The Use of Artificial Intelligence Tools in Systems Development

By Patrick Bongo

July 2003

Table of Contents

Abstract

A detailed analysis on the use of Artificial Intelligence tools in systems development has been the focus of this report. We looked at a selection of the AI tools used in systems development and identified what their functions were in order to view the contribution of AI in this field. The report also goes on to pinpoint what the problems are with the use of such tools and proposes solutions that would aid to eradicate those problems. The areas of systems development that might need the support of AI tools to facilitate current working practices and solve a number of encountered problems, have also been identified. An insight in some of the current R&D in relation to this topic, has also been presented in the report.

1. Aims and Objectives

The prime aim and objective of this research project is to present a broad view of how artificial intelligence can assist in systems development. Thus the relationship is as such that the field of systems development like any other scientific field benefits from AI tools.

Another objective of this project is to promote the development of new tools that could help in achieving a systems development where it is believed that human intelligence has failed or still struggling to get it right.

Identifying the problems surrounding the use of such tools is also a key aim in this project, in order to enable AI developers to build the right tool in the correct format.

2. Introduction

The contributions of artificial intelligence in the field of systems engineering are object-oriented programming and the influence of AI programming environments such as InterLisp on environments for software engineering and CASE[1] tools (Sommerville, 2003).

Although one might think that systems developers do not need intelligent systems to develop and integrate information systems since they are known to have the expertise and somewhat mechanical tools to carry out their work, Golubev (2001) states that it is much more fruitful to combine the abilities of experts with intelligent tools.

[1] CASE: Computer Aided Software Engineering.

This report focuses on the use of AI tools in systems development by implicitly looking at the various AI tools that are currently used in this field, the problems surrounding their utilisation and the solutions to those problems. Areas of the systems development process where such tools are used, were also looked at. The current areas of the systems development process where similar tools might be required, were also addressed, as for instance there are still problems with getting customers requirements right, which would often lead developers in not delivering the right product (Sommerville, 2003). The current R&D in relation to this topic were also reviewed with the intention to observe current trends and contemplate future developments.

The AI tools that were identified are used primarily for verification, usability testing, automatic programming and systems maintenance. Hence we therefore looked at the systems development life cycle and determined the phases of a typical systems development (Kendall, 1995) and positioned those tools in the corresponding phases.

The problems identified with the use of AI tools in systems development particularly in the area of testing, was that when errors are given, it is often not clear as to the exact location in the line of codes where the error occurred (Mugridge and Tempero, 2001). The other problem identified was the fact that AI researchers are perceived to have a different view of the systems development process, which explains the difficulties they have in meeting the core requirements of the systems development process, as for example the tools they develop are geared at making activities such as programming easier, but they do not address process modelling techniques and project planning issues (Sommerville, 2003).

The solutions to the above problems were the reformulation of the software engineers'systems development process in AI terms and also the selection of a subset of software engineering problems in order to identify how they could be solved in an AI approach (Sommerville, 2003). Moreover, the development of a testing tool with automatic execution and easier for customers to understand was also advocated as one of the solutions to the technical problems faced by verification tools (Mugridge and Tempero, 2001).

However, the areas of systems development that were found to be in need of the support of AI tools were the requirements analysis and definition, process modelling and process support, and project planning (Sommerville, 2003). Furthermore, the automation of GUI[2] testing for non-web based applications was found to be a pertinent issue that could be best tackled with the use of an AI tool (Crispin *et al*, 2001).

The current R&D in relation to the AI tools used in systems development were found to be mainly the development of a light-weight knowledge sharing tool to assist developers in open-source software development (Cubranic *et al*, 2003) and advanced tools for debugging and testing (Microsoft Research, 2003).

We are also very much inclined to perceive the fact that the systems development process is more effective with the use of AI tools, as the latter takes away the pressure from developers, and assist them in completing the development process within a short time frame.

