Web client server systems with advanced asynchronous communication, their architecture and applications

FILIP MALY, ANTONIN SLABY
Department of Informatics and Quantitative Methods
University of Hradec Kralove
Rokitanskeho 62, Hradec Kralove, 500 02
CZECH REPUBLIC
filip.maly@uhk.cz
antonin.slaby@uhk.cz

Abstract: - This article extends ideas and results presented in our contribution in Hangzhou WSEAS conference. In the paper we describe one way of solution of important and difficult problem of finding suitable ways of communications in special client server systems having hybrid and as to functionality point of view restricted client. We also describe basic improvements of client enabling running of more sophisticated software processes. In addition to it there is shown common concept and architecture of web client server systems that use asynchronous communication through one always open port (port No. 80). These systems have wide use and a lot of applications. On of them is mentioned in case study.

Key-Words: - client server systems, asynchronous communication, hybrid client, software processes, learning management systems.

1 Introduction

Basic purpose of this article is to describe common concept of web client server systems that use asynchronous communication through one port (port No. 80) and outline their architecture. The authors of this article tested four variants of solution of communication among various parts of such systems. These variants included communication through several ports, communication through two ports which does not require permanent connection, one port communication having permanent connection access to server and finally one port communication that makes use only temporary (on demand) server connection. The process of testing and assessing of various kinds of implementation resulted in reduction of number of communication ports so as the system could be used also in environments having strongly restrictive internet access policy. Design of such systems is rather easy unless there is not required implementation of processes that require asynchronous data access and data exchange emanating from both client and server site of the system.

2 Software used for creation of particular parts of web client server systems with asynchronous communication

Basic parts of the system that enables asynchronous communication of web client with server include client part which is in our case implemented as applet and server part of the system which consists of so called servlets. Third part of the system is some library which is in our case implemented as a set of web pages. These pages may set up asynchronous client response as a result of interaction with user of the system. Java script code for calling methods of client applet was used in tested samples of architecture.

3 Software client, its features, ways of modification of client part of the system and implementation of client

It is necessary to define and implement special interface able to accept requirements originating from web pages through Java script in the process of creation of client. Moreover it is essential to suppose that hybrid client will respond to several kinds of requirements and should be able to cope with more than one requirement. This problem can be solved by introducing a collection of so called listeners which are able to elaborate events rising in the environment of HTML pages with help of JavaScript. Possibility of applet event handling and elaborating by other GUI elements that are outside applet is spread in such a way. We suppose mainly the following two ways of implementation of listeners:

- Delegation of responsibility for event elaboration to server using applet
- Direct elaboration of the event by the client side

In the first case basic implementation of delegating listener may be given. In the second case we deal with the certain implementation of listener interface. Let us call it `JavaScriptListener`.

In case we consider and implement JavaScript functionality on the client side (typically web browser side) it is suitable to define simple package of listeners in Java programming language able to elaborate the events by client side of the software. In addition to it, it is suitable to create the library (in JavaScript pieces of code) looking after transferring events of browser to applet (hybrid client). The following sample of the code shows interface that defines heads of basic methods representing listeners. They are in fact methods implemented by client.

```
public interface JavaScriptListenerInt {
public void requestExample(String clientId, String strExample);
public void requestExample(String clientId, Object objExample);
public void requestExample(String clientId, InputStream isExample);
public void commandServer(String clientId, String strCommand);
public void commandServer(String clientId, String strCommand,String strData);
public void commandServer(String clientId, Object objCommand);
public void commandServer(String clientId, Object objCommand,Object objData);
public void commandServer(String clientId, InputStream isCommand);
public void commandClient(String clientId, String strCommand);
public void commandClient(String clientId, String strCommand,String strData);
public void commandClient(String clientId, Object objCommand);
public void commandClient(String clientId, Object objCommand,Object objData);
public void commandClient(String clientId, InputStream isCommand);….
```

These methods can be called through JavaScript. They can be divided into three groups.

- Methods that hand in commands (tasks) to client only. These methods serve the purpose of change the state of client or its graphical user interface (GUI). Methods can be in service of combined request for client and server at the same time.
- Methods that hand in commands to server only. These methods are of course also handed in through client as there is not possible or advisable to address server directly by JavaScript, as it might be safety hazard.
- Methods that contain various requirement are coming from web pages. In spite of the fact that they may be included into one of the two previous groups, they are dealt with separately, as they serve for solution of very specific requirements.

Software implementation of above mentioned methods successfully uses so called overloaded variants of methods, i.e. methods having the same name (identifier) but different number and/or type of parameters. Our implementation uses String, Object and Input Stream types in the role of parameters.

