FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

ENHANCING LIFE CYCLE SUSTAINABILITY IN SYSTEM OF SYSTEMS:

AN EVENT DRIVEN FRAMEWORK FOR CHANGEABILITY

Gil Manuel Magalhães de Andrade Gonçalves

U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

A dissertation submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy
(Electrical and Computer Engineering)

Doctoral Committee:
- Jorge Leite Martins de Carvalho (Full Professor) (chair)
- Norbert Link (Full Professor)
- Adolfo Steiger Garção (Full Professor)
- Fernando Manuel Ferreira Lobo Pereira (Full Professor)
- Raul Fernando de Almeida Moreira Vidal (Associate Professor)

October 2015

ENHANCING LIFE CYCLE SUSTAINABILITY IN SYSTEM OF SYSTEMS:

AN EVENT DRIVEN FRAMEWORK FOR CHANGEABILITY

Gil Manuel Magalhães de Andrade Gonçalves

Doctor of Philosophy
(Electrical and Computer Engineering)

30 of October 2015

This page was intentionally left blank

To family ..

This page was intentionally left blank

Abstract

This thesis presents a systems thinking approach to the definition of an event driven framework to enhance life cycle sustainability in System of systems.

Our world is a complex system of systems and customisation, agility and networked operations are keywords of the present. We can either be talking about industry, education, government or nature; in every domain evolution and connection are key to respond to continuous and rapid changes triggered by the complex, dynamic interconnection of the systems we live in.

Networked organizations or networked systems are concepts to cope with the needs and challenges created by this context. This new reality is highly relevant in multiple domains including the industrial domain.

European industry is active in all manufacturing fields, making Europe one of the strongest outfitters and operators of factories, mainly because of the high quality of the produced equipment and production systems. Industrial processing machinery and production systems cover a wide range of products destined to specific purposes in downstream manufacturing sectors and, as such, demand for these is closely linked to new products or product renovation in the downstream manufacturing sectors.

In downstream sectors, customization and make to order lead to smaller lot sizes, higher variability of products and reduced product life cycles. Global competition brings in cost pressure forcing European industry to re-think the costs their products as well as their investments in equipment, factory planning, ramp-up and operation.

Rapid changing product portfolios and process technology requires manufacturing systems that are themselves easily upgradeable, and into which new technologies and new functions can be readily integrated, creating the need for novel manufacturing control systems

able to cope with the increased complexity required to manage product variability and disturbances, and to implement agility, flexibility and reactivity in customized manufacturing.

Facing these challenges requires highly flexible, intelligent and self-adaptive production systems and equipment, which can react to continuously changing demand, can be smoothly brought into operation, and can extend equipment life. At the same time manufacturing control systems need to able to cope with the increased complexity and exploit these new functionalities of the system to its fullest, not only to maximise its efficiency but also to its utility throughout its entire life cycle.

In engineering we are primarily concerned with technics for the design, control and analysis of system performance based on well-defined quantitative measures. This is done using models and many systems, particularly technology based ones, have discrete state spaces and can be modelled based on state transitions that are observed at discrete points in time. These systems are referred to as *Discrete Event Systems*. Even complex systems with underlying continuous variable dynamics can often be modelled as discrete for the purpose of analysis.

Nevertheless, current discrete event systems theory and existing frameworks fail to fully cover the challenges posed by today's systems. Existing results on controllability, observability and supervisory control need to be extended to include additional concepts like system of systems and the need to continuously adapt. Moreover, the notion of life cycle and sustainability of the system throughout its entire life cycle also have to be introduced.

Starting from two simple observations like "*The Times They Are a-Chagin'*" and "*Our world is a complex system of systems*" the argument is built: an extension to existing models and tools to deal with systems composed of interconnected elements capable of adapting themselves to an ever-changing environment is required. An analysis of three different case studies coming from different domains – business, manufacturing and robotics lead to the definition of the main concepts that are missing: *system of systems*, *play* and *playbook*, and *changeability*. Once these concepts were formalised the first steps towards an event driven framework for changeability has been defined.

The application of the event driven framework for changeability started with the definition of the context of application. As previously mentioned, changeability requires systems composed of interacting smart components.

The selected industrial case study has been defined in the scope of two European projects – I-RAMP[3] and ReBORN – which are working in concepts related with plug'n'produce and smart components for manufacturing systems, involving variability in the production demand, fast ramp up times and re-use of production equipment. The event driven framework was applied in a case study that involves the design of a production line, involving new and re-used equipment, and the exchange of equipment in the production line during operation.

An additional case study, selected from the robotics domain, was used to further demonstrate the applicability of the event driven framework for changeability. The event driven framework was applied in a case study that involves the design of a maritime observatory, involving persistent operations in wide areas executed by teams of autonomous vehicles.

These case studies made possible to demonstrate the adequacy of the framework to the manufacturing and robotics domains and to the defined contexts. It was also possible to demonstrate that the framework can be applied in different phase of the life cycle and to realise the importance to include *evolution* in the framework.

The main contributions of the work presented in this thesis may be used to extend current discrete event systems theory and existing frameworks by including in the theory support for the concepts of System of Systems, Changeability and life cycle. The main scientific research objectives achieved during the course of the this work have been the following:

- Cases where system of systems thinking is necessary, in the three aforementioned domains, were identified and analysed.
- Cases were used to synthesise a definition of system of systems amenable to be treated inside the discrete event systems framework.
- A set of issues (common to the three domains) that require the discrete event systems framework to be extended in order to be addressed were identified, notably *changeability*.
- The first steps towards the definition of an event driven framework for changeability, contributing to the enhance life cycle sustainability in system of systems.
- The applicability of these results was demonstrated in two cased studies: one from the industrial domain, applied in a case study defined within the scope of two European projects, and another from the robotics domain, applied within the scope of an ocean observatory based on multiple autonomous systems.

This page was intentionally left blank

Resumo

Esta tese apresenta uma abordagem sistémica para a definição de um quadro formal orientado a eventos que promova a sustentabilidade de sistema de sistemas.

Vivemos num mundo complexo constituído por sistemas de sistemas e onde personalização, agilidade e operação em rede são palavras chave atuais. Indústria, educação, governo ou natureza; em todos os domínios evolução e sinergias são essenciais para responder às mudanças constantes provocadas pelas complexas interligações dos sistemas que construímos e nos quais vivemos.

Organizações em rede ou sistemas em rede são conceitos que nos ajudam a lidar com os desafios criados por este contexto. Esta nova realidade é relevante em múltiplos domínios incluindo no domínio industrial.

