A PLM APPROACH FOR DEVELOPMENT OF AUTOMOTIVE ELECTRONIC SYSTEMS

Martin Rohdin, Jan Söderberg

Abstract
The development of automotive electric/electronic systems imposes great challenges since different systems, technologies, processes, organizations and people all need to work together within strict project constraints. Traditionally, electronics development is performed using different document-based formats like word editors and spreadsheets, often with little dedicated tool support. When Volvo 3P\(^1\) faced the challenges of developing a common electronic system for the Volvo Group truck brands the concept of one integrated development system seemed as a promising but challenging approach. Existing PLM frameworks however focus on mechanical systems and CAD integration with no real support for software or electronics, so Volvo realized they had to define their own solution. The ambition with the chosen approach was to manage all E/E\(^2\) system information within one comprehensive information management system, offering a single source of information for concurrent and distributed development, and covering all activities from Use Case Modeling to final Testing. Many novel concepts were used, like component based architectures, multi abstraction levels, integration of system architecture and requirements, fine grained configuration management, and more. Recent challenges like the use of an AUTOSAR based architecture, and the application of the PLM system for the development of construction machinery, using different organizational structure and technical solutions, have demonstrated the versatility of the approach.

Keywords: Automotive embedded electronic systems development, Model based development, Product Lifecycle Management PLM

Introduction and background
Volvo 3P initiated the development of a new generation of its embedded real-time E/E architecture in the early 2000’s. It became evident that the new architecture would be greatly more complex than the proceeding generation, supporting much more functionality and a higher degree of integration between systems. The architecture should moreover support not just the Volvo Truck brand, but also the Renault Truck, Mack Truck and the North American Volvo Truck brands. Volvo 3P is a global organization so the development would be distributed at multiple sites located on different continents. In the case Volvo 3P would have chosen to

\(^1\) Volvo 3P is the organization within the Volvo Group that is responsible for truck development, purchasing, and product planning for the Volvo Truck, Renault Truck, Mack Truck, and Volvo North American Truck brands.

\(^2\) Here the term E/E refers to Electronic and/or Electrical (automotive system/architecture….)
continue the traditional document centric system development approach, the development work would have started by defining the black box requirements for system level functions. These functions would then be broken down into a coarse logical block structure, with a specific physical architecture in mind. The system level communication would be outlined using standard CAN messages according to the SAE J1939 standard, and the sub function blocks allocated to the physical subsystems or Electronic Control Units (ECUs). All this needed to be documented within the same requirement and system design specification, traditionally divided per system function, in all around 50 main functions. In the subsequent detailed design phase ECU specifications would traditionally have broken down the specifications for each ECU.

In the early project studies Volvo 3P found that this document based development method would not suffice. For example the logical information elements must be distributed and repeated in multiple documents, making the changes that are inevitable during development difficult to be performed in a consistent manner. Any analyses that the system interfaces were consistent had to be done manually. The document were also single points of access to the information so any editing of a document for one purpose would block out editing by other developers. The inherent problems with the file based document format would become even worse in a multi site, multi project context. It was clear that a more information centric approach was needed. Among the many features wanted from the information management solution was PLM capability, flexibility to adapt to specific needs of Volvo 3P, support for product line architectures, requirement management and systems analysis and design, and support for specific E/E technologies like CAN.

In 2003 a one year long evaluation project was conducted in order to find a suitable solution. Initially more than 20 different solutions were investigated, each with more or less PLM characteristics. At the end three solutions remained, all having clear PLM capabilities. In the last selection activities sample development cases were implanted in each solution. The last phase focused on performance figures, like scalability, access performance, and total system cost. Finally the SystemWeaver [12] [6] platform was selected as a base for further development aimed at adaptation of the system to the needs of the Volvo 3P organization. The information model, or meta model, to be used in the solution was highly inspired by EAST-ADL [1] and AUTOSAR [2], and the adapted PLM solution was named SE-Tool³.

