A Bridge from User Requirements to Forecasted Embedded Systems Technology
Sylvain Alliot1, Martijn van Veelen1
1 ASTRON, Oude Hoogeveensedijk 4, Dwingeloo 7990,Netherlands
Email : {alliot,veelen}@astron.nl

Abstract— The design studies of the embedded systems for future distributed data-processing applications assess the cost and performance of the hardware infrastructure. Specifications are derived from user requirements depending on system architecture; delivery time and enabling forecasted technology. Relating requirements to forecasted technology is complex because these aspects are much intertwined. We bridge the gap for the embedded system technology that are major parts of the next generation radio telescopes, LOFAR and SKA.

We were able to take the technology advances into account early in the project. This was possible using hierarchical and stochastic performance models. Hence we could derive hardware specifications from astronomical applications and assess cost-performance ratio for signal processing scenarios from specifications of forecasted technology.

Keywords— Embedded system technology, prediction models, viewpoints

I. INTRODUCTION

The new generation of radio-telescopes, LOFAR and SKA, are large scale distributed sensor networks. A large fraction of the infrastructure for these telescopes are embedded systems. Their adequacy relies heavily on high performance electronics. These embedded systems require innovative design breakthrough as their performance will scale the performance of the entire system. Similar infrastructures are switching fabric in telecom and radar. Large scale distributed embedded systems are distinct from consumer electronics in the sense that the time-frame for realization are in years rather than months and there is only one full system produced. Therefore, an additional difficulty is the one time industrialization experience. Given these difficulties and a fixed budget, various concurrent user requirements should be satisfied and this creates points of tension. Due to the size of the system, the points of tensions and the uncertainties are risks that cannot be mitigated by iterative prototyping of the full product. Therefore there is a preliminary design phase years prior to the role-out to address these risks. We aim in this phase at finding system specifications compliant with the user requirements and the limits of enabling technology.

In this context, the compliance of systems architecture with user requirements and it’s cost-efficiency is evaluated years before construction. The system architecture depends on embedded systems technology. System cost and performance have non-functional aspects too, such as development cost and capability, reliability, availability and maintainability; a meaningful analysis requires a choice of technology. Consequently, through the system architecture, forecasted embedded systems technology has to be related to the user requirements, despite technological uncertainties. The uncertainties are in terms of availability time (ETA), and performance/cost as shown in Figure 1. In addition, there are strategic technological choices to be made in order to anticipate the implementation phase: Typically domain specific board layouts emerge as a result of many years of engineering experience. Nonetheless to reduce development and maintenance cost we necessarily consider industrial standards and commercial off the shelf (COTS) solutions, see Figure 2.

In Section III we stress the difficulties in relating user requirements and technology. An approach to this problem is presented in Section IV. This solution was experimented during the preliminary design phase of the LOFAR telescope. We give in Section V some results of technology and requirement assessment using technology roadmaps for LOFAR.

II. RELATED WORK

Technology triggers new concepts. New requirements trigger the research on new technology. Therefore identifying the relevant technology and bridging technology to...
requirements is at the heart of innovating solutions. However, this is not fully supported by an engineering process. On the one hand, classical system engineering practices (i.e. high level description in UML) that help to separate views on the system or the viewpoint oriented conceptual modelling [1] do not assess the system performances and therefore cannot be used for decision support on the technology components. On the other hand, full system prototyping of detailed system simulation methods (e.g. in System C [2]) can assess the performance but are not applicable for very large systems when multiple options need to be investigated. Therefore in practice, the process is handled in an ad hoc way by an expert that will most likely fail to see all the dependencies between different disciplines and technologies. An alternative is to predict from an earlier product the performance/cost proportionally to the roadmap of a dominating component, e.g. Moore’s law, and estimate the requirement variations but then the predictions are too coarse.

The Massive project [3] pioneered methods and tools for the specification of large scale signal processing systems. We show here how these can be used to separate viewpoints on a large system and to integrate them avoiding discrepancies (conflicts and inconsistencies) for a real large system case.

III. PROBLEMS

In the feasibility assessment and technology selection for a large scale embedded system we are facing several challenges in the internal and external complexity of the system such as the ones listed in [4]:

- Multi-disciplinarity: intimate coupling of disciplines
- Heterogeneity: mixing different types of technology
- Scale: products are compositions of embedded systems

Too major problems are:
- The inherent interdependence of architectural decisions
- The conflicting stakes and views, i.e., not taking a system perspective

A. Interdependence

We chose a component for a system depending on different factors or other components in the system. These factors or components are interdependent. We give an example of interdependence for selecting a particular technology.

