USING TABLETS IN DISTRIBUTED SIMULATION

Josef Brožek[a], **Martin Jakeš**[b], **Lumír Gago**[c]

[a]Faculty of Electrical Engineering and Informatics, University of Pardubice, Czech republic
[b]Faculty of Electrical Engineering and Informatics, University of Pardubice, Czech republic
[c]Faculty of Electrical Engineering and Informatics, University of Pardubice, Czech republic

[a]mail@jobro.cz, [b]jakesmar@gmail.com, [c]lumir.gago@student.upce.cz

ABSTRACT

The current trend in the sale of computer equipment is a dramatic growth in sales of tablets and smart phones and a gradual decline in the sale of personal computers (PC). Since this market change marks a change in the users' focus, it is desirable to reflect this change in the preferences and advantageously use it for simulation applications.

This article seeks in particular to:

- explain the advantages and disadvantages of the change in the user's preferences and its implications for the development of computer simulation,
- familiarize the reader with the basic advantages and disadvantages of tablets and their comparison with PCs,
- familiarize the user with the classes of tasks, for which it is possible to advantageously use tablets,
- familiarize the reader with the basic methodologies, architectures and techniques of simulation and distributed simulation for tablets,
- give the reader a concrete solution that utilizes the above-mentioned technologies and advantages.

Keywords: Simulation, tablet, smartphone, distributed simulation, heterogenic simulation, HLA.

1. MOTIVATION

1.1. Introduction to the Issue

When creating simulation systems of some classes, we need to take into account user preferences - this is especially true for simulation applications of the simulator (trainer) type. For this type of applications it is assumed that the user efficiently handles the control device which serves as input-output simulation device.

By generalizing this principle, we achieve a thesis which says that for an simulation control/operation to be effective, the user environment must be user friendly as much as possible. At present, however, in the field of user-friendliness, methods of software engineering and software design are ceasing to play the first fiddle, and so does the focus on the hardware device that is to serve for operation.

The current development in the information technology market, especially the one focused on the end user, shows that there is a dramatic increase in the sales of tablets at the expense of standard computers and laptops. At the same time, it should be noted that most modern mobile phones sold works on the same principle as tablets. Most of the information in the following text that implicitly assume that a tablet is used can be generalized and applied also to smart mobile phones (for example: prototype testing and verification of most of the principles mentioned in the text were, in addition to the tablets, tested at the same time on the mobile phone Samsung Galaxy S4.).

In general, however, we can draw a conclusion that the end users are increasingly becoming experienced users of devices controlled by the touchscreen. To illustrate the issue, Figure 1 shows the development of the market in each period.

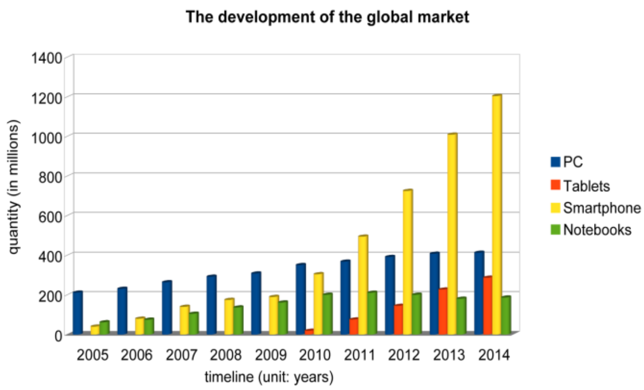

Figure 1: Development of the computer market

1.2. Comparison of Tablet PC Platform with IBM PC

If we want to seriously consider using tablets as primary runtime components for distributed simulation, we need to realize the fundamental differences and define the problems that may arise. For more information you can look to Schön (2013), Kovač (2012), Ku (2012), Mocný (2009) or The simulation interoperability standards organization (2001).

1.2.1. Performance Comparison

The first characteristic that we naturally focus to is the performance aspect of individual devices. It is certainly a legitimate requirement, but it is important to realize that only for a certain class of tasks. Tablet, PC or IBM PC is not a device that we choose for simulations for their brute force effectiveness, but for the user friendliness. Therefore, this article takes into account that the class of simulation tasks requiring enormous compute performance will run directly on a cluster, or a cloud.