**Single source of information**

SE-Tool provides the possibility of establishing one single source for all E/E systems information for all developed products. The users of this system, and the stakeholders of the contained information are found within many and diverse organizational areas, and in different project phases, all of which have different needs and different requirements. The system users that tend to get the most attention to their needs are the developers responsible for entering the information in the product development phase. The most obvious users of the entered information however are the suppliers who need extractions from the system viewed per sub-system or system component as a base for implementations. Other users, often getting less focus on their needs, are the users found in the test organization who should compare the stated product requirements to the delivered implementations and document the outcome of the tests and write their change

---

³ SE-Tool stands for Systems Engineering Tool (for automotive electronic/electrical systems at Volvo 3P)
requests into the system. In later product phases, the information would be used as a basis for product maintenance and after market service in the form of input to service manuals, fault trace methodologies and diagnostic systems.

One vital property of a system with so many potential users, spread globally as in the Volvo 3P organization, is its ability to support “Concurrent Engineering”. This means allowing multiple users, in parallel, to enter information in the system and provide them with easy and quick access to relevant and up-to-date information, without access conflicts and without endangering the integrity of the information. This puts demands not only on the IT infrastructure and access management principle of the system, but also on the structure of the user organization and on how the developed information is structured. A danger of using a PLM system with limited concurrent development possibilities to develop a complex distributed E/E system in a globally distributed organization is that efficiency may suffer by limiting access rules. Another, less obvious danger, is that the developed product architecture may be governed by these limitations, potentially resulting in a product which does not meet business expectations.

SE-Tool turned out to be a compromise between these needs, and has to a large extent been adapted to the Volvo 3P organization and its specific demands. The system is based on a client-server infrastructure including one single database with no replication or distribution of databases. Instead, the system makes changes on the atomic information level, made from any client, automatically available to all clients in real-time. The system is not based on traditional “check-in/out” operations to prevent users from modifying information while others read it, a feature which often leads to either delaying lock-out scenarios or version-growth followed by merging efforts. Instead, access conflicts are avoided by combining the system’s mechanisms for access rights management and concepts of “atomic item artifacts”\(^4\), atomic change operations and “fine-grained version control [3]. Since change operations are normally performed on atomic pieces of information, potential conflicts created by concurrent access are avoided. The risk that different developers, unaware of each other, perform conflicting changes is limited both by the instant updates to tool views, and due to that access is limited by tool setup to small sized development teams working on the same part of the system.

**Abstraction levels & separation of concern**

The investment in the E/E system model would be significant, and in order to maximize the utility of the model it was important that it was built and organized with reuse in mind. This means that more generic information, such as high level specifications need to be formally separated from more specific detailed design information. Typically software offers more flexibility than hardware and has a different lifecycle, so the software aspects should be separated from hardware aspects. The information must support traceability so that any changes can be analyzed for impact on later development stages prior to decision, and to facilitate the propagation when decided. The traceability must however not prevent reuse of the information. The information centric approach meant that graphical information used by developers during the development could be generated automatically from the system model. The EAST-ADL meta model devised by the ATESTST [1] project was developed with these concerns in mind, and was a

---

\(^4\) By the term atomic item artefacts, we here mean the smallest piece of information that the PLM system can identify and manage as a separate entity.
natural choice by Volvo 3P for implementation in the SE-Tool. The main structure of the EAST-ADL is abstraction levels, see the figure below.

The use of different abstraction levels, and the benefits offered by conciseness, support for reuse, and clear separation between phases in development has to be balanced by the cost of developing these abstractions, and the process and architectural complexity experienced by the developers. The final meta model became partially simplified, and partially extended to fit the specific needs and constraints of the development project.

Another way to increase the utility of the model was to separate the model from tool views. As an example the use of effective content centric Use Case modeling [7] would still support the graphical centric Use Cases of UML [5], using one single representation, see figure below.

The meta model was implemented into a schema set up in SE-Tool, and views were adapted and developed to support the meta model. Although the core of the model has remained since the start it has been continuously adapted and refined through the use in the project. For example, in Volvo 3P the analysis abstraction level was later on abandoned as part of this balancing, while in
another Volvo organization the analysis model was re-instituted since it offered the needed
distinction between high level architecture aspects and detailed software design aspects.

