# Defense Technical Information Center Compilation Part Notice

# ADP010662

TITLE: Environment for Signal Processing Application Development and PrOtotypiNg - ESPADON

DISTRIBUTION: Approved for public release, distribution unlimited

This paper is part of the following report:

TITLE: Commercial Off-the-Shelf Products in Defence Applications "The Ruthless Pursuit of COTS" [l'Utilisation des produits vendus sur etageres dans les applications militaires de defense "l'Exploitation sans merci des produits commerciaux"]

To order the complete compilation report, use: ADA389447

The component part is provided here to allow users access to individually authored sections of proceedings, annals, symposia, ect. However, the component should be considered within the context of the overall compilation report and not as a stand-alone technical report.

The following component part numbers comprise the compilation report:

ADP010659 thru ADP010682

# *E*nvironment for *S*ignal *P*rocessing *A*pplication *D*evelopment and Pr*O*totypi*N*g - ESPADON

Bob Madahar, Jan Hunink[1], Gilbert Edelin[2], James Smith[3], Brigitte Saget[4]

BAE SYSTEMS Research Centre
West Hanningfield Rd,
Gt. Baddow, Chelmsford CM2 8HN, U.K.
bob.madahar@gecm.com

**Abstract:** Defence industries are increasingly expected to field state-of-the-art products, at significantly lower costs, over significantly shorter time scales, and with significantly greater functionality. New designs, as well as design upgrades, are expected to keep pace with technology advancements, particularly in microelectronics. These constraints, and others, are forcing industry increasingly towards Commercial Off The Shelf (COTS) components (hardware and software). The advantages are reduced costs and state-of-the-art technology compared to proprietary in-house developments, and hard-wired solutions, which have long development times and are invariably out of date by the time the product is commissioned. The disadvantages are principally non-compliance with rigid military specifications of the COTS components and the inability of defence industry product design development and integration methodologies, established over many years, to accommodate the COTS components in an efficient and timely manner. Obsolescence (more acute for bespoke designs) created by COTS components for the long life-cycle military products, is also a key concern and leads to costly retrofits unless the potential design upgrade is included in the design methodology.

These major concerns are being addressed for defence embedded signal processing applications by the tri-national European EUCLID/Eurofinder defence programme called ESPADON. The primary objective is to significantly improve (reduced cost and timescales) the process, by which complex military digital processing systems are designed, developed and supported. A new design methodology, and a new development environment, has been reinvented to support this aim through reuse, concurrent engineering, rapid insertion of COTS technology and the key concepts of rapid and virtual prototyping. These techniques and developments are presented in this paper.

*Keywords: ESPADON, Methodology, Prototyping, DSP, COTS*

## 1 INTRODUCTION

Over the last decade there has been a sea change in the climate for the development of military digital processing systems. Principal factors forcing the change are:

*The New World Order* – the end of the cold war has seen a dramatic reduction in the defence budgets worldwide and changed the perceived future requirements. Political changes, economic globalisation, and technology advancements, sometimes on the back of 'local' conflicts, have also brought additional competitors (Israel, South Africa, India,..) into the market place.

*The Microelectronics Revolution* – the exponential growth in the performance of microprocessors and associated electronics (> 10M transistors/device and rising, Memory x 4 every 3 years [>256Mbit DRAM, >8Mbit SRAM], Clock rates x 50 every decade). This is continuing apace (1.5 order of magnitude increase in performance every decade, Moore's law survives!) with no immediate signs of abating or hitting the fundamental limits of physics (see Fig. 1)

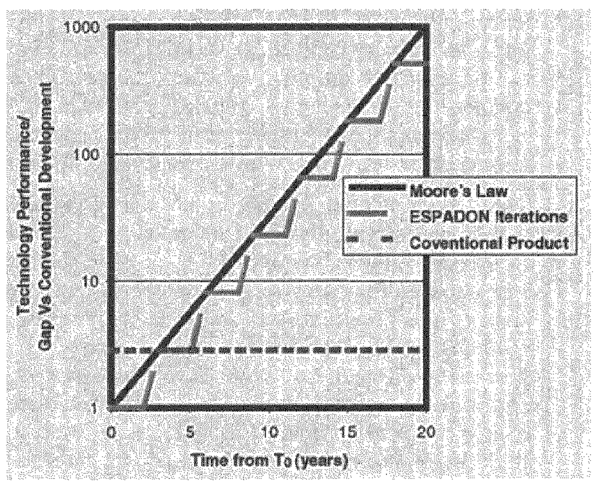


*Fig 1. Iterative Development Methodology*

[1] Signaal, Zuidelijke Havenweg 40, P.O. Box 42, 7550 GD Hengelo, Netherlands
[2] Thomson-CSF Detexis, 1 Bld Jean Moulin, 78852 Elancourt CEDEX France
[3] Thomson Marconi Sonar Ltd., Dolphin House, Ashurst Drive, Bird Hall Lane, Cheadle Heath, Stockport, Cheshire, SK3 0XB, U.K.
[4] Matra BAe Dynamics, 20-22 rue Grange, Dame Rose, B.P. 150, 78141 Velizy-Villacoublay, CEDEX France

*Paper presented at the RTO IST Symposium on "Commercial Off-the-Shelf Products in Defence Applications "The Ruthless Pursuit of COTS"", held in Brussels, Belgium, 3-5 April 2000, and published in RTO MP-48.*

*The New Age User* – the customer/user expectations are ever more demanding in terms of system interoperability, functionality, capability, and cost (better, faster, cheaper). Note this is conditioned, perhaps unfairly, by their daily exposure to the high performance, fast graphics, highly integrated and easily networked PC environment available on their desk top.

