Managing the Embedded Systems Development Process with Product LifeCycle Management

Eliane Fourgeau, Emilio Gomez and Michel Hagege

Abstract In industries like Transportation, Aerospace and Defense or High-Tech, the effective development of complex systems implies mastering multiple disciplines and processes: project, people and data governance processes, requirements based and model based system engineering approaches, System Architecture Design with trade-off analysis, cross-disciplinary detailed design, integration, test, validation and verification methods. These processes are interrelated; managing complexity calls for tight collaboration between development stakeholders and demand very high data consistency between system compositional elements, at all levels. Organizations face significant challenges to maintaining coherency, consistency and relevance of this information across domains as well as capitalizing on it from one complex project to another; Failure to do so though is often resulting in costly reworks, product recalls, worse: attractiveness and competitiveness drops and jeopardized growth. In today's highly competitive, fast changing and demanding markets, saving investment time, resources and money for the creation of higher market appeal products is vital. This presentation will outline a vision for an open and integrated System Engineering platform, that leverages the richness and efficiency of online communities, the effective composition and reuse of multi-disciplinarily assets and the value of virtual user experiences, to facilitate new smart E/E systems development. Architecture of Systems is therefore just a unified and homogenous framework that allows to see all these different architectural traditions as instances of a same and unique "abstract" discipline. This system-oriented discipline clearly emerges nowadays from the convergence that one can presently observe at the level of the scientific and methodological foundations of many architectural practices such as: Enterprise Architecture that allows to design integrated enterprise information systems, Software Architecture which is at the basis

E. Fourgeau · E. Gomez (✉) · M. Hagege
Dassault Systèmes, 10 Rue Marcel Dassault, 78946 Vélizy-Villacoublay, France
e-mail: Emilio.Gomez@3ds.com

E. Fourgeau
e-mail: Eliane.Fourgeau@3ds.com

M. Hagege
e-mail: Michel.Hagege@3DS.com

M.-A. Cardin et al. (eds.), *Complex Systems Design & Management Asia*,
Advances in Intelligent Systems and Computing 426,
DOI 10.1007/978-3-319-29643-2_11

of all software design processes, Systems Architecture that is key for designing efficiently complex industrial systems, resulting from software-hardware integration, Organization and Project Architecture which underlies agile development methods. Architecture of Systems is therefore just a unified and homogenous framework that allows seeing all these different architectural traditions as instances of a same and unique "abstract" discipline, which underlies agile development methods.

1 Introduction

For more than 20 years, manufacturers and suppliers of automobiles, buses, trains, planes, satellites, defense systems, industrial equipment, complex communication systems and more have relied on the power of 3D CAD solutions to help achieve concurrent 3D development, seamless integration and effective digital mock-up creation, at the same time as they implemented Product Lifecycle Management (PLM), establishing a single source of product data and product record and a collaborative repository providing CAD, manufacturing, and service departments a centralized and controlled environment to house product data (Fig. 1). In this context, product data generally refers to all the information associated to product specification, CAD models, drawings, test procedures, documentation, operating procedures, quality documents, compliance documents, assembly instructions, and service manuals.

Fig. 1 OEM's landscape

Simultaneously, in most industries, an explosive growth in electronics and networked computerized systems with more and more software has been developing and experts predict this growth in embedded systems complexity will continue unabated. This mounting sophistication is accompanied with increasingly focused customer demands for regional, customized products based on market segmentation worldwide, including new emerging markets in India, China, and South America. This is no easy task to deliver, given the ongoing trend to execute the design and production of products, manage complex product lines and product variants with reduced resources and numerous specialists, often located around the globe.

As a result, both OEMs and suppliers are under intense pressure to improve the efficiency and effectiveness of how they design and build products. OEMs must deliver what the consumer wants, suppliers what OEM requests, when and where they want it, at an acceptable price; to achieve this, they need to collaborate across all departments—and across continents—and create innovative products that meet their customer needs as well as comply with industry regulations.

Only a profound evolution of the industry's infrastructure, processes and core technologies can meet this demand for accelerated innovation and help manage multiple complex product lifecycles. Reduced time to delivery and reduced budgets permeate the competitive landscape calling for new solutions that support distributed product development with global collaboration between stakeholders, not only for 3D disciplines but also for the complete System Engineering including embedded systems development processes.

