

# Bricolage approach for the software development in SMEs -Case study of simulator manufacture

Radoslav Stojic\*, Olga Timcenko\*\*

\* Faculty of Information Technologies, Belgrade, Serbia

\*\* Aalborg University, Copenhagen, Denmark

[Radoslav.Stojic@fit.edu.rs](mailto:Radoslav.Stojic@fit.edu.rs), [OT@create.aau.dk](mailto:OT@create.aau.dk)

**Abstract -.** Bricolage approach, based on the idea that new product should be result of innovative assembly and adaptation of available technologies and tools in order to make them work in practice, may effectively be applied in software design in SMEs.

This paper presents a longitudinal case study of using bricolage approach in a SME producing flight simulators. The focus is on parts of the software for simulation of airplane flight, including development, testing and validation. The software is built from few commercially available tools, some free components, and in house developed specialized and integrating modules.

## I. INTRODUCTION

Modern ICT has enabled the Small and Medium Enterprises' (SMEs) to become more integrated, more effective across longer distances, and to operate with more efficiency.

However, SMEs can experience difficulties in adopting Information Systems (IS) and aligning them with their strategic development [1]. Introducing standard IS and Enterprise Resource Planning may impose a rigid structure on a company, requiring that a SME behaves like a large company with long-term strategic planning and lots of in-house experts for IS – two requirements that are typically not fulfilled for a SME [2].

SMEs possess abilities to innovate fast, to respond rapidly to changing environments and to satisfy customers' emerging and evolving requirements, that can give them a competitive edge over larger organisations. However, these attributes are rarely exploited when SMEs implement IS.

A useful concept is bricolage, as it deals with the need for SMEs to learn about the possibilities of IS in situ, simultaneously exploiting the can-do approach that is usually found in SMEs [1,2]. The bricolage is based on the assembly and integration of various existing technologies and devices in order to make them work in a novel product. In addition, IS may be implemented piece by piece, stretching IS to support administrative and other functions that were outside of the application's intended scope.

This paper presents the results of several years of development IT solutions in one SME - French flight

simulator manufacturer [3]. Starting at late nineties in the field of hi-tech IT technology, and promoting low cost technology products on market, the enterprise succeed to install their Flight Simulator Training Devices (FSTD) worldwide. One of its FSTD is installed in JAT Flight Academy in Vrsac.

The role of adopted strategy, combining existing commercial and free software components with in-house IS development, was significant factor for the enterprise success. Full control of the customer-tailored parts of the software makes it possible to remain flexible and dynamic, and to conform to the need of the customers at any time.

This experience may be useful for other SMEs, at least those working in the fields of innovative technological solutions.

## II. BRICOLAGE APPROACH AT SME

Bricolage (the term is borrowed from French meaning "do it yourself") is used in several disciplines, to refer to the construction or creation of a novel product from a diverse range of things that happen to be available. By strict definition the bricoleurs 'universe of instruments' is closed and he will make use of whatever is at hand [1].

For example, in education bricolage is described as a way to learn and solve problems by trying, testing, and playing around, very much like a constructivist approach to learning [16], contrary to the analytical style of solving problems.

The concept of bricolage has been adopted, among others, by information systems developers [2]. When considering IS bricolage 'materials at hand' are usually considered to be information technology hardware and software artefacts.

There is an increasing race in the IT business environment which requires faster and more efficient software development methodologies. Thus, many alternatives to classical waterfall development process model are suggested [17,18]. For example, rapid application development (RAD) and eXtreme Programming (XP) have in some cases allowed small teams to develop systems quickly in rapidly evolving requirements. In addition, outsourcing software development and combination of bought, in house

developed and open sourced software, which often happens in SMEs' practice, make scenarios not well addressed by formal software development methodologies.

In fact, improvisation is frequently a chosen business model of SMEs. It enables them for a fast reaction on immediate pressure, as they can adapt and change to suit market conditions with no long term direction or strategic planning. Sometimes this flexibility is giving them a competitive edge over large companies.

