Spring 3-27-2012

A Framework for Managing Information during the Design and Development of Complex Systems

Donna Champion
Loughborough University, d.champion@lboro.ac.uk

A Framework for Managing Information during the Design and Development of Complex Systems

Abstract

Many products developed today are becoming increasingly reliant on embedded software to facilitate customisation. Such products are becoming extraordinarily complex systems to design, update and maintain throughout the entire product life cycle. To support the design of complex systems there is a need for a complete re-conceptualisation of the traditional approaches to the management of information throughout the design and development process. This research is being undertaken within an action research framework and involves professionals from across the automotive sector, including participants from major OEMs; suppliers and support organisations and small start-ups. This paper sets out a new conceptual view for managing the information needed to support the design and development processes for complex systems. The framework has been evaluated through a number of workshops and is now informing work on a business transformation project in a large OEM for luxury cars.

Key words:

Information models Requirements Management Complex Systems Design

A Framework for Managing Information during the Design and Development of Complex Systems

1. INTRODUCTION

This research sets out a prototype framework for managing the information needed to support design and development processes for complex systems and products. The context for this research is the automotive sector, where for the past 40 years the car has changed from being a mechanical entity with simple electrical systems, to the point where it is now essentially a complex set of distributed inter-related software applications on wheels. Car manufacturers are now competing to become “the best software developer who also makes cars” (Lancaster, 2011). The increased complexity of the products, has created a step change in the level of difficulty in understanding the emerging properties of component systems and the potential interactions between systems at both a technical and human level (Bonjour and Micaelli, 2010; Ellims et al, 2011). For example, when creating a car, there is now a need to integrate many thousands of different software applications and to understand the emerging complex system properties from software and hardware integration on a scale of complexity not previously envisaged. The design and development environment for such complex products is becoming more difficult to manage and traditional approaches to managing the information around complex product design and development have reached the limits of efficacy. This research is part of a wide ranging research project being undertaken with a number of companies across the automotive sector to explore new approaches to managing the design and development processes for complex systems and products.

Working within an action research framework, a prototype framework has been developed to support the management of information (which includes requirements) associated with the design and build of complex systems. The framework is designed to support cross-functional communication and information transfer and also to offer a practical means of linking the necessary technical processes associated with engineering design to the supporting business processes. The creation of this framework has involved participants from across the automotive sector, including participants from a large luxury car manufacturer; supplier companies to the automotive industry; and IT analysts and green technology manufacturers who specialise in the automotive sector. The framework has just completed a first stage evaluation undertaken through a series of workshops again involving professionals from across the automotive sector and the work described here is currently informing a business transformation project underway in a luxury car manufacturer in the UK. Future research will follow the full practical implementation of the prototype and inevitable adaptation of the framework through practical use.

It is important to emphasise the scale and ambition of this project. In the IS field there is very little research undertaken on large projects due to the problems of access; academic time and also REF cycles where an academic is expected to have published a certain number of papers within a specific time frame. Large projects present unique problems and this research, in addition to investigating frameworks for information management in design and development, is giving insight into ways of managing the scale of information across large programmes, an area that is relatively unexplored in the IS field.

Section 2 of the paper sets out a brief review of the large literature in this field and summarises the problems and challenges associated with creating, analysing and tracking information for complex systems design and development, and also the difficulties of managing complex system design and development throughout a whole product lifecycle.

The third section sets out the new conceptual framework for managing the information and requirements, the fourth section of the paper discusses the evaluation of the framework and in the conclusions, the next phase of research currently underway is described.

2. LITERATURE REVIEW

2.1 Technical Challenges

There have been several useful conceptualisations of the design and development process, widely used across the engineering professions including the Waterfall Model (see Sommerville, and Sawyer, 1997); the Spiral Model (Boehm, 1986) and the NASA Vee Model (NASA, 2011). The waterfall model forms the underpinning of many of the traditional approaches to requirements management such as Jackson System Development (Jackson, 1982; 1995); Information Engineering (Martin, 1991); and of course SSADM (Structured Systems Analysis and Design Methodology) developed by the Central Computing and Telecommunications Agency (CCTA, 1990). The critiques of these approaches are well known in the academic community: the waterfall model and SSADM have both been criticised for failing to take sufficient account of the initial problem (Galliers, 1987; Mingers, 1995; Winograd and Flores, 1987). The Spiral model has been criticised as being just a "wound up waterfall model" (Graham, 1994) and the NASA Vee model has been criticised for only being "a waterfall model folded in half at the lowest level of decomposition" (Forsberg et al, 2005). All of the frameworks mentioned above have been designed to support the technical design process and can be criticised for not offering sufficient support for the integration and communication activities that are key to ensuring a theoretical design physically fits together in the real world once built. A further critique of all of these traditional approaches to systems development is that they do not offer any support for the complex process of integrating technologies that are being developed concurrently but in different teams and to different time scales.