A indústria Europeia atua em todos os domínios produtivos, tornando a Europa num dos principais fornecedores e operadores de unidades industriais, essencialmente pela qualidade do equipamento e dos sistemas de produção. Máquinas, equipamentos e sistemas de produção constituem um largo espectro de produtos destinados às mais diferentes utilizações em sectores produtivos a jusante e, como tal, a sua procura está diretamente relacionada com o lançamento ou a renovação de produtos nos sectores produtivos a jusante.

Nos sectores a jusante, personalização e produção para encomendas originam lotes de dimensões mais reduzidas, maior variabilidade nos produtos e ciclos de vida dos produtos mais curtos. A competição à escala global provoca pressão sobre os custos, obrigando a indústria Europeia a repensar a estrutura de custos dos seus produtos assim como o investimento em equipamento, novas unidades produtivas, custos de instalação e de operação.

Portefólios de produtos e tecnologias produtivas em constante mutação exigem sistemas produtivos que sejam eles próprios de fácil atualização, e nos quais novas tecnologias e novas

funções possam ser rapidamente integradas, criando a necessidade de sistemas de controlo inovadores com a capacidade de lidar com a crescente complexidade introduzida pela necessidade de gerir a variabilidade e alteração de produtos, de garantir agilidade, flexibilidade e capacidade de reação em cenários de produção à medida.

Para enfrentar estes desafios são necessários sistemas produtivos e equipamentos altamente flexíveis, inteligentes e auto-adaptáveis, com capacidade de reagir às alteações constantes no perfil da procura, de serem colocados em operação facilmente e aumentar o ciclo de vida dos equipamentos. Em simultâneo, os sistemas de controlo têm de ser capazes de lidar com esta complexidade adicional e de extrair o máximo destas novas capacidades dos sistemas produtivos, não apenas para maximizar a sua eficiência mas também a sua utilidade ao longo de todo o seu ciclo de vida.

Em engenharia uma das principais preocupações é o projeto, controlo e análise do desempenho de sistemas com base em medidas quantitativas bem definidas. Para isso são utilizados quadros formais onde muitos sistemas, em particular sistemas de base tecnológica, são caracterizados por espaços de estados discretos podendo ser modelados através das transições de estado observadas em determinados instantes temporais. Esta classe de sistemas é designada por *Sistemas de Eventos Discretos*. Mesmo sistemas com dinâmicas complexas associadas à evolução contínua das suas variáveis de estado podem ser modelados como sistemas discretos para efeitos de análise.

No entanto, a teoria e quadros formais de sistemas discretos atuais não endereçam completamente os desafios colocados por estes novos sistemas. Os resultados existentes relativos a controlabilidade, observabilidade e controlo supervisionado necessitam de ser estendidos de modo a lidarem com conceitos como sistema de sistemas e a necessidade de adaptação contínua. Adicionalmente é necessário introduzir a noção de ciclo de vida e de sustentabilidade do sistema ao longo de todo o seu ciclo de vida.

Tendo como ponto de partida duas simples constatações "*The Times They Are a-Chagin'*" and "*Our world is a complex system of systems*", o argumento para a necessidade de se alargar o quadro formal existente de modo a lidar com a realidade dos sistemas atuais é construído. A análise de três casos de estudo oriundos de domínios diferentes – negócios, indústria e robótica – conduziu à identificação dos conceitos necessários para lidar com esta nova realidade: sistema de sistemas, jogada e livro de jogadas, e capacidade de mudança. Uma vez formalizados estes conceitos, foram dados os primeiros passos no sentido da especificação de um quadro formal de capacidade de mudança orientado a eventos.

A aplicação do quadro formal de capacidade de mudança orientado a eventos começou com a definição do contexto de aplicação. Como já referido, a capacidade de mudança implica sistemas compostos por componentes inteligentes interatuantes.

O caso de estudo industrial selecionado foi definido no âmbito de dois projetos Europeus – I-RAMP[3] e ReBORN – que abordam conceitos relacionados com *plug'n'produce* e com componentes inteligentes para sistemas produtivos, envolvendo variabilidade no perfil de procura, tempos de instalação e colocação em operação reduzidos e re-utilização de equipamentos produtivos. Foi aplicado num caso de estudo que envolve a substituição de equipamentos numa linha de produção, utilizando equipamento novo e re-utilizado, durante a sua operação.

Um caso de estudo adicional, selecionado do domínio da robótica, foi utilizado para demonstrar a aplicabilidade do quadro formal de capacidade de mudança orientado a eventos em diferentes domínios. Este quadro formal foi aplicado num caso de estudo que envolve o projeto de um observatório marítimo, assente em operações persistentes em áreas amplas realizado por equipas de veículos autónomos.

Estes casos de estudo tornaram possível demonstrar a adequação do quadro formal apresentado aos domínios da produção e da robótica nos contextos definidos. Foi também possível demonstrar que este quadro formal em alturas diferentes do ciclo de vida e perceber a importância de incluir o conceito de evolução neste quadro formal.

As principais contribuições do trabalho apresentado nesta tese ajudam a estender o quadro formal atual ao permitir incluir na teoria o suporte para conceitos como sistema de sistemas, capacidade de mudança e ciclo de vida. Os principais resultados científicos alcançados na realização deste trabalho foram os seguintes:

- Identificação e análise de casos, nos três domínios de aplicação anteriormente identificados, onde a abordagem de sistema de sistemas é necessária.
- Síntese de uma definição de sistema de sistemas passível de ser utilizada no quadro formal de sistemas de eventos discretos.
- Identificação de um conjunto de questões (comuns aos três domínios) que requerem uma extensão do quadro formal de sistemas de eventos discretos de modo a poderem ser endereçadas, em particular a capacidade de mudança.
- Os primeiros passos para a definição de um de capacidade de mudança orientada a eventos, contribuindo para aumentar a sustentabilidade ao longo do ciclo de vida em sistema de sistemas.

- A demonstração da aplicabilidade destes resultados em dois casos de estudo: no domínio industrial, num caso de estudo definido no âmbito de dois projetos Europeus, e no domínio da robótica, num caso de estudo no âmbito de um observatório oceânico baseado em múltiplos sistemas autónomos.

Acknowledgments

This journey started nearly twenty years ago and was filled with breakthroughs, detours, setbacks and roundabouts. Along the way I had the luck to meet many fellow travellers, who made this journey of exploration and discovery a real pleasure. I will not take the risk of forgetting any by trying to name this countless bunch. But I would like to acknowledge the many times one of these fellow travellers backed me in a setback or helped me out of a roundabout. Many thanks for your support, advice and companion.

Last but not least, *family*. My wife – *Susana* – and children – *Guilherme*, *Tomás* and *Miguel* – for enduring absences, temper, and always standing by my side, giving me the drive to recover from the worst falls by the wayside. Thank you!