PLM must expand beyond physical data and manufacturing, and include everyone with a role in a product's life—from end consumers, to internal functions developers, functional architects, embedded systems developers, tests engineers, integration teams, manufacturing and post sales staffs.

Accelerating innovation with PLM 2.0.

Despite the hesitation and general desire of Embedded System engineers to be left alone by those telling them, "we're from corporate IT, and we're here to help," they usually hate wasting time and effort. After all, time is money, and due to the pervasive aspects of Embedded Systems, embedded system engineers are under tighter deadlines and more stringent constraints for higher quality, faster pace development cycles, and cost effectiveness:

- Accelerate innovation introduction and improve quality, in line with end users product experience and use cases expectations.
- Deliver environmentally compliant products that meet global market demands.
- Ensure delivery of projects within performance and timing guidelines.
- Enhance global collaboration at multiple sites and throughout supply chain.
- Master the entire Embedded System development workflow, including requirement gathering and authoring, functional architecture optimization, logical decomposition, trade-off implementation for Software and Hardware, unit tests and integration tests, documentation,… and more.

- Manage end-to-end software architecture and systems engineering processes, including functional analysis, logical systems design, multi-physics behavior modeling and simulation, configuration and change management.
- Integrate multiple domains and bring together contributors from diverse technical disciplines to provide a high-level, comprehensive definition of a product.
- Validate the designed Systems as early as possible in the development process, when changes are least expensive to make.
- Produce the safest, high-performance systems with comprehensive requirements traceability.
- Ensure systems optimization including Software architecture, Hardware and Communications optimization.
- Manage engineering change order effectively.
- Reuse design/production data and preserve methods and processes across projects/programs.
- Provide accurate, real-time reporting of program status.

Why then should they care about PLM?

Of what value is a PLM based development paradigm?

How should this PLM based approach look like to contribute a great impact on embedded system engineering productivity, quality of designs, and time to market?

To answer the above questions, one should look at some aspects of today's reality for engineering communities:

- What about spending days trying to find development artifacts (requirements, functions, models, code, tests, documents, etc.…) previously used in a project and finally giving up and redesigning new ones?
- What about trying to push for module based developments to leverage reuse while consistent data consolidation at module level is almost impossible?
- What about searching for the latest test results or the contracted product requirements from the customer, or trying to consolidate all tests run to validate the given function on all HW and SW product parts involved to fulfill the function?
- What about tracing requirements throughout all development artifacts data, tracking engineering change orders across the development cycle and analyzing change impacts on the different development teams' activities? Even though many companies use a formal change process to control changes to product data, as many engineers know, change processes aren't always as optimized or efficient as they could or should be, especially when they have to manage data changes across various heterogeneous files authored by as many independent, non-interoperable authoring tools. Some engineering change processes have become so bloated and time consuming that the change process itself becomes more arduous than the design process. How many engineers are still walking the engineering change through to each approver to ensure it gets priority attention? How many times are embedded systems engineers and others downstream, such as test engineers, waiting for the change before they can complete their tasks, address a quality issue, or respond to a new customer request?

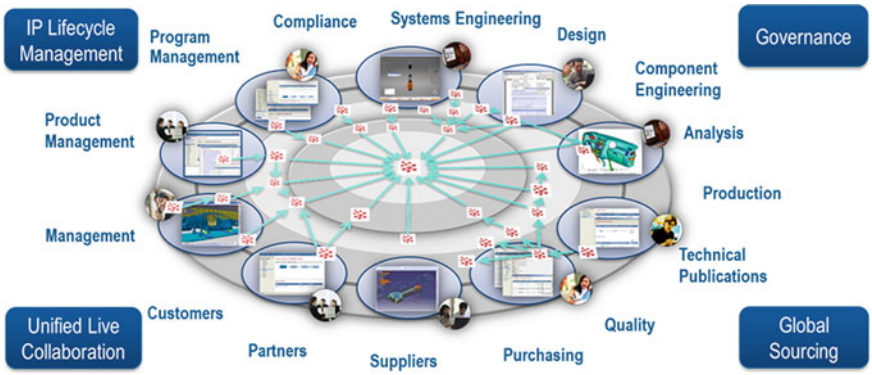


Fig. 2 PLM actors

Lack of real-time, comprehensive, efficient and accurate access to product data and customer/end user expectations has a great impact on engineering productivity, quality of designs, and time to market.

