Guidelines for Improvement Information Processes in Commerce by Implementing the Link Between a Web Application and Cash Registers

Julian Vasilev[a,*]

[a]*Varna University of economics, Department of Informatics, 77, Kniaz Boris I, blvd., Varna, Bulgaria.*

Abstract

The main task of the article is clarifying the content of different types of costs (which are associated with providing information about business processes) and on this basis an identification of opportunities for their reduction may be found. In this regard, this paper examines business processes in retail in rendering account of sales. A special place in the article is devoted to the analysis of existing models of information technology systems in commercial enterprises. The important question for the adaptation of web applications in commercial enterprises has been developed. Factors contributing adaptation of Web applications in business practices are taken in account. Factors hindering their adaptation (such as the relationship of a web application with cash registers a problem that is not widely discussed) are also reviewed. A discussion of various options for improving both the technological model of the information system in enterprises and specific guidelines for the quantification of the proposed approaches to reduce costs is made.

Keywords: Web application, cash register, fiscal printer, commerce, Delphi, Intraweb, Bulgaria.
CCS: D2.
JEL: I23.

1. Introduction

Reducing the cost of providing information for business processes can be made after a thorough analysis of the performed information processing. The study of information processes is performed in order to reduce the cost of hardware and software. An especially acute problem is the problem for measuring the cost of hardware and software (for purchase, maintenance and power supply). Under current conditions, operating costs of buying and maintaining hardware and software are distinct from an accounting point of view, but they are not subject of extensive study by a managerial perspective. This feature prevents their full, thorough and objective study to find

---

*Corresponding author
Email address:* vasilev@ue-varna.bg (Julian Vasilev)

specific ways for their reduction.

Rationalization of IT processes as an indirect effect gives rise to conditions for reducing managerial costs. Even though the cost of computer hardware and software declines steadily, companies do not report cost reductions. The article gives specific suggestions for reducing the cost of buying and maintaining hardware and software systems in commercial enterprises. Despite the fact that the costs of buying and maintaining hardware and software systems in commercial enterprises have relatively small share of total expenditures, these costs are subject to monitoring by managers. The article discusses concrete opportunities to reduce these costs in implementing the managerial processes. Several approaches for cost reduction are discussed. Options for rationalization of communication processes are considered.

At this stage, information systems used for sales in stores (known as POS systems) are generally associated with a cash register or a fiscal printer. The Bulgarian market offers a wide variety of both: (1) cash registers and fiscal printers and (2) software to track sales. It should be noted that some software products to track sales technologically implement the connection with a cash register. In the rest of software products for tracking sold goods the sales registering process is done twice at a computer and at a cash register. It is obvious that connecting an electronic cash register (ECR) with fiscal memory (or a fiscal printer) to a computer is not an easy task. There are examples of software companies where programmers applying for jobs are not approved because they do not know how to connect software to record sales with an ECR or a fiscal printer.

The *purpose* of this article is to improve the existing technology model of an information system for recording sales in retail outlets in order to reduce the cost of hardware and software. To achieve the objective we have to solve the following *tasks*: (1) to examine the current technological models, (2) to make a proposal for improvement, (3) to demonstrate the need to connect a web application with cash registers (4) to explore existing ICT and (5) to develop a specific program to connect a web application with cash registers. The *subject* of this study is information technology for development of web applications. The *objects* of this study are communication technologies (both low and high level) for communication between software (both desktop application and web application) and cash registers.

The paper is organized as follows. Section 2 expresses an analysis of existing technologies for providing information for retail business. Two existing technology models of information systems in retail are described. Section 3 presents several guidelines for improving the model of information support of commercial processes. Subsection 3.1 contains a new (enhanced) model the second technology model is further developed. Thin clients are used instead of desktop computers. The adoption of the enhanced model and the link between a desktop application and cash registers are described in subsection 3.2. Software aspects for the implementation the link between a web application and fiscal printers are illustrates in subsection 3.3. The conclusions are outlined in Section 4.