This page was intentionally left blank

Contents

List of figures

List of tables

Abbreviations and Symbols

OEM	Original Equipment Manufacturer
SoS	System of Systems
C2	Command and Control
CP	Console Profile
LOA	Level Of Automation
OODA	Observe, Orient, Decide, Act
PDCA	Plan, Do, Check, Act
CVSD	Continuous-Variable Dynamic Systems
DEDS	Discrete Event Dynamic Systems
DES	Discrete Event Systems
G	Automaton
AUV	Autonomous Underwater Vehicle
ASV	Autonomous Surface Vehicle
ROV	Remotely Operated Vehicle
UAV	Unmanned Aerial Vehicle
WSN	Wireless Sensor Netkor
USTL	Underwater Systems and Technologies Laboratory

ASV	Autonomous Surface Vehicle
AUV	Autonomous Underwater Vehicle
COTS	Commercial Off-The-Shelf
CTD	Conductivity, Temperature and Depth
DTN	Delay Tolerant Network
GPS	Global Positioning System
GSM	Global System for Mobile Communications
JAUS	Joint Architecture for Unmanned Systems
IMU	Inertial Motion Unit
ROV	Remotely Operated Vehicle
RPV	Remotely Piloted Vehicle
UAV	Unmanned Air Vehicle
UUV	Unmanned Underwater Vehicle

1. Introduction

"The Times They Are a-Changin'"

Bob Dylan

Standardization, specialization and concentration, are keywords of a not so distant past where the dominant organizational structure is a classic industrial bureaucracy: a huge hierarchical organization, permanent, mechanic and top down oriented, synthesized to make the same products or take the same decisions over an over again in relatively stable environments.

Things changed.

Markets constantly demand new, innovative and customized products or services; aggressive competition at a global scale; increasing productivity through highly optimized production processes, and environmental/societal pressures create a new environment which is everything but stable, and this demands radically different organizational structures. The new dominant keywords are now customisation, agility, re-configurability, flexibility and networked operations. From the 'Virtual Corporation' [1] to the 'Learning Organization' [2], including Toffler prophetic 'Matrix Organization' and 'Adhocracies' [3], there are a significant number of new organizational structures emerging. The intersection of these trends can be summarized in the following quote from Toffler's "The Third Wave":

" [..] a disorganized open system, opposed to an organized closed system [..] a system made of highly inter-related units, like the neurons in a brain, and not like the department organization in a bureaucracy."

1.1. Motivation and background

"Our world is fraught with inefficiency – US$15 trillion worth to be exact."

Peter Korsten, IBM Institute for Business Value

Virtual or Networked Organizations, and value-adding partnerships are some of the proposed concepts to cope with the need to rapidly bring new products into the market, with high quality and at competitive prices.

Multi-functional project teams is becoming one of the key approaches to business and problem solving, not only in high-tech but also in traditional industries. Multi-functional teams bring together members from different functional departments or even from different organizations. In organizations operating multifunctional teams structures, both permanent and temporary teams are used extensively to accommodate projects. Traditionally, these multi-functional teams (or consortia if we include the multi-organization dimension) were setup by a decision maker in charge of identifying the best members for the team. However, in an environment where the number and duration of these endeavours demands fast reactions and swift decisions it could be difficult to get prompt response from these decision makers.

This new reality also applies to manufacturing. Rapid changing product portfolios and process technology, requires manufacturing systems that are themselves easily upgradeable, and into which new technologies and new functions can be readily integrated [4], creating the need for novel manufacturing control systems able to cope with the increased complexity required to manage product variability and disturbances, effectively and efficiently [5], and to implement agility, flexibility and reactivity in customized manufacturing.

Increasingly, traditional top-down and centralized process planning, scheduling, and control mechanisms are becoming insufficient to respond to constant changes in these high-mix low-volume production environments [6]. These traditional centralized hierarchical approaches limit the adaptability [7], contribute to reduce the resilience of the system, as well as to reduce the flexibility of planning and to a corresponding increase in response overheads [8]. The ability of a manufacturing system, at all of the functional and organizational levels, to reconfigure itself in order to quickly adjust production capabilities and capacities in response to sudden changes in the market or in the regulatory environment is nowadays a major requirement.

When working to increase efficiency (or to eliminate inefficiency), most businesses, industries, and governments use traditional modelling and optimization approach centred on their own value chains, with little or no consideration to interrelationships with other value chains. But, this perspective creates opacity across “the system”. Organizations are often unaware of the indirect impacts of their own actions, because decisions are optimized for a particular organization, community or group, and the effect at the macro level is ignored. Much of existing inefficiency can be attributed to the fact that the world is optimised to work in silos, with little regard for how the processes and systems interact.

Business as usual continues to use our natural and financial resources. However, consumers, businesses and governments are increasingly focused on social responsibility and sustainability issues are being included into the decision process.

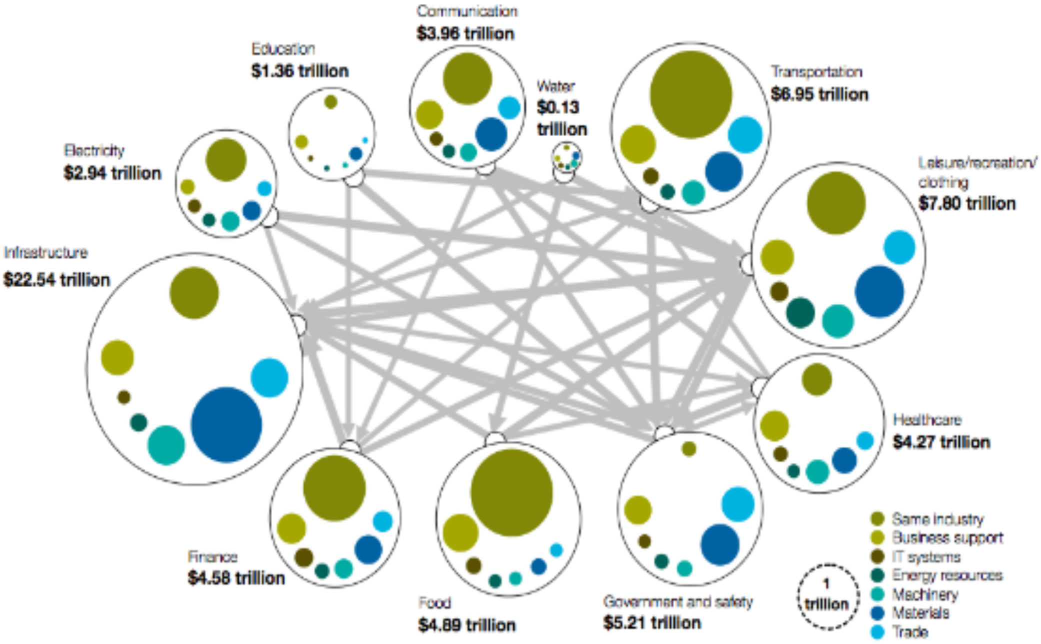


Figure 1: “Our world is a complex System of Systems” (source [9])

System of systems thinking is not new and it has been around for a while, primarily used by defence and aerospace industry. Several frameworks – e.g. TOGAF[1] [10], DODAF[2] [11], UML[3] [12], SysML[4] [13] – have been developed and extended to accommodate the System of Systems concept but are mostly focused on the architecture, design and implementation, missing support for other phases of the system life cycle.