To achieve these goals, increase productivity, meet timelines and innovate efficiently, companies must master disparate sources of globally distributed intellectual assets and facilitate open, frequent, streamlined collaboration across the enterprise and extended enterprise and into the marketplace.

The goal of a PLM based system engineering development approach is to provide the development teams, including fast growing embedded system teams, unambiguous and easy access to all product and product variants data at its latest release level, to ensure on one side that everyone is sharing and is working with the most up-to-date information, on the other side that enterprise knowledge is capitalized across engineering departments, structured in generic modules for reuse in different projects as much as possible (Fig. 2).

Hence, there is a need for a single platform that consolidates product-related information, throughout the development process steps, making it visible and actionable by participants from all areas of the enterprise and when relevant of the extended enterprise. Powered by a PLM of second generation "PLM online for all", this platform must allow users everywhere to work concurrently in real-time on the same data, via a simple web connection to a single server.

Product data is managed through a series of defined lifecycle states, baselines, configuration and release levels. User access and IP are secured and controlled according to user, group, and role levels, from everywhere, at every time. Finally, the platform provides tracking and reporting tools to monitor the collaborative development effort progress. All these program management capabilities are woven into an easy to access environment of reduced risk and increased effectiveness.

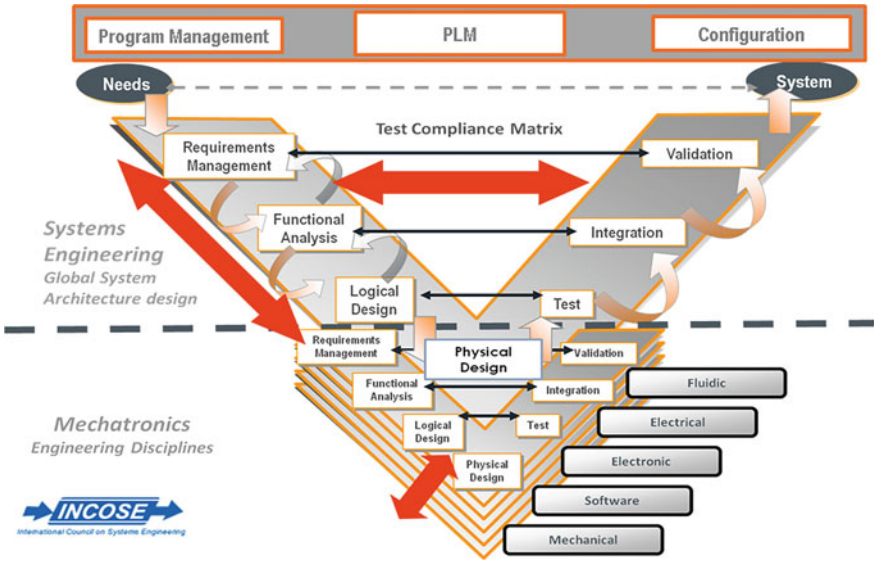


Fig. 3 PLM with MBSE scope

2 RFLP Data Model for Forward Looking Data Composition

The development of embedded systems involves multiple processes and disciplines (Fig. 3)—from project management and requirements engineering, through to configuration, mechatronics, HW and SW development, integration, test and verification management. The inter-disciplinary aspect of E/E and mechatronics development remains a challenge to all product developers, and especially for the sophisticated systems found in our today's life, making tight collaboration among the different design specialists mandatory.

These processes are interrelated and lead to a huge amount of systems data and models. Organizations face significant challenges and costs maintaining the accuracy and consistency of this information across all domains—with failure to do so often resulting in costly rework or even product recalls (Fig. 4).

To support effective embedded system development, the PLM based development approach calls for an innovative data model supporting seamless composition and easy reuse of complex product data, encompassing for a given product, system or sub-system, requirements and specifications documents, functions and functional interaction chains, logical compositions featuring HW and software trade-off architectures, eventually 3D Physical data and executable SW.

This is the so called RFLP (Requirement, Functional, Logical, and Physical) data model which in essence, helps guide the embedded system engineering teams chartered with the overall design (Fig. 5).