4 Elements of architecture of the system

Let us describe and comment one possible proposal of common software architecture of the system from the implementation point of view. Next figure shows basic concept of architecture.

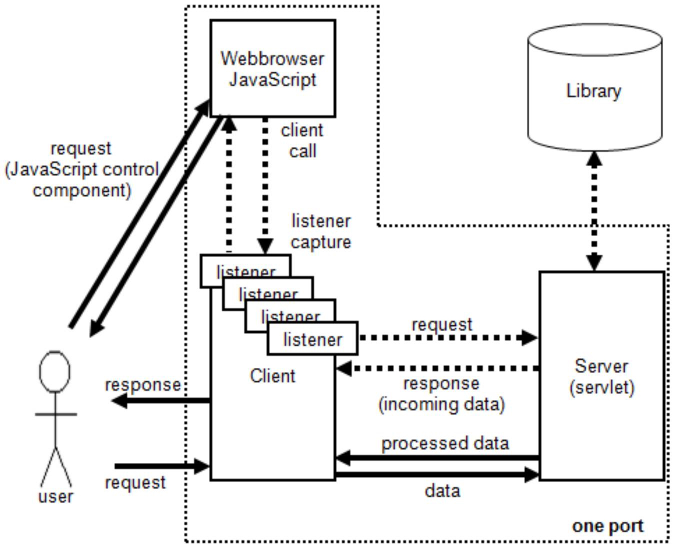

Fig. 1: Basic concept of architecture

The crucial task of the system is correct elaboration of response of client and server to the events delivered by Java Script. The result of such responses may cover up for example changes of content of selected window of the browser, some change affecting client only, server only or both client and server, various combinations of these tasks and many other responses. Due to these reasons the response is executed individually in the process of applications of various listeners. It is but also possible to assume partial solution which is analogous to that by listeners in applets, but is implemented on browser and through Java Script. Above mentioned solution offers the ideal possibility of binding applets with web pages in accordance with intentions and ideas introduced during time period when Java and JavaScript were invented and started to spread. Java Script that covers completely the whole area of processing events for both hybrid client and server side is necessary presumption for establishing common concept of the system capable to integrate more complicated tasks into the environment of web browser. Making use of Java script in the role of communication mean between applet and web page seems to be very reasonable possibility and may be enriched by some attitudes used in AJAX technology etc.

5 Way of communication between/among various elements (parts) of the system

Important step towards creating of common architecture is establishing (defining) common protocol for communication among various parts (modules) of the system. In case of implementation of our system is for this purpose established special class called Protocol. The communication inside the system is performed by special objects. These objects also serve for data exchange.

6 Class Protocol

Class Protocol has the following main characteristics:

- The class has ability of mobility among various parts of the system so that its descendents could be serialized.
- The class possesses special static method for verification of special marks of the protocol. It also implements methods for basic manipulation with specialized object HashMap that serves as information entry.

7 Class HashMap

Class HashMap proved to be very useful possibility and good help for solving our protocol problems. Class HashMap organization is similar to that of array. Classical array is indexed with indexes of type integer. HashMap but is indexed by so called key, which can be some object. Type string was used in our case for the purpose of indexing. String keys of course have to be unique. The advantage of this solution is in the fact, that in such a way we can include into HashMap some data or information. Keys possess role of meta information of some kind also. Consequently keys may be used for identification of elements in the HashMap and so almost any kind of information can be inserted into the HashMap object. Keys must be defined in the scope of protocol so that we could recognize what was inserted into the Hash map. There are two possibilities for making key definitions.

- All key are defined in predecessor of Protocol object.
- Keys are defined in ancestor objects of Protocol object.

The first possibility is more compact as all keys are available in the scope of one class. This variant is but less synoptic if we use it in the source code of the application. In case we need to set up a new module of the system, we have to define all new keys in this class. Any application will have all the keys for its disposal. It holds also for those keys that have no relation to application and are not necessary for description of data directed for communication between the new module and the rest of the system. The next figure shows hierarchy of Protocol classes in the second variant of solution.