One could also foresee the use of rapid application development tools that make extensive use of automatic programming, as a valid contribution of AI tools in systems development.

3. Background

3.1 What is Artificial Intelligence?

McCarthy (2003) defines artificial intelligence as the science and engineering of making intelligent machines, especially intelligent computer programs.

As for Howe (1994) artificial intelligence is an experimental science whose goal is to understand the nature of intelligent thought and action.

It is therefore possible to agree that artificial intelligence is perceived as the scientific study of intelligent behaviour and the interpretation of such behaviour into a machine.

3.2 What is an Artificial Intelligence tool?

Golubev (2001) states that intelligent tools help us to survive and multiply our intellectual abilities. However, we could also perceive them as expert systems, since they hold the knowledge of an expert in the relevant field area that they have been designed to support.

[2] GUI: Graphical User Interface

3.3 What is an Information System?

IS (Information Systems) are generally characterized by large data structures which are modified or queried by several users concurrency (Frappier *et al*, 2003). It could also be defined in my view as any computerized repository of information.

4. Examples of Artificial Intelligence tools used in Systems Development

4.1 Spin

Spin is a tool for analyzing the logical consistency of distributed systems, specifically of data communication protocols (Gerth, 1997).

4.2 C++ Test

C++ Test is another tool that automatically performs white-box and regression testing (tests if code modifications introduce errors) for programs written in C and C++.

4.3 IBM's Rational Robot

IBM's Rational Robot is a functional testing tool that is used to create, modify and execute automated functional, distributed functional and regression tests for applications developed with MS Visual Studio.NET WinForms and WebForms. It also supports testing of languages and implementation environments such as HTML, DHTML, XML, Java, MS Visual Basic and Visual C++, and applications built using Oracle Developer/2000, Delphi, PeopleSoft, and Sybase PowerBuilder.

4.4 WinRunner

WinRunner is another functional testing tool that captures, verifies and replays user interactions automatically.

4.5 Distributed Analyzer

Distributed Analyzer is an example of an AI tool that is used to detect and diagnose problems in web based distributed applications (Edward, 2002).

4.6 SMV

SMV is a model checking system used to verify hardware and software systems against their design specification (Grumberg *et al*, 2003).

4.7 Bobby

Bobby is a tool that is used for usability testing, particularly in the area of web development as it analyses the site and gives an expert opinion on the state of usability and provides recommendations on the changes that ought to be made (Bobby, 2003).

4.8 Watchfire

Watchfire is another tool that is used for usability testing in web development and works in the same manner as the Bobby tool (Watchfire, 2003).

4.9 Shamash

According to Canacho *et al* (2001) "*Shamash is a tool for modelling and optimising business processes. The main features that differentiate it from most current related tools are its ability to define and use organisation standards, and functional structure, and make automate model simulation and optimization of them"*.

4.10 RADvolution Designer

RaDvolution Designer rapidly builds applications without coding and with a high level of maintainability and Layout engine. Works by just by dropping controls and layout panels (Programmer Haven, 2003).

5. Problems with the use of AI Tools in Systems Development

5.1 Technical

Problems faced with AI tools such as the Sat which is an acceptance test system for testing socket-based servers, is that when an error is given it is not easy to identify the place in the test where the problem occurred (Mugridge and Tempero, 2001).

5.2 AI Researchers

Sommerville (2003) indicates that the major handicap to the application of AI in software engineering is that AI researchers are themselves software engineers, hence they think they know the problems of software engineering but unfortunately the type of software that AI researchers build and the development process that they follow is atypical. Furthermore, Sommerville also goes on to state that their idea of priority problems may not be the same as systems engineering practitioners.

6. Solutions to those problems

The solution proposed for the Sat system, is to use a format that is both suitable for automatic execution and which a customer can easily understand (Mugridge and Tempero, 2001). Thus, it has been altered to use a *Fit*, which is a testing framework that uses HTML tables for defining tests and reporting errors (Mugridge and Tempero, 2001).