2. Analysis of existing technologies for providing information for retail business

Commercial processes are widely known both at home and abroad. The need for rapid recording of sales leads to adaptation of sales software systems. In the course of time many experts have studied the business processes. They have offered a variety of improvements. The use of barcodes for automatic identification of goods and materials is a nice example. Sales are registered in a database. Nowadays analogues of barcodes and databases are not found.

In the early 90-ies of XX century, some software companies connect their automated information system (AIS) with a cash register. The technological achievement is significant and it is appreciated by retailers. For software developers (Application Software Providers ASP) the implementation of such a system leads to realization a significant revenue for a short period of time (having in mind that the software market does not offer similar products). Normally other software vendors also try to enter the market as an attempt to connect their AIS with a fiscal printer (FP). The available technical documentation offered 20 years ago by producers of fiscal printers (and ECRs) in Bulgaria is clear that a FP is connected to COM port via RS-232. Most modern laptops do not have a COM port.

Unfortunately manufacturers of FPs for a long period of time do not offer the communication protocol extremely necessary to developers of application software to connect their AIS with FPs. Users of specialized software fall into a situation where they have to enter one and the same data in two places in their AIS and on their ECR. Such duplicate data entry takes considerable time and it is a prerequisite for admission of technical errors. There are two possibilities to users of POS systems: (1) to ask the software vendor to enhance his software system so that it connects with a FP or (2) to change the POS system with another POS system offered by another software provider (ASP). Examples from both directions may be given.

There are two recognized process models of software systems for recording sales in retail outlets. It is typical for *the first technology model* of retail that sales are recorded in a cash register and in a local database. A cash receipt is printed. At the end of the day records from local databases are merged on the server database (Figure 1).

The mentioned approach guarantees quick registering of sales. *The first technology model of a software system for recording sales* in cash desks zone is adopted in many enterprises in Bulgaria. We have to highlight that each POS terminal consists of a computer with an installed operating system, a database management system (DMBS), a local database (for registering sales at each cash desk) and a software module which transfers sales transactions from local databases to the server database offline. The first technology model is characterized with high performance and reliability of the software system for registering sales transactions. The high speed of the software is due to the fact the software system is undependable from the local area network (LAN). A significant drawback is the operability of the transmission of data and up to date information on the server. Because data is transferred offline (not online), operational managers do not have updated information about sales. If a manager wants to obtain information online, he cannot obtain it. Such organization of work is suitable for small and medium-sized shops.

To overcome the shortcomings of the first technology model a number of companies apply a different technological model for recording sales that we would conditionally call *second technology model* (Figure 2).
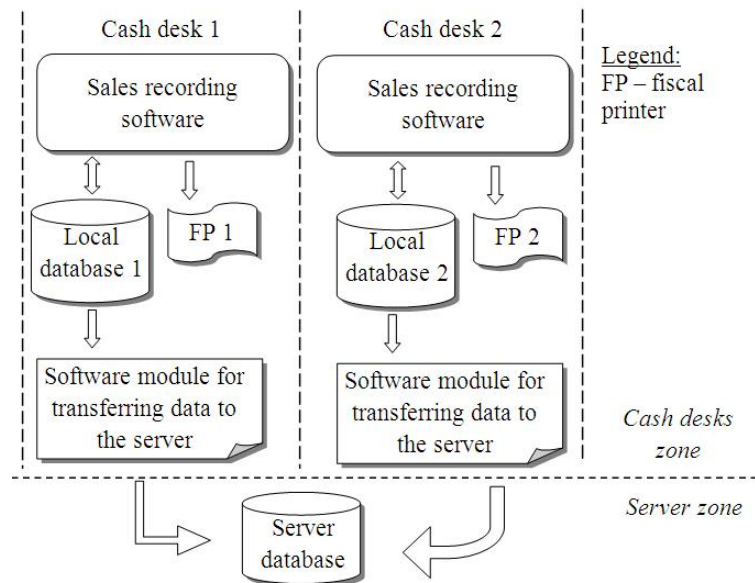
Figure 1. First technology model of a software system for recording sales in cash desks zone.
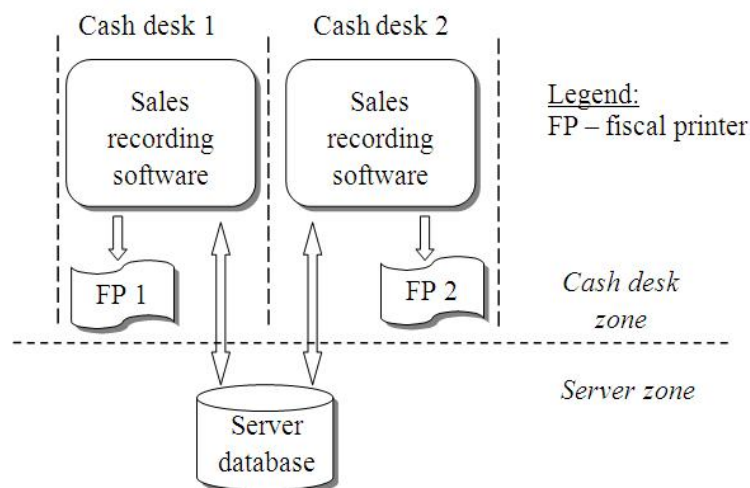