Notwithstanding the known problems and well established criticisms of traditional approaches to systems design and development, the NASA Vee model is the basis for the International Standard for Systems Engineering: ISO 15288 and hence is widely applied in engineering firms as a basis for creating and building complex systems and products. Across the automotive industry, the NASA Vee model is the framework most widely recommended (and applied) for managing the engineering design and development process. It is also important to emphasise that in large-scale development projects, the Vee model facilitates a component-based view of the systems under development which does make such projects manageable [Boehm and Basili, 2001; Valverde et al, 2011]. Again component based approaches to developing systems have well-understood drawbacks in that designs are broken down into detailed work units with the idea that products will be simply and seamlessly integrated again, though in practice of course this is difficult to achieve [Beuche et al, 2007; Ellims et al, 2011). This component based approach to systems design often creates significant challenges when building complex systems as unforeseen and unexpected relationships between systems often only become apparent when a design enters the build phase. This applies both to software systems and to complex products that combine physical hardware and software systems.

In the automotive industry there are a multitude of examples of systems on the car (both virtual and physical) that have passed verification and validation testing in the design phase, but that prove to be difficult to fit together during the build and have unexpected conflicts with other systems. Issues identified at this stage of the development are notoriously difficult and expensive to fix. For example, unexpected integration issues between software controllers embedded in 'black box' components sourced globally can lead to expensive changes and reworking of designs further down the work stream (Valverde et al, 2011).

Miller et al (2010) have pointed out that another consequence of adopting a component-based approach in complex product development is that 'whole-product' analysis is often only performed on high-level feature descriptions, because analysis at this level is easier to perform. In reality, continuous analysis needs to be undertaken at every level throughout the information architecture and the impact for other layers in the hierarchy understood and communicated. But when examining the literature on systems development, it is soon apparent that the focus of academic research has been overwhelmingly addressed at the problems associated with the design phase activities and in particular with requirements management. This focus has been in part for good reason, in that one of the most important and difficult processes to manage in complex systems development is the creation; analysis; prioritization and tracing of requirements and their associated test cases and methods through the design phase. Many small-scale tools have been developed to address specific local problems, but there has been little research undertaken on how to manage complex systems design and development across the whole product lifecycle, or on how to manage the vast scale of information that is associated with complex products.

There are a number of commercial tools available to help companies manage their requirements such as the IBM Rational suite that has developed from Telelogic's DOORS (Telelogic, 2011) or Borland CaliberRM (Borland, 2011). These tools focus on managing the processes associated with traditional ‘requirements management’ and can be visualised as focusing on the activities undertaken on the left -hand side of the Vee-Model of software development (NASA, 2011). The IBM and Borland tools are very widely used in OEMs (Original Equipment Manufacturers) and these tools link into other software tools that manage other parts of the process in complex systems development, such as tools for project management and gateway assessments. However an integrated, holistic process to managing the whole design and development process for complex systems with different rates of development across multiple product lines is not currently available.

There are further challenges to address. For example, when systems are being developed through traditional methods and are integrated with other systems following a more iterative, agile path of development, creating synergy between the two different sets of information in order to understand the impact of design decisions in one system on another can be very challenging. Today, where sensors and control units support multiple systems, it is only once a design is understood from a system (or functional) perspective that customer features can be safely linked to hardware and software components. This means that there is a need to create tools that can integrate different technologies at different stages of maturity; that are sufficiently flexible to cope with negotiation across system boundaries and that offer a means of managing business risk.

In addition, the inevitable changing of requirements (and associated information) must be managed, while incorporating mechanisms to allow feedback from manufacturing, service and disposal activities. There is also a tension between the need for flexibility and the need for strict discipline to achieve traceability. In the automotive sector there is also a need to work towards meeting the new functional safety standard for the automotive industry ISO 26262 (ISO 26262, 2011). Processes must be sufficiently rigorous to demonstrate traceability and to create a meticulous documentation trail (Boehm, 1986), whilst also permitting space for innovation and creativity. It is not surprising that such an approach has been elusive in practice.

2.2 Organizational Challenges

In addition to managing the information associated with product creation for complex systems, it is important to manage the associated organizational and resource issues. PRojects In Controlled Environments (PRINCE, 2011) is a widely used approach to managing projects that follow traditional approaches to systems design. Dynamic Systems Development Methodology (DSDM, 2011) has also been developed to support agile development processes, though it is only relatively recently that agile practices have been implemented in large-scale environments such as manufacturing organizations. The challenges associated with scaling up agile approaches have become better understood recently (Beuche et al 2007), but there are still frictions that occur in practice. For example, some sub-systems of complex products need to be developed with a formal methods approach in order to comply with safety requirements, but integrating the design and development cycles for these sub-systems designed according to formal and traditional approaches with other systems developed through an agile approach is difficult (Gil and Tether, 2011). Creating an accompanying document trail is even more challenging. However much of the literature suggests that if a holistic, systemic, more agile approach to managing processes could be implemented across the whole product lifecycle, some major business objectives would be achieved. For example, by allowing engineers more freedom to undertake design and testing within company processes the temptation to do off-process, ad hoc design would be reduced; the company would achieve a better documentation trail and also have more leverage to improve practice (Champion et al, 2005).