These conditions give rise to the adoption of COTS and the demise of the conventional military methodologies and bespoke military specific developments of signal processing systems for the following reasons:

a) For a specific application, at a given time, the optimum (performance) signal processing designs are likely to be bespoke non-standard hardware and interfaces, software optimised for the specific hardware, and a performance driven unique solution. Such 'company-centric' proprietary developments, and hard-wired solutions, have long development times and are invariably out of date by the time the product is commissioned. This is costly (time and money), the systems are difficult to reuse, and in any case quickly (a few years) overtaken by COTS technology, and emerging standards, and rendered obsolete (technology and components).

b) The support for military specific components by microelectronics vendors is declining, as they position for the significantly larger consumer market, thereby accelerating obsolescence problems and increasing the costs of military specific designs. The COTS components offer significantly better cost/performance ratios that the defence industry must try to adopt to remain competitive and offer a leading technical solution. Vendors recognise this and offer Military Off The Shelf (MOTS or COTS+) components that are slight variants of the COTS components to include extended environmental range of operation and higher quality components to improve the Mean Time Between Failure (MTBF).

c) The conventional timescales for product development (typically 4 yrs for Sonar and 10 yrs for Radar), followed by over a decade of in-service support, are disparately long compared to the very short (~ year) revision rates and technology refresh rates for COTS digital processors. Hence COTS processing technology will have increased in performance by one or two orders of magnitude over the typical lifetime of a product. To leverage these developments, defence industry must reduce development timescales and design for the rapid insertion of emerging COTS technology (design for upgrade) so as to maintain a leading solution for the customer.

d) The significant disparity in timescales discussed above presents defence industry systems with an acute obsolescence problem that is occurring earlier and earlier in the overall product lifecycle. At present there are two methods available to handle such a problem. The first is a lifetime buy and storage of components that are going to become obsolete. This requires capital outlay upfront, hence depreciating in value, based on exiting sales and estimated sales in the future. The latter may not materialise (capital loss) because the system by definition is obsolete compared to the current competing products. The second is an equipment retrofit with the current component technology. Unfortunately, design for upgrade is not integral to conventional military designs. Hence the retrofit is comparable in cost to the initial development and therefore competes unfavourably with current competing products. The solution is a new methodology where the development times are reduced, with design for reuse and design for frequent upgrades as an integral part of the process.

The new methodology, to ameliorate the above concerns, is being developed by the tri-national European EUCLID/Eurofinder defence programme called ESPADON [1] [2] as it is beyond the resources of a single company and Nation. The international consortium comprises Thomson-CSF and Matra BAe Dynamics from France, Signaal from Netherlands, Thomson Marconi Sonar Ltd and the Marconi Research Centre from the United Kingdom. The 3 year duration project, jointly funded by the consortium and by the Ministries of Defence of France, United Kingdom and the Netherlands, started in July' 98.

## 2 NEW METHODOLOGY & TECHNIQUES

ESPADON is the European analogue, albeit with only a few % of the budget, to the U.S. tri-service research programme RASSP (Rapid Prototyping of Application Specific Signal Processors) which was initiated in 1993 with a budget of $150M and lasted for nearly 5 years [3]. RASSP was a very broad programme involving government, defence industries, Electronic Design Automation (EDA) industries and Academia investigating three principle threads – Architecture, Methodology and the Education and Enterprise Infrastructure. The focus of ESPADON however is much narrower, towards the methodology and environment for embedded signal processing applications, and benefits from the lessons learnt by the RASSP programme.

The main project thrusts are:

1) Synthesis of an advanced design methodology and processes for the development of the next generation real-time signal processing systems;

2) Analysis and evaluation of COTS tools, emerging standards, signal processing and communications libraries, and associated techniques of direct relevance to the methodology;

3) Implementation of an ESPADON design environment (EDE), based on the integration of best of class COTS tools, standards and techniques,

within an extensible software framework, that can support the methodology;

4) Demonstration of the objectives through the implementation of real-time adaptive signal processors for Radar and Sonar applications on COTS hardware platforms;
5) Measurement of metrics to quantify the productivity gains and to validate the EDE, the techniques, and the methodology; and,
6) Dissemination of the project and results via the internet, seminar and workshops aimed at European companies.

Of these, the progress midway through the project is that, the methodology has been specified (1), the initial set of COTS tools for the EDE selected (2), and a preliminary version of the EDE implemented (3) and the benchmarking activity begun (5) as described in the sections below.

## 2.1 Methodology

The conventional methodology for signal processing system or component development is analogous to that for software engineering in the late 70's. It can be represented by the sequence of different activity steps, Requirements-Specifications-Design-Implementation-Testing, where a new step begins when a previous one has ended. The sequence is known as a 'waterfall', or with iteration to previous steps as an 'iterative waterfall' or the V model where hardware and software are co-developed for a system, Fig 2. These methods have been shown to be deficient for software engineering [4]. They fail to recognise the role of iterations in the overall process and the specifications are frozen at an early stage of the development process. The implication of the latter is that the cost committed to the program is large before the system concept has been adequately proven in terms of risk and performance. Iterations are to reduce risks, verify – are we building the product right?, validate – are we building the right product?, and test the outputs of each activity before proceeding to the next or to a previous activity to take corrective actions. Failure to do this results in validation late in the development process by which time corrective actions are costly as they propagate backwards through the process. For these reasons new methods have been developed for software engineering, and applied successfully, but have not as yet been applied to signal processing. A key method is the risk driven spiral model where risks are analysed, versus key criteria, at each step and the developments refined through successive iterations to eventually converge to the final solution [5].

### 2.1.1 The Iterative Development Process

ESPADON has, after careful analysis, adopted and modified this method and defined a new methodology for signal processing application development. This is shown in abstract form in Fig. 3 and applies to the development of the signal processing subsystem at the highest level. Central to the new methodology are the following key processes:

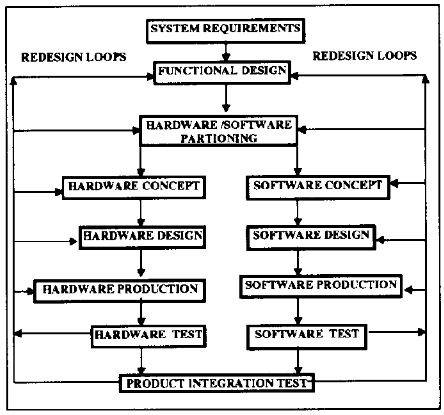


*Fig. 2. V Model of Development*

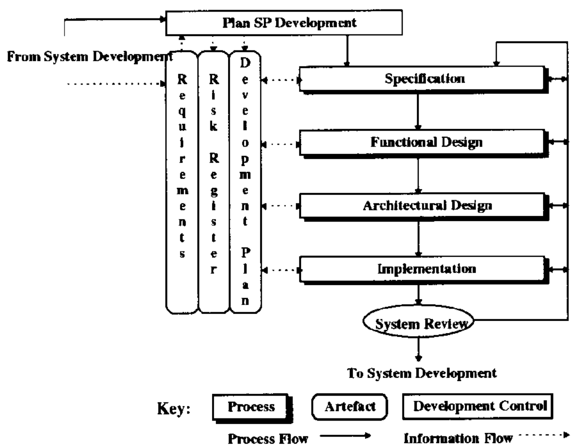


*Fig. 3 Abstract Iterative Development Lifecycle*

*Specification* – refinement of the raw requirements from the system development into an engineering specification that includes salient functionality, interfaces, physical attributes and performance and cost criteria.

