Proceedings of the 8^{th} International Conference on Applied Informatics
Eger, Hungary, January 27–30, 2010. Vol. 1. pp. 79–86.

Verified Mobile Code Repository Simulator for the Intelligent Space[*]

Zoltán Istenes[a], Máté Tejfel[a], László A. Jeni[b]

[a]Eötvös Loránd University,
e-mail: {istenes,tejfel}@inf.elte.hu

[b]University of Tokyo,
e-mail: jedi@hlab.iis.u-tokyo.ac.jp

Abstract

This paper describes a method for formalisation and verification of properties of mobile code and presents a simulator for multiple mobile robots. Mobile robots can be controlled with the mobile code technology. In our model, the mobile code – with its properties – is created, verified, stored and transmitted to the robot through the Certified Proved Properties Carrying Code (CPPCC) architecture. We integrated the architecture in the Intelligent Space, containing mobile robots and several communicating Distributed Intelligent Network Devices (DINDs).The mobile robots in the iSpace handle real objects and interact with humans to support them. Humans can give tasks to the robots and the iSpace selects the most appropriate robots to execute the given task. A simulator was created to support both the simulation of the iSpace with several robots and the execution and transmission of the mobile code to the robots.

Keywords: Mobile Robots, Intelligent Space

MSC: 68U20, 93C85

1. Introduction

Mobile robots can be useful in wide spectre of application areas such as industrial, military, domestic robots. This paper illustrates a model of a system using mobile robots which work in a human environment, can execute various tasks, and are safe, in the manner the actions of a robot are verified against a set of explicit and formally expressed security requirements (for example: each task has to be started from, and has to be stopped in a given state, the robot is prohibited to go to dangerous

[*]Supported by KMOP-1.1.2-08/1-2008-0002

places, lift, stairway etc.). In our model, the mobile code – with its properties – is created, verified, stored and transmitted to the robot through the Certified Proved Properties Carrying Code (CPPCC) architecture. A correspondence analysis is executed using a formal verification system, in order to verify the mobile code properties correspondence against the robots' requirements. This analysis may refuse to execute those mobile code tasks, that violate the robots' requirements.

We integrated the above described method in the Intelligent Space, this iSpace contains multiple mobile robots and several communicating Distributed Intelligent Network Devices (DINDs) which share their information about the human environment. The DINDs monitor the dynamic environment of the iSpace, process the captured information and communicate to provide the cooperation of different DINDs through a network. The mobile robots in the iSpace handle real objects, cooperate with each other and with other components of the system and interact with humans to support them. Humans can give tasks to the robots – with the help of the correspondence analysis – the iSpace selects the most appropriate robots to execute the given task. A simulator was created to support both the simulation of the iSpace with several robots and the execution and transmission of the mobile code to the robots.

2. The Intelligent Space

The Intelligent Space (iSpace) is a space (room or corridor), which has ubiquitous distributed sensory intelligence (various sensors, such as cameras and microphones with intelligence) actuators (TV projectors, speakers, and mobile agents) to manipulate the space [7]. The information from the sensors is used by computers and robots connected through a communication network, in order to provide of various services to humans. The various devices cooperate with each other autonomously, and the whole space has high intelligence based on ubiquitous computing, which is used manly for welfare support. Figure 1 introduce the basic concept of iSpace.

A space becomes intelligent, when Distributed Intelligent Network Devices (DINDs) are installed in it [8]. A DIND has a sensing function through devices such as a camera and microphone that are networked to process the information in the Intelligent Space. The iSpace consists of humans not only sensors cameras or robots. The DINDs monitor the space, achieve data and share them trough the network. For instance, the iSpace can recognize humans, track their movement to identify the walking areas and learn the shortest safest path in the environment [14, 6].

Mobile robots become physical agents of the Intelligent Space and they execute tasks in the physical domain to support people in the space. Task includes movement of objects, providing help to aged or disabled persons etc. Thus, the Intelligent Space is an environmental system, which supports people in it electrically and physically [11].