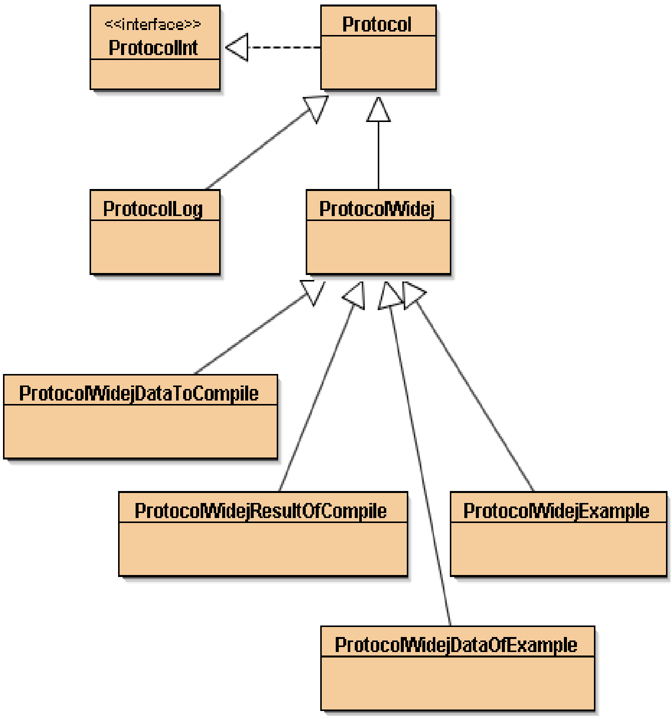
<<interface>>
ProtocolInt
Protocol
ProtocolLog
ProtocolWidej
ProtocolWidejDataToCompile
ProtocolWidejResultOfCompile
ProtocolWidejExample
ProtocolWidejDataOfExample

Fig.2: Hierarchy of Protocol classes

The second variant is less compact. In this variant it is supposed that descendents of basic Protocol class will be set up gradually. In this case only basic keys destined for all classes are defined in predecessor class. Ancestor Protocol classes define keys necessary for specialized modules. In such a way more classes will rise, but on the other hand within a specific protocol will always be available only keys that are developed for the required module. It is necessary (after the adoption and verification of the protocol showing that accepted object is an instance of the class Protocol) to retype it to a specific type in this variant. The following sample source code shows one possible declaration of the Class Protocol.

```
public class Protocol implements Serializable,
ProtocolInt {
private HashMap data;
private int counter;
public static final String DELIMITER = " -> ";
public Protocol() {
        data = new HashMap();
        counter = 0;
}
public static boolean verifySelection(String
unknownSymbol,
        String protocolSymbol) { . . . . . . . }
public int getCounter() { . . . . . . . }
public void incrementCounter() { . . . . . . . }
public String getTime() { . . . . . . . }
public void setAllData(HashMap data) { . . . . . . }
public void addNewData(String key, String d) { .. }
public void addNewData(String key, String d,
String time) { . . . . . . . }
public void addNewData(String key, Object o) { . }
public void addNewDataWithCounter(String key,
String d){ . . . . . . . }
public void addNewDataWithCounter(String key,
String d, String time) { . . . . . . . }
public boolean addDataToSpecificKey(String key,
String d{ . . . . . . . }
public HashMap getAllData() { . . . . . . . }
public Set getKeys() { . . . . . . . }
public Iterator getKeyIterator() { . . . . . . . }
public boolean isDataEmpty() { . . . . . . . }
public String getDataAsString(String key) { . . . . . }
public Object getData(String key) { . . . . . . . }
```

Class Protocol implements the interface for serialization and interfaces defining some methods. It consists mainly of methods that allow manipulation of HashMap, particularly addition of other elements with specific keys, returning values for the key. It also includes possibility of returning set (Set) and the iterator to obtain an overview of all the keys in the HashMap. Class implements static method enabling to compare keys. (Especially in case that it is necessary to ask for a specific element identified in HashMap key it is crucial to find out if there exists a named key, which is to be used for returning of the element. In other words, it is appropriate to determine whether such a key is defined as part of the protocol. If the key is not found, it does not exist even in the HashMap and therefore the element (value) to this key does not exist. If there is raised a requirement to return an element that has not a key, then methods GetData method getDataAsString return null or empty string. It is extremely dangerous to require an element of the HashMap based on the key that does not exist. The following source code fragment shows the use of a specific protocol on the client side.

```
Protocol p = new ProtocolWidejDataToCompile();
p.addNewData(ProtocolWidejDataToCompile.PRO
TOKOL_REQUEST_WIDEJ,
ProtocolWidejDataToCompile.PROTOCOL_REQ
UEST_WIDEJ);
p.addNewData(ProtocolWidejDataToCompile.PRO
TOCOL_COMMAND_COMPILE,
ProtocolWidejDataToCompile.PROTOCOL_COM
MAND_COMPILE);
. . . . . . .
p.addNewData(ProtocolWidejDataToCompile.PRO
TOCOL_COMMAND_COMPILE_CONSOLE,
ProtocolWidejDataToCompile.PROTOCOL_COM
MAND_COMPILE_CONSOLE);
p.addNewData(ProtocolWidejDataToCompile.PRO
TOCOL_PARAMETERS_MAIN,widej.getParamet
ersMain());
p.addNewData(ProtocolWidejDataToCompile.PRO
TOCOL_MAIN_CLASS,
sourceProcess.getMainClass());
```