*Functional Design* – the functional parts of the component specifications are modelled by assigning the appropriate algorithmic and control processing blocks, functional libraries and description models of computation, to a functional model. The model is independent of the implementation and is simulated to prove functional correctness or raise any corrective actions for further refinement of the overall process.

*Architectural Design* - The critical characteristics of the reference functional model (computing power, rate, etc.) and the non-functional requirements (costs, volume, etc.) are identified. A risk analysis is performed to determine the critical characteristics to be taken into account in identifying candidate architectures. Through trade off studies, the most effective architecture is chosen. If no appropriate solution can be found, the model and/or the system requirements are refined.

*Implementation* - the result of the current design iteration, a Rapid Prototype, a Virtual Prototype (section 2.1.3) or a Production Component. This process includes production and test of hardware and software, integration of the software on the target hardware and validation of the component. Co-design, Hardware/Software synthesis and co-verification are essential techniques to use in this process.

The other nodes shown in the diagram control these processes and the development lifecycle for the signal processing component being developed. Namely,

*Plan Development* – input is the requirements and the output is the plan and risk register

*System Review (Control Point)* – a system level review, with all the system design authorities, at the end of each complete cycle in accordance with the plan. Outputs are, a) exit with the results (the appropriate artefacts) to the overall system development team for integration with the system, or b) reiteration of the cycle with changes to the control artefacts as necessary.

The control artefacts are the *Requirements* – handed down from the overall system design process, *Risk Register* – severity and priority ordered list of current identified risks, and the *development plan*.

Each of the key processes above is itself composed of the generic abstract iterative process shown in Fig. 4 where the nodes either represent generic activities, described below, or the control artefacts described previously. These generic activities are the five phases that embody the ESPADON iterative design methodology:

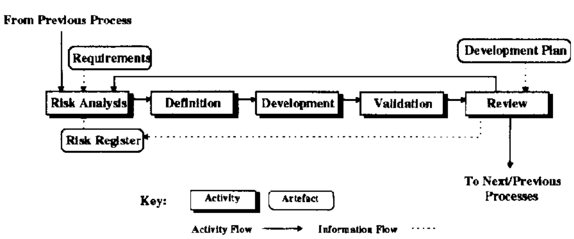


*Fig. 4 Anatomy of an Iterative Process*

*Phase 1: Risk Analysis* - analyse the requirements, any available process artefacts, the risk register and the development plan to determine what should be achieved in the current pass through this process.

*Phase 2: Definition* - the definition and documentation of the objectives for this iteration. This will include creating or updating any design and test documents and/or data involved.

*Phase 3: Development* - develop the object(s) (one or more of the design artefacts) of this iteration according to the definition made in the previous activity.

*Phase 4: Validation* - validate the object(s) produced by this iteration against the objectives and the component requirements using the defined tests. Analyse the results and update the risk register.

*Phase 5: (Exit OR Refine) Review* - review the requirements, any available design artefacts, the risk register and the development plan to determine what course of action needs to be taken next. Possible actions are: a new iteration of the same process (introducing new requirements or refining existing ones) or move onto the next process. If a new iteration of the process is required and this is not compatible with the current development plan, then a Development Review must be initiated and the plan updated.

Design artefacts, not shown in the diagram, will be produced and modified by the activities as the development iterations proceed. As the iterative process proceeds these artefacts will grow in content and become more refined. At the end these artefacts, with the control artefacts, will be the signal processing components complete design archive which can be reused for the development of similar components and mid-life technology updates.

The abstract iterative processes described above are equivalent to a spiral model for signal processing development at the component level, sub-spirals or fractals of the spiral model, or the signal processing development at the system level as shown in Fig. 5. Embedded within this development process are the key ESPADON design concepts defined in the next section.

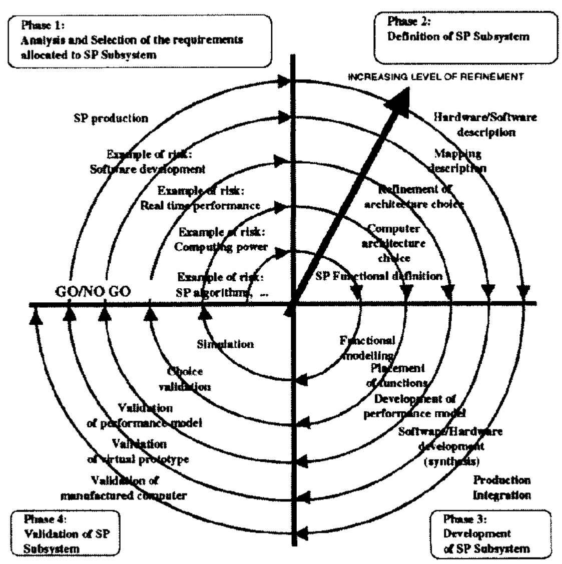


*Fig. 5 Spiral Model of ESPADON Development*

### 2.1.2 Reuse & Capitalisation

Reuse, along side the iterative development process, is the other element of the signal processing methodology implemented to decrease development time and cost. Reuse applies at two levels:

*Reuse between iterative processes of development cycle* - use elements developed in an iterative process with a certain level of refinement for the development of the next iterative process having a higher level of refinement. The strategy with reference to the generic iterative process is:

Definition activity - the same modelling formalisms or functional models are used at different levels of refinement but with dual libraries of components,

Development activity - hardware is synthesised and code is generated for different target machines with the same synthesis techniques. These targets may be, for example, a workstation or a real time multiprocessor machine according to the development stage,

Validation activity- the virtual prototypes of the previous iterative process are used as a reference for the virtual prototype of the next iterative process.