Distributed Intelligent Network Devices are the building blocks of the Intelligent Space. The key concept of DIND is consisting of three basic elements. The dynamic

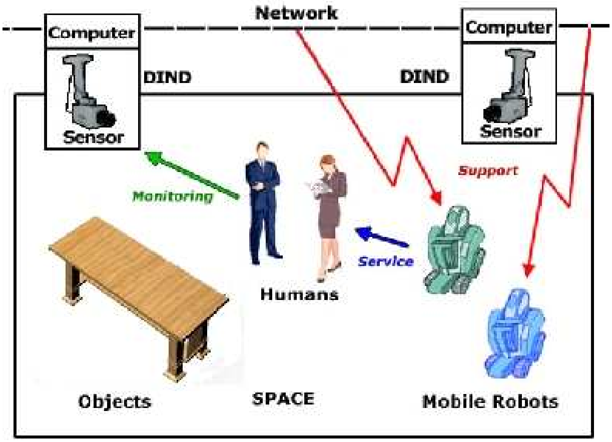

Figure 1: Intelligent Space Concept.

environment (which contains people, vehicles and robots, etc.), is monitored by the *sensor*, the information is processed into a form easily captured by the clients in the *processor* and the DIND communicates using some kind of *communication device* with other DINDs through a network. Robots are able to use resources of DINDs as their own parts. However, robots with their own sensors may be considered mobile DINDs.

The real power of the Intelligent Space is the network of DIND, that can sense the whole space, and share information between each other. Network of DINDs form a distributed sensor and processing network, to percept the space, interact with the inhabitants, and extend the capabilities of the mobile agent [15].

3. Controlling the robots using mobile code technology

Mobile code is a program or a program-component obtained from a remote system, transferred across a network, dynamically downloaded and executed on a local system [2]. Such an example is the execution of plugins, JavaScripts and ActiveX controls in a web browsers or codecs in multimedia players.

One of our goal is the usage of robots capable to execute various tasks. These tasks are not known in advance, the iSpace environment assigns them to the robots "on the fly", dynamically. Several different technologies can be applied to control robots. The most basic method is called teleoperation. The robot sensor data is transmitted to a remote operator and the operator send each command separately back to the robot. These commands are very simple and command the robot "step by step".

In the second method the robot controlling program is running directly on the robot. The robot can execute only one program which is initially installed on it. The robot executes the program autonomously and it can only receive control data, for example coordinates to have to attain. In this method while the robot executes his control program it access its sensor data, makes decisions and controls its actuators, motors. The advantages of this method are the autonomous robot and the small amount of the communication. However the robot controlling program can be adequate for some situations, when the robot environment changes, the robot control program can become outdated and not well suited for the changed situation.

The third method tries to overcome the problems of the second method. The robot executes a framework which downloads and executes the robot controlling program dynamically. The dynamically downloaded program can be referred as "mobile code". This method does not require continuous connection and provides high flexibility in executing various tasks.

From these we use the mobile code technology to achieve the mentioned goal. In our model the robots are parts of the iSpace, they cooperate with each other and with other part of the iSpace and share their informations. This architecture does not contain central controller.

Humans and DINDs of the iSpace can send mobile codes to the robots. During the execution of their tasks robots can communicate with the ISpace and receive informations about the environments. The iSpace contains inhomogeneous robots which has different hardware construction, sensors, actuators.

The mobile code technology is well suitable for the presented model hence it provides the adaptability in robot controlling by using a uniform framework to reprogram and reconfigure the mobile robots. Different program can be downloaded to the robot for each different task.

4. Safe mobile code

As we mentioned earlier in our model mobile code technologies are used to control some functionality of a robot. The control code of a robot can extend with components by dynamically linking the obtained piece of code to the application. Such technologies are extremely vulnerable against malicious code, as well as against accidentally erroneous or improper code, especially because these mobile components can also be created by a second party. For this reason it is extremely important to verify the correctness of the components.

Since the mobile robots exist, work, navigate and operate in human environment, it is natural to request them to satisfy certain basic requirements. In most of the case these requirements exist only implicitly in the robot controlling code (for example the robot stops when its ultrasound distance sensor detect an object close to him). In our model we would like to express these requirements explicitly, and verify if the mobile code satisfies the requirements. The requirements and the properties of the mobile codes are expressed in a formal way (for example using the B-method [1]). This make possible to use a formal verification system.

Some examples of these formal requirements are the following.

- The robot may not go to places from where it is not able to go back to its service station.
- The robot is prohibited to go to objects or peoples closer as 5 cm.
- The robot has some resource bounds (memory, time, power consumption etc.).
- There are prohibited places in the space (lift, stairway, dangerous places etc.).

In our model the robots can refuse tasks (mobile codes) which do not satisfy they requirements. The explicitly and formally expressed requirements are verified against the properties of the mobile code. The properties of the mobile code are expressed also explicitly, using the same formalism.