When considering the academic literature, one gap that becomes apparent is the lack of guidance for managing communication issues and working practices during complex systems design and development. The only communication issues that are specifically identified in the literature have been those associated with prioritization issues and also release activities and most authors in the Engineering literature recommend a 'single-capture process' to deal with this issue (see Boehm, 1991; Charette, 1989; Egbert and Neve, 2001; Keil et al, 1999). In practice, in large manufacturing environments with simultaneous design and development of several product lines, a 'single-capture process' approach to requirements is infeasible. One of the impractical assumptions that is associated with adopting a single-capture requirements process is that it is considered a relatively easy task to assign 'content ownership' of any requirement (Gil and Tether, 2011). Identifying content owners for each requirement is, of course, essential, but it is equally important to identify those responsible for communication activities; cross-stream collaboration and feedback mechanisms. Such activities may, or may not be undertaken by the 'content owner'. For example, some sensors and electronic control units manage several different systems and so changes in one system can have unpredictable emergent properties across several sub-systems. It is over-simplistic to just decide that the 'sensor owner' has responsibility in such situations as this person may lack the experience or knowledge necessary to specify all the necessary potential faults. Such activities need to be managed by groups of engineers with sufficient qualifications and experience across a range of disciplines and with experience of the systems potentially impacted by change. Any signing off of such multi-system requirements then needs to be managed by a more senior, suitably qualified and experienced person, often as a collaborative exercise. This hierarchy of management and control in the technical process is rarely considered in the academic literature as it adds an unwelcome layer of complexity.

It is also important to state that project management activities do not address the sorts of technical issues described above. Project managers often have little engineering design experience, and with the increase in complexity in systems and products, such decision points are becoming increasingly frequent. There are of course many 'gateway and evaluation' processes to oversee and manage the design activities across programmes, but these activities are not set up to deal with technical problems in the level of detail and frequency that is becoming required in order to develop safe systems. This increased detail and frequency is as a direct result of increased complexity of systems in a vehicle.

In practice, when creating a framework capable of addressing the multi-layers of resource and organizational management; the underpinning information model; overarching project management processes and strategic direction all need to be considered. This holistic consideration is essential in order to gain some understanding of the otherwise unexpected relationships and unanticipated emergent properties that can occur later in the development cycle.

In the Information Systems literature a number of frameworks and design and development approaches have been offered to address some of the problems associated with the technology focused approaches discussed above. (See Avison and Woodharper, 1990; Checkland, 1981; Checkland and Holwell, 1998; Champion et al, 2005; Stowell, 1991). However, all of these approaches have been developed to address systems design for relatively small-scale development projects and can be criticised for not offering sufficient support for managing the organisational issues associated with the more technical design activities. Recent research in the automotive sector suggests that current 'soft-systems' approaches do not offer comprehensive and practical support for an engineer in addressing the scale of information that design for such complex systems entails on a daily basis (Champion et al, 2012).

The challenge of managing design and development work for complex systems across an organization is significant. This applies both to small start-up companies that are developing new technologies such as new motors or batteries and to large OEMs. From the perspective of a large OEM, creating a framework that can support the development of several concurrent product lines, each with differing development timescales, is a serious challenge. It is also important to integrate financial and project management systems and facilitate joint development work and cross-stream communication with both internal company stake holders and external participants such as suppliers and component manufacturers. Considering process alone is not sufficient. There is a need to consider the core competencies and skills available to a company if implementation of any new procedures is to be successful. Managing the scale of information associated with such tasks is an area that remains largely ignored in the IS literature. There is a need to understand the relationships between diverse information sets, across thousands of embedded systems and to be able to understand and predict the impacts of change when handling millions of pieces of information.

2.3 Traceability and Documentation

A key business aim of companies involved in complex systems design and development is to be able trace quickly and accurately all the requirements, information and change management processes associated with a particular programme. To achieve traceability it is necessary to provide links and documents trails between all of the different requirements associated with a design. This may require information to be linked across many different software applications. For example, a requirements management system will hold the information on the programme; project and technology development that is of interest to senior managers, but the information associated with the design work will often be created and managed using other applications. For example, model based safety case analysis or simulation of braking systems needs to be undertaken using specialist software applications, but a record of which tests have been completed and the results of the tests, has to be recorded within a central location and made available to engineers working on other inter-related systems. In order to achieve full traceability of requirements and changes, it is also necessary to specify all the associated roles and responsibilities to ensure this essential task is completed. In practice, it can be unclear who ought to be responsible for a particular system or sub-system during the design phase. For example, where a system has been designed by one group of engineers, but sub-systems such as sensors and control units have been designed by a different team there can be ambiguity or even disagreement as to who was responsible for a particular system during design and development. In addition, the focus on cost reduction, meeting financial targets and a pressure to deliver, all means that ‘owning’ some complex systems is a difficult role, and not one that every employee is willing to undertake (Champion et al, 2012).