*Reuse of existing components (SP algorithms, components, hardware architectures, PCBs, etc.)* – use in-house components already developed or COTS components for the development of an activity (or an iterative process) of the development cycle. The development strategy is:

*Development with reuse* - development of an application must be able to reuse already-developed existing constituent parts.

*Development for reuse (or capitalisation)* - the new constituent parts of an application are developed in order to be reused in other systems.

The above reuse objectives are integral to the ESPADON development process and enables,

a) *increasing productivity and decreasing development time,*

b) *providing additional architecture choices,*

c) *using better quality constituent parts since they have already been tested and validated, and*

d) *capitalising on existing know-how.*

### 2.1.3 Rapid and Virtual Prototyping

An iterative development method necessarily implies the use of prototyping, at some level, such that requirements and functional solutions (the prototypes) can be validated and verified by measurement and improved through successive refinement to arrive at the final solution. The value for complex systems development was recognised in software engineering a few decades ago and high level environments to support prototyping, and faster iterations of prototyping (rapid prototyping), developed [6].

*Rapid Prototyping* - Unlike software engineering where the functions are compiled and executed to run on a workstation, signal processing requires the functional solutions to be partitioned, mapped and implemented on the embedded multi-processor hardware for meaningful performance measurements and validation. For conventional developments this is a specialised and expensive activity as the code is hand mapped and hand crafted for optimum performance on, as explained earlier, rapidly obsolete custom computers. Instead we need a prototyping environment that is fast and can support the insertion of the available commercial technologies based on COTS boards or COTS computers integrated with any necessary proprietary hardware (I/O, display etc.). The rapid prototype will enable the functionality to be properly tested in terms of dependencies, performance and real-time behaviour. This can be applied to any signal processing component development and associated requirements. It provides an opportunity for the early and frequent involvement of the customer to refine the requirements and common understanding of a signal processing component or iterations of the signal processing system. Rapid prototyping for signal processing is therefore the ability to seamlessly move from the functional design to the architectural design *(the modelling & simulation domain)* to the implementation, through automatic code generation, on real-time COTS test beds *(the execution and measurement domain).*

Note that the prototyping is an iterative development process where the results are used to refine the successive iterations as per the generic development processes described earlier. Clearly as the iterative process proceeds, performance and behavioural data are amassed, the functional models grow in content and become more refined. These successive prototypes, together with their associated functional models and performance and behavioural data, provide the basis for virtual prototyping.

*Virtual Prototyping* - is the ability to model and simulate in *the software domain*, the complete signal processing application, including hardware at different levels of abstraction, to validate the architecture selection prior to technology implementation. Rapid prototype measurements and information feeds into the virtual prototype, which enables component and system libraries and data bases to be built so as to construct virtual models of signal processing systems for iterative cost/performance and other trade off and analysis studies. Integral to the studies is the concept of hardware and software co-design discussed next.

*Co-design* – this is implied in the prototyping described above but has particular relevance to virtual prototyping as follows. Co-design is defined as the concurrent and co-operative design of information processing sub-systems composed of hardware and software components operating together. It is central to the iterative prototype developments discussed earlier. In the traditional 'V' model (Fig. 2), the hardware and software developments are partitioned early in the development lifecycle. Hence they diverge in terms of engineering or design interaction, and cross-validation, until the integration, test and validation phase. This phase however is much further downstream leading to potentially costly (time and money) reengineering of solutions, often in software as the hardware is by then fixed, to overcome deficiencies with respect to the initial requirements and specifications (as per the issues in Section 2.). Co-design provides a method to overcome these deficiencies by closely coupling the hardware and software developments within an iterative design framework. The main phases are shown in Fig. 6. The important points are that the design starts with a system or sub-system *specification and functional model.* This specification may be independent of the future implementation and the partitioning of the system into the hardware and software components. The specification has to be captured in a functional model that can be simulated and verified. This model is *partitioned* into hardware models and software models that make up an overall architectural model of the system (the virtual prototype discussed above). These models will at the lowest level be described in high level languages such as C and VHDL and at the highest level be described by graph based objects.

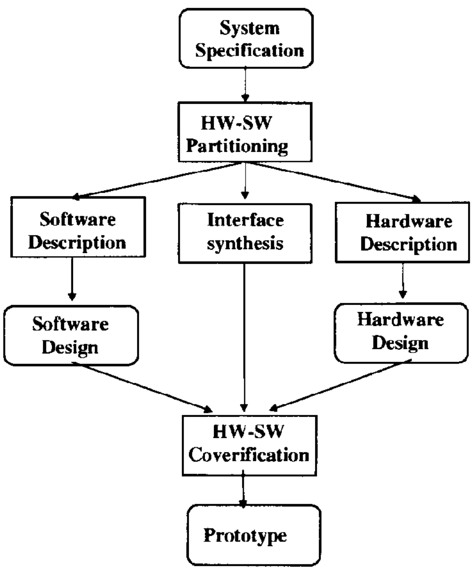


*Fig. 6 Co-design Process flow*

With the emergence of large reconfigurable and reprogrammable devices (>> Millions of Gates), and system on a chip (SOC) devices, co-design offers a very powerful technique for encapsulating by design software functionality onto hardware devices through partitioning studies and trade-off studies so as to arrive at an optimal architecture. Because it is model based, it is easier to modify and refine the models and architecture for the latest implementation in the succession of prototypes.

Hence the co-design methodology provides the ability to model the system specification, to model the architecture solution and to perform trade-off studies (performance, cost, power consumption etc.). A key application in signal processing application development is in the partitioning and mapping of time and performance critical signal processing functions, that would otherwise run on COTS general purpose processors, onto COTS or bespoke FPGA or SOC arrays.

### 2.1.4 The Model-Year (MY) approach

The concepts of rapid and virtual prototyping for signal processing are fundamental to the MY architecture concept developed under RASSP [7] and is integral to the ESPADON iterative development methodology. A MY approach expects that the signal processing system can be fielded with the latest digital technology in less than a year if the architecture has been developed iteratively through a succession of prototypes. In fact the MY concept is to deal with obsolescence and provide systems with the latest COTS digital technology when fielded. Key attributes of the MY concept are;

a) *the MY mitigates the risks of the development of an equipment by rapidly validating its requirements through a succession of prototypes (Rapid Prototyping); and*

b) *the implementation of a MY architecture of the signal processing application uses the available digital technology.*

Indeed, instead of developing expensive and rapidly obsolete custom computers, the rapid prototype integrates available commercial technologies based on COTS boards or COTS computers. On the other hand, the final equipment, taking into account the constant digital component improvement, is developed with the latest technology. Therefore with the MY architecture, a retrofit of the equipment due to an obsolescence of components is only another iteration in the life cycle of the equipment.