Facing these challenges requires highly flexible, intelligent and self-adaptive models of systems and equipment, which can be used throughout the entire life cycle, react to continuously changing demand, can be rapidly and smoothly brought into operation, and can extend equipment life, contributing this way to enhance the systems sustainability.

---

[1] The Open Group Architecture Framework

[2] DoD Architecture Framework

[3] Unified Modeling Language

[4] Systems Modeling Language

In engineering we are primarily concerned with technics for the design, control and analysis of system performance based on well-defined quantitative measures. Usually this is done using models and many systems that are found in our everyday life, particularly technology based systems, have discrete state spaces and can be modelled based on state transitions that are observed at discrete point in time (associated with *events*). These systems are referred to as *Discrete Event Systems* (DES). The higher-level behaviour of complex systems with underlying continuous variable dynamics can be simplified and often modelled as a DES for the purpose of supervisory control, monitoring, and diagnostics.

However, some of the issues found in System of Systems are not fully covered by current discrete event systems theory and existing frameworks. Existing results on controllability, observability and decentralised control have to be extended to encompass these additional concepts like *System of Systems* and *Changeability*. Moreover, the notion of life cycle and sustainability of the system throughout its entire life cycle also have to be introduced.

1.2. Scientific research objectives

Motivated by ever-changing environments and emergence of the system of systems thinking to address many societal and business challenges, this thesis will focus on the framework of discrete event systems to understand what extensions are necessary to this framework. The purpose of these extensions will be to make this framework adequate to support system of systems thinking in many domains, notably in the industrial, business and robotics domains.

As such the scientific research objectives of this thesis are the following:

- Identify and describe cases where system of systems thinking is necessary in the three aforementioned domains.
- Use these cases to synthesise a definition of system of systems amenable to the discrete event systems framework.
- Identify an issue or a set of issues common to the three domains and that can be treated using the extended framework, notably related with the need for the systems to change and adapt to ever-changing environments.
- Formalise an event driven framework for changeability that extends current discrete event systems framework and contributes to the life cycle sustainability.

- Demonstrate the applicability of these results in two of the previously identified domains.

1.3. Original contributions

This thesis deals with contributions to extend current discrete event systems theory and existing frameworks by including in the theory support for the concepts of System of Systems, Changeability and life cycle. The main expected results of the work are the following:

- Support for System of Systems modelling in the DES paradigm.
- System of Systems, play and playbook definition.
- Event driven changeability framework.
- Application and demonstration of the framework in an industrial scenario.

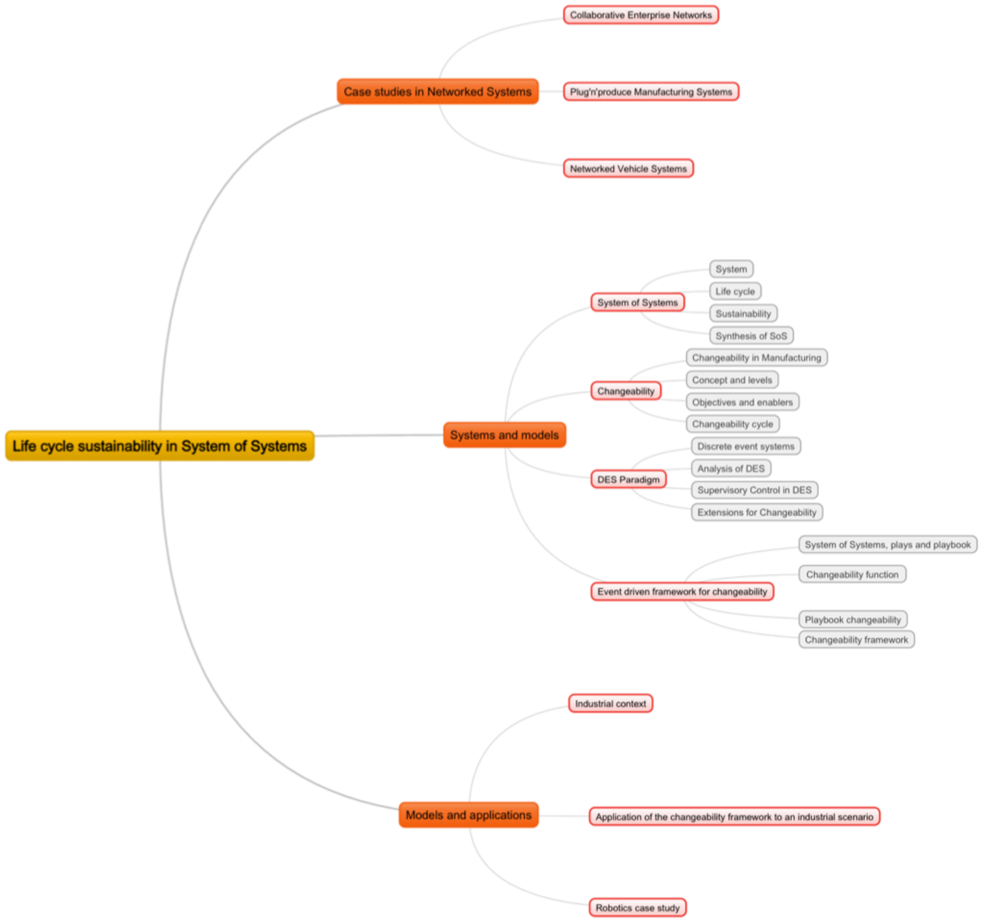


Figure 2: Contributions mind-map

Contributions also include three case studies in networked systems, coming from three different application domains (business, industry and robotics), and the methodology followed to derive the results.

1.4. Methodology and thesis organization

Synthesis in systems thinking [14] (or systemic thinking) is deliberately finding repeating patterns (or common themes) across a system or situation, whilst analytical thinking is more focused on identifying the differences. The idea of systems thinking is to list as many different elements as possible, then look for the similarities between the different elements. Unlike systems thinking, the basic idea behind conventional analytical thinking techniques is to list a handful of elements, compare them, rank them and then select the most valuable one(s) discarding the remaining.