Figure 2. Second technology model of a software system for recording sales in cash desks zone.

In the second approach, data is maintained in a centralized database, which improves the operability of data. In this case, the manager receives updated information about sales. A significant advantage of the second approach is the absence of the need for: (1) the use of a local database and (2) transfer of sales transactions made at the end of the day to the server database.

It should be noted that the second technology model uses a server database. There is no need to install client side DBMS on local workstations. At each cash desk there is a computer with an installed operating system and application software that communicates with the server database (it derives price and the name of a specific item by its barcode and it records sale transactions in

the database). The second technology model is a typical example of an online recording system of sales working in client-server mode.

For its reliability of great importance is the availability of secure network connection between the server (which stores the database) and separate workstations. The cost of implementing the second technology model (which is popular in business practice) requires fewer resources than the first one. Several saving of costs are achieved. As an example we mention costs for licenses of DBMS on individual client machines and the cost of software which periodically transfers the data from different workstations to the server. At the first glance, the second model has no disadvantages. That is why it is widely spread in practice. However, our research continues to seek improvements in the development of the second model. Later in the text we provide guidelines for improvement the model of information support of commercial processes.

3. Guidelines for improving the model of information support of commercial processes

3.1. The use of thin clients

The current status of information systems used in retail outlets is an adequate implication of the second approach. Despite many years of experience of the application of information systems in retail, we can seek guidelines to improve existing technological models. Some possibilities for cost reduction can be found in the following areas: reducing the cost of application software, system software, hardware and power supply. Further, the text puts forward concrete proposals to enhance the second model, in which some costs for both hardware and software may be saved.

It makes an impression that a centralized database (located on the server) is maintain in both approaches. As noted, an analogue of the centralized database cannot be recommended. In terms of software, desktop applications are mostly used in Bulgaria. It is typical for them that they are installed on each workstation. This feature requires the use of a personal computer on every work place. In order to reduce the cost of electricity it is possible to use *thin clients*.

Using thin client does not allow the installation of desktop applications. To be able to use software system for recording sales on hardware devices such as thin clients, there should be a change in the application software to shift from desktop applications to Web app to record sales. In this case, the technological model for recording of sales is as follows (Figure 3).

The recommended third technological model for sales recording is used in some European retail shops, but it is not popular in Bulgaria. The use of an intranet software system for recording sales allows the work of web applications within a corporate local area network. The application software is installed only on a server. It is accessible to workstations through a web browser.

The third technology model is done to reduce costs in the following areas: (1) energy (one thin client spends significantly less electricity than a desktop computer), (2) cost of installation and maintenance of a license for application software on each workstation (application software (software product for recording sales) only be installed on server) and (3) the maintenance costs of hardware thin clients have significantly fewer parts than a desktop computer hence their tendency to damage is lower than a personal computer. Indirectly labor costs are reduced because fewer people are needed to maintain the hardware and software in the third model.