These iterative technology insertions are shown in Fig. 1 as the 'ESPADON technology staircase'. The signal processing system prototypes are refreshed with the latest COTS technology at regular intervals which in practice will be determined by the planned refresh rates for the pre-production, delivery, and post production phases of the signal processing system as part of the iterative development methodology.

## 3 THE ESPADON DESIGN ENVIRONMENT (EDE)

Having defined an ESPADON methodology and development process, the next technical development was the ESPADON Design Environment (EDE) to support this new method [8]. Figure 3 described earlier, shows the key development activities which need to be supported by the EDE. For each, the technical requirements, pertinent techniques, and scope was identified and defined. Technical studies were undertaken to provide up to date information on key techniques such as software synthesis, hardware synthesis, rapid and virtual prototyping, libraries, tool interfacing techniques, etc. Each study summarised the current status of the technology areas and potential

COTS tools that were available to support it. These COTS tools were evaluated further as described below.

## 3.1 COTS Tools Selection

Having identified the potential COTS tools in the domain areas of interest, a tool selection process was defined [8]. As part of this process, and to ensure consistency, a tool function coverage grid, Fig. 7, matching the methodology requirements was designed and used to rank the tools in each domain [9]. Non-functional requirements, not shown, were also added to the assessment. In parallel, a vendor questionnaire, consistent with the grid, was also sent to the vendors for completion and the results used to update the ranking. The key factors that were taken into consideration for the ranking were; *1) Commercial factors (size of company, licence costs, history, support etc.), 2) Features and functionality support, 3) Interface with existing and to future tools and designs, 4) Methodology support, and 5) Usability*

PROGRAMME : ESPADON — Tool Name : SPW 4.1
Creation Date : 4/3/99 — By : I.D. Alston — Reference :
Tool Contact Details : Supplier ALTA Group of Cadence Design Systems Inc. — Contact Telephone: +44 (0) 1344 360333

| Activity / Process | Risk Analysis | Definition | Development | | | Validation |
|---|---|---|---|---|---|---|
| | | | S/W Synthesis | I/F Synthesis | H/W Synthesis | |
| Specification | | | | | | |
| Functional Design | | Yes | Yes | Yes | Yes | Yes |
| Architectural Design | | Limited | Limited | Limited | Limited | Limited |
| Implementation - Rapid Prototyping | Yes | Yes | Yes | Yes | Yes | Yes |
| Implementation - Virtual Prototyping | Yes | Yes | Yes | Partial | Yes | Yes |
| Implementation - Production Standard | Partial | Partial | Partial | Partial | Partial | Partial |

*Fig. 7 Tool Function Coverage Grid*

Following the collation and analysis of the results, the best of class tools were selected for detailed evaluations with representative test applications [10]. As the first release of the EDE is directed towards functional design and rapid prototyping, the detailed evaluation stage has concentrated on these domains only at this particular stage in the project. The selection process for co-design, co-simulation and virtual prototyping tools has just commenced.

From the results of the evaluation, the best of class tools for the first release of the EDE, and rapid prototyping are:

- GEDAE [11]Currently technically the best of class tool. It is also the recommended tool from the RASSP programme.
- Ptolemy [12] An extensive research & development software suite covering many domains of signal processing and considered to be the father of signal processing simulation tools. It is research quality software, the output of many students and many years of research at the University of Berkley. It is free open source software available directly from the University.

In addition the following has been selected for mathematical work, algorithm development and prototyping.

- Matlab/Simulink [13] Matlab is widely used among project partners

Other than the above tools, the evaluation studies also recommended the use of signal processing libraries and standards and associated APIs, for example for algorithms and communications, to support reuse and capitalisation and provide tool independence for the future. ESPADON has therefore evaluated the following standards;

*Vector Signal Image Processing Library (VSIPL)* [14] – This standard is being developed by representatives from Industry, with representation from ESPADON, and academia with the goals to:

- Catalyze the formation of an Industry Standard Working Group for Vector/Signal/Image Processing Libraries.
- Create a widely (industry) supported standard API/library for vector/signal/image processing primitives.
- API/Library for single processor and parallel version.
- Foster standardization for sensor software portability such as reuse, interoperability, low cost COTS upgrade path, lower life cycle costs, etc.

ESPADON is adopting the VSIPL API, and investigating the efficient implementation of the VSIPL standard on the ESPADON target platforms and future evolutions, so as to enable reuse and capitalisation of the algorithm developments. These developments are focussed towards application domain libraries, such as for Radar and Sonar.

A draft of the VSIPL standard has been written and has been distributed for final comments and approvals by the VSIPL core members.

*Message Passing Interface Real-Time (MPI-RT)* [15] – Inter-process communications (IPC) are the glue that binds processing in the ubiquitous multi-processor embedded signal processing systems. An IPC standard offers the potential for code portability, and hence reuse. MPI-RT is such a standard, and like VSIPL, is being developed by representatives from Industry and academia.

MPI-RT is neither a subset nor a superset of MPI or MPI-2 but part of the process to develop a message passing interfaces standard for real-time applications. It has been developed as a middleware API standard to support the real-time paradigms of TIME-DRIVEN, EVENT-DRIVEN (low level and high level), PRIORITY-DRIVEN processing. The Quality Of

Service (QoS) is a key attribute in each case. In fact the delivery of the QoS is central to the MPI-RT philosophy.

The adoption of MPI-RT by ESPADON raises many issues that need to be investigated further. These are concerned with the delivery of the QOS and the efficiency of implementations. Since it is a standard, its implementation is left to the systems or hardware vendors. At present, to the authors knowledge, no such implementations are available for detailed study, except for emulation on a workstation, though most of the major vendors do have MPI-RT in their future road maps. Hence ESPADON is keeping a watching brief on vendor's developments and investigating how the MPI-RT API may be implemented within ESPADON to provide a possible interface to future implementations. A draft standard for MPI-RT has been issued and is available on the web [15].