Analytical thinking breaks things down into their component parts; systems thinking finds the patterns across those component parts. Analysis is about identifying differences; synthesis is about finding similarities. Synthesis needs analysis – how can you find the similarities across different things, if you have not listed the different things first? Analysis needs synthesis – understanding how things behave in isolation is pointless.

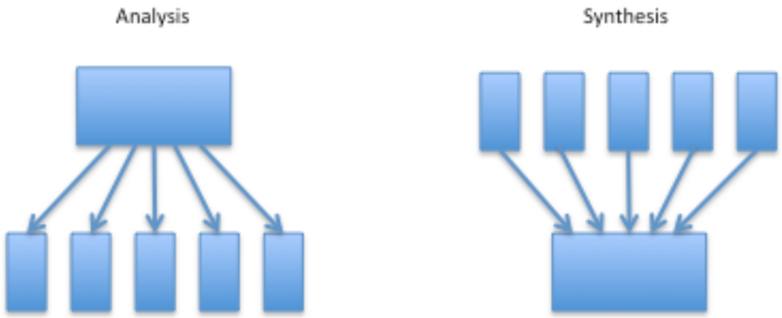


Figure 3: Analysis Vs. Synthesis

To understand how complex systems, systems composed by many independent and interacting components, behave we need to understand how the different components behave in concert and not in isolation. Analysis, in the context of systemic thinking, is different from analysis outside of that context. Outside of the systemic thinking context, the tendency is to list only a manageable number of elements, in order to reduce the effort[5]. Within the systems thinking context, it is desirable to list as many different elements as possible to ensure the most representative pattern possible. Systems thinking combines analytical thinking and

---

[5] Analysis breaks things down into their component parts, so you get more and more things to think about, and the tendency to list only a few elements.

synthesis. The first step is analytical: list as many elements as you can think of. The second step is synthesis: find the common theme / repeating pattern across those elements.

Figure 4: Systems thinking approach

The organization of the thesis follows this methodology. After this introduction, part I – Case Studies in Networked Systems – provides an analysis of a set of motivational examples of networked systems, from different domains, in order to list as many different elements that characterise this class of systems (System of Systems) as possible. Part II – Systems and Models – finds the common theme and repeating patterns across those elements in order to synthesise a representative definition for this class of systems and to define an event driven framework to life sustainability. Part III – Models and Applications – demonstrates the applicability of this framework in two application scenarios, the first from the industrial and the second from the robotics domain.

The thesis concludes with a chapter describing the main results and contributions of the work and defining a roadmap for future research.

Part I: Case studies in networked systems

"Every synthesis is built upon the results of a preceding analysis, and every analysis requires a subsequent synthesis in order to verify and correct its results."

Tom Ritchey

This part of the thesis will introduce carefully, but rather informally, the basic concepts of networked systems, with three illustrative case studies of these concepts coming from the areas of business, manufacturing and robotics. As we go along, fundamental criteria by which this class of systems can distinguished and classified are identified. The main goal of this section is to set out the motivation for the System of systems approach and for the need of an inherent capability to change in this class of systems.

2. Collaborative enterprise networks

New organizational forms like “Networked Organizations”, “Virtual Organization” and “Joint Venture”, are temporary consortia between organizations that get together and cooperate in response to a specific business opportunity, dissolving right after the opportunity ceases to exist.

What are networked organizations? Snow *et al.* [15] describe three types of networked organizations: internal, stable and dynamic. In an internal networked organization, a single company owns most of the required assets and is very little exposed to outsourcing. In a stable networked organization there is already a significant level of outsourcing (typical situation where there is an OEM supported by a set of suppliers). In a dynamic networked organization there is a leader that plays the role of broker, identifying potential partners that are owners of assets and selects the best team for a particular endeavour.

Virtual organizations are very similar to networked organizations: an alliance of independent business processes (or assets) contributing with a different set of competencies/capabilities (e.g. design, manufacturing, marketing, distribution, etc.) to achieve a common goal. Similarly to dynamic networked organizations, there is no single company that owns all the assets (or has all the competencies). Virtual organizations are built upon temporary collaborations to take advantage of specific business opportunities. “Joint Ventures” are a common example of this type of organizations, formed by two or more distinct organizations, typically used for international expansion.

The added value of this type of organizations is hindered by what Coase [16] defined has the “transaction cost”: a cost incurred in making an economic exchange or, in other words, the cost of participating in a market. One of the main “cost drivers” is related with the selection of the right partners: competent, compatible and complementary partners (the 3Cs). It is not sufficient to guarantee the 3Cs of the selected partners, but most important is to insure that all partners are efficient in their individual and collective contribution, so that together they form a networked organization in which the whole is greater than the sum of its parts. This is one of the principles in holonic organizations [17].

The process of setting up this type of organization may be organized in four phases: selection of partners, negotiation to setup the consortia, definition of the consortium agreement and operation of the consortium.

The role of all partners is instrumental in every phase, since it their characteristic (individual and combined) determine the success of the partnership. Therefore, the selection of the adequate partners is of critical importance. Although the thematic of partnerships is extensively discussed in the literature (see for example the literature review in Mat *et al.* [18] and Tseng *et al.* [19]) there are a limited number of formal approaches supporting the selection of partners in environments with uncertain information.

The assignment problem is a special type of linear programming problem where resources are being assigned to perform tasks. There is a simple algorithm to efficiently evaluate the solution. This algorithm is known as the Hungarian Method [20] and is able to allocate the best set of resources to a set of tasks. However, this approach is not helpful in the present context mainly due to the fact that the data made available by the partners (assuming that each potential partner provides a self description with its typical capabilities, times and quality levels) does not take into account the impact of working together with other companies. To be effective, the Hungarian method would have to run several hundreds of different partner configurations.

The partner selection problem is not a simple allocation between resources and tasks, and several approaches are available to address this problem.

Zakarian and Kusiak [21] present a conceptual approach for selecting, between members of a team with different capabilities, the most adequate based on the requirements of the client and product characteristic. Using a three stage approach that applies Quality Function Deployment (QFD) matrix [22], to organize the critical factors to use in the selection of the team members, followed by the Analytical Hierarchical Process (AHP) [23], to determine the importance of each element in the team, and by a mathematical programming model to determine the optimal composition of the team.

Tseng et al. [19] propose a methodology based on fuzzy logic [24] that is applied to the formation of teams when there is no clear relation between the characteristics of the project and the requirements of the client.

Hajidimitriou and Georgiou [25] propose a quantitative approach to the problem of partner selection in order to deal with the multiple variables and criteria within a goal programming model [26].

Talluri et al. [27] present a quantitative approach with two stages to support the selection process of compatible and efficient partners in a consortium. The first stage uses a filtering process to select the more efficient candidates [28] and the second stage uses an integer programming model [29] to determine the more efficient combination based on a set of pre-defined criteria.