Bricolage approach formalizes and systematizes this process of IT systems development.

### III. IT APPROACH AT SIMULATOR MANUFACTURER

Aeronautical industry is usually related to cutting edge technology products, large systems, large investments, very systematic approach and rigorous standards. The solutions are often exclusive and expensive. Flight simulation industry is not an exception.

In many branches of industry, flexible low-cost products, adaptable to current market demands are produced by SMEs. This might be expected in the flight industry, too, but that seldom happens. Namely, practice, standards, experience and development methodology in aeronautical industry leave little or no room for small firms with limited resources, unless they find their way to produce high-tech low-cost solutions.

Although "high-tech low-cost" seems to be an oxymoron, it may not be the case if a SME succeeds to find a way outside well known schemes and procedures. A conceptual approach of bricolage is not only ideological but just practical framework to find new innovative solutions.

ALSIM [3] is French simulator manufacturer started to build Flight Simulation Training Devices (FSTDs) for pilots' initial training [4] in late 90'ies. Independent firm with 25 to 30 employees including management, finance, sales and marketing, development and production, succeeded to develop its own products and its own methodology. What happened just corresponds to what is meant by the concept of bricolage.

The text to follow will highlight an aspect of the IT development usually considered as hi-tech in simulator technology. Its major components are:

- airplane flight modelling;
- collecting of representative data;
- implementing the flight model on FSTD;
- validation of the flight model;
- qualification of the flight model and FSTD.

It will be explained how the company under study has integrated all these components into a successful marketable product, combining off-the-shelf parts with in-house developed components, thus successfully implementing bricolage development process in practice.

### IV. SOFTWARE CONCEPT FOR FLIGHT MODEL DEVELOPMENT

The development concept in the company under study in this paper is centered around real time (RT) simulation software implemented on the training devices.

#### A. Real time IT support

Only few commercially available tools are used on PC Windows platform:

- Visual C++ 6.0 for real time programming;
- PC anywhere, for access of distant computer
- Adobe Illustrator, Corel Draw, 3D Studio Max for graphical content (assets) of synthetic instruments and indicators in the cockpit and visual system creating computer generated imagery of outside world;
- Microsoft Word and Excel for documentation.

This completes RT informatics support.

A distributed system architecture shown in Fig. 1 (autonomous processes interact with each other by message passing) is adopted for RT software on FSTD.

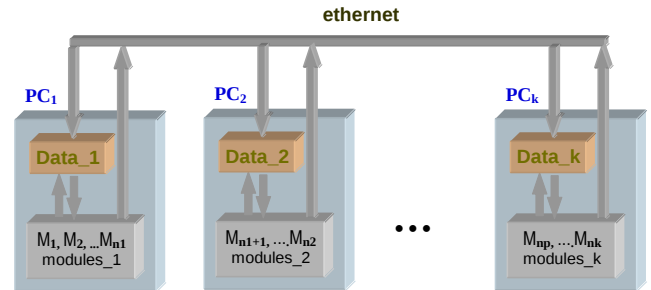


Fig 1 Distributed computing architecture

The processes are running either on the same physical computer or different computers.

Peer-to-peer architecture is exploited where there is no special machine or machines that provide a service or manage the network resources. Instead all responsibilities are uniformly divided among all machines, known as peers. Peers can serve both as clients and servers. This is in fact application of the concepts of DIS (Distributed Interactive Simulation) i HLA (High Level Architecture) [5,6,7] which offer high level of modularity and flexibility.

Thus, FSTD are built primarily from low-cost massive production components and common basic software tools are used.

#### B. Non-real time IT support

Software for Flight Model Development (SFMD – running in non-real time) is composed from in house developed modules or third party products. The latter is either free software (e.g. Octave – Matlab clone, or GCC compiler), or bought for a fee of the book with the software attached (e.g. Sidpac [8], Sdaer [9] etc.).