The other tools required for the EDE are more concerned with the infrastructure, requirements, cost estimation, EDE control and configuration management. The final list of tools selected for the whole EDE are shown in Fig. 8. These additional tools are not critical to the success of ESPADON but need to be interfaced to the EDE to support the overall signal processing application development lifecycle.

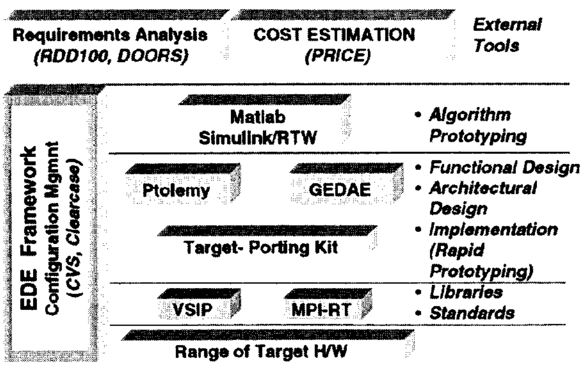


Fig. 8 EDE Tools Selected

## 3.2 The EDE Framework

The ESPADON Design Environment (EDE) consists of the tools and libraries connected through the EDE infrastructure as discussed above [16]. A technical working group has been established on the project to progress the EDE development through the various revisions, starting with 0.1, the first realisation of a Rapid Prototyping framework, 0.2, the first update, after a 'hands on' evaluation and review by benchmarking teams, and 1.0, the first realisation incorporating Virtual Prototyping. The requirements placed on the EDE are that it is based on :

- a modular approach consisting of standard interfaces etc., an open architecture, and basic services (e.g. key elements).
- existing capabilities/tools: e.g. virtual prototyping tools provide capabilities of co-design, co-simulation and co-synthesis,
- low intrusion into existing tools

It should be portable and extensible and provide functionality that can support the following key attributes:

*Simplify tool usage - the new user should have a gentle learning curve*

Familiar GUI (rather than command line); On line manual pages – tool selection, usage and style guide; Common tool start up procedures including user profiles.

*Make tools appear more 'professional' - some tools have an academic/research origin*

Login security, tool usage trace logging; Data security, backup, archive; Design deposition - change control, code management; IP data/design repository, capitalisation and reuse; Multi–user support.

*Multi tool management and data exchange - some difficult problems will require the use of several tools at once*

Concurrent use of tools for co-design and co-simulation; Sequential use of tools - avoid manual re-entry of intermediate design data.

*Tool automation – in some cases existing UNIX or NT tools can be used but they may need a wrapper*

'Scripting' language for driving low level tools.

Other than these attributes, the EDE has to clearly support the iterative system development methodology of ESPADON. Consequently there are three specific viewpoints (users) which govern how the functionality of the framework is accessed and by whom. These are self-explanatory, strictly hierarchical, and are the System Viewpoint (the overall signal processing application composed of a number of component developments assigned to particular project groups), the Project Viewpoint (the signal processing component developments being undertaken by a project group), and finally the User Viewpoint (one of the project group members undertaking a specific task).

To support the above the key elements, *Graphical User Interface, On-line guide, Tool Management & Control, Repository, Data Exchange, and Trace Information* were identified and designed to build the first version of the EDE . The GUI is shown in Fig. 9 with the trace window that records information useful for collecting metrics and the history of the development.

## 3.3 V0.1 Version of the EDE

This version has been integrated with the GEDAE tool and the Ptolemy Tool and will be used for the benchmarking of an example Sonar application and example Radar benchmarking respectively [17]. The choice of the two tools is deliberate so as to provide performance and efficiency measurements for cross-comparison and mutual improvements. The main objective however is to benchmark the V0.1 version for rapid prototyping by using two representative applications.

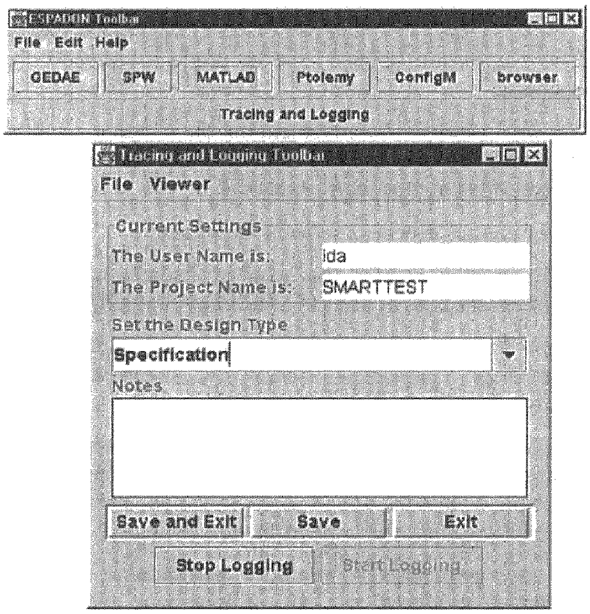


Fig. 9 EDE User Interface

The Ptolemy tool is being ported to a Mercury platform for the benchmark. The GEDAE tool supports a number of target platforms (Mercury, Ixthos, Sky etc.) but is being ported to support a subset of the EUROPRO platform [18]. Key to both is the board support package for the target architecture, its adaptability to support other targets and the overall efficiency. Hence work is underway with GEDAE to support commonly used real-time operating systems such that additional processors (RISCs and DSPs) can be supported. This will enable a board porting kit to be developed to support a range of hardware test beds and potentially heterogeneous systems. An example of the design flow with GEDAE is shown in Fig. 10.

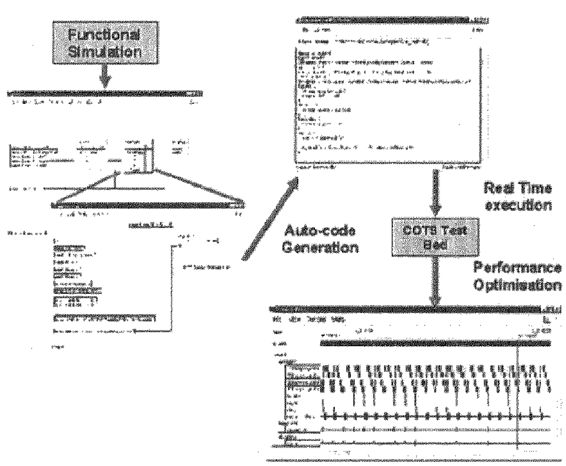


Fig. 10 Typical Rapid Prototyping Design Flow

The first application of the EDE is towards the benchmarking of the ESPADON methodology and development process. Two benchmarks were identified at the outset of the project. These are the implementation of a beam-former for a Sonar and a Radar applications (see Fig. 11). Beamformimg is a generic processing function for which metrics for conventional developments are known or can be estimated. A technical document defining the rationale for, and the definition of the metrics to measure the ESPADON objectives has been written and a benchmarking plan drawn [19]. An overview of the radar benchmarking application is provided in the next section.

