Reasons for Success (and Failure) in the Development and Deployment of AI Systems

Nestor Rychtyckyj
Ford Motor Company
Advanced & Manufacturing Engineering Systems
Dearborn, MI

Alan Turski
Ford Motor Company
Ford Motor Credit Company
Dearborn, MI
(Retired)

Abstract

The AI community annually gathers at conferences such as the Innovative Applications of Artificial Intelligence (IAAI) to celebrate the achievements of those AI applications that were successfully deployed in a production environment. However, the applications that do not get deployed successfully are mostly ignored and not discussed publicly. There has been a large amount of research done from both the software engineering aspect as well as the project planning view to determine why the failure rate of software projects remains so high (Standish 2004). The use of innovative technologies, such as AI, complicates the software development process even further. At Ford Motor Company we have had our share of success and failure with AI applications – in this paper we will describe both types of projects and try to draw conclusions of what factors are most relevant in the successful deployment of AI projects. The paper is organized as follows. Section 1 will discuss the history and eventual failure of an AI system due to implementation issues. The next section will describe the organizational challenges that need to be addressed by an organization that wants to utilize AI technology. This is followed by a discussion of the technical challenges that are faced in the development and maintenance of AI applications. In the last section, we will summarize the lessons that we have learned over time and discuss how they can be applied to future AI system development.

Introduction

As stated in the overview for this workshop, it is very difficult to find papers in the literature that focus on failures and errors that were made in the development of AI systems. In most cases, authors have very little incentive to publish the results and there are not too many conferences or workshops that are looking for papers on failed applications. However, it was very refreshing to find a paper written by a non-AI person that describes the development, deployment and eventual failure of a system known as the "Vibration Advisor Expert System" (Pavlovcic 2005). The paper is written more for engineers than for AI practitioners, but it eloquently captures the essence and frustration of a motivated domain expert in the field of vehicular noise and vibration. The author goes on to describe the knowledge acquisition phase from the point of the human expert as being asked to describe 30 years of knowledge and expertise to a group of knowledge engineers. The expert system gets built and tested successfully with a 100% diagnosis accuracy and a 75% correct recommendation for repair. Everything seemed to be all set for a successful deployment when a corporate reorganization removed the executive champion from the developing organization. This, along with some serious technical deployment issues, prevents the system from ever being deployed to production. The frustration of the author is evident throughout the paper; a system that can bring huge success and benefit to a company is developed at great cost in time and labor, successfully tested and then never deployed, due mainly to the lack of "political horsepower". The author does point out some of those critical factors that can easily derail the best designed and developed systems: loss of management support and deployment issues, both technical and organizational.

From a system development perspective, the "Vibration Advisor Expert System" had several critical drawbacks that needed to be addressed before the system could be successfully deployed into a corporate environment. As was the case with many expert systems that were developed, there was very little or no in-house expertise with the development of AI systems. Typically, outside consultants were brought in to do the initial requirements definition, project scope, knowledge acquisition, system development and validation testing. The organization that would actually do the deployment was often not involved until the expert system was ready to be deployed. At this point, all of the issues with hardware and software standards, programming languages, coding standards, database and security access would come to the forefront. The consultants would not be particularly interested in changing their system to conform to corporate IT standards and the local IT organization did not have the expertise or knowledge to integrate a system written in some unfamiliar programming language into their corporate data center. These problems could be solved with sufficient management support, but the deployment effort and costs could be extremely high, especially if the system needed to be deployed at local user sites, which often did not have current hardware or software installed.

The second critical problem described in the Pavlovcic paper dealt with the loss of management support. One of the given conditions in any organization is the fact that the structure and people will be changed frequently. This must be addressed from the beginning and the "corporate selling" of the system must be an on-going task by itself. Other executives in both the development and deployment organizations must be kept aware of the project benefits and progress, so that the loss of one person will not stop the project in its tracks. This is no guarantee that corporate objectives cannot change, but the political issues surrounding a project need to be followed as closely as the technical issues.

In the example of the "Vibration Advisor Expert System" – the following steps could have been taken:

- The organization that was doing the deployment should have been brought into the development loop as soon as the project was approved, so that their input could be included in the development process.
- Executives other than the project champion should have been kept informed of the project status and proposed benefits.

Certainly, the project could still have been cancelled even if these steps were followed, but these actions could have helped in getting the system deployed. The paper concludes with a list of the proposed benefits that the system could have provided to the company and states that it is still available to be updated and deployed, if only a new project champion can be found.