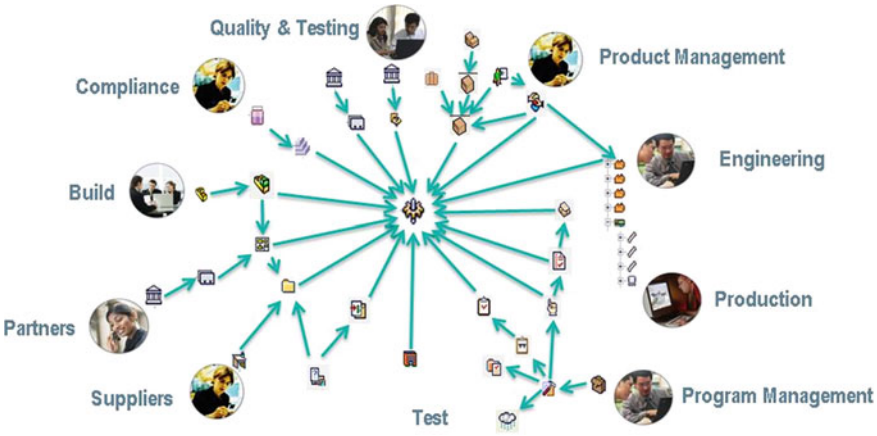


Fig. 4 Actors around PLM component

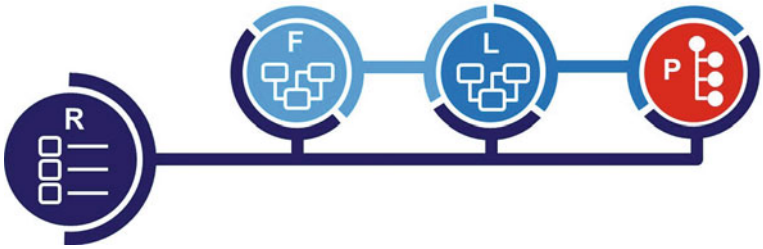


Fig. 5 RFLP framework

The Requirement and tests (R) defines customer and stakeholders needs and as the results of their analysis in technical terms: what characteristics, activities the system shall satisfy. The test defines the requirement verification protocol, the acceptance criteria, and the results.

The Functional (F) defines the mission that the system shall perform, what the system does, what actions or activities that must be accomplished to achieve a desired outcome, or to provide desired capabilities. The Functional formalizes the understanding of the intended use of the system (operational functions, use case scenarios, services) and its decomposition into elementary functions that can be allocated to technical solutions.

The Logical (L), or logical/organic defines how the system is, the technical solution based on subsystems, components and their interrelations according to stakeholders/customer needs and requirements, technical requirements, and expected functions.

The Physical View (P) defines a virtual definition of the real world product, including 3D representation to "visualize" the targeted system concept in the early phases.

The Functional and Logical include behavior models to define how functions or logical components transform inputs into outputs, react to events, or respond to excitations.

The Process and Methodology approach necessitates capturing the "voice of the customer" requirements and then applies them to make system level decisions downstream the development cycle, at the functional architecture level. These systems then get instantiated through subsystem-level functional structures and the logical systems that control them. Individual designers can then model and simulate these complex structures and the connections between them. Finally they are realized by actual physical components, whether mechanical, electrical, or software controls.

Functional and logical structures that have been defined between requirements and the physical components produced to satisfy those requirements are thus captured and aggregated in the same RFLP data structure.

These intermediate structures allow designers to analyze, model, simulate and virtually "Experience" the system and validate that the requirements are met before significant resources are engaged to its detailed design, development, validation and operation.

The structures produced by this RFLP based process allow them to understand, improve and capitalize the design for their reuse in new products developments.

3 Requirements and Model Based Engineering for Robust Process

3.1 Requirement Based Engineering

Taking into account the current and well established practices of various OEMs and Suppliers, effective support of Requirement based engineering processes means capturing requirements from many files and database sources (Windows/Office, Doors, ALMs, ReqIF,…) in a wide variety of data and file formats and link them (with easy to implement traceability links) to development and verification artifacts, through the provision of a large number of interfaces to common systems engineering tools. These traceability links should not only operate at files level, but truly at fine-grained engineering artifacts levels, enabling effective requirement coverage and change impact analysis.

