre. Rose 2 has a complete new architecture based on our test experiences with Rose 1. Rose 2 is a preproduction prototype that we are looking to develop further in the coming three years by placing several robots in different care organizations and different homes. Hereby we gain further experience and further develop robot Rose to a complete product.

ACKNOWLEDGEMENTS

This work is subsidized by the Dutch Ministry of Economic Affairs and the Dutch Province of Noord-Brabant.

We like to thank all project partners of the TSR project: Inroads BV, Technische Universiteit Eindhoven, Fontys Applied University, Tegema-group, NTS-group, V.E.D.S.-group, Sioux Embedded Systems, Exact Dynamics BV, Van Berlo Design, and Zuidzorg. Furthermore we like to thank test our operators and all students of Eindhoven University of Technology and Fontys Applied University that worked within the TSR project.

References

1. Bohren, J., “Towards Autonomous Robotic Butlers: Lessons Learned with the PR2”, *ICRA 2011*, 2011.
2. Graf, B., Reisner, U., Hägele, M., Mauz, K., Klein, P., “Robotic Home Assistant Care-O-bot 3 – Product Vision and Innovation Platform”, *“IEEE Workshop on Advanced Robotics and Social Inpacts”*, pp. 139-144, 2009.
3. Srinivasa, S., “Herb: A Home Exploring Robotic Butler”, Autonomous Robots, Vol. 28(1), pp. 5-20, 2010.
4. Robocup, http://www.robocup.org/robocup-home/

5. Durrant-White, H., Bailey, T., “Simultaneous Localization and Mapping: Part 1”, *IEEE Robotics and Automation Magazine*, 2006.
6. Robot Rose, http://www.robot-rose.nl
7. The 2009 Ageing Report: economic and budgetary projections for the EU-27 member States, 2009.
8. European Commission, “Active & Healthy Ageing”, http://ec.europa.eu/research/innovation-union/index_en.cfm?section=active-healthy-ageing
9. Siciliano, B., Khatib, O, *Handbook of Robotics*. Berlin: Springer-Verlag, 2008.
10. Da Vinci Surgery, http://www.davincisurgery.com/
11. Gerkey, B., “ROS: an open-source Robot Operating System”, *ICRA Workshop on Open Source Software,* 2009.
12. Girault, C. Valk, R., Petri Nets for System Engineering, 2003.
13. Jensen, K., Kristensen, L.M., Wells, L., “Coloured Petri Nets And CPN Tools for Modeling and Validation of Concurrent Systems”, *International Journal on Software Tools for Technology Transfer*, Vol. 9(3), pp. 213-254, 2007.
14. Bera, D., Hee, K.M. van, Osch, M. van, Werf, J.M.E.M. van der, “A Component Framework where Port Compatibility Implies Weak Termination”, *Proceedings of the International Workshop on Petri Nets and Software Engineering (PNSE 2011)*, 2011.
15. Bera, D., Hee, K.M. van, Werf, J.M.E.M. van der, “Designing Weakly Terminating ROS Systems”, *Proceedings of the 33rd International Conference on Applications and Theory of Petrinets and Concurrency (ATPN 2012)*, 2012.