These issues underline the need for a standardised information structure to underpin product development. There are many different levels of complexity in information required across different areas of the technical process and the management process. In addition, there are different levels of 'information need' across different groups of engineers. For example, the hardware development teams, who tend to work with geometric CAD systems, create and manage a much smaller information resource during product design and development than the electronic and software teams who need to integrate various sub-systems across the whole vehicle. Industry best practice is also currently being updated with the publication of ISO 26262 (2011), the new Functional Safety standard for the Automotive industry, which requires a much more iterative approach to managing information and requirements, so that manufacturing and service functions are fully integrated into the main design and development processes; the standard also requires that disposal issues are considered during design work, creating the need for total lifecycle information management for a vehicle.

The integration of new complex technologies into the already highly complex 'system of systems' that comprises a vehicle is managed through software. Some software applications in a vehicle manage functionality across many disparate systems and the implications of software change, particularly late into a programme, can be underestimated by project managers with no software development experience and by engineers involved only in mechanical design work. Additionally, if testing is undertaken using a specific 'black box' component in a sub-system, but then this component is changed during the build phase by a purchasing department, as they believe they can source 'the same component' for a few pence cheaper, problems can arise as unanticipated behaviour manifests in the vehicle which only becomes apparent very near to the expected delivery date. These late stage errors are expensive to correct. Tracing the cause of any errors requires an approach to design where the interrelationships of all the information and data sets are mapped and understood. Currently there is no framework that can facilitate the integrated management of information across the design and development processes for complex systems, and certainly managing information across the whole product life cycle and supply chain, necessary for building modern vehicles seems very distant.

3. A FRAMEWORK FOR MANAGING INFORMATION IN COMPLEX SYSTEMS DESIGN AND DEVELOPMENT.

3.1. Background to the Information Model

In a traditional model of systems development for the automotive industry, at the start of a new programme of product development, thousands of requirements are cascaded in a single event across the engineering teams for them to work through. This constitutes a large and significant data set that has to be managed and co-ordinated. For an information model to be capable of fully supporting the design and development of complex systems, including the development of embedded software systems across the product (in this case a vehicle) it is essential to change the way that information and requirements are structured and managed across the product lifecycle. The aim is to create a configurable information architecture that will support:

i. the complete lifecycle of the product, including diagnostic tools, service procedures and disposal;

ii. the appropriate use of information at different stages of maturity;

iii. the management of business risk.

A Framework for Managing Information during the Design and Development of Complex Systems

When considering the necessary information support for design and development activities there are broadly two main categories of information. The first category is related to various standards for design and testing. The information and requirements based around the application of standards and testing protocols is often well-structured and requires less frequent updating than design information, this sort of information is frequently referred to in the field as ‘non-functional’ requirements. The second category of information relates to the dynamic design work and associated test results and this information is subject to discussion, negotiation and frequent change (this information describes systems behaviour and if often described as ‘functional’).

For this second category of information and particularly in integration activities it is essential to manage both the information and the risks. Cycles for software design, hardware design, for integration issues and for the design of new technologies and complex systems all have different risks and timing issues to consider. It is necessary to also understand which activities in the process are subject to the most uncertainty and so are the most risky and also to build in a prioritization process for managing design information and requirements based on the assessed risk. If risk is not managed through the process then the senior managers cannot monitor or assess progress from concept through design and build activities. For example, many engineers faced with a mass of information downloaded from a single cascade event, will focus on the ‘easy’, familiar activities and ‘carry over’ requirements first. If software is being developed, then it is possible for a system or product to appear in a requirements management system as 90% complete, but in fact, still have significant and potentially programme disrupting problems to overcome, which is not registered. In addition, in a commercial world, the pressure to include new technologies onto programmes quite late into the development cycle, to maintain competitive advantage, is intense. Incorporating business risk into the information model is important to ensure if a new or 'immature' technology is incorporated late into a programme, that senior managers can use this information to flag up additional issues and questions at gateway evaluation meetings and at key decision points.

The framework for managing information described here is based in part on the idea of the Universal Description Discovery and Integration protocol (UDDI), originally proposed as a web-standard (Sabbouh et al, 2001) that describes the protocols and message formats required to facilitate connectivity across diverse web services. The idea of universal descriptors to manage services (UDDI) has been adopted by OASIS (2004) and although the UDDI standard is applied to web services, the underpinning idea of a set of standards being associated with key pieces of information, that are applied in order to define relationships and information exchanges across a network, is directly relevant to the problem of facilitating connectivity across the software systems in a complex 'system of systems' (in this example, a vehicle). In the information model presented here, this concept of a universal descriptor has been coupled with the idea of a 'knowledge asset'.