Key benefits are:

- Capture of requirements at any level
- Intuitive, quick and easy requirement capture from various existing sources (Doors, Word or PDF documents,…)
- Coverage analysis and traceability
- Management of requirement changes and lifecycle information
- Upstream and downstream impact analysis for regression risk management
- Automated report and documentation generation.

3.2 Model Based Design

The typical E/E design process usually begins by generating specifications to describe the system, subsystem, component, and interface requirements. The system design is partitioned into functions and the functions partitioned into hardware and software. During development, the hardware and software components are usually developed individually, and then brought together late in the development cycle, when the system was integrated. In most cases, the final integration step, when discrepancies between concept and implementation are discovered, turns out to be the schedule driver: issues have to be resolved that were unforeseen, and thus not planned for!

Since recently, a novel methodology is emerging that incorporates the best features of the typical development cycle and adds additional visibility and risk reduction to the schedule-driving integration phase. This is accomplished by using, early in the design cycle, advanced hybrid simulation techniques that allow a lifelike, "virtual" integration of multi-faceted component models that can be launched while the design is still in its early phase and easy to change. This virtual integration eliminates the costly reworks that are frequently required when problems are not discovered until late in the product integration and validation phase. It also drastically reduces the time spent integrating hardware and software, sensors, actuators, processing units and communication networks, because many of the problems have already been discovered and corrected during development.

Each element at each level of hierarchy will have its own development flow, which will resemble the development flow of the overall system. Eventually many interoperable models will have to be developed to support the various operational behaviors that need to be simulated and validated (finite element model, multi-physics/mechatronics dynamic model, dysfunctional model for a safety behavior assessment, thermal model, control model, etc.…) (Fig. 6), ultimately with cross-correlation parameters reflecting impact of 3D metrics on behavior dynamics for instance, or of logic trade space effects on physical dimensioning and vice et versa, or of non-functional constraints on COTS selection, etc.…

It is useless to stress that complete systems depend on software (as well as hardware), and that system software design can make or break any system in performance, schedule, and cost. Software architecture and performance can be modeled just as hardware can. Real time simulations, hybrid simulation techniques such as MIL, HIL and SIL are to be used, as early as possible in the development cycle, with the capability to interoperate with other simulation techniques, which are necessary to virtually assess the complete system operational behavior.

Key aspects of a real system can be modeled and simulated completely in a lifelike simulation of complex multi-disciplinary systems, addressing the known characteristics of the system that relate to its multi-dimensional behavior and performance with regards to specific real life use cases and environmental conditions. This is known as the lifelike digital experience paradigm, where functional

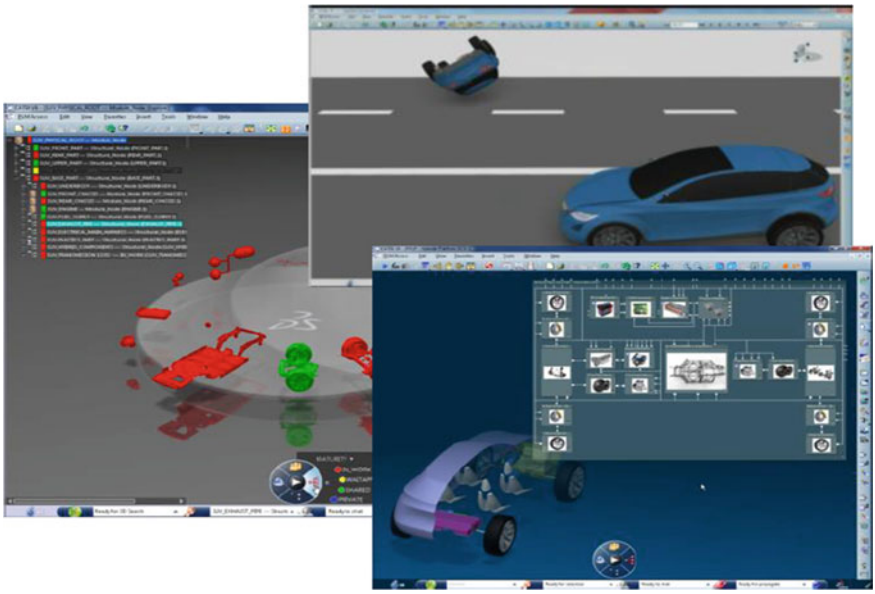


Fig. 6 3DEXPERIENCE platform

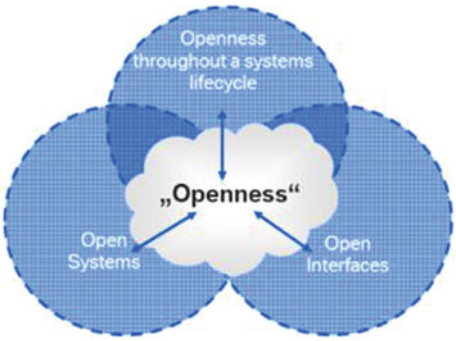
mock-up meets digital mock-up, both in one, to virtually express a user experience and reason on it from different Engineering perspectives.

