ARCHITECTURE OF INFORMATION SYSTEM FOR INTELLIGENT CASH MACHINE

Krystian Ignasiak, Marcin Morgoś
Institute of Radioelectronics, Information Technology & Electronics Department, Warsaw University of Technology
Nowowiejska 15/19, 00-665 Warsaw, Poland

Surachai Ongkittikul
I-Lab/Centre for Communication Systems Research (CCSR), University of Surrey
Guildford GU2 7XH, Surrey, UK

Keywords: Cash machine simulator, software architecture, Computer Vision Markup Language.

Abstract: The paper summarizes the idea of the intelligent cash machine simulator that can be a very flexible environment to test new algorithms in image processing. The use cases for such a cash machine are discussed. Some data workflow is presented, and finally, an architecture is proposed for integration of different software modules without imposing constraints on software platforms and tools. The proposal is based on XML as a common language to exchange data.

1 INTRODUCTION

Nowadays, all electronic equipment may become *intelligent* if appropriate logic is implemented inside. Cash machines are sensitive to many situations occurring in front of them. This concerns safety of cash transactions and the safety of the user of the machine – a bank customer. A cash machine can also be perceived as an element of a surveillance system.

This paper presents some system concepts of a software architecture, proposed for a cash machine simulator that will become user-friendly, *intelligent* and safe. The proposed cash machine simulator has the same functions as an ordinary cash machine but extends them with new features. The intelligence of the simulator is enhanced and realized by:

- Introducing a voice communication channel; the cash machine can be instructed by voice commands, as well as being able to inform the user about actions to take and the status of the transaction via synthetic speech.
- Introducing a visual communication channel from user to the machine; the user is *observed* by a set of cameras; the software keeps track of the user's behavior, analyses the behavior, interprets their gestures, and takes proper decisions.
- Introducing elements of a surveillance system into the machine; audiovisual sequences can be recorded and stored in the database; the AV material can be enriched by metadata about anything unusual that happened during ordinary cash machine transaction.

On the other hand such an automatic teller will increase the safety of transactions and personal safety of the customer. The cash machine uses advanced customer authentication based not only on standard PIN codes but also on face recognition and verification, speech recognition and other signals from different *modalities* available at the cash machine (e.g. fingerprints). Unusual behavior by the user or the occurrence of an unusual situation in front of the cash machine can be detected and security services can be informed about that. In such a case the audiovisual (and *multimodal*) sequence is stored in a database and it documents the marked disturbance.

Imagine that you approach the cash machine and it welcomes you, recognizes your gender, age, maybe mood, and it modifies its behavior with respect to these factors. The machine helps you to carry out the basic operation—withdraw the cash

Ignasiak K., Morgoś M. and Ongkittikul S. (2007).
ARCHITECTURE OF INFORMATION SYSTEM FOR INTELLIGENT CASH MACHINE.
In *Proceedings of the Second International Conference on Signal Processing and Multimedia Applications*, pages 391-395
DOI: 10.5220/0002139903910395

from your bank account—faster and with greater ease. For example, users can define simple gestures for common tasks, such as withdrawing money. The proposed system enables also the bank to offer new services.

We want to start with a simulation of such an *intelligent cash machine*, hoping it would be implemented as a real device once its usefulness is proved and tested. On the other hand our goal is to create an environment to test new algorithms and ideas in a manner that does not impose on use of particular software platforms and software tools. However we propose a particular solution to test the assumptions. The *simulator* is a very flexible experimental environment to test new algorithms and ideas, prior to implementation in real not only in cash machine environment.

We start from the discussion of the use cases for the cash machine to set up the requirements of the system, then we present the workflow of data and tasks within it. Finally we propose the software architecture to be used for the cash machine simulator.

2 USE CASES FOR CASH MACHINE SIMULATOR

The analysis of use cases (OMG, 2005) for cash machine should start from the user's perspective. The main cash machine services offered to the user are withdrawal of money from his/her bank account, checking the account's status and—for our purposes—logging into the system to access advanced functions (cf. Figure 1).

In fact we can identify modules of the system preparing the use cases and set the system's initial requirements. The first module identified is the cash machine itself. It is used by ordinary user (bank's customer) as well as it is used by bank's service to configure the cash machine, calibrate devices such as cameras, microphones, fingerprint sensor etc. (cf. Figure 2).

The most important module for the simulator is the Conversational Agent module. Use cases for this module are depicted in Figure 3. The Conversational Agent represents the artificial intelligence of the system. The agent is updated by the environment analysis module and welcomes and invites pedestrians to use the machine. The agent controls the interaction between the user and the machine following the sequence of appropriate actions needed to service the user.

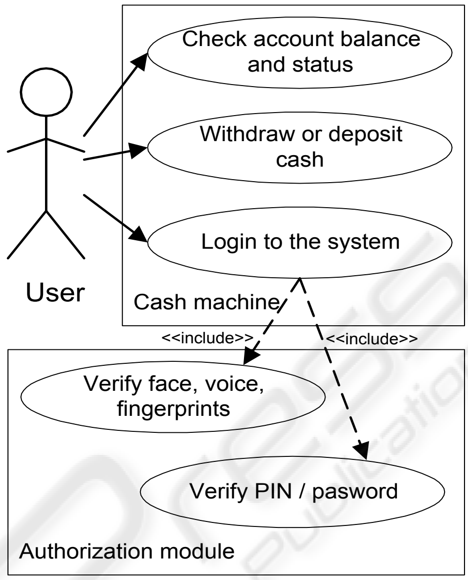

Figure 1: Use cases for the user.