The idea of a knowledge asset is that the asset collects together all of the information and history associated with particular aspects of a complex system, a wider concept than the traditional idea of a requirement. In the information framework suggested here, a Knowledge Asset (KA) is made up of a number of sections, which encapsulate the ideal order of generating information, ensuring the appropriate questions for information and requirements creation are asked in the right order. The example in Figure 1 shows a conceptual knowledge asset for application in the automotive industry (where vehicle attributes summarise system and sub-system performance):

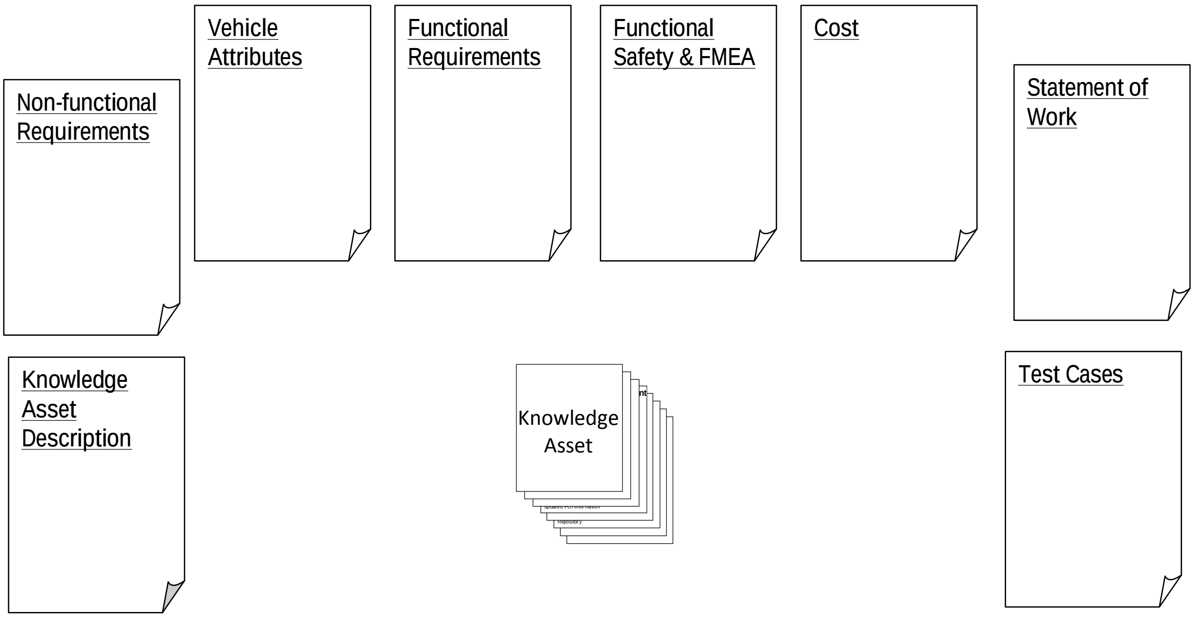

Figure 1: A Knowledge Asset for Complex Systems Design and Development: An Automotive Example

Knowledge Assets can support the definition of a complete design of a complex system from concept to physical components and are designed to facilitate the examination of a design from the associated information alone, prior to expending effort on a physical build and implementation. Such an approach to information management will facilitate the re-use of knowledge across programmes, help to improve quality and also facilitate the management of business risk associated with information that is not fully mature.

Knowledge Assets are designed to act as a shared resource of the organisation, created, modified and subscribed to through their deployment. One problem that recurred across design and development activities was when engineering teams created boundaries for their team and kept information within that boundary meaning other teams took decisions without an understanding of possible impacts or conflicts. To prevent this it is suggested that ownership of Knowledge Assets should not be assigned to specific groups but be a whole organisation resource. The creation and modification of Knowledge Assets will still need to be moderated. The idea is that over time a 'Library' of Knowledge Assets can be created to be available for re-use. By combining the idea of a 'Knowledge Asset' and of a 'universal descriptor' setting out protocols, services and messages it is possible to reconceptualise the information model for complex systems design and development.