Sommerville (2003) proposes two approaches to solve those problems:

1. The **AI / SE** (Software Engineering) **approach**. Reformulate software engineering processes in AI terms and attempt to solve them entirely within AI.

According to Sommerville (2003), the traditional Systems Engineering Process applied by software engineers is made up of the three stages outlined in fig1.

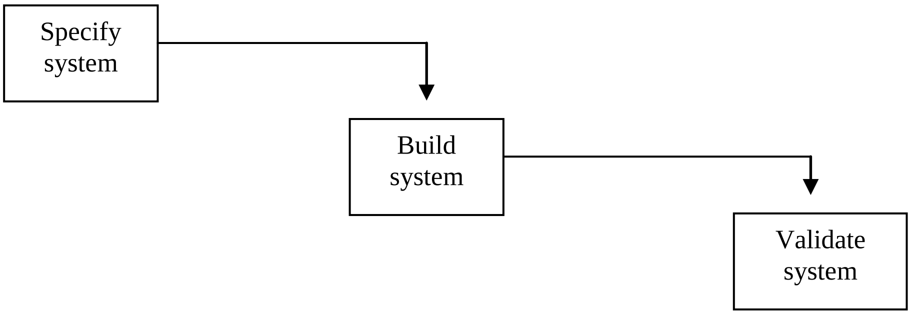

Figure 1: The Systems Engineering Process

Whereas the development process used by the AI community is an Explorative development outlined in fig 2.

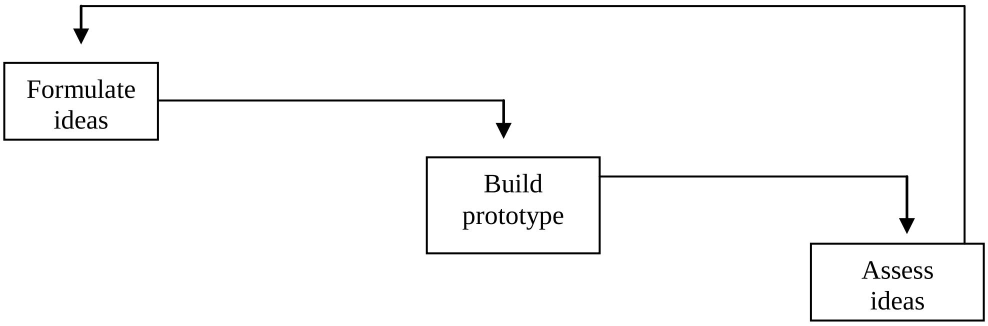

Figure 2: The Exploratory development

Hence, the AI/SE approach is aimed at restructuring the software engineering process outlined in figure 1 to the AI development process outlined in figure 2. But Sommerville predicts that this approach cannot be applied effectively as it does not recognize the importance of an agreed specification and validation against that specification. It is also possible to say that the exploratory development method in figure 2, does not promote the use of structured methodologies such as SSADM as it is too rapid at developing systems. Whereas the traditional systems engineering approach (fig 1) could be perceived to satisfy at least all methodologies including the DSDM[3] framework for RAD (Rapid Application Development) since it does not meet the core stages of the systems engineering process.

> 2. The **SE / AI approach**. Select a subset of software engineering problems and apply ideas, techniques and representations taken from AI in the solution of these problems.

As for the SE / AI approach, Rich and Walters (1992) cited by Sommerville (2003) suggest that tools to support software engineering must become knowledge intensive if they are to make an improved contribution to the software development process. Therefore, they believe that we have to look at other areas of systems engineering to see what knowledge-rich tools might be developed.

7. Areas of Systems Development where AI tools are used

The Systems Development Life Cycle provided by Kendall (1995) comprises the seven stages outlined in figure 3.