A bit more specifically to the performance, performance of a standard PC processor is about 80 GFLOPS[1], in contrast to tablets, where the standard performance is about 80 MFLOPS. At first glance, the absolute value says that the performance of the tablet should be approximately 1000 times lower, compared to PC. At this point, it should be stressed that both architectures use completely different processors. A standard IMB PC uses the x86 architecture processors that may appear to be considerably ineffective (information on the very principle of CISC, internal RISC structure and the resulting information can be obtained from [1]). On the contrary, the ARM processors (for more information see [2]) used in Tablet PCs have better transfer of compute performance to instructions applicable in programmes.

The possibility of an illustrative comparison is presented to reader in the following figure.

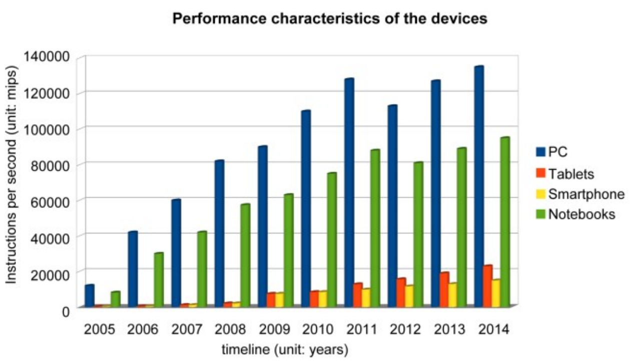

Figure 2: Various theoretical performances

The comparison clearly shows that tablets are not appropriate means for the class of tasks requiring brute force (on the other hand, below, the reader will be acquainted with the fact that even this problem is solvable with relatively easy approaches.)

1.2.2. Comparison with Regard to Peripherals

Although this directly begs comparison of the physical peripherals and input-output devices, motivation for comparing peripherals is a bit different. Therefore, let's move from the intuitive comparison of user input-output devices to the network communication technologies.

A standard PC network connection is carried out through a cable line, which has a (relatively) high level of reliability. In contrast, a standard tablet will be most often connected during the calculation through a wireless connection, which has a high quality only under ideal conditions - otherwise, there is an increase in latency, or loss of parts of the data. Therefore, the creation of models for tablets is more challenging (on the other hand, it is about as challenging as creating models that are synchronized in, for example, the internet, via web services, etc.).

Wireless communication (either Wi-Fi or Bluetooth, at worst, GSM) has its own merits, which is the portability of the device.

1.2.3. Physical Comparison

Since the comparison is very intuitive, just a few words about a strength that we can advantageously use.

Easy portability of tablets allows using them outdoor in support of ad-hoc decision-making (which is the topic of the practical application presented in this article). The risk is the limited battery operating time, but this problem is gradually eliminated. In addition, for certain classes of tasks do not require a long-lasting battery life.

The physical portability implies another major advantage and that is localization - because if we have an active Wi-Fi or GSM receiver, using certain principles (detailed in [3]) we are able to perform accurate localization of the device (this is possible even with GPS, but not in enclosed spaces).

A result of the portability and localization is, for example, that if we create online simulation of factory operation, the tablet will automatically visualize only the part of the plant near to our current location, i.e. the one where a malfunction is being dealt with, for example. Likewise, these principles can be used to collect data from relevant sensors (as the tablet can also serve as a central point for the collection of values collected with the Bluetooth technology and external sensors that support this technology).

1.3. Practical Programmer's Perspective

This chapter is rather intended for software engineers, it may not be interesting for other professionals engaged in the simulation. The chapter can be skipped without losing the context of the entire thesis of the article.

For the purpose of implementation, we need to take into account several important criteria, such as the demands of development (particularly in terms of the programming language support), the existence of libraries, support, and size of the community engaged in the development.

In order to discuss these issues, it is important to realize that there are (essentially) two main representatives of tablets, namely Tablet PC (with the most common operating system being Android,

[1] Flops is a measure of computer performance, indicating a count of basic floating-point operations per second.

followed by Windows) and the Apple platform – i.e. iPad and iPhone (running on iOS).