Huang and Chen [30] propose and approach based on risk criteria for partner selection. The leader organization knows (or estimates) the associated risk to each candidate partner and uses this information to evaluate the risk level of different consortia.

2.1. Collaborative enterprise networks in knowledge intensive scenarios

Project Management looks at the formation of multidisciplinary teams as one of the critical aspects for successful projects. Nonetheless, there are very few attempts to solve the problem on selecting the most adequate elements for a multidisciplinary team in a context with incomplete or uncertain information. This is true when we consider collaborative research and development (R&D); there is no literature on this topic. Notwithstanding, some of the works described in the previous section can be used as reference to address the problem of partner selection in collaborative R&D projects. Collaborative R&D involves businesses and researchers working together on innovative projects in strategically important areas of science, engineering and technology – from which successful new products, processes and services can emerge, contributing to business and economic growth. The selection of the most adequate partners, with the right competencies, complementarities and which are compatible is a critical success factor.

Selection of partners is usually conducted by a lead partner (coordinator) or by a set of core partners who have previously worked together. The task of identifying the remaining partners is not simple and a platform was setup in order to support the decision maker.

The goal of RDNET platform is to provide a set of decision tools that can support the decision maker in the selection of partners for collaborative R&D projects. At the same time the platform will evaluate all involved organizations based on their interactions overtime, by adjusting their reputation and capabilities based on the feedback from partners. The ultimate aim of RDNET is to create an ecosystem that fosters networked organizations for collaborative R&D projects.

Due to the volatile and sporadic character of the interactions in these kind of organizations, and to the fact that knowledge is dispersed through the various partners of the network, the partner selection mechanisms have to be able to cope with is incomplete and uncertain information. Moreover, and since the selection of partners can not be made isolated only based on the competencies, but needs to guarantee the complementarity and compatibility, the solution space grows exponentially with the number of candidates in the ecosystem creating additional requirements for the mechanisms.

The RDNET platform integrates three type of support mechanisms: partner selection, partner evaluation/reputation and cooperation.

Based on these mechanisms, the RDNET platform will be able to support the decision maker in the selection of the most adequate consortium for the implementation of a collaborative R&D project (based on a characterization of the project).

The RDNET platform requires mechanisms that help organizations, or decision makers inside the organizations, to select the best set of partners to exploit an opportunity for a collaborative R&D project. This consortium (or set of possible consortia) is suggested to the user, which can adjust the selection and afterwards make the contacts with the prospective partners to explain them about the opportunity and their role.

During the execution of projects, the RDNET platform will provide mechanisms that can be used by participating organizations to rate the behaviour and performance of their partners in several perspectives (scientific-technical, quality, time, etc.). These ratings are then used by the system to evolve the profile of the organizations. These profiles, containing criteria that measure the perceived value of the organization to its partners, are the basis for the partner selection mechanism.

The partner selection mechanism, that supports the user in setting up a consortium to exploit a cooperative R&D project opportunity, is of paramount importance in the RDNET platform. A good consortium is vital in all phases of the life cycle of a cooperative R&D project:

- In the proposal preparation, due the importance of the active collaboration of all partners for a successful application.
- During the evaluation of the proposal, because the capabilities, complementarity and adequacy of the consortium is one of the important factors under analysis.

- During the execution of the project, since successful completion is only possible with competent, timely and quality contribution from all partners.
- In the exploitation of the project results, where all partners must contribute and agree on a common exploitation plan for the results.

Albeit the importance a good consortium is undisputed, the characteristics that make a good consortium for a given project are not consensual. Its highly subjective, considering that the consortium should have, on top of the required scientific-technical capabilities, other competencies that guarantee the correct management and fluid execution of the project. Most of the times these others factors (also described as soft factors and include communication skills, language and cultural barriers, company culture, individual strategies, etc.) are very difficult to measure and to quantify.

The main difficulties when implementing this partner selection mechanism are related with the selection of the properties that best describe the project and in the selection of the criteria to use in the analysis of the consortium adequacy. Some are more obvious, like the degree of expertise in a certain area relevant for the project implementation, others are less apparent, as for example the cultural compatibility of the partners in an international consortium. It would be rather simple to list and use a basketful of criteria, but such choice would make not only the selection mechanism to complex and cumbersome, but also the gathering of the all the information needed to characterize the projects and the organizations impracticable. The selection of the most relevant criteria is mandatory.

Dijk et al. [31] present the result of a study conducted for the European Commission with the goal to identify the critical factors of success in high-impact projects in ICT. The approach used started with the selection and analysis of several case studies with the goal to identify the factors in common. The main conclusions point out the creation of critical mass (i.e. representativeness of partners and stakeholders), involvement of the end-users and of the whole value chain, and a range of partners with competencies and strategic vision in the focus area of the project.

Wagber et al. [32] describes the example of a consortium composed of several health care systems that come together for specialized studies in patients with cancer, dealing with prevention and management of these diseases. Once again, the critical success factors highlighted are related with the critical mass and the competencies of the partners involved.

In a slightly different perspective, the ACTeN project [33] identifies the major advantages in being part of collaborative projects (e.g. the transfer of methods, knowledge and ideas) and the context where the advantages materialize (e.g. the need for permanent communication between partners, consortium dynamics that promotes cooperation, etc.).

2.2. Selection approach

Once an opportunity is identified and the requirements for the project are defined, the selection of the partners is not a simple match of the required activities and competencies. Other criteria are used, related with the effectiveness of the potential partner, its known track record in collaborative R&D, and previous activities between potential partners, amongst other. Some characteristics required for the potential partners are derived from the opportunity and from the future project, whilst other may be set by the decision maker. All these characteristics together form the project requirements, which is the baseline for the partner selection.

The IDNet project [34], funded by the Portuguese Government, conducted a European wide survey to identify the most relevant criteria used in the selection of partners for R&D projects. This survey identified 18 criteria, organized in 3 groups, which will be used for the selection mechanisms:

- Idea related criteria (6): scientific-technological area, scientific-technical competencies needed, technology readiness level of the needed technologies, degree of innovation, degree of novelty in the market and market size.
- Partner related criteria (6): level of expertise in the area, previous collaborations, experience in cooperative R&D projects, size of contact network, track record in cooperative R&D projects, degree of pro-activity in cooperative R&D projects.
- Consortia related criteria (6): coverage of the needed scientific-technological areas, country coverage in terms of number of countries and balance between countries, good balance in the type of partners, number of partners is adequate to the size of the projects, knowledge between the partners from previous R&D cooperative projects.