This is a piece of code that is typically called in the process of creation of data translation protocol. Client adjusts data to desired form and establishes an instance of the Protocol class. Instance is then sent to the server. Let us note that the instance of the class ProtocolWidejDataToCompile is created. Reference to this instance is stored in a variable, which is typed to class Protocol. In one version of our system information all activities are collected and gradually saved it in pre-defined log files. Saving data to a file in Java is very similar to sending such data over the network. In essence it consists of opening a stream, sending data and closing a stream. Location where the data should be sent, is determined by type of used stream. Moreover, special modifiers so-called decorators may be used to influence or affect the stream. For example, we can send a common object and through the combinations of various decorators and other streams, such object can be sent over a network through the socket associated with the server (or client), or such an object can be saved to a file. This similarity may be also easily used to create log information. Original function of the class responsible for making logs was to store text data in a file (either by overwriting the entire file, or by adding such data to the end of the file). Class was gradually refined (improved) to be able to create a log file only on the basis of log data. This means that data and information that need to be recorded to a log file on the disk server are no longer collected in plain text form.

The new instance of the class `ProtokolLog` is created at the beginning of each process. Necessary information is stored during running of the process into this instance. This instance is forwarded to the logging class that takes the protocol and saves it in the text file at the end of the process. Advantage of this solution is obvious. In particular, it simplifies handling logs and makes the information stored in the log files more transparent. As the class `ProtokolLog` is descendant of the class Protocol, it can be sent together with the socket to the client while maintaining (preserving) the structure of data and log information. The recipient does not need to know the structure, because the information about the structure is included in the protocol. On the server side (the following example), the situation is similar to that of the client. It is known that the incoming data are sent by the client. Let us imagine that client requires for example translation of source files. Instances of the protocol and the protocol log are created. Processing of incoming data follows. The next program segment shows more details.

```
ProtocolWidejResultOfCompile compileData = new
ProtocolWidejResultOfCompile();
informationProcess = new ProtocolLog();
informationProcess.addNewData(ProtocolLog.PRO
TOCOL_CLIENT_ADDRESS,
mySocket.getInfoAboutConnection());
. . . . . .
 .
informationProcess.addNewData(ProtocolLog.PRO
TOCOL_INFORMATION_PROCESS,
"Compilation process of console",
informationProcess.getTime());
. . . . . . .
MakingFilesToCompilationProcess mfcp = new
MakingFilesToCompilationProcess(informationPro
cess);
mfcp.saveClassesWithPackages(pathToRootOfNew
Appl,
(HashMap)p.getData(ProtocolWidejDataToCompile
.PROTOCOL_SOURCES));
. . . . . . .
ProcessingOfMainParameter pmp = new
ProcessingOfMainParameter();
p.getDataAsString(
ProtocolWidejDataToCompile.PROTOCOL_PARA
METERS_MAIN));
. . . . . . .
String classNameWithPackage =
p.getDataAsString(ProtocolWidejDataToCompile.P
ROTOCOL_PACKAGE_
MAIN_CLASS_OR_APPLET_WITH_DOT)
+p.getDataAsString(ProtocolWidejDataToCompile.
PROTOCOL_MAIN_CLASS);
```

In case we want to design sufficiently common and widely useful architecture we have use special adaptable mechanism. One possibility is to use systematically class `Protocol,` which is ancestor class of all protocol classes. The class instance can and need to be retyped only in the moment, when it needs to obtain data. This is the only possibility to ensure that the architecture is as much independent on particular data exchanged between modules of the system as possible.

8 Basic requirements placed on the architecture and essentials of its functionality

We will describe requirements and principles imposed on the architecture so that it could be used for various purposes mentioned in this part of the text. Basic requirements placed on the architecture are the following ones:

- Possibility to send requirements to the server with the help of web browser client.
- Possibility to execute such requirement on the server i.e. possibility to execute specific processes on server. Processes may for example translate/interpret source code of programming language, establish various calculations, create database table, and do many other tasks.
- Possibility to send back and show results of these executed processes on the web browser, on the client etc. (It depends on specific requirements placed on the system and on specific situation).
- Possibility to create easily simple testing examples.

The architecture is based on the following essential facts/principles:

- The client transforms data to the form of `Protocol`.
- Server can accept everything which is instance of the class `Protocol`.
- Server can run everything which is instance of the class `Requirement`. Instance saved in the

variable `Requirement` specifies what is to be done on the server with data (translation/interpretation calculation etc.).