There is a further level of complexity that must be managed for design and development of complex systems. When creating complex products there will possibly be many new technologies being developed that will incorporated onto different platforms for the product, and a product released to meet a particular market demand might be made up of several different platforms, with platforms being updated in future product releases. For example, in the automotive industry, new technologies such as new batteries might be incorporated into a platform such an engine, that may then be incorporated onto several different vehicle lines. The information model for a manufacturer of complex products must be capable of managing the information and the different stages of maturity of information across many technologies, platforms and products as set out in Figure 2:

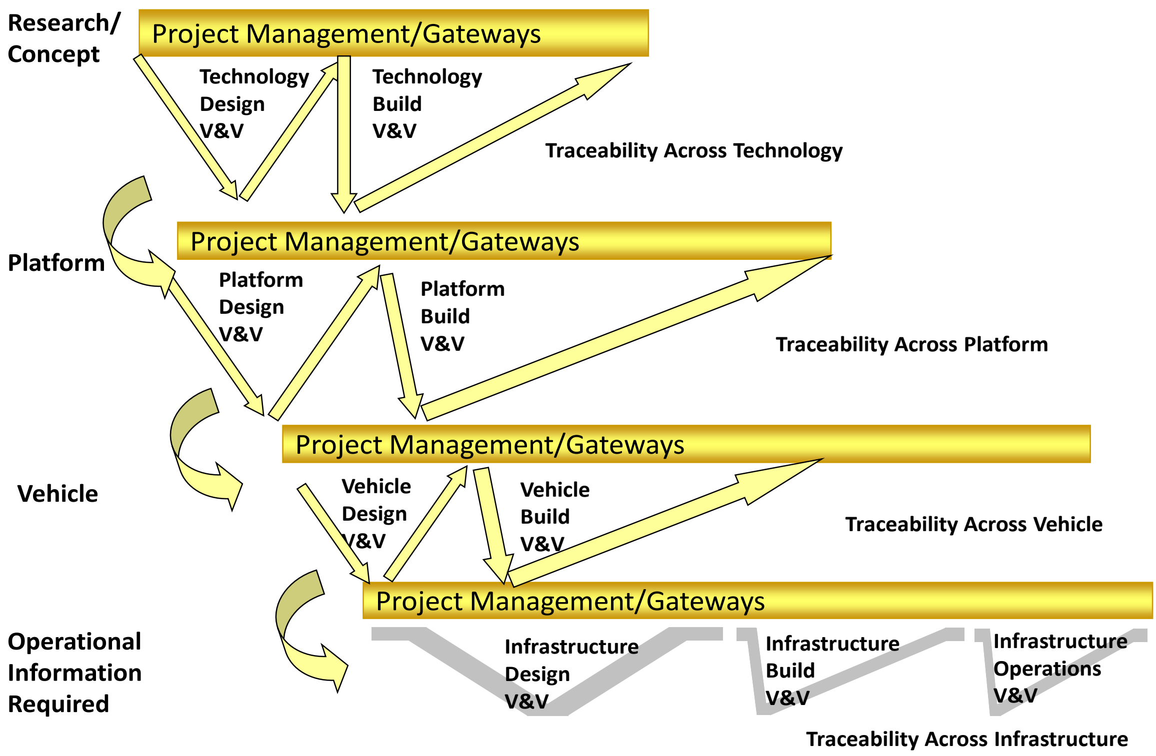

Figure 2: Managing concurrent design and development of technologies; platforms and products in the automotive industry.

To facilitate the idea that technologies, platforms and products might be managed in a flexible manner, a 'Fishbone' conception of an information model has developed. Initially information for a new technology is created and managed as the research, design and development process progresses following the Vee model process for Systems Engineering. Once a technology is sufficiently mature it can be integrated onto a platform which will then be validated before being made available for inclusion onto a product –in this case a vehicle. Figure 3 below sets out the idea of information gathering in maturity and then being included onto a platform and eventually being included in a product (vehicle) programme:

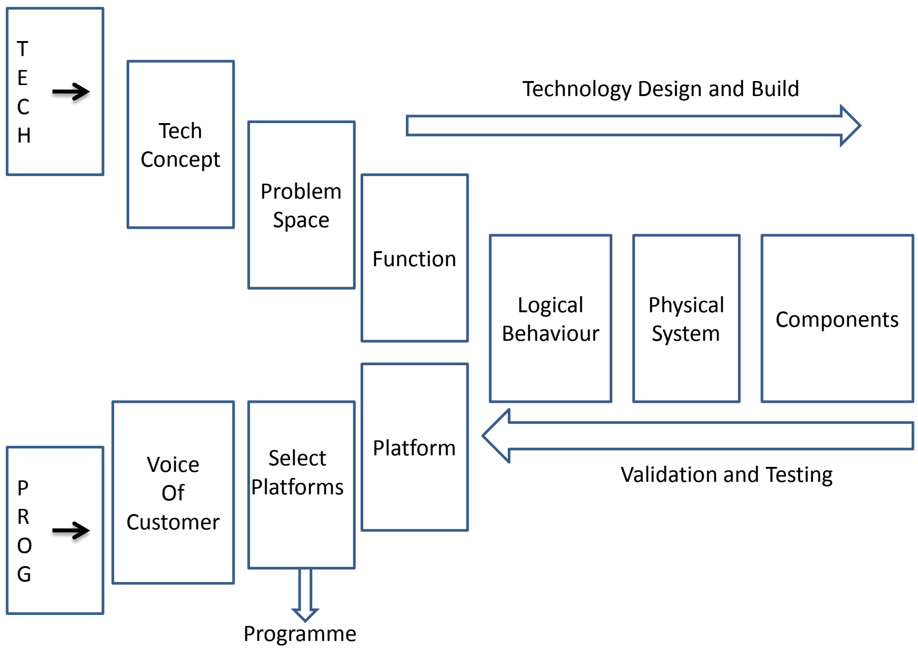

Figure 3: The Fishbone Information Model to manage technologies, platforms and products.