In terms of costs, the application of the represented third technology model leads to cost savings for hardware, software and power supply. Hardware savings are in the following areas. In
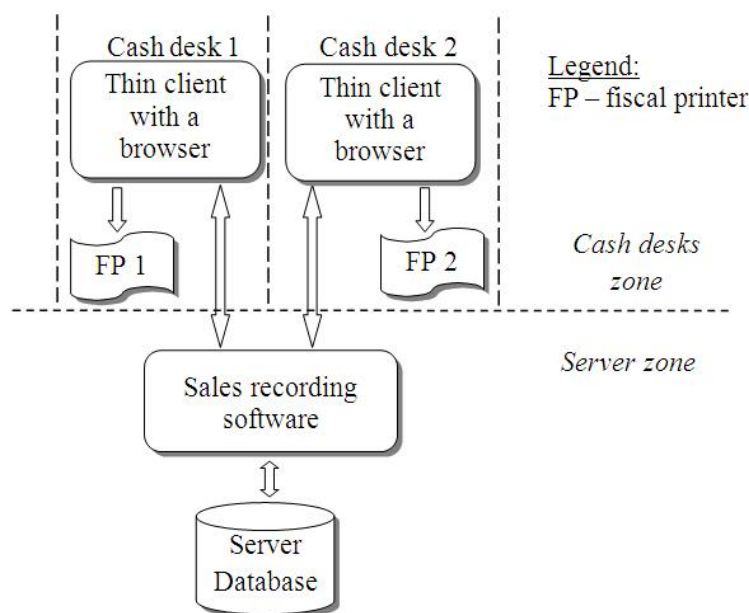
Figure 3. Third (enhanced) technology model of a software system for recording sales in cash desks zone.

the first and second model each workstation computer is configured with a hard disk, a central processing unit (CPU), RAM memory, a monitor, a keyboard and a mouse. The third approach suggests the installation of a thin client (which does not have a hard disk drive) and it is much more compact than a desktop computer. Furthermore, the thin client consumes significantly less power than a standard computer. In terms of software, the time for installation of application software is saved, because it is only installed on the server.

In all three approaches (three technological models) sales data are recorded in a database and a cash register receipt is printed. For the end user the technological model of the information system for recording of sales remains hidden, but the manager is interested in the costs and reliability of the software system.

3.2. Adoption of the enhanced model. The use of middle tier software for implementation the link between desktop applications and cash registers

Elaboration of the software, so that it is connected with a FP (fiscal printer) is not an easy task. When ASP do not provide specification of FPs, including protocol for communication and describing how to send commands to the FP, the form and content of the response, received from the FP, it is a difficult task to write sales software working with a FP. In this case, developers use a specialized software system that communicates with a FP. To monitor the communication between the PC and a FP specialized software may be used called sniffer to eavesdrop the communication between two devices.

When using a software system to record sales, it sends commands to the FP and the program sniffer captures packets from the computer to the FP and packets from the FP to the computer. Using software from the class of sniffers is considered as a hacking technique, but it can be used to

establish communication between the operating system and the FP, in case there is no documentation for FPs.

Observations in business practices indicate that some of the POS systems for recording sales in Bulgaria are connected to fiscal printers (FP). Some software systems are not connected with FPs. In 2012 the situation on the FP market changed. Many businesses need to use a new type of FPs directly connected to the NRA (National Revenue Agency) by GPRS (because the taken by the Government of Bulgaria legislative initiative aimed at displaying the light of the informal sector). This situation has led to an increase in sales of FPs, but also an increase in competition both on the FP market and the software market.

In response to increasing competition some companies, selling cash registers, have published on their website the communication protocol for communication between software for registering sales and a cash register (or fiscal printer). In this case, software developers can download the specification, inspect it and test the connection (communication) with FP on COM port. For testing of the software it is recommended to use a non-fiscal printer.