In this example a filter is used after digitization. The filter length and thus the signal processing quality is chosen such that our application is integrated in a single component. There is a relation between the bandwidth of the signal delivered and the performance of the device needed to transport the signal. An alternative to cut on the cost for the communication device is to reduce the bandwidth but using more filter coefficients. Consequently we might need an external memory. We could also relax the transition edge of the filter if the signal quality loss is acceptable to the users.

We identify here with this example the interdependence in the architecture of a system.

- The refinement of requirements across subsystems
- The processing functions and algorithms
- The partitioning of the processing tasks
- The technology for processing and communication
- The distribution of processing tasks across the system

An architecture is the result of a decision process given the dependencies listed above.

B. Conflicting views and stakes

Various design iterations are needed before converging to a solution. In particular, the technology available contributes to a solution. Deriving detailed requirements for the hardware or estimating system performance and cost from hardware specifications, often a back of the envelope activity, becomes a nuisance as the number of implementation options grows rapidly with different experts shining their light on the problems, preventing the convergence. The stake holders are the users, managers, subcontractors and technology providers that want their services and products to be used. Domain experts have by definition a limited view of the system and they cannot see always the impact of their decisions on the system. They introduce a bias in the evaluation of this impact. Moreover, in this context, it is difficult to keep track of options and to trace the modifications if any if the views are imbricated.
IV. Solution

A. Approach

In order to take decisions on the technology, we need to assess its impact. This is done by composing variants of the system architecture in the form of scenarios and modelling the performance/cost of the system. The performance results provide rationale for design decisions. Another important aspect is to separate the various concerns through views across the system architecture supporting entry and visualization for the multiple disciplines and stakeholder. The views on the system clarify the relations between decisions in the different domains of expertise. Important views are architecture, application, mapping, performances/cost, instantiation, requirements, and constraints at different levels in the system hierarchy.

The approach followed in Massive is summarized in Figure 3. In this figure, a library of parameterized constructs in both application and architecture domains are used to compose a system scenario. On the left hand side, a scenario is derived from a set of requirements that are refined into requirements for the various subsystems. The requirement set is related to constraints on the system architecture (a). A scenario can be modified as in (b). In such case, the dependencies between components of the system were already established and the constraints are re-assigned automatically following the procedure detailed in [5]. The requirements are related to the constraints and, therefore, the changes in the requirements sets 1 and 2 are also traced automatically.

On the right hand side, the library constructs are associated to performances that can be obtained from prototyping or simulation experiments. This information is used for evaluating the performance of the components in the larger systems given aggregation rules and the technology specification [6]. In Figure 3 (c), the constructs are parameterizable and can be instantiated with current or forecasted technology. Finally, the impact of the technology can be related to other decisions in the architecture. It supports an iterative process in which the system composition can be rapidly modified and the performance evaluated and validated. This allows for:

- The identification of and prioritization of issues and conflicts by creating different scenarios.
- The assessment of the impact of architectural changes.

B. Views in the Massive tool set

A prototype software tool set has been implemented by LIACS and Astron for architecture exploration. The tool consists of five software modules: a knowledge database, a system database, a topology editor, constraints and instantiation processor and a performance model, see figure 4. We explain only the user interaction with the tool set here, the interested reader is referred to [7] for more details on the tools.

The objective of the tool set is to separate the different aspects of the architecture through different views, while maintaining a consistent architectural model. The supporting views are enumerated here:

**Domain Experts** The knowledge base offers three views:
1) signal processing techniques, e.g., filtering and array processing algorithms; 2) technology, e.g., processors, memory, communication devices, power supplies and housing; technology is organized by type and device family; 3) platforms, that are architectural templates and their
instances, e.g., astronomy specific processing boards and commercially available boards.

**Application View** The signal and control flows of the application are process networks composed in the Ptolemy [8] Vergil user interface. The application elements are selected from a library, e.g., FFT or FIR Filters.

**Architecture View** Similarly, the system architecture is hierarchically composed in Ptolemy, the elements are platforms and components selected from a library.

**Mapping View** The process network in the application view is assigned to the platform network in the architecture view. The three views: application, architecture and mapping enable the hierarchical composition of a system at a high abstraction level.

**Constraints View** The top-level requirements for a system can be entered in an excel sheet, which is imported to derive constraints on every element of the system.

**Instantiation View** The parameterized elements are instantiated by selecting instances from the database.

**Performance view** Derived performance and validation results can be visualized along the system hierarchy and compared to other scenarios.