A technology concept will be made up of a potentially large number of Knowledge Assets that become more mature as design and development moves from the logical, to physical world before being specified through particular components. A platform will be made up of many technologies and a programme for a new complex product might be made up of many different platforms.

The fishbone model describes the route through design and development for the various Knowledge Assets that constitute a complex product. Traditionally Fishbone Models (Ishikawa, 1990) have been used to demonstrate cause and effect. Here the fishbone structure is being used to manage the integration of information at different levels of maturity and where the same information is being used for different purposes. There will be many thousands of Knowledge Assets associated with a particular platform and potentially over a million KAs that make up a vehicle configuration. The universal descriptor associated with each KA keeps track of the inter-relationships between KAs and only permits a KA to be incorporated onto a platform once validation has been completed. To facilitate innovation and experimentation, workspaces and problem spaces will be used where KAs can be incorporated into a design without interfering with a platform or programme allowing engineers to decide on the most efficient and safe means of designing new technologies and platforms before a new programme is released. It is envisaged that this approach will allow new programmes to come to market much faster than using traditional information models and also that traceability will be significantly enhanced.

It should be emphasised that the design of the information framework is being been undertaken as part of an Action Research project and the design work has been collaborative. This paper sets out the framework as part of a ‘Declaration in Advance’ (Checkland and Holwell, 1998) in order to gain some initial feedback from the academic community and has been written with permission from all parties. A journal paper is currently being approved that will allow all authors to be named and credit given. Working on a cross-industry project offers many advantages such as access to broad experience and insight but also has challenges when validating and writing up research for publication. The preliminary evaluation of the framework is described in the next section.

4. EVALUATION OF THE FRAMEWORK

The framework described above was developed iteratively by creating a number of conceptual models and discussing these with professionals working across the automotive sector. Three separate workshops were held to discuss the practical relevance and usefulness of the framework. The participants at the workshops had considerable experience of working across the automotive sector, and most had experience of working across globally distributed supply chains and in internationally distributed design and development teams. Each participant in the evaluation process had either a post-graduate qualification relevant to engineering or to business; or a minimum of ten years' experience in the industry and in some cases participants had both. In each workshop the conceptual framework was presented and a facilitated debate followed with each practitioner offering their views and insights as to what were the strengths and weaknesses of the framework.

In discussing the framework, the idea of managing requirements through the fishbone model and the associated information through KA and descriptors was seen to be of value to facilitating through-life traceability; supporting innovation; understanding and communicating the maturity of information and additionally for managing business risk. Additional business benefits could be gained from the virtualisation of testing; potential removal of redundant parts leading to significant cost savings (achieved through better logical modelling thereby addressing late integration issues); complete specification of the product configuration early in a programme, and enhanced programme costing. The potential difficulties were perceived to come from problems associated with the practical implementation of such a framework. From a practical perspective the information model would require all engineers to work according to Systems Engineering principles and in a reasonably standardised manner. Building the capacity across companies to work in this way would require business transformation through training and increased cross-functional working.

One of the outcomes of these workshops is that a major UK luxury car manufacturer is using the ideas presented here to inform a business transformation project and a full proof of concept exercise is being planned where a virtual car will be built underpinned by the information model described here.

REFERENCES

Agresti, W.W. and Harris, R.M. (2009). "Practical Profiles for Managing Systems Engineering R&D" in *IEEE Trans. Eng Man*, Vol. 56, pp. 341-351.

Avison, D. E. and Wood-Harper, A. T. (1990). Multiview: An Exploration in Information Systems Development. New York: McGraw-Hill.

Beuche, D., Birk, A.; Dreiger, H.; Fleischmann, A.; Galle, H.; Heller, G.; Janzen, D. John, I.; Kolagari, R.T.; von der Massen, T.; Wofram, A. (2007). "Using Requirements Management Tools in Software Product Line Engineering: The State of the Practice" in Software Product Line Conference, 2007. Pp. 84-96.

Boehm, B. (1986). "A Spiral Model of Software development and Enhancement" in *ACM SIGSOFT Software Engineering Notes"* Vol. 11 (4) pp. 14-24.

Boehm,B. (1991). "Software risk management: Principles and Practices" in *IEEE Softw.*, Vol. 8, no. 1, pp. 32-41.

Boehm, B. and Basili, V. (2001). "Software defect reduction: Top 10 list" in *IEEE Computer,* Vol. 21, No. 5, pp. 61-72.