Table I: Consortia Related Criteria

Criteria	Variable
Scientific-technological areas (capabilities)	C_i
Size of the consortium (interval)	d
Scope of the project (national, regional, European)	A_k
Type of instrument for the project implementation	I_j
Number of partners per type (university, SME, ...) *	T_k
Number of partners per scientific-technological area *	A_i
Role of the partner (research, development, ...) *	P_i
Number of work packages in the work break down structure of the project *	W_i

By using these criteria, the proposed approach is based on a three-stage decision model. The first stage is a filtering process that selects all candidate partners (i.e. those that have at least on of the required capabilities) organizing these candidates by (area of) competencies. The second phase applies a new filtering process that uses relations between internal variables of the candidate partners. This approach helps to select the most efficient candidates for each of the competencies, thus reducing the solution space by eliminating the least efficient candidates. This is implemented with a Correlation Component Regression (CCR) model [28] which is a technic that is used to identify efficient candidates using internal variables and performance. As a consequence, the number of candidates is reduced, thus reducing the solution space (which is a product space). The third and last stage uses a multi-objective integer-programming model that uses exogenous decision variables (compatibility criteria) to select the most efficient combinations.

Consider an example. For a certain collaborative R&D project three distinct capabilities – C_a, C_b and C_c – are identified as part of the project requirements. If, as a result of the first stage, there are 10 possible partners with capability C_a, 10 for capability C_b, and 8 for capability C_c, there are 800 possible solutions (i.e. number of different combinations). The second stage allows reducing this solution space, by identifying the more efficient partners per capability for example to 4, 3 and 3. Stage 3 would have to solve the multi-objective problem for this set of candidate partners and provide the most efficient consortia.

A possible limitation of this approach is related with the filter applied in the second phase. If we optimize this filter the final solution might be sub-optimal (candidates removed at this stage might allow for better consortia). One way to address this limitation is with the relaxation of the constraints for the selection of the most efficient partners.

2.3.Selection mechanism

This selection mechanism is going to identify the most adequate partners to form a consortium for a cooperative R&D project, to which the project requirements have been previously defined, and uses a 3-stage approach (Figure 5):

- Select potential partners.
- Select the best candidates for the consortium.
- Select the best combinations/solutions (consortia).

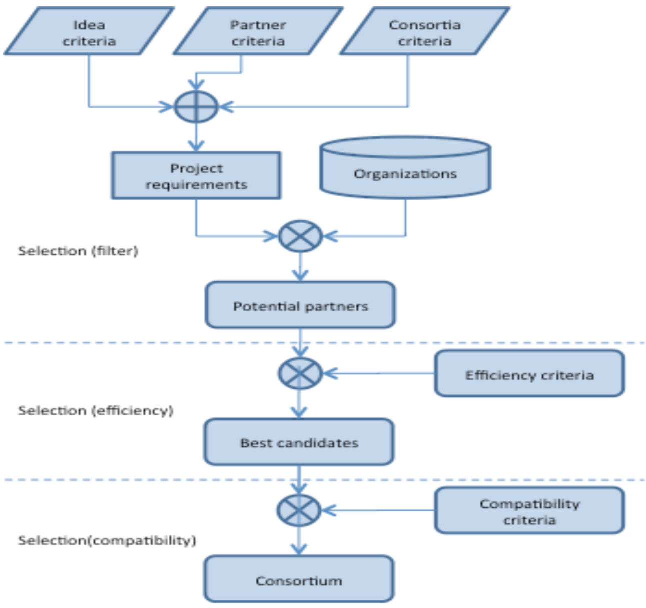


Figure 5: 3-stage approach for partner selection

This three stage approach follows the process: 1st stage filters candidates, 2nd stage selects the best candidates based on efficiency criteria and finally the 3rd stage selects candidate consortia based on compatibility criteria. The criteria used in the filtering and in the efficiency calculations are based on the criteria presented previously. The consortia related criteria (Table I) are derived from the idea and from the funding instruments that can be used to implement the project (own funding, national program, international program, etc.). The criteria marked with * may be undefined at first, i.e. these criteria are not mandatory.

The other criteria are related with the partners (Table II), organizations that are registered in the RDNet platform and have an associated profile. This profile defines their areas of scientific-technological interest, their capabilities and the perceived level of these capabilities in the platform.

Selection of potential partners

The selection of potential partners for a project (1st stage) is a filtering process based on the capabilities needed for the project implementation:

Equation 1

$$EC_i = \begin{cases} \{E_j : AC_{ji} = 1\} \Leftarrow C_i = 1 \\ \emptyset \Leftarrow C_i = 0 \end{cases} \quad i = (1, \cdots, n), j = (1, \cdots m)$$

Where:

- EC_i is the set of potential partners to fulfill capability i.
- E_j is the entity with a capability vector AC_j.
- C is a vector of the required capabilities for the project.
- n is the number capability areas and m the number of entities.

This 1st stage will select the potential partners and organize them in sets (1 set per capability area needed in the project). One organization can be part of many sets.

Table II: Partner related criteria

Criteria	Variable
Type of partner (university, SME, …)	t
Scientific-technological interests	AI_i
Scientific-technological Capabilities	AC_i
Level of experience (per scientific-technological area)	ACE_i
Level of experience in R&D cooperative projects	e
Track record in R&D cooperative projects (perceived Quality)	q
On time delivery of results (perceived)	p
Size of the contact network	C

Selection of best candidates partners

The selection of the best candidates for the consortium is done by capability area. This stage of the partner selection procedure uses a measure of cross efficiency, with the goal to reduce the size of the solution space without removing potentially good solutions.

By potential partner set (EC_i) and by organization, the idea is to maximize the efficiency (relation between inputs – e.g. number of R&D staff, annual budget for R&D – and outputs – e.g. number of projects, number of patents) of an organization k in relation to a set o reference organizations (EC_i) for a specific capability area i. This efficiency is computed using optimal weights for the input/output measures by finding the solution to the following problem:

Equation 2

$$\max \mathrm{E_{kk}} = \sum_{y} \mathrm{O_{ky} v_{ky}} \qquad \mathrm{s.t.} \begin{cases} \mathrm{E_{ks}} \leq 1 \; \forall \; \mathrm{E_s} \\ \sum_{x} \mathrm{I_{kx} u_{kx}} = 1 \\ \mathrm{u_{kx}}, \mathrm{v_{ky}} \geq 0 \end{cases}$$

Where:

- O_{ky} is the output y of organization k.
- I_{kx} is the input y of organization k.
- v_{ky} is the weight of output y for organization k.
- u_{ky} is the weight of input y for organization k.
- S is the set of organizations that have the required capability.