The case of the "Vibration Advisor Expert System" is by no means unique to anybody that has spent time in the development and deployment of AI systems in industry. Many of the same factors that prevented this system from being deployed to production are present in many business settings and can also lead to failures to deliver AI systems. In this paper, we will elaborate on the critical success factors that need to be addressed in order to successfully deploy and maintain AI systems. Both the success factors and reasons for failure will be illustrated with examples from some of the applications that the authors have dealt with over the last 20 years.

Organizational Aspects of AI System Development

In many ways building an AI system is not much different that building any reasonably complicated computer system. Virtually all of the factors that need to be addressed in building a conventional system are also needed in developing an AI system. According to the Standish Group (Standish 2004) – the top ten reasons for the successful implementation of a system are the following:

1. User Involvement
2. Executive Management Support
3. Clear Business Objectives
4. Optimizing Scope
5. Agile Process
6. Project Manager Expertise
7. Financial Management
8. Skilled Resources
9. Formal Methodology
10. Standard Tools and Infrastructure

This list is directly applicable to the development of AI systems with several notable differences, mainly dealing with the last 3 points. The issue of skilled resources has and will continue to be a major stumbling block in the development of any applications that rely on innovative technologies that do not have a large number of skilled practitioners. The obvious result is that the resources that can develop an AI system are going to be scarce and expensive. Another related issue deals with the management and deployment of these types of resources. A company will have several choices if they want to build and deploy systems in the AI arena. They can do one of the following:

- Hire external consultants that are familiar and skilled with AI technology, but do not understand the business very well. These consultants will work on-site with the company employees and are expected to transfer their knowledge to the company employees and leave at a certain time.

- Develop in-house expertise for AI system development through the creation of a specialized group whose primary responsibility is to develop and deploy these systems where needed throughout the company.
- Develop expertise as needed in specific areas of the company, where specialists in AI may work on one particular type of business, such as manufacturing or financial applications.
- Work with external vendors that will develop a solution and deploy it at the company site. The vendor will be responsible for all aspects of the system development and maintenance.

All of these approaches have some advantages and disadvantages and have been tried with varying rates of success at both Ford Motor Company and other corporations. In many cases, one division in the company may not know that another part of the company may be working on a similar application, but using a different methodology and approach.

The use of external consultants is a very appealing strategy when a company is getting into a new area of technology and does not have much in-house expertise. This option can quickly get experienced people to "hit the ground running" without spending time on training employees or trying to hire new employees. This was the strategy that was used by Ford and others during the early years of the "expert system boom". Management wanted to develop and deploy AI systems quickly and hiring an AI consulting company, such as Inference Corporation, was the fastest way to get results. This resulted in the development of many AI systems at Ford including the Direct Labor Management System (O'Brien et al 1989), TIES (Vora et al, 1990), CAPE (Cunningham and Smart 1993) as well as others that were not published. The external consultants work with the company employees and are expected to train them and turn the system over at some point. This model was successful for a period of time, but was also very expensive in terms of the costs needed to keep specialized AI consultants working on site. The automotive industry is quite cyclical and consultants are usually one of the expenses that is reduced when the company goes through a cost-cutting exercise. This transition from outside consultants to company employees did not go smoothly in many cases. Budgets were cut to respond to external business pressures and consultants were frequently released before knowledge transfer was completed. The employees that did learn the details of AI development would often get moved to other assignments and the system maintenance would be neglected with the result usually being that the system would get scrapped. Another related issue was the fact that during this time period, AI system certainly did not run on standard hardware and use standard software.

Another common approach that was used at Ford and other companies including Dow Chemical (Kordon 2005) was the development of a centralized team that would work on AI systems throughout the corporation. This was a very successful approach at Dow where all of the available resources involved in the development of "soft computing" were consolidated in a specialized research group in 2000. This led to some extremely successful applications of soft computing technologies, such as Neural Networks, Support Vector Machines and Genetic Programming (Kotanchek, et al 2002). This approach has many advantages, including the fact that there is a "critical mass" of skilled people to do project work, the projects will benefit from the synergy and common standards and practices that are in place and there are fewer issues with projects being cancelled or delayed due to people being moved to another job. The problems with this approach once again lie in the area of how this project gets passed over from "development mode" to "maintenance mode". At some point, the organization that will be using this system will need to take responsibility for the maintenance and support and the centralized team will move along to other projects. A process must be developed and followed that specifies exactly how the project will get passed from the developers to the maintenance organization. The need for system enhancements, rewrites, ports to different architectures and platforms will require a very deep and thorough knowledge of the system and this cannot often be done without the continued support of the AI group. Our previous experience has shown that many projects failed or were abandoned when the host organization did not have the skills necessary to change the system and the AI development team was no longer available to work on this project. Issues completely unrelated to the AI systems, such as platform changes, operating system upgrades, database upgrades, changes in security standards and other common occurrences in a corporate IT setting will also require changes to the AI system. Another common problem happens due to the fact that much of the AI development software was produced and sold by very small companies.