For the safe usage of the mobile code technology it is also important to ensure a safe manner of the transmission of the mobile components. This safe transmission can be done, for example the use of Certified Proved-Property-Carrying Code [3] (or CPPCC, for short) technique. In this architecture the code receiver can make a decision on whether or not to accept and utilize the received code based on the declared properties of the code and on the opinion of a third-party, creditable certificate authority, which verified that the code indeed has the declared properties.

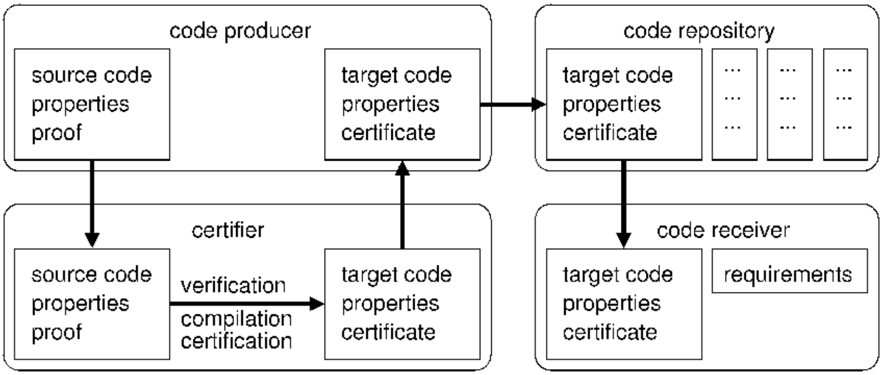

Figure 2: Overview of the Certified Proved-Property-Carrying Code technology

As Figure 2 illustrates, there are four different participants in a CPPCC system, the code producer, the certificate authority, the code repository and the code receiver. The scenario for producing and receiving safe mobile components is the following.

1. The code producer creates a program component and additionally it formulates and proves the properties of the created component based on the source

code of the component. Finally the producer pack together the source code, the properties and the proofs and send the package to the certificate authority.

2. The certificate authority checks that using the received proofs the specified properties can be proved for the source code of the received program component. Then it creates the target code from the source code, packs the target code and the properties together, signs the package and sends it back to the code producer.

3. The code producer uploads the signed package to a code repository.

4. The code receiver obtains the mobile code from the code repository and checks the certificate attached to the received package. If the signer is trusted, it verifies that the properties of the code match its requirements. If they match the received code is linked into the code receiver and gets executed. If the certificate is not correct or the properties do not match the receiver refuse the execution of the code.

In our model the CPPCC architecture has been integrated with the iSpace environment. The code repository becomes part of the iSpace. With the help of some human interaction the mobile code is sended from the repository to a mobile robot playes the role of the receiver.

5. Simulation

For modeling the described architecture a simulator was created. The simulator contains two cooperating parts, the control part and the execution part.

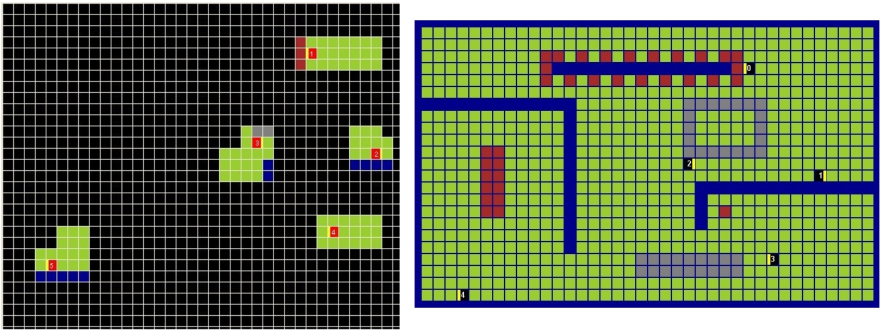

Figure 3: The two parts of the simulator.

The control part – according to some user interaction – can create simple mobile codes using a set of simplified instructions and sends them to the robots of the iSpace. For simplification this part involves each of the code producer, the certifier and the code repository of the CPPCC architecture and we simulate only executions where the certification is correct. This component also has some visualization role. During the simulation this component visualize that part of the physical environment which is already detected by the robots.

The execution part simulates the behaviour of the robots. The corresponding robot receives the sended mobile code, checks its properties (and the certification, which will be always correct here) and if the properties of the code corresponds with the own requirements executes it. During the simulation this component visualize the whole physical environment. In the current version of the simulator the environment can contain walls, robots, movable and unmovable objects.