1.3.1. Development for Android

First, let us focus on (at least in Europe) the most widespread variant, devices that run on Android. The programming language that is used for programming is, de facto, the programming language Java; we just need to use different libraries. The development itself is relatively trivial. Part of external libraries, written in the Java programming language and designed for standard computers, are also applicable in the solutions for Android.

The community of general programmers is relatively broad and focuses mainly on the graphical environment issue, which cannot be considered a problem as most architectural problems is the same as in the standard JAVA.

However, a major disadvantage and a great risk in creating applications for Android is the issue of operating system versions. It is because it happens very often that the things that worked with ten different versions in a standard and predictable way will fail to run with an eleventh version. Therefore, with every new update of the operating system we should re-verify the functionality of the solution. The actual verification is performed though verifying the functionality of each module. It does not happen that the application would not work and return invalid results. The need for testing a new version is still a major inconvenience.

1.3.2. Development for iOS

Even for the actual development for iOS it is necessary to have the relevant physical device - emulation is very difficult, practically impossible.

Another major disadvantage is the programming language. It is an Apple company's own language, which is a certain mutation and an extension of the programming language C. For programmers accustomed to standard programming languages (Java, C#, C++), the actual syntax is relatively counterintuitive and we need some time to get accustomed to the language (as well as to the paradigms that the programming follows in that language).

The community is relatively large, but the collective problem solving takes place only to a limited extent. This is probably due to the fact that currently there is a transition to another programming language, so the community is slightly fragmented.

Another major problem is the non-existence of libraries that could be correctly linked to the program that will run on iOS devices, so it would be necessary to create the runtime environment for the simulation virtually from scratch.

1.3.3. Development for Windows

A brief mention should also be given to the development for tablets running on the Microsoft Windows operating system. The development is performed in a standard way that developers are used to, so it will not be discussed in the article, for more information you can go to [4].

A moderate risk of this solution is the fact that it is not that easy to disseminate applications or to network owing to the intricate settings and security policies in this operating system. Thus, even though the programming is simpler, we can often encounter various types of runtime problems caused by the settings.

1.3.4. Recommendation

The results of our work show that it is preferable for the simulation to use devices that run on Android OS or Windows. With the experience, we cannot recommend creating simulators for iOS.

2. WAYS TO USE TABLETS

The actual tablet can be used in simulations in several different ways, some of which are very intuitive, while some on the contrary deserve more attention (for example, due to their considerable development potential).

2.1. Used to Run the Monolithic Simulation

Currently, this method of use is purely a marginal issue. The performance is still insufficient and, even with calculations running on graphics accelerators, tablets experience memory issues. Therefore, the simulation thus triggered would be relatively undemanding and it would be rather a trivial demonstrator without much practical use.

On the other hand, it can be expected that with the increasing computer performance of tablets and modifications in the architecture of processors (number of kernels, higher parallelisms on graphics accelerators), even this approach will be broadly applied in the future.

2.2. Distributed Simulations solely on Tablets

Distributed simulations solely on tablets can be run in two different modes, namely:

2.2.1. Based on the Sharing of Performance

If we have a certain amount of tablet users who have access to the network and enabled a specific application, it is possible to run a computationally oriented simulation with a part of distributed computing running in the background of each tablet (even more logical processes in one tablet).

This solution is especially useful for calculations with a high degree of parallelism and it is an alternative to the solution in the cloud. The advantage, however, is advanced debugging options of the application.

2.2.2. Standard Distributed Simulation

In the case of this mode, there is just one logical process running on each tablet, in the foreground of the simulation. Most commonly the method is used in an application of the simulator/trainer type. Each tablet thus serves not only as a computational node for the simulator, but also as an input-output device, which can be used to parameterize the simulation calculation directly during its run.

2.3. Combined Operation of Distributed Simulations

It is the combined (according to some sources, hybrid) methods of the operation of distributed simulation that currently have the greatest potential for practical use. Specific usable solutions are listed below.