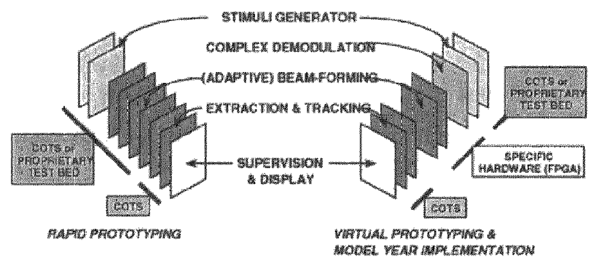


Fig. 11 Outline of the beamformer benchmark.

### 3.3.1 The Radar benchmarking application [20]

For the Radar benchmark, an adaptive digital Beamformer (BF) application for multibeam radar, Fig. 12, will be used.

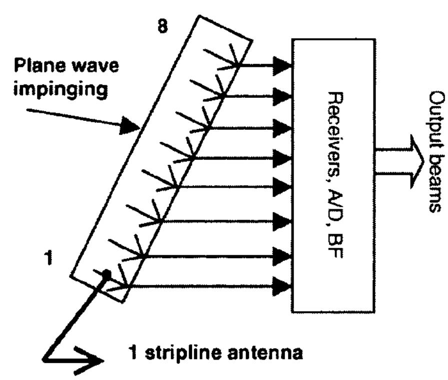


Figure 12: Multi stripline receiver antenna array signals are transformed into a beam pattern in elevation

The function beamformer is part of the functional chain of an X-(H new NATO) band air surveillance radar. The antenna of the radar comprises a vertical array of, for example, 8 elements each of which is a horizontal linear stripline array of dipoles. The array is used as a transmit antenna as well as a receive antenna. As a transmit antenna the power splitter distributes the RF power among the elements (linear arrays) via phase shifters and circulators. This results in a transmit beam which illuminates targets within the desired elevation coverage envelope. As a receive antenna, each of the 8 array elements is connected directly to an individual receiver and an A/D converter. Each array element is sensitive over the desired elevation coverage. Elevation beams are formed by the digital beamformer that performs an 8 point FFT or FIR algorithm on the outputs of the 8 receiver channels. In this way a multibeam receive system is formed, Fig. 13. The benchmark concerns only the receiver beamforming function, the transmit beamforming function is implemented by an analogue system.

The beamformer is adaptive with respect to the ships course and speed, and the ships roll and pitch movement. This results in a phase correction that is applied to the complex data stream prior to the beamforming, together with windowing and calibration correction.

The application contains all aspects of a radar signal-processing element as it is found in modern radar systems nowadays. This includes mode switching, synchronous/asynchronous data and control flow.

Adaptive beamforming is characterised by high data rates (up to 20 Mbytes/sec for each channel/beam) and corner turn processes. The signal processing architecture on which the algorithm will be implemented therefore asks for high-end multi-processor machines with high-speed crossbar interconnect between processing nodes. The selected crossbar interconnect has a peak throughput of 267 MB/s per crossbar connection and also gives the desired flexibility needed for rapid prototyping in the sense of ESPADON. For the processing element the 4th generation Motorola PowerPC processor is selected: the AltiVec processor. This processor is similar to the previous version of the PowerPC with a 128-bit vector-processing unit added, which is well suited for signal processing algorithms.

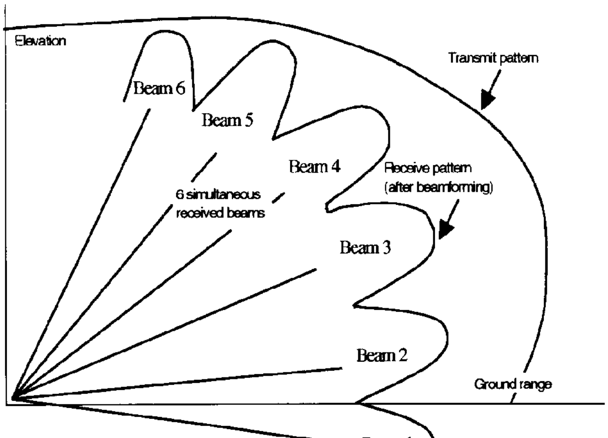


Figure 13: Example of resulting multi beam pattern in elevation for an eight channel to six beam beamformer

### 3.3.2 Principle Benchmark Metrics

As per the iterative development methodology, the benchmark of the ESPADON process will also be carried over successive iterations. For each of the benchmarks, the principle metrics directly related to the performance of the ESPADON process and performance will be collated. These in summary are:

*Design Cycle Metrics* – the reduction in development time, through software and hardware reuse, productivity, iterative refinement etc.

*Product costs* - the reduction of the development costs. These costs are defined cost to produce and cost to support.

*Product quality* - improvement in the product quality measured by the number of hardware and software defects, the time to repair, and MTBF.

Other metrics deeemed to be important are:

*Tool oriented metrics* -the level of integration of the tools and the ease of use and uniformity of the EDE.

*Application complexity metrics* – try to capture the benchmark application complexity, independent of hardware and software implementation

*Product complexity metrics* - for each product, for example, software, hardware, and documentation, complexity metrics are required to weight the product efficiency against the implementation difficulty

*Product performance metrics* - performance of the products produced is not synonymous with the ESPADON performance itself. Hence it is important that the appropriate metrics are collected and analysed.

These metrics will be collated as part of the benchmarking activity which will be carried out for each of the three releases of the EDE. The first step will be to evaluate the preliminary version of the EDE (V0.1) described above. The results will be fed back to improve the tools, the EDE framework and the integration. These steps will be repeated for the next version of the rapid prototyping EDE and then proceed to the virtual prototyping EDE.