From document driven to model driven development

Another challenge for the Volvo 3P organization was the transition from a document driven
development process to a more formal model based process manifested in the development of a
system model for the E/E system.

Paper based information drives the creation of redundant information, as each piece of
information must be repeated on many places in a document, or in many different documents, to
suit different needs and stakeholders. Thus, updating the information will be inherently difficult,
because it will be virtually impossible to update all occurrences at the same time and in a
consistent way. A PLM system provides a better control of the information, information is reused
instead of repeated, occurrences are traced and changes made in one place are visible in all
occurrences and to all readers at the same time.

Volvo 3P saw many motives to head for a PLM based approach; the organization had grown to a
globally distributed team of engineers, the foreseen complexity of the next product was much
greater than the previously developed, the planned system was expected to become more
distributed in the product, the pieces of product information were expected to be heavily
depending on each other, and the needs for reusing information in a controlled way to keep cost
under control grew.

In the document driven approach users are often not aware of any distinction made between the
activities to develop and collect the information from the activities to structure it and to select a
layout for presenting it to the readers of the final document. In a model based approach the
activity to develop and collect the information is often clearly separated from the activity to
present it on paper, and the development activity is traditionally supported by different views
adapted to develop different aspects of the information, none of which reveals the complete
layout and structure of the final output. Consequently, in the Volvo 3P organization where by
tradition the most important readers belong to the own organization, many Volvo 3P developers
felt doubts that the efforts to enter the information into the system should pay off in the end and
result in understandable and informative specifications. In the chosen approach, the ambition is
that no E/E system information shall be managed by paper documents, but shall be entered in SE-
Tool using different model views adapted to the specific information characteristics and to the
needs of the developers responsible for the entered information. Besides being viewed directly in
SE-Tool in the same format as it was entered, the information may be processed and combined by
the system to generate a wide range of outputs such as graphical diagrams and customizable
reports for consistency and completeness statistics. The system also provides an API for the
extraction of unprocessed information for external manipulation. At the end of the day, paper
based specifications seem to be a necessity to communicate in a project, at least as long as not all
stakeholders in all project phases share the same models and semantics, why SE-Tool also
supports extracting information in different word processing formats e.g. to auto-generate
specifications for software implementation.

The introduction of a model based development process and the deployment of SE-Tool at Volvo
3P brought a new challenge to the organization, the need to conform to a semantically well-
defined schema for system representation – a meta model. The customized meta model is adapted to Volvo 3Ps needs and model conformance is governed by SE-Tool. The system strictly dictates what type of information that is allowed to be entered into the model, and how this information may be combined into specifications of structures and systems.

The information centricity and focus on details inherited by introducing a governing meta model led to that much of the information traditionally managed by, or often even hidden by, the document driven process, now was challenged and must be thoroughly analyzed, restructured, and most often detailed to fit into the new model. This in turn pushed for changes in process and organization. Also, as the ambition and scope of SE-Tool grew, the meta model had to be developed and updated accordingly. And the scope still grows…

Architecture, components & requirements

Proper management of requirements is recognized as paramount in the development of complex systems [4]. This is especially the case in the automotive context where external suppliers perform development and manufacturing of components and complete subsystems, since the system specifications will be part of the contract between the supplier and the manufacturer. This situation has been recognized by the German HIS organization, and supported by its RIF standard [13], one of the data interchange formats supported by the SE-Tool. The traditional use of requirement specifications being detached from the system architecture definition is not the most suitable one for the case of E/E systems, or indeed any other complex systems and systems-of-systems. Instead the properties of the architecture should rather be expressed and managed as formal requirements and integrated with the building blocks of a formalized system architecture. The architecture description is organized into abstraction levels, and the requirements of one level are traceable to higher levels using derived-from links to higher-level requirements, quite similar to the SysML standard [14]. Optionally a more coarse-grained block-level traceability is used where traceability is made from building block to building block. One difficult decision in defining the meta model is to settle an appropriate level of semantic richness and formalism. Although the preciseness of a detailed model may be attractive, it also requires increased effort to be maintained throughout the life cycle of the system. Semantic richness can also be a threshold to overcome for the developers involved in projects. This aspect is especially important in the heterogeneous automotive environment, where the developers may be short-term consultants, or resident engineers with little time to dig into the details of a rich meta model.