The new situation provides tremendous opportunities for software providers (ASP) to extend the functionality of their software. It should be noted that a number of other AIS are built to input data and then a cash receipt is issued (as an example we note the payment of interest and fees on credit institutions). A wide range of software vendors can extend the functionality offered by their software so that they connect with a FP. The initiative for further developing the software can be both from the software vendor and from the end customer (the user of the software product). To connect sales recording software with a FP, developers can implement two approaches.

First, communicating directly to a COM port by sending hexadecimal commands directly to the FP (www.daisy.bg, 2013*a*), (www.datecs.bg, 2013*a*), (www.tremol.bg, 2013*a*). The FP returns response: (1) with a successfully executed command or (2) an error code. In this case, the AIS should use a pointer to a COM port that is in standby mode to receive messages from the cash register. After sending the command from the AIS to the FP, the AIS should wait for 100 ms, to get an answer from the FP. In the *first approach* programmer sends "low" level commands form the AIS to the FP. To record a sale at one cash register, a series of hexadecimal commands should be sent. The reply by the FP has to be read.

The specification of some FPs describes the communication between a FP and a PC. The format of the messages between the Host (PC) and Slave (FP), between Slave and Host is given. A description of various types of commands, error codes and status flags for FP are also given. Because the format and content of commands is rather complex, we do not include an example. It has to be marked that only a high qualified programmer can make the communication between Host and Slave. He will write the software for communication between the sales software and the FP. He has to be very familiar with COM port communication. From his side it is required considerable effort and energy to connect the POS system to record sales with a FP, which means that the cost of the software system (Total Cost of Ownership TCO) is increased.

Second, by using intermediate software (middle-tier software), developed by the FP provider and available for: (1) downloading from the company's website (selling FPs) and (2) free to use (www.daisy.bg, 2013*b*), (www.datecs.bg, 2013*b*), (www.tremol.bg, 2013*b*), (www.orgtechnica.bg, 2013). The middle-tier software stands between software for recording sales and the FP. In the second approach, the sales recording software prepares a text file with the sale and the software

copies the text file to the folder where the middleware software is installed. It stays resident in RAM on the workstation. Every 100 ms it inspects for a text file with a sale. If a text file is found the middle tier software prints the sale the middle tier software sends the sale to the FP. After sending sales to the FP, the sales recording software deletes the text file.

The second approach is much easier to implement from the programmer's perspective. A programmer sends a set of commands as a text file. It is structured very similarly to the sale. After that the file is printed. Individual lines in the text file are filled with, items, quantities and prices. It is possible to generate a cash receipt for a sale by departments. In the second approach, the work to print a cash receipt to a FP is significantly easier than the first approach. The cost of developing the software for the second approach is lower than the first approach.

Writing the program logic in the second approach is "high" level. The number of lines of code (LOC) that must be written to communicate with FP is significantly less than the number of LOC for communication with the application of the first approach. The fewer lines of code to write, the likelihood of errors is less and less time to implement fixes (design, programming, debugging, testing and deployment) is shorter.

Most of the software products offered on the Bulgarian market which can communicate with a FP are typically desktop applications (Graphic User Interface - GUI applications). Despite the serious boom in the development of Internet technology, leading to the development of a number of web-based applications, a majority of the POS systems continue to be GUI applications. Most of the web-based e-commerce solutions do not offer a traditional connection with a cash register. If necessary to update the software (replacing EXE file on the server), the application software (installed on POS terminals) must be closed on all workstations. It means stopping work with the sales software system.

For now we can say that single software companies attempt to connect their web-based applications for tracking sales with FPs. The reason is quite simple. To print a cash receipt, desktop applications generate a text file (the second approach) or they communicate directly with the COM port (the first approach) on the local machine that is running a GUI application to track sales.

3.3. *Linking a web application for registering sales with fiscal printers. Software aspects for the implementation the link*

Proposals for improvement (third (advanced) technology model) can be adjusted in business practices. There is a problem with the communication of web application for reporting sales with cash registers.