This trend is nurtured by novel modeling and co-simulation technologies.

The industry has been striving for years to provide a unified open standard language (Fig. 7) to describe multi-physical systems behaviors. Eventually such a language, the Modelica language, has been developed and is starting to enjoy a successful adoption by the E/E mechatronics community worldwide.

Furthermore, to support Model based Design, the emerging FMI standard brings multiple interoperable simulation capabilities: the Functional Mock-up Interface (or FMI) defines a standardized interface to be used in computer simulations. This interface is implemented by simulation models or software libraries called FMU

Fig. 7 Openness axes



(Functional Mock-up Unit). FMI development was initiated by Daimler AG and is developed by the MODELISAR consortium. FMI is the runtime environment that was created to allow simulation data exchange (see also http://functional-mockup-interface.org/). By supporting the open source Modelica language, and leveraging the FMI interoperability of models and simulation kernels (co-simulation), The PLM based platform will also help Systems Engineers execute and analyze system and sub-systems models, while mixing dynamic and state logic behaviors, adding control by linking to Matlab Simulink or other control models. Modelica makes it possible for users to generate deterministic behavior and reuse modular mechatronics components. The open Modelica Standard Library, which contains mechanics, thermal, electricity including electronics, fluidics and control components, greatly improves collaboration across engineers of various disciplines, as well as enhances modeling effectiveness and throughput.

Built-in openness for scalable adoption.

Requirement based Engineering and Model based Design are two main pillars of effective system engineering processes. To support them, built-in openness mechanisms have to be implemented in the platform, with manifold objectives:

- Scale requirement traceability and change impact analysis throughout most popular systems engineering tools
- Support E/E and mechatronics industry standards.
- Integrate or interoperate with complex and heterogeneous System Engineering and Simulation environments by using standards as FMI.
- Be open for partners' solutions including customer preferred partners.
  Enable controlled and stable integration to other systems such as PDM, ERP, xCAD using the xPDM and xCAD platforms.

4 Conclusion

Market forces are driving the quantity, quality, completeness, and rate of change of Embedded System Engineering environments. Primarily, better integration and traceability between product requirements and lower level technical requirements down to system/software engineering environments are called for. An immediate powerful result of this integration will be the comprehensive traceability between high level driving product/modules requirements, the resultant technical configurations, and their cost and schedule impacts (design-to-cost, just-in-time …).

The requirement to integrate the trade-off space across technical, program and financial boundaries is likely to grow up. Models authoring will populate rich reusable model libraries if these engineering environments support open, standard languages and description formats, fostering their adoption and ease of inter-operation by a large community of domain skilled engineers worldwide.

These collective, multi-disciplinary efforts would never be harnesses nor capitalized upon without a true open, multidisciplinary, collaborative system engineering

management platform that facilitates comprehensive models elaboration, interoperation, cross correlations, global parameterization, configuration and reuse.

To accelerate the development of innovative products with higher rates of market success, online, collaborative PLM based solutions are mandated for both OEMs and Suppliers.

Such an open and integrated E/E PLM portfolio will enhance business processes and facilitate global collaboration, transforming information into actionable and reusable knowledge in a lifelike digital experience and effectively manage strategic activities, all while collaborating with global teams and key decision makers.

References

1. Dr. Morkevičius, A.: Integrated Modeling
2. Henneau, P.: A pragmatic approach to SE practices at Schindler elevators
3. Krob, D.: Architecture des systèmes complexes
4. Krob, D.: Eléments d'architecture des systèmes complexes
5. Krob, D.: Eléments de systémique—Architecture des systèmes 1
6. Resser, A.: Model based systems engineering management plan
7. V-Model Views, INCOSE