One consequence of using the EAST-ADL and AUTOSAR in a fine-grained PLM approach is the complexity and amount of configuration items to be managed by configuration management processes. Although platforms like SystemWeaver support this, the amount may still be perceived as a challenge.

Initiating and managing change

An organization that implements a model driven development process will be forced to undertake many changes. These will not only encompass the basic tools and methods by which the product is developed, but will also have effect on established principles on how projects and line organizations are structured and how responsibilities are shared in the organization. Thus, they are often seen as a challenge to established structures and as threats to existing routines, and may
be received reluctantly by individuals of the organization. One of many possible reasons may be found in the common consequence of model driven development, that work is broken down to developing small pieces of a complete model at the same time and by the same individual. Thus, it may be difficult for both developers and project managers with a background from projects driven by more traditional development methods to see the big picture or understand how and where the pieces add to the benefit of either the organization, or the individual. Another reason is that the big benefits from the improved information quality and structure brought by the model based approach are often not achieved in the first project, which have to adapt to the change and normally will take much of the costs, but in coming projects that may reuse already developed models.

Managing these non-technical challenges is of outmost importance. Failing to do so will most probably lead to a situation where more time is spent on finding arguments to persuade people who do not want to be persuaded, of the benefits of the new development process, than on developing and introducing it. One key to the successful implementation at Volvo 3P was the introduction of a strong support organization with a clear and unquestioned mandate to define the rules and schema, basing the decisions about meta model and process rules on a close and broad cooperation with the SE-Tool users, project managers for the end product, and the supplier of the system.

An often suggested approach to manage the confusions while introducing complex changes is to introduce them gradually. The possible options would then be either to introduce the changes isolated per engineering domain, like “infotainment”, “comfort functions” or “powertrain control”, or isolated per different development process or project phase. These proposals are all good and may be a way ahead. However, introducing model driven development gradually is inherently difficult since tools, methods and roles are so tightly interweaved and because the
targeted E/E system is often quite distributed from a functional aspect and/or by the physical topology, which was often one of the motives for introducing model driven development in the first place.

Also, when introducing model driven development processes to develop a distributed system, a gradual introduction divided per engineering domain have the drawback that some parts of the model will need to interface or utilize parts of the system (functional or physical resources) that are outside the scope of the model. This will make it hard to draw a clear border to what is included or not, require much effort to be spent on modeling “environment stubs”, and on defining and guarding the border of the scope. Later, when the change is finally introduced on a full scale, this approach will also require some amount of re-work to replace the stubs with real data. Also, the gradual approach may lead to that a sub-optimal system model is introduced and manifested, as was described above.

Another common method used to approach uncertainties and manage risks, is to try out planned changes in a simulated environment, in the case of introducing model driven development often referred to as “pilot activities”. In such pilot activities, a small team representing “all” skills planned to be involved in a real development organization is assigned to take the lead in the development activities. By practical trial-and-error work, preferably using realistic system data, but in a test environment, that team simulates the proposed changes to tools, methods, and processes to find potential pit-falls and come up with alternative solutions. Thereby paving the road ahead in front of the main troops and avoiding production critical issues to hit the main development activities. Pilot work like this can be initiated both before a PLM system is deployed, or even chosen, as some kind of “proof of concept” or “user validation” activity, or in the deployment phase to fine-tune methods or processes, or even in the process maintenance phase to come up with and elaborate on enhancements. A critical factor for the outcome of any pilot activity is the availability of correct resources, limitation of the pilot scope, and access to realistic system data. The experience from Volvo 3P shows that the ideal resources for pilot activities are often the organization’s key-resources, usually heavily involved in multiple other on-going projects and activities, why it was quite difficult for them to find the time to focus on pilot activities. Another experience was that one single pilot can not focus on too many issues, otherwise its outcome will not be in enough depth to identify all pit-falls, or it will not be able to deliver its output in time before the main organization is forced to face the unfinished issues. Finally, if the piloted data is too simplistic, all difficulties will not be revealed in the pilot, why, again, the outcome will not be detailed enough, and there will be unfinished issues to deal with in the sharp project.