- Client accepts and elaborates data in case they are encapsulated in an instance of the class `Protocol.`
- Client runs, redirects etc. obtained data according to its function.

This architecture is very flexible. Any new functionality of the system is established relatively simply by introducing the new client implementing extensional purpose.

9 CASE study

Described software system with rich asynchronous communication can be of wide use in the area of education. It can help to eliminate some insufficiencies of widely used learning management systems (LMS) and in such a way enable efficient distance and combined education of some subjects. The system in fact enables running any software application in the environment of LMS.

Let us summarize some important problems and limits of present LMS. We will reflect only the problem of effective teaching of programming in some programming language.

- Current systems lack some important features and functionalities, which include mainly the possibility of the presentation and manipulation with the source code of some programming language. If we take into account that the LMS is typically available only through a Web browser, the issue of manipulation with the source code is complicated. User - the student of some course, that requires such manipulation (for example course of online LMS supported programming) should be allowed to view the source code and at least modify it. This question can be satisfactorily solved by integrating a text editor or text components in the system.
- Another specific feature that actual LMS lack is the ability to manipulate comfortably and efficiently with source codes of software packages, for example programming languages environments. In case of Java and/or other programming languages environment it comprises, loading, modifying, translation/interpretation etc. As indicated in the previous paragraph, editing can be partially solved by integrating text components in a Web page, for example in the form of a text box using JavaScript technology. The problem that concerns compiler is rather more complicated. The issue of integration of compilers seems challenging, as it is certain, that the compiler will not be possible to implement on common Web sites. Therefore, we will talk about a "mediated translation", i.e. the compiler that from the user point of view will appear as integrated in the website, but in fact its main components have different location and the user will meet only with the intermediary in the form of a graphical user interface that he or she uses and controls. So the system is or may be distributed in several places. The client site is typically by the user and the interpreter is located either as part of the LMS, or somewhere on other distant server.
- Next specific feature, which arises particularly in connection with the support of teaching of programming is the ability to run translated applications in the LMS environment in user interface of the web browser only. Even if we have a system that is accessible only via the web interface it should allow us to search a sample application (i.e. its source code) through this interface, enable us to read/ modify this code and finally, allow such to translate the code. All these operations are performed inside a Web browser environment, i.e. in an environment of not very operational Web site. Then comes the question of how the compiled application in a web browser transforms into application and runs it on the web. This feature is also of prior importance for students and influence substantially efficiency of education process. It is very important for students to be able to create simple applications in Java or other programming language or modify existing one, to compile it and run it and see the results not leaving LMS environment. Otherwise the efficiency of education process is very limited as students have to do important parts of their work outside the environment of LMS. Students should be enabled to perform all the activities only in the scope of the LMS and access it via a web browser. These activities include: making the source code or editing existing one (for example, some sample code), transfer of the source code and building executable application (provided the source code written by a student is error free) and last but not least, run it via Web browser. This step is very important and its implementation can be quite challenging.
- Another required feature is capability to store and retrieve students work in-process. In our case of the course of programming it is usually

the source code of some program. LMS should posses the ability to store the code easily in the file and enable reverse manipulation, i.e. retrieve the source text from a file into the LMS environment. These processes may have several options and variations and is rather problematic as it is very implementation-dependent.

- Finally LMS has to address the problem of sample examples, i.e. the sample source codes. Such source code should be available again in LMS environment and accessible through Web browser, and students should be able to choose suitable sample examples, view them and manipulate them (loading to the editor of source codes, carry out modifications of the code, and then translate and run them).
- The last feature is certainly easy work with the environment (system or component). Emphasis will be placed mainly on a clear, easy to use and friendly environment.

The solution of these problems is in including new software component capable to add the required functionality to existing functionalities of LMS. Such component should enable to run (execute) Java environment and other software packages from the environment of LMS in such a way that students are not aware of difference between working inside LMS and working in Java environment and/or other native software packages.

The software component described in the article and based on asynchronous communications proved to be fully capable to integrate into LMS and enable to fulfill mentioned demanding tasks. Software system was capable either to be executed from LMS or to run without dependence on LMS. System is easily accessible from any place in the internet via web browser and web interface. System is easily capable to run any set of test examples and respect all standards and environment details students got used during direct work with Java environment and/or other software packages. The installation and configuration requirements revealed to re very minor. The testing was carried out at the Faculty of Informatics of Management of the University of Hradec Kralove and repeatedly used in distance education of subject Object oriented programming in Java. Java environment was addressed and executed from LMS Moodle and all additional features and functionalities were achieved. Other testing included successful running software very complex and extensive software package Matlab from LMS Moodle again without problems. System was partially and successfully tested also with a variety of mobile clients.