These companies would often be bought out, go out of business or change their strategy; if you had developed critical applications that used this software – you were forced to scramble to modify or rewrite your system because the software would no longer be supported

A typical example of this occurred with Ford's DLMS system (O'Brien 1989, Rychtyckyj 1999).. The system was originally developed using the LISP programming language as well as the ART (Automated Reasoning Tool) from Inference Corporation. The system ran on a network of Texas Instruments Explorer LISP machines and it used screen emulation to connect to a mainframe system written in COBOL with an IMS database. Within a few short years, the TI Explorer platform stopped being supported and DLMS was ported to the HP Unix platform still using ART. Shortly afterwards, Inference stopped supporting the LISP-based ART platform and instead concentrated on the C-based ART-IM and later ART Enterprise product. At this point we needed to make a choice: DLMS either had to be rewritten from LISP/ART to either ART-IM with C and ArtScript or the ART had to be replaced with another expert system shell that was integrated with LISP. It was much less work to rewrite the rulebase from ART to KnowledgeWorks and keep the LISP programming language. During the same time, the mainframe COBOL/IMS system was replaced by a client server system that used the Oracle database. The AI communications was redone to use SQL to connect to the database. Since then, the AI system has been ported from HP-UNIX to a Linux platform. This kind of maintenance and deployment history is not unique to many AI systems that were deployed in those times. Systems that did not have strong support from their user community and management, as well as having people with the appropriate skills, would have been abandoned or replaced. That is exactly what happened to the other three Ford AI systems that were published in IAAI (Cunningham & Smart 1993), (Burney et al 1990), (Vora et al 1990).

AI systems, because of their frequent use of non-standard technology and unique skills, are extremely vulnerable do the type of organizational change that occurs often in corporate settings. Therefore, these systems must have specialized development and deployment methodologies and processes built into the organization. These processes should have the following components:

- A specialized group that is responsible for the development of AI systems and is available to be consulted and used throughout the system life cycle. This group needs a core of people that are employees of the company and who have a viable career path that rewards them for their technical excellence.
- Technical employees need to have a career path inside the company that gives them an opportunity to advance their career while become more proficient in their field. At Ford, this is done through the Technical Specialist program; it is imperative that employees don’t feel that they need to switch jobs often in order to advance their career.
- The use of consultants and contract employees to fill out the development group is beneficial, but it must be pointed out again that budgets for consultants and contract employees can be quickly reduced and the critical mass of knowledge must be kept in-house.
- The use of outside companies for the development of projects and systems for a company is an acceptable practice, but a process must be put in place to maintain or use these tools after they are developed. Management must be sensitive to the perception that they pay more attention to outside consultants than they do to their own employees. This perception is very damaging to employee morale and will often lead to very good employees leaving the company.
- The corporate culture must reward innovation and risk-taking, even if the idea or system does not work as intended. Advanced technology innovations, such as AI systems are risky and employees should not be afraid to get punished if they take risks and fail.

In many cases, the same group of developers will have success in one application and failure in a similar application. The technical skills of the developers did not degrade over time, but the underlying project dynamics were not in place. The probability of success increases markedly when the projects are not started and approved until they have both the management support and technical expertise committed. In the next section, we will discuss the technical challenges that need to be faced.

Technical Aspects of AI System Development

We need to preface this section with the following familiar axiom from the AI world: "It is very rare that the AI component of the system is responsible for the failure of the entire project". In most cases, AI developers spend much more time integrating their system, building the user interface, developing database access routines, documenting the system than do they do in building the AI into the system. The initial challenges lie in the development of the user requirements and in the selection of the architecture and software development tools. In many cases, this is where the success or failure of the project is decided.