The final release, Virtual Prototyping version, will not be benchmarked against two applications, and by two teams, but against one benchmark application (Radar or Sonar) and one benchmark team. This is expected to be sufficient to demonstrate the concept and advantages of the virtual prototyping process. Virtual prototyping is a complicated concept to disseminate in a production environment and to find suitable reference baselines to compare against.

## 4 CONCLUSION

The ESPADON project expects to significantly improve reduced cost and timescales, the process, by which complex military digital processing systems are designed, developed and supported. A new design methodology, and a new development environment, has been reinvented to support this through reuse, concurrent engineering, rapid insertion of COTS technology and the key concepts of rapid and virtual prototyping as described earlier. The key attributes of the methodology are a Risk driven spiral lifecycle, encapsulation of the

"Model Year" concept to mitigate risks by the iterative development over successive rapid prototypes integrated with the latest COTS technology, and support for component Reuse and Capitalisation.

The preliminary version of the EDE to support the methodology has been implemented, with the GEDAE COTS tool, and supports the concept of Rapid Prototyping. Key features are the data flow signal processing paradigm, the EDE framework and GUI, the support libraries, and the efficiency of code generators (communications and processing). The first EDE is targeted for a range of real-time COTS test beds, the first being a Mercury system. A benchmarking process to evaluate the EDE and provide valuable feedback towards its improvement has begun. It will enable real behavioural, performance and timing measurements to be made to feedback into the iterative process so as to arrive at an optimum implementation.

Such an EDE and Rapid Prototyping environment provides a number of advantages for signal processing application development. It enables the collection of measurement data to provide as an input to virtual prototyping. The performance data can be used to correctly size the overall system requirements (hardware and software). Data can be collated with respect to benchmarking other COTS components. The prototype can be used with real data or in the field to validate the processing. It enables the early and frequent involvement of the customer so as to adjust requirements over the development and field experimentation stages. These advantages offer a significant improvement compared to the conventional methods for signal processing application development.

## 5 ACKNOWLEDGEMENTS

The work has been carried out under the WEAG EUCLID/Eurofinder programme, Project RTP2.29, with support from the UK, French and Dutch MoDs and the participating companies. The authors are grateful for this support and would also like to acknowledge the contributions of all the ESPADON team members.



## 6 REFERENCES

[1] D. Aulagnier and B. Saget, "The ESPADON programme: Environment for Signal Processing Application Development and prOtotypiNg", Actes des Journees Thematiques Universites/Industries du GRAISyHM-AAA-99, Lille, france, 23-24 March, 1999.

[2] D. Aulagnier, J. Hunink and D. Muller, "The ESPADON programme", Proceeding of RADAR' 99 International Conference, Brest, France 17 –21 May' 99.

[3] M. A. Richard et al, "The RASSP Program Origin, Concepts, and Status – An Introduction to the Issue", Rapid Prototyping of Application Specific Signal Processors, Jnl. Of VLSI SIGNAL PROCESSING SYSTEMS for Signal, Image and Video Technology, Kluwer Academic Publishers, Vol. 15, 7, Feb, 1997.

[4] G. R. Gladden,"Stop the lifecycle – I want to get off", ACM Software Engineering Notes, Vol. 7 (2), 35, 1982.

[5] B. W. Boehm,"A Spiral Model of Software Development and Enhancement", Computer, 61-72, May1988.

[6] R. Balzer, "A 15 year perspective on automatic programming", IEEE, Trans. Software Eng., Vol 11, 1257, 1985.

[7] J. Pridmore et al, "Model-Year Architectures for Rapid Prototyping", Rapid Prototyping of Application Specific Signal Processors, Jnl. Of VLSI SIGNAL PROCESSING SYSTEMS for Signal, Image and Video Technology, Kluwer Academic Publishers, Vol. 15, 83, February 1997.

[8] I. Alston and B. Madahar, "The Tool Selection Process for the ESPADON Design Environment", Proceedings of the BAE SYSTEMS Conference on Signal & Data Processing, 1-3 March, 2000, BAE SYSTEMS Research Centre, Gt. Baddow, CM2 8HN, England, Report No. YD/000140, March 2000.

[9] ESPADON Programme Deliverable, 'ESPADON: COTS Tools Assessment Summary', BAE SYSTEMS Research Centre, Gt. Baddow, CM2 8HN, England, Report No. YD/992197/D1348, Issue 2, 10th January 2000.

[10] ESPADON Programme Deliverables, 'ESPADON: Tool Evaluation Reports', BAE SYSTEMS Research Centre, Gt. Baddow, CM2 8HN, England, Report Nos. YD/992344/D1348, YD/992375/D1348, SIGNAAL 9501 114 059, Thomson 215 PV 0165, February 2000.

[11] GEDAE, Lockhheed Martin ATL, USA, http://www.gedae.lmco.com/.

[12] Ptolemy, University of California, Berkeley, USA, http://ptolemy.eecs.berkeley.edu/.

[13] Matlab, The MathWorks Inc., http://www.mathworks.com/.

[14] VSIPL, http://www.vsipl.org/.

[15] MPI-RT, http://www.mpirt.org/.

[16] ESPADON Programme Deliverable, 'ESPADON: System/Segment Design Document', Thomson-CSF DETEXIS, 1 Bld Jean Moulin, 78852 Elancourt CEDEX France, Report No. TBU.TN/ELA/ETNI, 98/1104/SSS, 28th June 1999.

[17] Alston and B. Madahar, "Rapid Prototyping", Proceedings of the BAE SYSTEMS Conference on Signal & Data Processing, 1-3 March, 2000, BAE SYSTEMS Research Centre, Gt. Baddow, CM2 8HN, England, Report No. YD/000140, March 2000.

[18] EUROPRO Consortium, EUROPRO Final Report, Project P21040-HPCN/EUROPRO, V1.1, Thomson Marconi Sonar, Sophia Antipolis, France, May 1999.

[19] ESPADON Programme Deliverable, 'ESPADON: Validation Plan - Definition of Metrics and Reference Baselines', Signaal, Zuidelijke Havenweg 40, P.O. Box 42, 7550 GD Hengelo, Netherlands, Report No. 9501-113-932, October 1999.

[20] H. Schurer and J J. Hunink, "Rapid Prototyping of Radar Signal Processing Algorithms", IEEE, ProRISC 99 Conference Proceedings, ISBN 90-73461-18-9, 405-412, October 1999.