Web applications are server applications. Web applications are run on a server. Web applications have access to hardware resources of the server. They do not have access to hardware resources on the client machines (workstations). Web applications have access to each user session. *The problem is how to create a text file on the workstation by the web application or how the web application communicates with a COM port on the client machine*. Most of the integrated development environments (IDEs) do not allow web applications (1) to generate a text file on the client machine and (2) cannot communicate with a COM port on the client machine.

As we know, web applications can be created through a number of IDEs. For this study we should seek an appropriate IDE for developing web applications that allows the generation of a text file on a client machine or sending commands from a server to the COM port (again on

the client machine). Previous experience in web applications allows us to choose the Intraweb technology (or VCL for the web developed by Atozed software (www.atozedsoftware.com, 2013) and Embarcadero technologies (www.embarcadero.com, 2013)) for developing web applications. *Web applications created by Intraweb technology can generate a text file on the client machine.* The paper shows how a web application for recording of sales (installed on the server) can generate cash receipt on a FP connected to the client machine. The approach is new, innovative and it is still not popular in Bulgarian business practices. Its implementation will lead to a significant multiplier effect. In order to clarify our proposal, the third technology model of the software system for recording sales is further developed.

Suppose that at a POS terminal sales web-based application is used. By pressing a button (or link) "Save sale and print a cash receipt", the sale shall be recorded in a sales database (DB) by sending an INSERT request to the DB) and a cash receipt is printed. Saving data in a DB through a web form is described in most textbooks on development of web applications. That is why we will not present the information process on generating SQL clauses and sending them to the database. More interesting is to show how to connect a web application with FPs.

When entering sales in a web form, they are recorded in a tabular form and they can be seen on the screen of a workstation. The data from the tabular part of the document (of the sale) can be saved in a special type of variable: a list of strings (*TStringList*). When using the method *SaveToFile* the list of strings is stored on the server, which is not a suitable option in our case. Here most developers give up their assignment or leave the software company, because they do not know how to save a list of strings (containing information about a sale) on the client machine.

If a desktop application (running in client-server mode) is used, the *SaveToFile* method writes on the client machine. It means that there is no problem: (1) a sale to be recorded in the server database and (2) a text file may be generated on the client machine and it can be passed to the middleware software. And it prints a cash receipt on the FP, which is connected to the client machine.

Quite differently information processes are carried out in web application. As noted, web apps are server applications. It means that they are executed on the server. There is not a problem with writing to a server database. Web applications for recording sales are server applications. In this case, data is recorded on the server. *A significant problem remains how to generate a text file on the client machine from a web application.* If the web application calls the method *SaveToFile* for an object of type *TStringList*, strings (contained in the object type *TStringList*) are recorded on the server. This approach is working very embarrassing because a web application can write to a text file on the server. The web app can be associated with only one cash register and not at a cash desk, but in a separate room where the server is located. This approach is extremely uncomfortable and it is not applied in business practice. Therefore the research has to be continued in order to find a solution of the marked problem how to generate a text file on the client machine from a web application installed on a server.

Most software products for recording sales are desktop applications (GUI). Software applications from the class of web applications built to record sales in Bulgaria are not connected to FPs so far. Therefore, to solve the problem posed in this work, we suggest a specific approach for connecting a web application with FPs. To realize the information link (between the web application (which is located on the server) and FPs, which are at workstations) the web application

for reporting sales has to generate a text file on the client machine. The technological solution (offered in this work) was developed in IDE Delphi using the technology Intraweb. A temporary file (with a unique name for the session) is generated on the server. Then the file is sent to the client machine. On closing the web application, all temporary files are deleted.

In the procedure that saves web form data in the database, the following variables are declared (Listing 1).

Listing 1. Declaration of variables.

```
aRow: Integer;
f: TextFile;
Line, File_Name : String;
```

The source code is as follows (Listing 2).