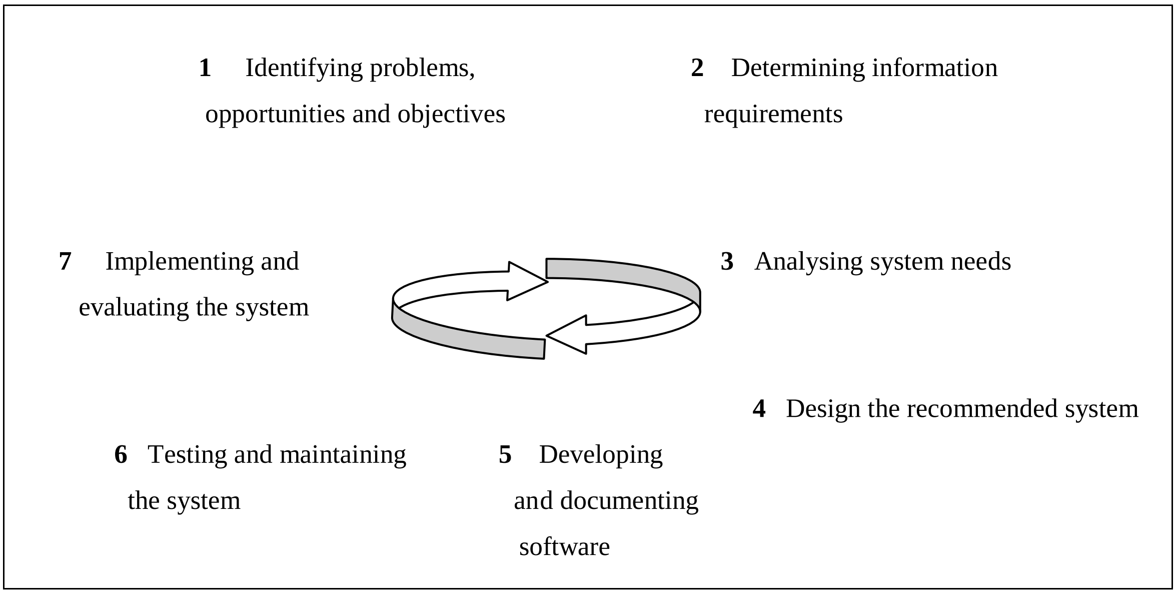
1 Identifying problems, opportunities and objectives
2 Determining information requirements
3 Analysing system needs
4 Design the recommended system
5 Developing and documenting software
6 Testing and maintaining the system
7 Implementing and evaluating the system

Figure 3: System Development Life Cycle as adapted from Kendall (1995).

Hence as referred from the features of the tools outlined in section 4 of this paper, we could easily place them in the phases below:

(5) Developing and Documenting Software (e.g. Shamash)
(6) Testing and Maintaining the System (e.g. Spin, C++ Test, WinRunner)
(7) Implementing and Evaluating the system (e.g. RADvolution Designer)

8. Areas of Systems Development that could be best supported by AI

Crispin *et al* (2001) state, "*We haven't found a cost-effective way to automate JavaScript testing (so, we just avoid using JavaScript). And we're also struggling with how to automate non-web GUI testing in an acceptable timeframe."* It is true to say that at present most of GUI testing tools such as Bobby and Watchfire are mainly aimed for web applications, as none of them seem to address the testing of GUI for non-web based applications. It is also true that current GUI testing for non-web based applications is done using various usability evaluation techniques such as user walkthroughs, field trials, questionnaires and observation.

With reference to Sommerville (2003) it is also suggested that AI could best support areas such as the Requirements analysis and definition, Process modelling and process support, and the Project planning. As it has always been difficult to get the requirements right. Moreover, Sommerville goes on to say that Process modelling tools have to provide informed suggestions on how to proceed in a process or how to adapt a process to a particular set of circumstances. For the Project planning, Sommerville perceives that existing management tools do not address the problems that project managers must face during the course of the project.