What all solutions have in common is that part of the distributed computation is performed on standard computers, servers, or even in the cloud (it is with the cloud solution that it achieves the best results). The combination of the computing performance of more robust computers with the advantages of tablets (portability, intuitiveness) is also used in the demonstrator presented in this paper in Chapter 3.

Only four basic methodologies are listed, which can be combined.

2.3.1. Used as a Visualizer

Simple use the tablet as a visualizer is beneficial for those classes of problems where it is desirable to animate the simulation calculation. This has practical applications rather for presentations of the simulation process to a customer and for validation of processes, where a simulation specialist consults realism of the established processes with an expert from the modelled field.

2.4. Used as a Driver

If we want to use the tablet as a distributed simulation driver, we just need to create such a simulator that will be able to parameterize the conditions prior to the start of the simulation, and then also the conditions for termination of the simulation run.

The solution is useful in situations where it is necessary to parameterize the simulation according to the information that can be obtained only in the field, i.e. beyond the standard stationary device running the simulation.

The ideal is to use this principle in decision support systems, where we can relatively easily parameterize the simulator, without being limited by our physical location.

2.4.1. Used as an Output Device

The second possibility is to use a tablet as an output device. Since it is not necessary to perform stable synchronization as in the case of online animation (2.3.1), this solution is relatively trivial. Like the previous approach, this solution is advantageous for decision support systems. Users are often not interested in the simulation, but only in the result.

2.4.2. Used as an Input Simulator

The most demanding method of using the tablet occurs when there is a request to implement the DIS methods (Distributed Interactive Simulation, IEEE1278, more in [8]).

It is especially challenging to secure such transmission problems in the network in order to make sure that all interactive interventions have been performed correctly and at the right time. Synchronization methodology is equally challenging. It should be noted, however, that despite some complications in the development we can achieve excellent results due to the high user-friendliness of such simulators.

2.5. Tablet Used as a Preprocessor for Data Processing

Being a portable computer that has a number of wireless connections (Bluetooth and Wi-Fi), the tablet can be advantageously used as a preprocessor for data processing. It is thus possible to obtain relatively cheaply a complex measuring station. If we have a number of sensors or limit sensors that support the Bluetooth technology, it is possible to retrieve data from these sensors through the tablet and process it into such a form that is required for distributed simulation. The actual distributed simulation is then connected to this node.

3. DEMONSTRATION SOLUTIONS

3.1. Description of the Problem

The chemical metal coating plant is characterized by several facts. The whole plant is automated (human labour is only at the beginning of the process to place the products to be coated on hooks and then during the storage process). The high degree of automation allows the factory to process a large number of products. The factory itself contains several lines for different types of chemical metal coating (bluing, electrolytic and galvanic lines), while the actual chemical bath are subject to change. It is also true that one chemical bath can be used for a number of metal coating lines (i.e. if the chemical bath has a size of 20 x 4 x 2 meters, it can be used for 4 metal coating lines and each line can process different types of products with the same surface finish).

The problem of the plant, however, is that due to the chemicals and enormous currents (thousands A at 3-5 V) there are frequent malfunctions on the lines. The chance that a fault occurs in one line, in any shift is up to 7 percent (up to 14 percent with lines fully loaded). If the factory has 40 lines, it is a relatively high number.