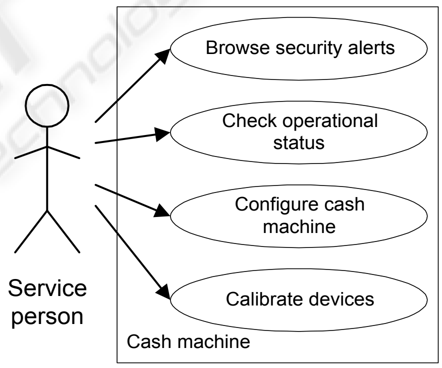

Figure 2: Use cases for the bank's service.

The Conversational Agent controls the behavior of the 3D avatar that represents the agent itself in the process of user-machine interaction. It controls the speech synthesis by sending proper signals to speech synthesis module, as well as controlling the visualization of the avatar by sending signals to visualization module. This includes rendering the 3D avatar and animating its face together with emotions to be expressed on the avatar's face.

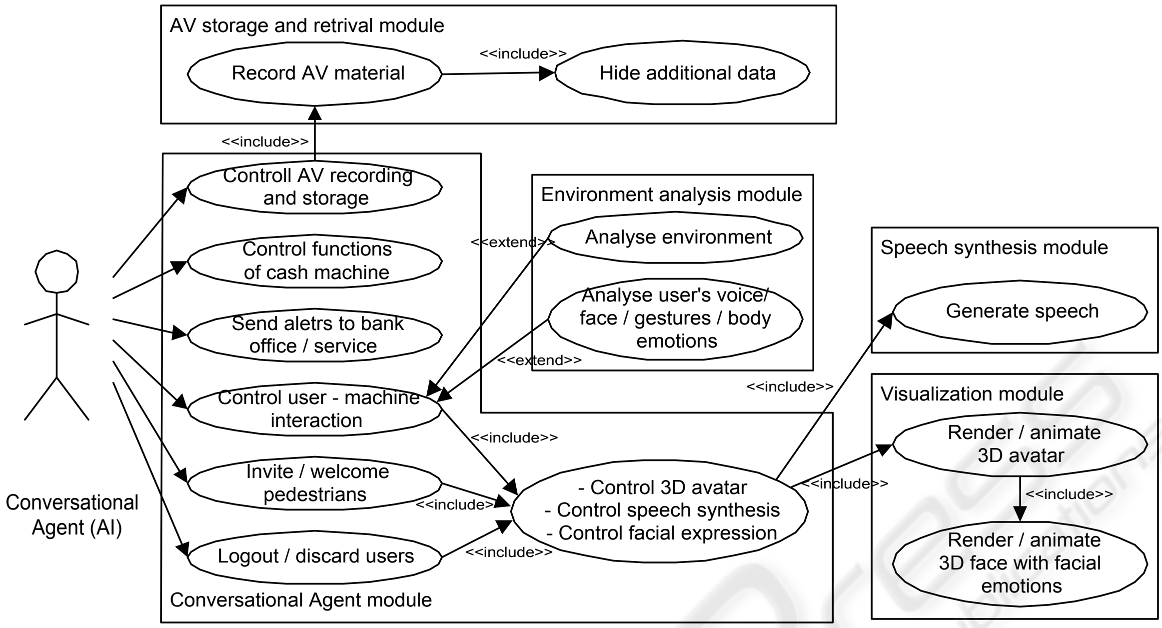

Figure 3: Use cases for Conversational Agent.

The environment analysis module keeps track of user's voice, user's body and face emotions to update the Conversational Agent and to identify an unusual user's behavior at the cash machine. Besides recognizing unusual user behavior, this module is responsible for detecting unusual behavior in front of the cash machine at all. It can detect pedestrians carrying dangerous objects etc. Once such an unusual behavior is identified, the agent sends alerts to bank or police station, also it can change the status of the machine into so called *anti-attack* mode.

The gesture analysis part of the environment analysis module aims to provide a convenient human-machine interface for users. This gesture analysis function is responsible for extracting the user's hand gestures. Gestures can be used by the user to access the cash machines services, and can also be an indication of mood to the machine.

There is a very important module present in the system from surveillance point of view: AV storage and retrieval. This module is responsible for storing the audiovisual material taken by cameras and microphones at the cash machine. One of its functions is additional *data hiding* in such a way that the material can not be modified by intruders. Metadata hidden in the audiovisual material allow the bank to state the origin of it.

The use cases in Figure 3 do not cover actions for close interaction with bank services, because it is beyond the scope of this paper. For example, monitoring the amount of cash in the machine or monitoring the funds available to the user are functions that are not considered in the simulator.