9. Current R&D in relation to the use of AI tools in Systems Development

R&D never stop, and at present Microsoft is one of the companies that is currently researching on ways to build and demonstrate advanced tools to improve the design, development, debugging and testing of software (Microsoft Research, 2003).

Cubranic *et al* (2003) is involved in the development of a light-weight knowledge sharing tool to assist developers in open-source software development. As current computer-mediated communication and coordination mechanisms such as mailing lists, Usenet

[3] DSDM: Dynamic Systems Development Method.

newsgroups, CVS source management repositories, and Bugzilla issue tracking systems do not offer much support to developers to learn and benefit from the past experiences of others working on the same project. Since it is expected that the knowledge portal should be able to gather information relevant to a task, organise it, search it and analyse it, synthesize solutions with respect to specific task goals, and then share and distribute what has been learned to other knowledge workers (Cubranic *et al*, 2003).

10. Discussion

Despite the fact that AI researchers are perceived to have a different systems development approach as referred to Sommerville's view, we must at least acknowledge the reality that the current AI tools have contributed greatly to the efficiency of the systems development process. But I believe that in order for AI researchers or developers to succeed in developing the tool in the manner that would be highly beneficial to systems developers and less criticized, they should work in partnership with systems developers from the beginning until the end of the development, rather than assuming that they have already enough knowledge of the problems experienced in systems development since they are themselves software engineers.

There is however the tendency to believe that the use of AI tools in systems development could supersede the human intelligence and intervention in certain circumstances of the development process where those elements are required to make decisions and exchange views with clients, but nobody can deny the fact that the support they provide allows systems developers to focus on ground rules of the development process (Sommerville, 2003).

It would be advisable for AI researchers to move away from the tradition of building automatic programming tools to develop core systems development process tools, that would provide a solid bench to achieve successful programming for the required system (Sommerville, 2003).

Despite the enormous variety of verification tools, it is wise to say that more research needs to be undertaken to develop GUI testing tool for non-web based applications, as this seems to be quite a growing concern particularly in the aspect of complying with user interface guidelines and standards which form a predominant part of human-computer interaction (Crispin *et al*, 2001).

The use of AI in supporting information systems evolution is also a feasible research issue. Although as referred to Cook (2003), when asked if an AI tool could be developed to assist in systems evolution, he replied by saying that information on systems evolution would normally come from people within the organisation and not from a tool, it could be possible to develop an AI module in every system, so that the module could predict the direction of evolution for that system.

11. Future Developments

It would also be worth conducting further research in order to find out the impact of Artificial Intelligence on human knowledge, as it is possible to say that humans could place more faith in the knowledge contained in the system, to such an extent that if the system commits an error, they might not be able to recognize it, since it is expected that its intelligence is supreme than the one of man kind.

Verification tools have proven to be quite effective in supporting the major parts of system testers tasks, but we are still a long way to automate the software evaluation process in accordance with user-interface guidelines and standards, thus perhaps this could be an area worth exploring.

12. Conclusion

If there is anything we could learn about the use of AI tools in systems development, certainly is its indispensability in the verification and validation process of the development. As we could not ourselves work out what the errors are in a line of codes. Nevertheless, an evaluation of such tools is also encouraged in order to cope with the problems faced in their utilisation.

We have limited ourselves in the field of systems development, but if other fields where AI tools are used were explored, we could have easily foreseen AI tools as saviours rather than supportive tools to particular organisational information processes.

In terms of the emphasis of AI researchers and developers in building more automatic programming tools for the field of systems development, we could argue that such tools facilitate the rapid delivery of the system to the client, but at the same time too, they do influence developers to miss out some important elements of client requirements that could only be captured and monitored if a well structured systems development approach that promotes all the phases of systems development is followed.

We all believe that prototyping is a good thing, thanks to recent AI inventions (automatic programming tools for RAD), but it has its downfalls in managing customer's expectations, since the latter are shown parts of a working system in different stages of the project for feedback, therefore, they anticipate that developers could do more sophistication to those components according to their appetite.