Figure 5 gives an example of view interactions with a visualization of the application network breakdown (bottom left of Figure 5), a view on the constraints (bottom right) corresponding to an element of the system (i.e. a filter), and a list of possible instances (top of figure 5).

The views implemented in the MASSIVE tool separate the concerns for the different experts. Modifications can be made in the different views independently, the tools ensure consistency of the architecture across the views though the automatic re-assignment of constraints, validation of the topological correctness, and performance computation.

V. RESULTS

The methods and the tools sketched in the previous section have been used in the preliminary design phase of the LOFAR radio telescope. They helped to conduct the investigation and developments related to the technology. We conducted a targeted prospective survey of the technology over three years [9] for high speed communication devices, processing and memory components. As a result, relevant components could be anticipated and specified in detail, including their cost information and forecasted performances. The exploration quality depends highly on the information collected. Unfortunately, in the high technology market, the above issues can fluctuate rapidly. Trends must then be estimated from technology roadmaps on a short term and the database updated accordingly.

A. LOFAR Station Scenarios

The subsystem investigated was the stations of LOFAR and the impact of technology on the performance/cost of the systems. Tradeoffs could be made in the architecture after conducting an extensive architecture exploration. The results of the exploration are reported in details in [10] considering about 200 realistic scenarios. We illustrate the assessment of technology in the present section with a few examples. The outcome is a ranking on the criticality of the decisions choices, trade-offs, preferred technology.

B. Example I: technology impact

One of the architectural option in LOFAR was to use standard PCs interconnected in a cluster. After evaluating the performance of the entire system we show that the bot-
tle neck in the architecture is the critical I/O usage between a co-processor board and the CPU. The bus technology on a PC at the time of the study is PCI-32 in 2002Q1, however the system needs to be built before 2006. Looking at the technology roadmap for the PC busses as shown in Figure 6, a number of standards have been announced together with a probability of availability. In this figure and posterior to the evaluation we give an example of uncertainty in the predictions for the PCI-X 266 standard in 2004. Based on the roadmap and the predicted performances, different system performances are derived at different deadlines of the system integration. In particular, as shown in Figure 7, the number of processing nodes in the network can be reduced from 46 nodes to 23 in 2005 with a critical use of the bus with 80% capacity. However the target in terms of system cost is not reached by 2006 using this type of technology and therefore an alternative architecture was selected using high speed backplane technology instead.

C. Example II: dynamic range

The impact of a particular processing component in another architecture scenario later selected as a baseline is illustrated here with the choice of cost efficient FPGA families. In particular, the dynamic range of the signal processed can be limited by the internal architecture of the components. The selection of the processing arithmetic is a difficult task because there are no ideal representation formats. Effectively, the relative size of the architecture increases quadratically with the size of the mantissa. The architecture is hence optimized through a minimum assignment of coding levels after an operation. Arithmetic can be floating point, integer or block floating point, depending on the architecture support; all of these generate a quantization noise. Fortunately, the effects of quantization are only significant above a threshold that is fixed. We show in Figure 8 that a slight change in the format of the mantissa increases the cost by a factor two. In this particular scenario, the arithmetic is block floating point and only marginally satisfies the application quality criteria. Modifying the arithmetic to floating point leads to the case of a mantissa with more than 18 bits. On a different architecture support that is less cost-optimized (no embedded multipliers), this gap can be closed with modified floating point formats (9 bit mantissa only).

VI. Conclusions

The method experimented here for LOFAR is an attempt to rationalize the decision process and guide the technology developments. We have identified a level of detail for describing a system that can be related to acceptable performance measures. This avoids detailed system modeling as proposed in SPEAR [11] for radar systems. In particular, we could relate emerging system properties to...
individual system components and quantitatively evaluate specification decisions based on specification space metrics.

However, this approach imposes a number of constraints. The knowledge database needs to be updated regularly and consistently. The scenarios specified also need to be maintained along the project timeline, which requires much discipline.

Other non-functional aspects are also not measured at system level (e.g., reliability, maintainability). Some non-decomposable aspects, like electro-magnetic compliance for instance are not modelled, and the impacts of changes on EM cannot be estimated. In addition, the decomposition and verification of the functional requirements for signal processing quality should be facilitated. The non-functional aspects and requirements analysis for signal processing quality will be pursued in future research.

ACKNOWLEDGEMENT

This work has been conducted under the MASSIVE STW Progress project LES.5028 in collaboration with Leiden University. The authors would like to thank Ed Deprettere for his guidance and Laurentiu Nicolae for his review comments.

REFERENCES