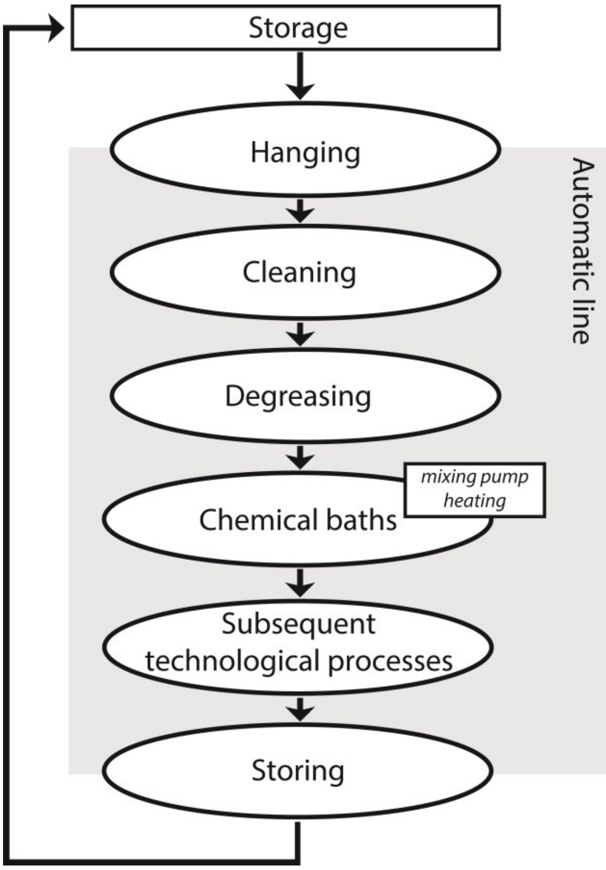

Figure 1: Chemical Plant Line Diagram

3.2. Motivation for Solving Simulations

If there is a malfunction, is it better to stop the operation and remove the error, is it advisable to have the production completed, or is it ideal to switch the production to another line? And what happens with the products? Will there not be unreasonable queues? Will there not be a plethora of rejects? Alternatively, will the products degrade that have gone though the preparatory process and are still waiting for a galvanic bath?

For such complex problems encountered analytical methods to their limits, and better way is to use simulation, as is noted by Manling (2009).

At present, all the problems that do not require acute repair are postponed until the production is completed and only then they are dealt with. However, the problem is that some of these errors reduce the efficiency of the process (type restrictions of the functionality of some pumps for mixing the baths) when it is necessary to slow the production by about 20 percent and there is an increased risk of poorly metallised parts.

By applying online simulation (i.e. simulation that uses actual operating data - statuses of the input queues, warehouse and individual lines - but from the start, the simulation runs with a time base much faster than the real time) it is possible to evaluate the best solution method immediately after detecting a problem.

It is because if we know the malfunction, due to the frequent repetition of identical malfunctions we know the time needed for the repair. Based on the parameters of the line (temperature, type, degree of metal coating, input queue, availability of other lines), the simulation can then be used to create an ideal scenario for the repair (so far, service engineers have usually not been able to evaluate all parameters only by expert judgment, as there are many variable parameters).

The simulation then can recommend redirecting the production to another line for the time need for repair (this solution minimizes the loss of unprocessed elements that have already undergone preparatory processes), suspending the line production and repairing (there will be an increase in the line capacity, if there are large input queues), or recommend continuing the production without any repair (with small input queues, or expected change in the production).

By including the simulation model to support decision making in the work of service engineers, we can increase efficiency of the total production under standard conditions by up to 2-5 percent (in excess of orders and with busy input warehouses and all lines, it is possible to increase efficiency and production throughput 7-10 percent).

3.3. Physical Simulation

The actual simulation is distributed and usually uses 4 types of nodes.

- The computing node is located on a powerful computer that is designed just for this single activity,
- The node for parameterizing, defining scenarios, launching the simulation and displaying the results is running on the tablet so that the operator can use any device in any production hall, without having to go back to the service centre once a malfunction has been detected
- The node for data collection is connected to the main automation computer and receives online data from the production,
- The node for obtaining historical data is placed in the company's data warehouse and is used to collect historical data on production processes, repairs, etc.

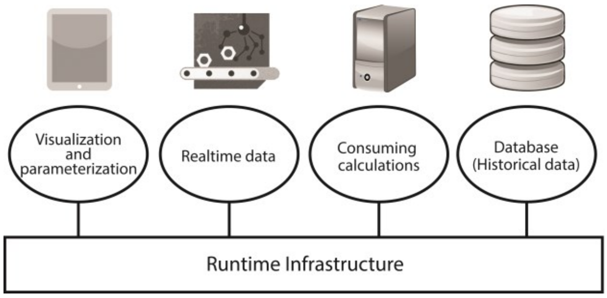

Figure 2: Architecture of Simulation Model

Individual nodes can run in the simulation in a larger number (e.g. two independent nodes for parameterization running in tablets are a natural part of the solution).