Each of these modules has been tested and validated prior to its integration in the SFMD package.

The integration of third party software was not simple due to variety of programming languages, input-output formats, and a mixture of source and executive code. Generally, none of these modules has been compatible with others.

The principle of distributed computing is exploited, because data-centric architecture enables computing without any form of direct inter-process communication.

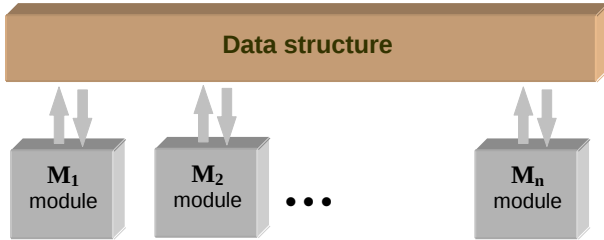


Fig 2 Data centric Architecture

Again, the ideas from HLA are exploited. All modules of SFMD are independent but it is assumed that communicate with rest of the system through communication interface (which corresponds to middleware in HLA terminology).

In howse developed modules are already built compatible, and third party modules are wrapped by an auxiliary software layer to obtain FSMD compatibility (see Fig 3).

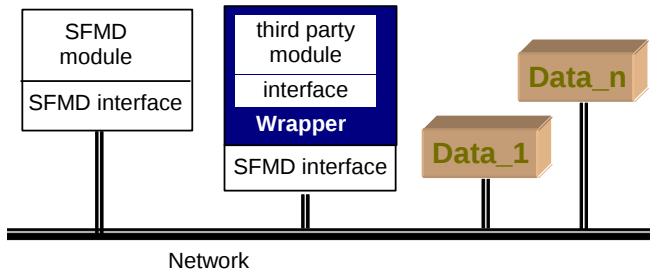


fig3 integrating third party modules into SFMD

The following sections give more details on particular tasks, methods and modules in SFMD.

## V. FLIGHT MODELLING AND SIMULATION MODULE

Flight model or mathematical model of airplane motion is commonly adopted system of differential equations representing rigid body with six degrees of freedom in quasistationary aerodynamic flow field [ 10-12]. Solution of this system may be represented in the form

$$Ch = f(p) \quad (1)$$

where **Ch** is set of airplane characteristics (fixed values as airplane maximal speed, ceiling, or functions like motion time histories e.g. flight altitude versus time), **p** is set of airplane parameters (fixed values as weight, wing area or functions like engine thrust vs. airspeed and height) and **f** is function .

Parameters **p** are input data to the flight simulation model while characteristics **Ch** are output data, i.e. results from the simulation model due to eq. (1).

The set of parameters **p** (known as data package) is sold by airplane manufacturer, since these data are normally predicted as a part of the airplane design. For some simulated airplanes there is no data package to buy, so simulator manufacturer has to build it. The practice at ALSIM was to build its own data for simulated airplanes.

Thus, the task to be solved by use of SFMD is: a) Build the set of representative parameters **p** for a simulated airplane, b) Obtain the set of airplane characteristics **Ch**, both of them validated and conform to the standards [4].

### A. Data collection

Primary sources for the airplane flight model definition are: certification data sheet, Airplane Flight Manual and Airplane Operating Manual as officially approved documents. Additional data may often be found in some of textbooks [8-14], and in various technical reports.

The next level of data collection concerns the similar airplanes (i.e. those having similar or same purpose, certification category, propulsion, weight and size). Using principle of aerodynamic similarity and model equations [8-14], this data may be used to predict parameters of the target airplane.

### B. Data consistency check and flight model parameter prediction

Data collected from various sources may be more or less reliable and should be checked for consistency.

For data consistency checking, the following simple statistic principle is often exploited. The values of the same parameter coming from various sources are compared. If some value is outside the pre-set level (which may be defined in terms of standard deviation  $\sigma$ ), its source may be eliminated from further considerations, or taken into account with the confidence level depending on deviation of parameter mean value.