Regardless of which introduction strategy is chosen, introducing a model driven development process in a target project will require much support to end users. Depending on the organization’s level of maturity and previous background from model driven development, object orientation and information management, this support may be needed during a very long time, or in the case the staffing of the project is volatile or the target project time spans over many years, a support organization may be needed as a permanent part of the organization. In the Volvo 3P case, traditional training in tools and methods was only regarded as a basis, on which a continuous and more dedicated “per user” support, was added. This “Modeling Coach” organization has proven successful and was made a permanent part of the Volvo 3P organization. It has grown into a two-way communication interface both to provide users with immediate, directed and hands-on support, and as a way to collect both new needs and enhancement proposals from end users to the
maintenance and enhancement activities for tools and methods. It has also proven a good way to communicate changes to tools and methods made by the maintenance organizations to the user community.

The introduction of a PLM system usually does not start by the challenge of training end-users in a tool or providing support to the methods that comes with model based development. Long before that, the challenge would be to get acceptance from the management for that there is really a need for introducing new processes and methodology in coming product development projects, and to get their attention, access to funding and resources to initiate the change process. The needs may spring from a foreseen increase in product complexity, from organizational changes or, as in the Volvo 3P case from a combination of these. These needs may not be recognized by everybody in the organization though. Members of an organization in a phase in between projects focusing on maintaining existing products developed by more traditional methods which, in retrospective, seems to work quite fine for those products, may not see these needs as clear as people involved in early phases of coming product development projects, analyzing the organizational challenges and technical risks. Thus, it is vital to really define the needs and find unambiguous motives for each change, which can be understood and accepted by people on all levels in the organization.

In these initial phases it is a common mistake to think of the introduction of model based development as an isolated tool issue. Even though it may be easier to find funding and management’s attention for a new “silver-bullet” tool, the complexity of the changes is too big for this simplistic approach, at least if applied in full scale, to an organization with none or just sporadic experience to model driven development processes. A change of tool must go hand-in hand with the definition and introduction of applicable and efficient methods and processes, as must the organization get access not only to tool support, but to support to apply methodologies and execute process activities.

A major question to address early in a project aiming to introduce model driven development, is if the best approach is to adapt the organization, process and methods to a tool, or the other way round, to adapt the tool to existing processes. I practice a mix of these approaches may be the best way.

Conclusions and Future Work

In the project we showed that it was indeed possible to manage most parts of the system specification in one single model, used as one single source of information by the developers. By using a model that allows entire object structures to be used and viewed for different purposes a minimum of redundant information was used, greatly improving the management of changes. Dividing the information into different levels of abstraction improves the possibility for reuse between projects, and protects the integrity of information during changes. By integrating the requirement structure with the system architecture many integration issues could be addressed already during the specification stage. The complexity of such a model and the transition to model based specifications rather than document based specifications however requires specific tool support for authoring and review, as the model as such does not support specific aspects in the way hand written documents do. One of the novel concepts used in the model, fine grained model based versioning rather than file based versioning proved to be a challenge to many developers, since the users have to be aware of the multi-dimensional version aspects during all
development activities. One conclusion was that the semantic richness of a meta model has to be balanced by the need for simplicity and usability. One important issue that needs future investigation is the support for version operations, especially in multi project contexts. The level of support towards the development project is a critical success factor when introducing an information management system of this kind, since many things are changed at the same time.

References
[9] Linda Northrop “Product Line Architectures”, SEI

Corresponding authors:
Martin Rohdin, Volvo IT, Electrical & Systems Engineering Services, martin.rohdin@volvo.com
Jan Söderberg, Systemite AB, jan.soderberg@systemite.se