10 Mobile phone as a client device

Mobile phone has some very special features. These features can be also successfully used in distance education:

- Mobile phone is the first very personal device. Mobile content is typically accessed in total privacy, it is not borrowed or shared. Other media are often consumed in groups. Sharing is restricted by size of the apparatus and mainly size of display.
- Mobile phone is always handy. Vast research documents it by frequency with mobile phone is used.
- Mobile phone is activated almost all the time (at least in stand by mode). The reason for it is simple Telephone is directed for on line communication. So it is always open channel for consuming entertainment, messages and also information of any kind.
- Mobile phone has built in paying mechanism. It enables to arrange payment from bank account or by special SMS by several clicking. SMS enable extremely simple transfer of money and a large layer of entertaining industry is based on it.
- Mobile phone enables rich work with media such as taking snaps, videos, music and spreading it by media channels. User made content is very popular recently for example in the scope of YouTube or Facebook social networks.
- Mobile phone can provide the most precise information about its user and mobile operators may very precisely focus on target groups. AMF Ventures measured the amount of data about user that could be obtained about user from various medias in the year 2007: 1 per cent was gained from TV, 10 per cent from the internet and 90 percent from mobile phones.
- Mobile phone can record social connection of consumption. Social connection does not measure what we do but with whom we do something. The frequency and addressee of telephone calls and SMSs enable to model social networks. It is possible to assume that members of social networks will have a lot of common interests.

11 Specificity of mobile applications

Mobile applications should be able to make use of its main advantages in comparison with computers. It is mobility (possibility of connection in any tine and on any place) and rich connectivity. These advantages are but accompanied by some limits. Low cost of hardware signalizes of course significantly less computation performance. Applications requiring high computational performance (on client mobile side) are not suitable for mobile devices. Even applications requiring less computational performance should be optimized. Substantial limit in many kinds of applications (including educational applications) is small size display and its different shape that computers have. Height of mobile display is higher that its width. As only several text or graphics elements can be displayed, it is crucial to simplify mobile outputs as much as possible. Task to select only main graphical elements and to omit/ shorten displayed text and not to restrict intelligibility and functionality of application is not an easy one.

Mobile phones have also restricted keyboard and even in case that have full keyboard its not very comfortable. So management of application should be also restricted as to the number of required inputs and outputs. Often it is necessary to reorganize or restrict subsidiary functions so that to gain more straightforward achievement of major functions. Navigation should be carefully designed. Each application should have in any situation easily accessible possibility to undo previous navigation step and easily accessible possibility to exit the application. Application should be able to save its content state so that it could be in any time interrupted or exited. Serious problem is number of various mobile devices and absence of common standards. Application should be designed as universal as possible. For example position of elements on the display should be calculated proportionally to size of the display. Problem is also existence of several platforms. They differ by their purpose, programming language, firm that has invented them etc.

Well known are for example platforms – JME, Android, BREW, Adobe Flash Lite or Microbrowser, platforms for Symbian operating system Windows CE, iPhone, BlackBerry or Palm. Platform Java ME is widely supported by Nokia, Sony Ericsson, Motorola, Samsung, LG or Siemens and at present hundreds of telephone kinds support it. Even very cheap kinds of telephones support Java applications. These applications can run on various mostly spread operating systems including Symbian or Windows CE/Mobile etc. Java virtual machine provides also good environment for running these applications. Java applications should have big advantage in independence on host environment and portability. Each mobile device has but some specific libraries and specific functions and so real portability is restricted. Platform JME uses Java language and enables application development also for other devices such as set top boxes. Android is the platform and operating system for mobile devices developed initially by Google company and later by Open Handset Alliance (OHA), which consists of 47 mobile companies including mobile operators and software firms. Their operating system is based on Linux kernel and Java language is directed for application development. Android is developing and is not spread very much now As among members of OHA are powerful firms such as Google, HTC, T-Mobile or Intel, it can be supposed good future of this platform.

.NET Compact Framework (.NET CF) is the platform of Microsoft company used for small devices such as mobile phones or set top boxes. It is based on .NET Framework, similarly to JME of J2SE, and adds some specific libraries. .NET CF platform is accommodated to limited performance of smaller devices. . Programming on this environment typically uses storing data in Microsoft SQL Server Compact Edition and classes for communication with Microsoft Office Outlook Mobile application etc. Consequently .NET CF is closer to desktop applications that JME profile MIDP) with its RMS and sandbox model.

.NET CF are typically used on more expensive mobile phones having operating system Windows CE (Windows Mobile a Windows Embedded CE). Applications for this operating system can be written in C# language or in Visual Basic.NET.