Flight model parameters are predicted using standard techniques for airplane design, available in the textbooks [8-14] and technical reports. Often the books are accompanied by software ready to use.

The principal software packages exploited in the airplane flight modelling are the following:

- DATCOM. The U.S. Air Force Digital Datcom is a computer program that uses flight conditions and aircraft geometry to estimate the aerodynamic stability and control characteristics of aircraft. Digital Datcom follows the methods in the U.S. Air Force Stability and Control Datcom [13]
- Smetana codes is a software accompanying the book [9], permitting to estimate aerodynamic parameters of a light airplane – just corresponding

to the class of airplanes used in the initial pilot training and simulated on the low-cost FSTD.

- Lowry codes [14] represent Excel sheet implementing simple techniques for predicting light airplane performances. It is used in inverse fashion: parameters are tuned until predicted performances fit the real airplane performances.

## VI. FLIGHT MEASUREMENT MODULE

Demonstration of the simulator fidelity, i.e. that simulated airplane behaviour is sufficiently close to real airplane behaviour require flight measurement records done at specified test conditions by standards [4]. Flight records may be bought from airplane manufacturer (if available) or done in house. The latter is the option adopted by ALSIM.

The equipment developed in house, dedicated to the flight measurement for the airplane model development is based on the following principal low-cost components [15]:

- Laptop computer (Sony Vaio PCG-141C notebook)
- PCMCIA data acquisition card (National Instruments DAQ Card AI-16E-4) collecting up to eight analog input signals and eight digital IO signals.
- Attitude and Heading Reference System (Crossbow AHRS400CA-100)
- Video and voice recording camera
- Accessories, position and force sensors completes the system hardware.

The software is written in C++ (hardware communication routines) and Matlab.

## VII. FLIGHT MODEL VALIDATION AND TESTING MODULE

At a first stage the flight model is validated in non-real time context. It consists of validation of model equations which are common to all airplanes in a simulated class and validation of model parameters, specific to simulated airplane.

Model equations are implemented in software modules and they are validated prior to use in SFMD as well as during each flight model development. Namely, they may be considered as source of the of model parameters and data consistency check described in last section also attributes the confidence level to the module.

The simplified principle of the consistency check may be stated as: "If the results of two different modules are identical or sufficiently close, then both of them are correct. In opposite case they are both considered not correct and further validation is necessary".

In Fig 4. a simulation results are compared with flight records. Although match is relatively good, the deviations may be noted.

One may not a priori state that neither simulation results are correct (comprising equations, parameters and simulation algorithm) nor the flight records are correct (since they suffer from measurement and data processing errors).

Flight model parameters and equations are considered

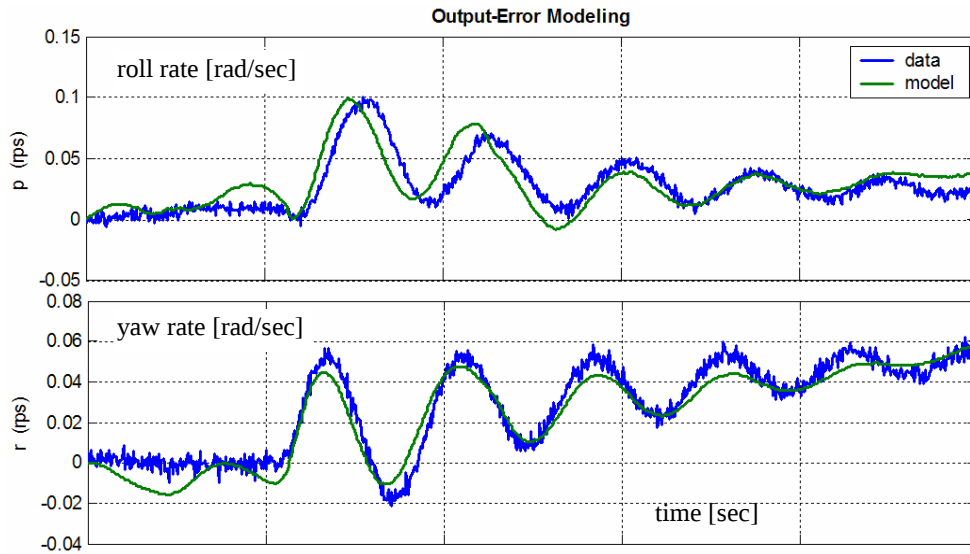


Fig 4. Comparison of simulation results with flight records

validated after achieving acceptable confidence levels.