Much of the architecture solutions and development platform selection is driven by the framework that is already in place. If a new system is to be deployed, it is much more likely to be successfully maintained if it fits into the corporate standards that are already in place. AI and expert systems can be written in almost any programming language; the main advantage to using specialized languages like LISP or Prolog or development tools like ART Enterprise or KnowledgeWorks is in the speed of the development process and the ease of maintenance when the system must be inevitably changed. The use of open source development tools like CLIPS or JESS may lead to issues with the IT policies at the corporation and will have lower levels of support or maintenance than purchased software. In most cases, people will be much more comfortable with familiar technology and tools and will usually first try to adapt this approach. These types of decisions can and should be made in the context of what the corporate standards support with an eye toward future maintenance and support. AI systems have been written in virtually in every conceivable programming language, using purchased software packages, open source packages or internally-developed packages. The critical aspects that need to be addressed deal with the ensuing support for that programming language and tool in the future.

Another issue that has become critical now is the lack of communization and interoperability between the various knowledge bases and ontologies that have been developed by various organizations over time throughout the corporation. This has resulted in a state where corporate knowledge is isolated in different islands without any hope of being used outside the area or system where it was initially developed. For instance, we have knowledge bases that have been developed using with rules using similar, but different syntax. As presently written, it is extremely difficult AI systems in Manufacturing to access the AI knowledge that was developed in Product Development. Fortunately, there is a solution that incorporates Semantic Network technology by using XML, RDF, OWL and RuleML to develop ontologies in a format that is easily accessible from other systems. This will lead to a much more synergy and increase the benefits of applying AI technology on an enterprise level.

Many business users already have a preconceived notion of the capabilities and limitations of AI technology. These expectations need to be properly understood and managed if the project is going to succeed. It is very easy to oversell the technology and underestimate the required work when approached by enthusiastic business customers that want a quick and easy solution. This was frequently a problem when outside consultants were brought in to "evaluate" the business problem and determine if their assistance was needed. Obviously, there was an enormous advantage to be gained by giving an excessively optimistic opinion on the project feasibility and cost with the underlying assumption that the customer would not walk away once the project was started and resources were committed. This temptation is also present for regular employees, who may be very eager to work on a new exciting application and do not completely explain to the customer what the potential risk may really be. This problem is magnified tremendously when the person making this initial assessment will not be doing the project work.

We have a lot of experience in managing user expectations as it relates to Machine Translation. This is a very well-understood technology and many customers already have a high expectation of what the results should be. Our development of Machine Translation systems for manufacturing ran across the whole range of customer expectations as we worked to complete this project (Rychtyckyj 2007). Our customer base was located in Germany, Belgium, Spain, Mexico and South America. We quickly found out that there were wide cultural differences between our customers in terms of what constituted an "acceptable translation". Our evaluations showed that the users in Germany expected a much higher degree of accuracy to accept a translation than users in other parts of the world. Some users expected that Machine Translation technology should work without any additional input on their part, which is not true, especially when the source text contains engineering and manufacturing terminology. Our biggest problem with the machine translation system was in getting good user feedback to improve the translation quality. In most cases, users were very quick to point out mistakes in the translations, but were very reluctant to spend time working with us to correct those mistakes. The degree of feedback also varies greatly with the organization and the location where the translations are being used.

The biggest challenge with the translation of technical documentation is with the terminology. The terminology that is used in our manufacturing applications contains a large amount of specialized part and tool description information that must be translated exactly in order to be useful. In most cases, a part description, such as "insulation assembly body pillar" must be translated as en entire phrase; therefore, these entire terms must be put into our translation glossaries. If the translation system cannot find the entire phrase in the glossary it will do a word-by-word translation, which is usually not acceptable. The development of these technical glossaries requires the use of bilingual engineers who understand the terminology in both languages. These types of people are usually very busy and cannot be spared to do translation work. We used various alternatives such as professional translation agencies, automotive bilingual dictionaries, on-going support from engineers that had some free time, but our best results were accomplished through the use of retired Ford engineers. We have also found that Machine Translation accuracy can be increased significantly by ensuring that the source text is written as grammatically as possible. Therefore, we developed a system that utilizes Natural Language Processing (NLP) techniques to improve the grammatical quality of the input text prior to translation. This “pre-translation cleanup” process has been automated and has been integrated into the translation process.

However, the maintenance of the Machine Translation systems remains an on-going issue. The technical terminology is constantly changing and the translation systems must also be kept up to date to prevent degradation in translation quality. Systems to enable translation into languages such as Turkish, Russian, Romanian and Chinese need to be developed to support our growth in those parts of the world. The Machine Translation systems are evolving from a strictly rule-based approach to incorporated statistical translation technology. All of these factors must be addressed as we try to integrate this technology into our corporate manufacturing systems.