12 Client WideJ for asynchronous communication for mobile phone client

Mobile software clients enabling rich asynchronous communication and possibility to adopt described software client for various mobile devices were tested during solution of a project granted by the Czech grant foundation GAČR No.. 402/08/1046 „Models of firms having mobile architecture. Possibility of various connections of mobility and educational process were assessed in the environment of mentioned software system WideJ with various mobile devices. Advanced mobile phones and PDA having their own operating system and wide variety of functionalities begin to prevail in the market. These apparatuses can master a lot of built in functions and enable installation of other applications. One of them is already mentioned mobile Java (J2ME). Java platform could enable enough functions to run in the article described mobile client on mobile phone or PDA. We should also mention last development in the area of m learning. Area of programming is not very suitable and comfortable for systematic use of mobile devices and restriction to only easy applications is advisable. Client WideJ described in this article enable to create any application of the type console, applet or frame, but mobile phone using Mobile Java enables to run only so called MIDlets. MIDlet is application similar to frame application but is restricted mainly by its graphic user interface GUI and by the range of functions it can offer to the user. This fact seems to be both advantage and disadvantage. Three types of applications are reduced to only one type-MIDlet. Advantage is in the fact we can run MIDlets only. Disadvantage is in transformation of the three types of application to MIDlets. This fact is not big problem in case we succeed in redirecting inputs and outputs and in fact emulate the terminal.. Applets and frames possessing more complex GUI may cause difficulties, as it may be impossible to transform such GUI to the environment compatible with mobile phone environment. Efficiency of education of programming languages is also important practical problem. Bearable organization of education assumes using the library of ready or almost ready examples but idea of preparing the whole source text on mobile phone is unreal as the process of making source file is very uncomfortable. Source code of programs are not plain common texts but comprises vast proportion of commas, asterisk, semicolons, slashes and other special characters not widely used or completely omitted in common SMS messages or e mails. As Java is case sensitive frequent switching between lower cases and upper cases is typical.

Consequently major tests were performed mainly with different application making some server mathematic calculations. This application proved to be more comfortable for practical use of testing. Some computer algebra system (Matlab, etc) may serve for this purpose on the server side. Lack of mobile phone calculation capacity is avoided and capacity of server side is made use of. Distance course of mathematics recorded in LMS enabled to access to this course through web client of mobile phone and sample calculations were demonstrated by MIDlet connected to server side of the WideJ system. Such use of client proved to be more useful that the more complicated but due to restrictions of mobile devices rather uncomfortable client enabling all activities in education of programming. The most comfortable and adequate to mobile phone showed the client enabling complicated calculations not performable on mobile client due its lack of computation performance. In this case good solution was connecting to the client (MIDlet) of WideJ system and handing over the calculation to it.

13 Netbook as a client

Very favorable netbooks are very adequate and promising solution covering the whole area of distance education activities. Netbooks of course have some restrictions in comparison with bigger computers. They make use less source demanding operating systems such as Linux, Windows XP or Windows CE) and low power consumption processors such as Intel Atom, VIA C7, VIA Nano or AMD Geode. Due to their limited dimensions and weight and stress on electricity consumption they don't have CD-ROM/DVD-ROM.
(It may be external accessory only) only and are used so called Solid State hard discs (2,5 or 1,8 inches). On the other hand they have rich possibility of connection to the Internet (Ethernet, Wi-Fi), communication with other devices either wire (USB, FireWire) or wireless (Bluetooth, EDGE, IrDA)..Typical display dimensions are 7, inches – 10, inches and distinction 800 × 480 pixels to 1 280 × 768 pixels, weight t 0,9–1,5 kilograms, battery can last 2–10 hours.

14 Client System for asynchronous communication for netbooks

Using WideJ client for netbooks does not qualitatively differ from using this client for common computers and so all conclusions of parts 2 to 7 of this article holds for netbooks with only minor changes. In the wide area of application in education the restrictions are minor in comparison with bigger computers.

15 Experience gained

Mentioned case study and other practical projects can be partly summarized.

- Mobile solutions enable to manage more efficiently the burden of information system in time and reduce peaks thanks to unlimited on-line access. For example total (the whole day covering) orders submitted in the evenings are minimized. In case of education mobile solutions enable students to organize their study in as flexible way as possible from the point of view of place and time.
- Customers and staff should be equipped with necessary mobile devices for reasonable starting of mobile application. Various marketing activities should serve as good motivation and the expenses should be covered by the company. Substantial client penetration of comparable parameters and software is necessary condition of success of the project of mobile education
- Investment to mobile communicators brings benefit in the form of more effective communication inside firm (Devices include mobile telephone, e-mail terminal,…), Effectiveness of the mobile solution increases rapidly if making use of mobile solution is planned in more projects of the firm. On the other hand using mobile solution in one project only will not pay. In case of universities it is very supportive for university to have mobile

access for more kinds information systems and other software packages

- Liking for mobile solution will increase if the application uses mobile device which has built-in telecommunication functions. In case of education this function is almost necessity as it is condition for access to information sources and alternative education resources.