References

McCarthy, John. (2003). *What is Artificial Intelligence*. Stanford University. [Online]. Available at: http://www-formal.stanford.edu/edu/jmc/whatisai/whatisai.html
[Accessed 16 May 2003].

Howe, Jim. (1994). *The nature of Artificial Intelligence*. University of Edinburgh. [Online]. Available at: http://www.dai.ed.ac.uk/AI_at_Edinburgh_perspective.html
[Accessed 20 May 2003].

Frappier, M. *et al*. (2003). *Automatic Production of Information Systems*. [Online]. Available: http://www.dmi.usherb.ca/~frappier/SSSMFrappier602.pdf
[Accessed 20 June 2003].

Programmer Haven (2003). [Online]. Available at:
http://www.programmersheaven.com/zone28/cat1022/27396.htm
[Accessed 02 June 2003].

Golubev, Konstantin. (2001) *Is there any future for Artificial Intelligence*? International Society for Professional Innovation Management. [Online].
Available at: http//gkm-ekp.sourceforge.net/gkm-ai.htm
[Accessed 21 June 2003].

Gerth, Rob. (1997) *Concise Promela Reference*. Eindhoven University. [Online]. Available at: http://spinroot.com/spin/Man/Quick.html [Accessed 21 June 2003].

Para Soft C++ Test. COMPUTER, September 2000.

IBM Rational Robot. [Online]. Available at:
http://www.rational.com/products/robot/index.jsp?SMSESSION=NO [Accessed 21 June 2003].

Mercury Interactive WinRunner. [Online]. Available at:

http://www-svca.mercuryinteractive.com/products/winrunner/ [Accessed 23 June 2003].

Edward, Tim. (2002) *Why Distributed Analyzer?* VNU Research Library. [Online]. Available at: http://vnu.bitpipe.com/data/detail?id=1051711833_424&type=RES&X=838887211 [Accessed 23 June 2003].

Grumberg, Orna *et al*. (2003) *Learning to Order BDD Variables in Verification*. Journal of Artificial Intelligence Research. Vol 18.

Cadence Berkeley Laboratories. (2003). [Online]. Available at: http://www-cad.eecs.berkeley.edu/~kenmcmil/smv/ [Accessed 23 June 2003].

IBM Bobby. [Online]. Available at: http//bobby.watchfire.com/bobby/html/en/index.jsp [Accessed 24 June 2003].

Watchfire. [Online]. Available at: http://www.watchfire.com/ [Accessed 24 June 2003].

Camacho, David *et al.* (2001) *SHAMASH*: *An AI tool for Modelling and Optimizing Business Processes*. 13th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'O1). [Online]. Available at: http://www.computer.org/proceedings/ictai/1417/1417036abs.htm [Accessed 27 June 2003].

Mugridge, Rick and Tempero, Ewan. (2001) *Retrofitting an Acceptance Test Framework for Clarity*. University of Auckland. [Online]. Available at: http://www.agilealliance.com/ [Accessed 28 June 2003].

Sommerville, Ian. (2003) *Artificial Intelligence and Systems Development*. Lancaster University.

Kendall, Kenneth and Kendall, Julie. (1995) *Systems Analysis and Design*. Prentice Hall.

Crispin, Lisa *et al*. (2001) *The Need for Speed: Automating Acceptance Testing in an Extreme Programming Environment*. [Online]. Available at: http://www.agilealliance.com/ [Accessed 04 July 2003].

Microsoft Research. [Online]. Available at: http://research.microsoft.com/pprc/

[Accessed 10 July 2003].

Cubranic, Davor *et al*. (2003) *Tools for Light-Weight Knowledge Sharing in Open Source Software Development*. [Online]. Available at: http://www.sourceforge.org/
[Accessed 10 July 2003].

Cook, Stephen. (2003) *Making Sense of Evolution in Information Systems*. University of Reading.