Borland (2011). Borland Website Accessed 9/11/11 at: http://www.borland.com/us/products/caliber/

Bonjour, E. and Micaelli, J. (2010). "Design Core Competence Diagnosis: A Case from the Automotive Industry" in *IEEE Trans. Eng Man* Vol. 57, no. 2, pp. 323-337.

Charette, R.N. (1989). *Software Risk Analysis and Management*. New York, McGraw-Hill.

Checkland, P.B. (1981). *Systems Thinking, Systems Practice.* Wiley, Chichester.

Checkland, P.B. and Holwell, S. (1998). *Information, Systems and Information System*. Wiley.

CCTA (1991). *SSADM Reference Manuals*.Manchester, NCC Blackwell.

Champion, D., Stowell F.A. and Callaghan, A. (2005). "Client-Led Information System Creation (CLIC): Navigating the Gap", *Info Sys Journal*, Vol 15, pp. 213-231.

Champion, D.; Belton; C., Lathbury, M.; Howell, A.; Monkhouse, H.; Kawalsky, R. (2012) "Integrating Technical and Business Processes in Complex Systems Design and Development: Lessons from the Automotive Industry" Submitted to: *IEEE Transactions on Engineering Management.*

Egbert, C. and Neve, P.D. (2001). "Surviving global software development" *IEE Softw* Vol 18, no. 2, pp. 62-69.

Ellims, M., Bridges, J and Ince, D.C. (2006). "The Economics of Unit testing" in *Empirical Software Engineering*, Vol. 11, pp 5-31.

Forsberg, K. Mooz, H. and Cotterman, H. (2005). *Visualizing Project Management, Third Edition.* New York, NY: J. Wiley & Sons, Inc.

Galliers, R. (ed). (1987). *Information Analysis: Selected Readings*. Reading, Addison Wesley.

Gil, N.A. (2010). "Language as a Resource in Project Management: A Case Study and a Conceptual Framework" in *IEEE Trans Eng. Man.* Vol 57. Pp. 450-462.

Gil, N. A. and Tether, B.S. (2011). "Project Management and Design Flexibility: Analysing a Case and Conditions of Complementarity" *Research Policy*, Vol. 40, pp 415-428.

Graham, I. (1994). *Object-Oriented Methods.* (2nd Ed.). Wokingham, Addison Wesley.

Ishikawa, K. (1990). *Introduction to Quality Control.* (Translated by J. H. Loftus), Productivity Press, New York.

Jackson, M.A. (1982). *Systems Development.* New York, Prentice Hall.

Jackson, M.A. (1995). *Software Requirements and Specifications.* Harlow, Addison Wesley.

Keil, M., Cule, P.E., Lyytinen K. and Scmidt, R.C. (1999). "A framework for identifying software project management risks" in *Commun ACM*, Vol. 41, No 11, pp. 76-83.

Lancaster, G. (2011). Senior Manager Electronics and Software, Jaguar Land Rover, personal Communication.

Martin, J. (1991). *Information Engineering.* Vols. 1, 2 and 3, Englewood Cliffs, NJ, Prentice Hall.

Miller, J.A., Ferrari, R. and Madhavji, N.H. (2010). "An exploratory study of architectural defects on requirements decisions" in *J. of Sys and Software*, Vol. 83, pp. 2441-2455.

Mingers, J. (1995). "Using Soft Systems Methodology in the Design of Information Systems", in *Information Systems Provision: The Contribution of Soft Systems Methodology.* F.A. Stowell (ed), pp. 18-50, London, McGraw-Hill.

NASA (2011). NASA *Systems Engineering Handbook.* Available at: http://education.ksc.nasa.gov/esmdspacegrant/Documents/NASA%20SP-2007-6105%20Rev%201%20Final%2031Dec2007.pdf (Accessed, 4th January, 2012).

OASIS (2004). "Introduction to UDDI: Important Features and Functional Concepts" Available at: http://www.oasis-open.org/ (Accessed: 10th December 2010).

Sabbouh, M.; Jolly, S.; Allen, D.; Silvey, P. And Denning, P. (2001). "Inter-operability" presented at *The Workshop for Web Serices"*, 11-12th April, San Jose, CA, USA. http://www.w3.org/2001/03/WSWS-popa/paper08 (Accessed: 10th December, 2010).

Sommerville, I. and Sawyer, P. (1997). *Requirements Engineering: A Good Practice Guide.* Chichester, John Wiley and Sons.

Telelogic (2011). Telelogic.com Accessed on 9/11/11at: http://www.telelogic.com/corp/products/doors/index.cfm

Valverde, R. Toleman, M. and A. Cater-Steel, A. (2011). "A method for comparing traditional and component-based models in information systems re-engineering" in *Info Sys E-Bus Man* Vol. 9, pp. 89-107.

Winograd, T. and Flores, F. (1987). *Understanding Computers and Cognition: A New Foundation for Design.* Reading, Addison Wesley.