Many users remain very enthusiastic about the system and have spent considerable time and effort to improve the accuracy of the translations. As word spread about the technology, we started getting a lot more interest and feedback from other groups within the company that wanted to get access to our system. The end result has been successful, but we recently went back and looked at our initial forecasts and timelines for the implementation of Machine Translation from 1998 when this project first started. We found that we never did achieve the levels of translation accuracy in the original time frame; luckily our management and customers were sufficiently satisfied with the progress of the project and gave us additional time to succeed. Our understanding of the work needed to deliver a successful translation system is much better now and we can build and deliver these applications according to schedule.

Conclusions

In this paper, we presented a summary of our experiences with the development and deployment of AI systems at Ford Motor Company. In many cases, "What Went Wrong and Why?" in AI system development is more of a case of not having the proper foundation in place before beginning the development process. Any organization that expects to have success in developing and using AI technology needs to develop the proper prerequisites in place before beginning this work. This involves putting a team together that understands both the technology and the business requirements. This team needs to have a "critical mass" of people that are well-versed in the technology to overcome the inevitable changes in personnel and assignments that take place in the course of business. There must be an effort made to integrate the receiving organization into the process as soon as possible to ensure the smooth transition of the project from development mode to maintenance mode. The choice of the development software and architecture is not nearly as important as it is to have the proper process in place to support the AI application throughout its lifecycle. As shown in the examples above, AI systems will be need to be modified frequently in order to keep current with the changes in system platforms, databases, operating systems, software upgrades and other changes. Both the development and maintenance teams need to collaborate closely to keep the system current and viable.

Another important success factor is to keep the customer involved throughout the development process. It is critical to manage this process carefully, both to keep the customer expectations in line with the deliverables, but also to see what other enhancements can be added to existing systems and to find new opportunities for applying AI technology. Satisfied customers provide the best opportunity to increase the use of AI technology throughout any organization and lead to more opportunities for making AI a more viable and visible problem solving methodology. By looking at "what went wrong with AI" we can take the necessary steps to make AI be the right solution to many critical business problems.

References

Bunney, W., Curson, S., Lemmer, J., Scollard, J., (1990), "Ford Motor Company's Expert System for Claims Authorization and Processing ESCAPE, *Proceedings of the 2nd Annual Conference on Innovative Applications of Artificial Intelligence*, AAAI Press, (presentation only).

Cunningham, A., Smart, R., (1993), "Computer Aided Parts Estimation", *Proceedings of the Fifth Innovative Applications of Artificial Intelligence Conference*, AAAI Press, pp. 14-25.

Kordon, A.K., (2005), "Application Issues of Industrial Soft Computing Systems", *Proceedings of the North American Fuzzy Information Processing Society (NAFIPS-2005) Annual Conference: Soft Computing for Real World Applications*, Ann Arbor, MI, June 22-25, 2005, pp. 110-115.

Kotanchek, M., Kordon, A., Smits, G., Castillo, R., Pell, R., Seasholtz, M., Chiang, L., Margi, P., Mercure, P., Kalos A., "Evolutionary Computing in Dow Chemical", *Proceedings of GECCO-2002, New York, vol. Evolutionary Computation in Industry*, pp. 101-110.

O'Brien, J., Brice, H., Hatfield, S., Johnson, W., Woodhead, R., (1989), "The Ford Motor Company Direct Labor Management System", *Innovative Applications of Artificial Intelligence*, ed. Herbert Schorr and Alain Rappaport, AAAI Press, pp. 333-346.

Pavlovcic, Frank, (2005), "Lessons Learned Through Working With General Motors", *SAE 2005 Noise and Vibration Conference and Exhibition*, Traverse City, MI, May 16-19, 2005, SAE Publication 2005-01-2551.

Rychtyckyj, N., (1999), “DLMS: Ten Years of AI for Vehicle Assembly Process Planning”, *AAAI-99/IAAI-99 Proceedings*, Orlando, FL, July 18-22, 1999, pp. 821-828, AAAI Press.

Rychtyckyj, N., (2007), "Machine Translation for Manufacturing: A Case Study at Ford Motor Company" *AI Magazine*, vol. 28, no. 3, (Fall 2007), pp. 31-44.

Standish Group Inc, (2004), "Chaos 2004 Survey Results", http://www.standishgroup.com

Vora, L., Veres, R., Jackson, P., Klahr, P., (1990), "TIES: An Engineering Design Methodology and System", *Proceedings of the 2nd Annual Conference on Innovative Applications of Artificial Intelligence*, AAAI Press, pp. 131-144.