By model based testing approach, a real time simulation model is validated using standard unit and integration test techniques and driver-stub arrangement.

Finally a series of automated tests is executed on the simulator and compared with the data collected during model development. The test results together validation data are delivered with device.

Each of FSTD in use is subject to periodic evaluation (called FSTD qualification according to standards [4]) by flight and technical inspectors which run the tests and compare with validation data.

## VIII. CONCLUSION

The standard IS development approach tailored primarily for policy and operation of large companies might not be

always the best solution for SMEs which usually operate with limited resources, but are typically more flexible and adaptable.

Introducing IS piece by piece from already available products, combined with in house development, may result in IS better adapted to SME company needs. This approach is formally called bricolage. Experience and IS solution described in this paper indicate that bricolage approach of combining technologies at hand may result in successful products developed by SMEs, although those products are usually considered to be hi-tech and exclusive for large companies..

#### REFERENCES

- [1 ] Ferneley E., Bell F., Using bricolage to integrate business and information technology innovation in SMEs, Technovation, No 26, pp. 232–241, Elsevier, 2006
- [2 ] Ciborra, C.,. The labyrinths of information: challenging the wisdom of systems. Oxford University Press, Oxford, 2002
- [3 ] ALSIM Simulateurs, France, [www.Alsim.com](http://www.Alsim.com)
- [4 ] JAR-FSTD A, Airplane Flight Simulation Training Devices, JAA-Joint Aviation Authorities, 2008
- [5 ] Jolivald B., La Simulation et ses Techniques, Presses Universitaires de France, Paris, 1995
- [6 ] R. Fujimoto, Parallel and Distributed Simulation Systems, Wiley Interscience, 2000.
- [7 ] F. Kuhl, R. Weatherly, J. Dahmann, Creating Computer Simulation Systems: An Introduction to the High Level Architecture for Simulation, Prentice Hall, 1999.
- [8 ] Klein V., Morelli E. “Aircraft System Identification: Theory and Practice”, AIAA education series, AIAA, USA 2006.
- [9 ] Smetana Frederick, Flight vehicle performance and aerodynamic control, AIAA Education Series, AIAA, 2001
- [10] Roskam J., Airplane Flight Dynamics and Automatic Flight Controls: Part I, Part II. DARcorporation, Kansas, USA, 2001
- [11] Roskam J., Lan E., C., T., Airplane Aerodynamics & Performance, DARcorporation, Kansas, USA, 1997
- [12] Roskam J., Airplane Design Part I to VII. DARcorporation, Kansas, USA, 1997 to 2001
- [13] Hoak, D. E., et al., USAF Stability and Control DATCOM. Flight Control Division, Air Force Flight Dynamics Laboratory, WPAFB, Ohio, 45433-00001, 1978. Revised
- [14] Lowry J. Performances of a light aircraft, Aiaa Education Series, AIAA, 1999.
- [15] R. Stojic, J. Binachon, C. Joslin AL-SIMA Système Intégré pour Mesures sur Avion, ALSIM Rapport Technique, AL RT 050, ALSIM Simulateurs, 2001
- [16] Papert, Seymour (1980): "Mindstorms: Children Computers and Powerful Ideas", 1980, New York, Basic Books
- [17] Steve McConnell, "Code Complete", Microsoft Press, 1993.
- [18] Kent Beck, "Extreme Programming Explained", Addison-Wesley, 1999