The key aspect for the sustainability of a mobile solution is an "uncertainty of the location", and the need for cooperation with external resources (from the process-point of view). In case of "certain location" the mobile solution is usually later replaced by more comfortable computer terminal. Especially in case of education mobile solution is of wide use for distance form of education and for self study but has much less use for presence form of education.

The partial investigation shows that in the future mobile information systems formed by a set of mutually connected components which collect information, administer processes, store and distribute information and make use of mobile ICT technology will gradually become more popular and in certain situations will prevail. More and more existing processes will be mobilized in near future i.e. part of processes or even the whole processes will be prepared for mobile use. Transformation of processes into mobile ones will include: software processes, service processes, operational processes, and most of business oriented processes. Thinking of mobility features of processes and transformation of processes into mobile ones results in changes of their structure, changes of supporting technologies, administration and management of processes. The main distinction is in the way of decision about what data should be accessible from mobile terminal with reference to efficiency of mobile communication and the whole management of mobile processes. Transformation of processes into mobile ones usually concentrates on selecting their concrete parts. More fundamental transformation may result in higher efficiency of the whole business processes and these changes should go hand in hand with transformation of process administration into SOA architecture (Service oriented architecture). Mobility has also large potential in future types and forms education.

16 Conclusion

The system WideJ for asynchronous communication described in the article proved to be very flexible, easy to modify. In such a way it promises wide range of applications. Various variants of the system contain some core ideas that can serve as design patterns in the area of asynchronous communication. Other applications are prepared and tested.

This contribution is supported by the grant project No 402/08/1046 of the Czech grant agency (GACR): Models of firms having mobile architecture.

References:

[1] Malý, F., Kozel, T., Slabý, A.: Mobile approach, trends and technologies in modern information system. *Proceedings of the 7th WSEAS International Conference on APPLIED COMPUTER & APPLIED COMPUTATIONAL SCIENCE (ACACOS '08).* Hangzhou, China, April 6-8, 2008. ISSN 1790-5117. ISBN 978-960-6766-49-7

[2] Malý, F., Kadavová, M., Slabý, A.: Key Factors Involving the Design of the System of Virtual University. *Proceedings of the 7th WSEAS International Conference on APPLIED COMPUTER & APPLIED COMPUTATIONAL SCIENCE (ACACOS '08*). Hangzhou, China, April 6-8, 2008. Published by WSEAS Press, s.678 – 683. ISSN 1790-5117. ISBN 978-960-6766-49-7

[3] Malý, F., Kozel, T., Slabý, A.: Mobile access, trends and technologies in modern information systems. *WSEAS TRANSACTIONS on BUSINESS and ECONOMICS.* China, 2008. ISSN 1109-9526

[4] Gaurav Sethi and Manoj Kumar, Simulation and Comparison of Communication Protocols in Adhoc Networks", *Proceedings of 7th WSEAS International Conference on ELECTRONICS, HARDWARE, WIRELESS AND OPTICAL COMMUNICATIONS*, Cambridge, UK, February 2008.

[5] Abdullah Gani, Qi Han, Nor Badrul, Anuarand Omar Zakaria, Enhancing DSR ProtocolPerformance in Mobile Adhoc Network usány ACK Reply", *WSEAS TRANSCATIONS ON COMMUNICATIONS*, Vol.8,No.2, pp.227-236,February 2009

[6] J.-Y. Choi, S.-J. Yim, Y. J. Huh, Y.-H. Choi -A Distributed Adaptive Scheme for Detecting Faults in Wireless, *WSEAS TRANSACTIONS on COMMUNICATIONS*, Issue 2, Volume 8,February 2009

[7] A. Gani, Q. Han, N.B. Anuar, O. Zakaria -Enhancing DSR Protocol Performance inMobile Ad Hoc Network Using ACK Reply,*WSEAS TRANSACTIONS on COMMUNICATIONS*, Issue 2, Volume 8, February 2009

[8] Khayati N., Lejouad-Chaari W., Moisan S.,Rigault J.P., Distributing Knowledge-BasedSystems Using Mobile Agents, *WSEAS TRANSACTIONS ON COMPUTERS*, Vol. 5, No. 1,January 2006.