Listing 2: Source code in Delphi for generating a text file from the web application and sending it to the client machine.

```
// The name of the file is the session number + 'txt' extension.
// The file is saved on the server in a folder where the web
// application is started.
File_Name := ExtractFilePath( ParamStr( 0 ) ) + WebApplication.AppID +'.txt';
// The temporary file is deleted.
DeleteFile( File_Name );
// Assigning the pointer F to the name of the file, stored
// in the File_Name variable.
AssignFile( F, File_Name );
// The file is created and opened for appending data.
Rewrite( F );
// The sale is written in the text file on the server.
for aRow := 0 to ClientSideDataset.Data.Count − 1 do
  Begin // a cycle by rows of the table in the web form
    // a sale by department
    Line := 'E,1,_____,_,__;;%.2f;;1;1;%d;0;0;'
    // Writing the sum of tax group (department) in the sales row
    // Line := Format( Line, [ Sum, Departament_sale ] );
    // Writing one row in the text file
    WriteLn( F, Line );
  End; // for
// End of the fiscal receipt (cash receipt)
Line := 'T,1,_____,_,__;';
// Writing a marker for the end of the cash receipt in the text file
WriteLn( F, Line );
// Closing the file
CloseFile( F );
// Sending the file from the server to the workstation
  WebApplication.SendFile( File_Name, '', '' );
// Deleting the temporary file on the server
DeleteFile( File_Name );
```

4. Conclusion

This work proposes an advanced technology model of a software system for recording sales in retail outlets. The problem for connecting a web application with cash registers is solved. The proposed innovative approaches have significant multiplier effects because several costs are reduced energy costs, computer hardware and software costs. Preconditions for software enhancements of wide range of web applications are created. A new functionality of Web applications (to connect a web based system with fiscal printers) may be added.

Increasing the effectiveness of commercial enterprises depends not only on good managerial and logistics practice, but also on the costs for computer hardware and software. With the implementation of the proposed advanced technology model of companys information system a serious problem occurs how to connect a web app with cash registers. Because last literature sources do not provide a technological solution to the problem, the solution is given in this work. A "white spot" in the field of informatics is enlightened. Specific attention is given to the source software code in Pascal language (Delphi), describing how to connect a web application with cash registers.

As a result, the study concludes that many costs can be reduced. In particular it comes to energy, hardware and software costs. An argument proposal to streamline the existing process models of an information system in stores is made.

The most important factor in reducing the cost of adopting new technology models is the cost of providing information for commercial business processes. The proposed technology model and a software solution enable to make guidelines that can be perceived by businesses to streamline IT processes and reduce a number of expenses.

The result of the study showed that in times of crisis managers can find ways to reduce costs. Therefore, in parallel with the proposal to improve the technological model of an information system in the trading business a proposal for communication between a web application and cash registers is formulated and thoroughly described. The proposals for improvement do not cover all possible ways to reduce the cost of providing information for commercial processes. In the future it is necessary to explore new approaches to reduce costs and streamline IT processes. A new research should be made to look for other ways to reduce costs and other ways to improve the implementation of information process and in terms of the managerial process. These approaches (which we do have not covered in this work) may be the subject of a further systematic and thorough study.

The adaptation of the proposals in business practice makes the processes for maintenance of software systems simpler and easier. The ideas formulated and grounded in this paper may have broader continuity not only because of their innovativeness but mostly because of cost savings. As a guideline for future development of this work we may note the improvement of managerial processes in adaptation of new technological models of software systems in the trading business.

References

www.atozedsoftware.com (2013). *VCL for the web (Intraweb) documentation*.

www.daisy.bg (2013*a*). *Communication protocol for link between fiscal devices and computers*.

www.daisy.bg (2013*b*). Middle tier software for communication between sales software and a fiscal device.

www.datecs.bg (2013*a*). *Communication protocol for link between fiscal devices and computers*.

www.datecs.bg (2013*b*). Middle tier software for communication between sales software and a fiscal device.

www.embarcadero.com (2013). *Delphi XE3 documentation*.

www.orgtechnica.bg (2013). Middle tier software for communication between sales software and a fiscal device.

www.tremol.bg (2013*a*). *A Communication protocol for link between a fiscal device and a computer*.

www.tremol.bg (2013*b*). Middle tier software for communication between sales software and a fiscal device.