Figure 3 illustrates a simulation, where the iSpace contains five robots.

6. Related work and conclusion

The ongoing research activities about Intelligent Space achieved several results and solutions in the fields of motion control [12], feature extraction [9], recognition and tracking of moving objects [14] and a component based approach of the iSpace implementation [13].

This paper described a method for formalisation and verification of properties of mobile code and presented a simulator for multiple mobile robots. The described framework uses the Certified Proved Properties Carrying Code architecture and it has been integrated with the iSpace environment. To model the architecture a simulator was created which is capable to simulate both the simulation of the iSpace with several robots and the execution and transmission of the mobile code to the robots.

References

[1] Abrial J.-R. The B-Book Cambridge University Press, 1996.

[2] Ghezzi, C., Vigna, G. Mobile Codes Paradigms and Technologies: A Case Study In Proceedings of the 19th International Conference on Software Engineering, LNCS 1219, Springer-Verlag, Berlin, Germany, 1997., pp. 39-39.

[3] Horváth, Z., Kozsik, T. Safe mobile code - CPPCC: Certified Proved-Property-Carrying Code. G. Czajkowski and J. Vitek, Resource Management for Safe Languages (in: ECOOP 2002 Workshop Reader, LNCS 2548/2002, Springer-Verlag), 2002, pp. 8-10. Full position paper is available at http://www.ovmj.org/workshops/resman/.

[4] Istenes, Z., Kozsik, T., Hoch, Cs., Tóth, L. A. Proving the correctness of mobile Java code 6th Joint Conf. on Math. and Comp. Sci., July 12-15., 2006, Pécs, Hungary, submitted to Pure Mathematics and Applications, SAAS Ltd.-SAAS Publishing, Budapest, Hungary.

[5] Istenes, Z., Kozsik, T. Commanding a robot in a safe way Proceedings of the 10th Symposium on Programming Languages and Software Tools (SPLST 2007), Budapest, 2007, pp. 167-177., ISBN 978 963 463 925 1

[6] Jeni, L. A., Istenes, Z., Szemes, P., Hashimoto, H. Robot navigation framework based on reinforcement learning for intelligent space. Proceeding of Conference on Human System Interactions, 2008 (HSI 2008), Krakow, Poland, 2008

[7] Korondi, P., Hashimoto, H. Intelligent Space, as an Integrated Intelligent System, Keynote paper of International Conference on Electrical Drives and Power Electronics, Proceedings, 2003, pp. 24-31.

[8] Lee, J., Morioka, K., Ando, N., Hashimoto, H. Cooperation of Distributed Intelligent Sensors in Intelligent Environment, IEEE/ASME Transactions on Mechatronics, Vol.9, No.3, pp.535-543, 2004.09, ISSN 1083-4435

[9] Morioka, K., Lee, J., Kuroda, Y., Hashimoto, H. Hybrid Tracking Based on Color Histogram for Intelligent Space, Artificial Life and Robotics, Vol.11, No.2, pp.204-210, 2007.07, 1433-5298

[10] Necula, G. Proof-carrying code. Conference Record of POPL '97: The 24th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, Paris, 1997, pp. 106-119.

[11] Niitsuma, M., Hashimoto, H. Spatial Memory as an Aid System for Human Activity in the Intelligent Space. IEEE Transactions on Industrial Electronics, Vol. 54, Issue 2, 2007, pp. 1122-1131, ISSN: 0278-0046.

[12] Padhy, P., Sasaki, T., Nakamura S., Hashimoto H. Modeling and Position Control of Mobile Robot, The 11th International Workshop on Advanced Motion Control, Niigata, Japan, pp.100-105, 2010.3.22.

[13] Sasaki, T., Hashimoto, H. Design and Implementation of Intelligent Space: a Component Based Approach, Mechatronic Systems Applications (Edited by Annalisa Milella Donato Di Paola and Grazia Cicirelli), INTECH, Chapter 6, pp.81-98, 2010.03, ISBN 978-953-307-040-7

[14] Sasaki, T., Brscic, D., Hashimoto, H. Human Observation Based Extraction of Path Patterns for Mobile Robot Navigation. IEEE Transactions on Industrial Electronics, Vol.57, No.4, pp.1401-1410, 2010.4.

[15] Szemes, P.T., Hashimoto, H., Korondi, P. Mobile agent Control in intelligent space based on observed human behavior CEAI journal, Vol.7 No.3, pp.15-23, ISSN 1454-8658, 2006.03.