3 WORKFLOW WITHIN THE SIMULATOR

Having defined the use cases for the cash machine one can define tasks to be performed within the system and a workflow of them within the system. It is not an easy task, as we do not want to specify the algorithms to be implemented and tested within the system.

However, we want to use the simulator to test selected algorithms in the image processing field that cover our interests within the Visnet II Network of Excellence (Visnet II, 2006, Sadka, Skarbek, 2004).

We can start from calibration of the devices connected to the cash machine – cameras for example. They need to be calibrated prior to use while some of algorithms to be implemented require a set of three cameras for example. We have to ensure stable cameras positions and parameters so that the task of cameras calibration can be performed only once.

The other exemplary task that is led up continuously is the observation and analysis of the environment of the cash machine to detect potential user approach. This could be performed by human face detection in the frames of visual sequences captured by cameras built into the machine. This is the task performed by the environment analysis module and it can be illustrated as the sequence UML diagram (cf. Figure 4).

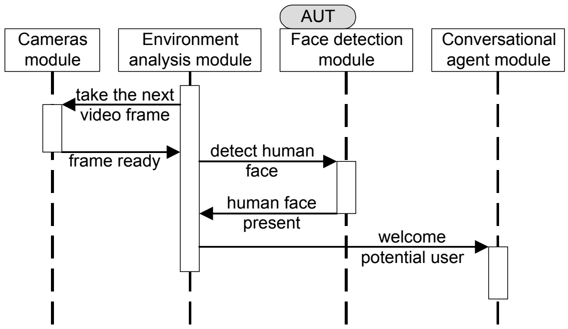

Figure 4: Exemplary sequence diagram within the cash machine simulator system, AUT stands for Algorithm Under Test – the part of the system being tested in the particular experiment.

It is obvious from the Figure 4 that there is a skeleton of the system that is present in every particular implementation and there are modules that are included for testing purposes, i.e. AUT – Algorithm Under Test. However the system has to define a common language for exchange the data flowing between the modules. The best choice is some XML application. This XML application must provide the modules with the ability to exchange the audiovisual data and the ability to exchange control data. Audiovisual data create massive streams while control data are rather sparse.

4 THE ARCHITECTURE

The architecture constitutes some software modules, that are the core of the system and modules that can be plugged into it for testing purposes. The core modules cover the following:

- Cash machine module,
- Authorization module,
- Conversational Agent module,
- Audiovisual storage and retrieval module,
- Environment analysis module,
- Speech synthesis module, and
- Visualization module.

The modules communicate with each other using Computer Vision Markup Language application, CVML (List, Fisher, 2004). The media data and the control signals are both encapsulated in XML syntax that is understood by every module. The AUT being plugged into the system must recognize a set of commands from this syntax to process media and to produce its output that will be probably sent to another module for further processing.

The modules themselves have internal structure to allow finer processing and controlling the AUT being plugged into the system. The communication between elements of modules relies on the same idea of sending AV data and control data encapsulated into CVML application.

Such a general architecture does not impose any particular software platform and any particular software tool that has to be used. The only requirement is that the used tools are to understand XML encapsulating media data and control data.

However we would propose a model solution. It includes Java as a platform for generating graphical user interfaces. The SWT (Standard Widget Toolkit, http://www.eclipse.org.swt/) would be used instead of AWT and Swing. For rendering purposes we would propose some OpenGL package for Java, for example JME (Java Monkey Engine, http://www.jmonkeyengine.com/) – one of the most efficient implementation of OpenGL to Java mapping, that supports many OpenGL extensions.

As the IDE for building software we would propose Ecplise platform (http://www.eclipse.org). It has many advantages over other IDEs available. For instance, from collaborative point of view, again, it does not impose use of any particular programming language, however it is fine suited for Java as a result of many anonymous Java programmers contributions.

The Eclipse IDE defines Rich Client Platform (RCP) that enables close integration of tools being created within the IDE. For example the JME OpenGL engine is prepared within RCP concept.

Massive computations needed by tested algorithms can be processed in other programming languages or external mathematical packages, for instance Scilab (http://www.scilab.org/). The Java Native Interface can be used for this purpose or any other method offered by the mathematical package itself.

5 CONCLUSIONS

We have discussed the idea of the intelligent cash machine through its use cases, data flow within the system and sketch of the architecture. We have proposed a set of tools well tailored to the needs of people testing their algorithms and ideas in image processing field in general.

ACKNOWLEDGEMENTS

The work presented was developed within VISNET II, a European Network of Excellence (http://www.visnet-noe.org), funded under the European Commission IST FP6 programme.

REFERENCES

List, T., Fisher, R. B., 2004, CVML – An XML-based Computer Vision Markup Language. In *International Conference on Pattern Recognition* Proceedings, 1, 789-792.

OMG, 2005, Unified Modelling Language: Superstructure, version 2.0.

Sadka, A. H., Skarbek, W., 2004, European Union Network of Excellence on Networked Audiovisual Media Technologies – Visnet. In *Visnet@KKRRiT2004, Special Session at National Conference on Radiocommunications and Broadcasting*, Warsaw, Poland.

Visnet II, Network of Excellence, 2006, http://www.visnet-noe.org/.