Anyway, the demonstration example is a great example of how a distributed simulation using tablets can effectively support decision making and thus improve the efficiency of industrial production.

3.4. Software Solution

Since it is a distributed solution that is to run on heterogeneous hardware platforms and it was assumed that it would be necessary to program various parts of the system in various programming languages (e.g.: the tablet requires the JAVA language, the data acquisition system in the automation computer requires the C programming language coding), the use of HLA seemed to be ideal as it has its internal processes adapted to all these contingencies.

For more information on HLA, its principles and potential usage, we recommend to consult Fujimoto (2000), Robelo, Sala-Diakanda and col. (2013), Kuhl, Dahmann, Weatherly (2000) or IEEE1516:2010 standard (2010*).*

The software architecture thus uses the central computer not just to run the simulation calculations, but also to run RTI.

It uses synchronization provided directly by HLA - specifically its conservative variant.

4. POTENTIAL OF FUTURE DEVELOPMENT

Currently, the prototype works only for a limited class of malfunctions – the article seeks to present an idea that the systems can deal with using tablets.

It is desirable to extend the system itself with the capability of automatic detection and conjecture of malfunctions. Ideally also with neural networks to deliver the learning ability so that the system is better able to work with various malfunctions, their diagnostics and solutions and able to create scenarios for malfunctions of a new type.

In general it may be said that further development is rather a low-level issue, and it does not change anything significant on the very ideas and principles of using combined simulations and tablets for decision support.

REFERENCES

Fujimoto, Richard M., c2000, *Parallel and distributed simulation systems*. New York; John Wiley & Sons. ISBN 04-711-8383-0.

Kovač, Pavel, 2012. Reflection: *Where are we going tablets?*. Available from: http://www.svethardware.cz/uvaha-kam-smeruji-tablety/34506-3 [accessed 15 July 2014].

Ku, Andrew, 2012. *Lenovo's ThinkPad X230T Tablet PC, Tested And Reviewed.* Available from: http://www.tomshardware.com/reviews/thinkpad-x230t-review-benchmark,3229-2.html accessed 15 July 2014].

Kuhl, Frederick, Dahmann, Judith, Weatherly, Richard, *Creating Computer Simulation Systems: An Introduction to the High Level Architecture*, c2000, Upper Saddle River, NJ; Prentice Hall PTR. ISBN 01-302-2511-8.

Manlig, František, 1999. *Computer simulation of discrete events.* Available from: http://www2.humusoft.cz/www/archived/pub/witness/9910/manlig.htm [accessed 15 July 2014].

Mocný, Ondřej, 2009. *Real-time physics simulation for mobile device.* Bachelor thesis. CHARLES UNIVERSITY.

Rabelo, Luis, Sala-Diakanda, Serge, Pastrana, John, Marin, Mario, Bhide, Sayli, Joledo, Oloruntomi, Bardina, Jorge, 2013. *Simulation Modeling of Space Missions Using the High Level Architecture.* Available from: http://www.hindawi.com/journals/mse/2013/967483/ [accessed 15 July 2014].

Schön, Otakar, 2013. *How powerful is your PC? New 3DMark test measures the ability of PC and tablets and phones.* Available from: http://tech.ihned.cz/hry/c1-59252020-3dmark-pro-pc-i-iphone-a-android [accessed 15 July 2014].

The institute of electrical and electronics engineers, Inc, 2010, *IEEE1516:2010: IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Framework and Rules*. New York; IEEE. ISBN 978-0-7381-6251-5.

The institute of electrical and electronics engineers, Inc, 2010, *IEEE1516:2010: IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Object Model Template (OMT) Specifications*. New York; IEEE. 2010.ISBN 978-0-7381-6249-2.

The institute of electrical and electronics engineers, Inc, 2010, *IEEE1516:2010: IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-Federate Interface Specification*. New York;IEEE. ISBN 978-0-7381-6247-8.

The simulation interoperability standards organization, Independent Throughput and Latency Benchmarking for the Evaluation of RTI Implementation*s*, 2001, *The Simulation Interoperability Standards Organization*. Fall. DOI: SISO-01F-SIW-033.