E_{ks} is the cross efficiency of organization s with respect to organization k, computed using the weights of the organization whose efficiency is being evaluated (k) using the following formula:

Equation 3

$$\mathrm{E_{ks}} = \frac{\sum_{\mathrm{y}} \mathrm{O_{sy}} \mathrm{v_{ky}}}{\sum_{\mathrm{x}} \mathrm{I_{sx}} \mathrm{u_{kx}}}$$

The optimal of the value function E_{kk}^{*} given by eq. (2) is the efficiency of organization k. If $E_{kk}^{*} = 1$ than no other organizations is as efficient as k (for the selected weights). If $E_{kk}^{*} < 1$ then there is at least one organization more efficient than k for the optimal weights determined by eq. (2).

This optimization problem has to be solved as many times as the number of organizations in the set EC_i, for each of the capability areas required.

At the end, this 2nd stage selects, for each of the capability areas, the set of most efficient candidates for the consortium EC^{*}_{i}. One organization can be part of many sets.

Selection of the best candidate consortia

The selection of the best combinations/solutions uses the sets of most efficient candidates EC_i^{*} $(i = 1,..,n \wedge C_i \neq 0)$ and computes the set of all possible combinations (i.e. all possible consortia):

Equation 4

$$\mathrm{CO} = \{\mathrm{CO_k} \in \mathrm{EC_1^{*}} \times \mathrm{EC_2^{*}} \times \cdots \times \mathrm{EC_n^{*}} : (\mathrm{ec_{1k}}, \mathrm{ec_{2k}}, \cdots, \mathrm{ec_{nk}})\}$$

Since one organization can be part of several candidate sets, solutions may exist in which the same organizations plays multiple roles (i.e. is responsible for providing more than one capability in the project).

Once all the possible solutions are determined, the best ones are selected by a multi-criteria decision problem. This problem uses the following criteria to rank the feasible solutions:

- Coverage of the project requirements.
- Strength of the consortium (a measure of existing trust between the partners).
- Confidence in the final result (inverse to the risk of project failure).

These are called the compatibility criteria, and are determined for each combination of n organizations C_{it}, where t is the compatibility criteria and i is the solution index.

The coverage of the project requirements by a feasible solution is determined by

Equation 5

$$\mathrm{C_1 = \frac{ar + geo + dim + tip}{4}}$$

Where *ar* is the coverage of scientific-technological capabilities needed, *geo* is a measure of the geographical coverage, *dim* is the size of the consortia and *tip* is the type of partners involved:

$$\mathrm{ar = C/AC}$$
$$\mathrm{geo} = \begin{cases} \mathrm{p/P \Leftarrow p < P} \\ \mathrm{1 \Leftarrow p \geq P} \end{cases}$$
$$\mathrm{dim} = \begin{cases} \mathrm{0 \Leftarrow n < n_{min} \vee n > n_{max}} \\ \mathrm{n/n_{n_{max}}\ every\ otlherelse} \end{cases}$$
$$\mathrm{tip = 1 - \frac{\sum_k |T_k - t_k|}{m} \quad (k \in \{1, \ldots, m\})}$$

Where

- C is the number of capability areas (required by the project) available in the consortium.
- AC is the number of capability area required by the project.
- p is the number of partner from different countries in the consortium.
- P is the number of required different countries required.
- n is the number of organization in the consortium.
- n_{min} is minimum number of recommended organizations.
- n_{max} is maximum number of recommended organizations.

- T_k is the number of *k* type organizations in the consortium.
- t_k is the number of *k* type organizations recommended.

The strength of the consortium, a measure of the trust between the partners, is determined as a function of the distance between the partners in a social network (a graph with the relations between the organizations with unit weight on the edges; the minimum distance between organizations is 1) by

Equation 6

$$\mathrm{d}_{\mathrm{x,y}} = \min\{\mathrm{n} | \mathrm{A}^{\mathrm{n}}[\mathrm{x}, \mathrm{y}] \neq 0\}$$

The strength of the consortium is the inverse of the distance between organisations in the network

Equation 7

$$\mathrm{C}_2 = \frac{\mathrm{n} - 1}{\sum_{\mathrm{x}} \min\left(\mathrm{d}_{\mathrm{x,y}} : \mathrm{y} = 1, \cdots, \mathrm{n}\right)}$$

The confidence in the final result of the project is determined based on the degree of confidence (reputation) the members of the consortium have in each other. This reputation is a value, a measure of the perceived value for each organization, computed using the degree of experience in the area, the experience in cooperative R&D projects, the track record and the on-time delivery of results:

Equation 8

$$\mathrm{C}_3 = \frac{\sum_{\mathrm{x}} \mathrm{confr}_{\mathrm{x}}}{\mathrm{n}}$$

with

Equation 9

$$\mathrm{confr}_{\mathrm{x}} = \frac{\mathrm{ACE}_{\mathrm{i}} + \mathrm{e} + \mathrm{q} + \mathrm{p}}{4}$$

Where

- ACE_i is the degree of experience of the organization in capability area *i*.
- *e* is the experience in cooperative R&D projects.
- *q* is the track record in cooperative R&D projects.

- *p* is the on-time delivery of results.

Once the values of the compatibility criteria are determined for each of the feasible solutions, the selection of the best solution (i.e. the most adequate consortium) is done by solving the following integer programming problem:

Equation 10

$$\min \sum_{t} w_t v_t \qquad \text{s.t.} \qquad \begin{gathered} \sum_{i} x_i = 1 \\ \sum_{i} x_i C_{1i} - v_1 = C_{1\min} \\ \sum_{i} x_i C_{2i} - v_2 = C_{2\min} \\ \sum_{i} x_i C_{3i} - v_3 = C_{3\min} \\ C_{t\min} = \min\{C_{ti} : i = (1, \cdots, m)\} \end{gathered}$$

Where:

- x_i is either 1 or 0 (selected/not selected).
- C_{ji} is the value of criteria *j* for solution *i*.
- w_t is the weight of criteria *j* for the best result of *t*.
- v_t is the value of the best result of *t*.

This concludes with a sensitivity analysis, that using the concept of Pareto optimality [35], verifies the robustness of the solution to variations in the selected weights. This allows determining for which intervals in the variation of w_t is the solution still optimal and what are the new best solutions outside these intervals.

The results from this process may be presented to the users in two distinct approaches:

- Result from the optimization and sensitivity analysis – presents the best solution and a measure of its robustness.
- List of best solutions with the possibility to order it by different compatibility criteria – allows the decision maker to select the solution that best fits his perception.

In advanced scenarios, more experience users would like to set a few constraints or boundary conditions on the solution space by, for example, fixating some partners from the start and influencing the final solution with this decision.

2.4. Performance monitoring mechanisms

The partner selection mechanism relies heavily on the profiles of the organizations that exist on the platform and that evolve due to updates performed by the organizations themselves (e.g. new capability) and due to interactions over time with other organizations. In the course of interacting in the preparation of proposal or execution of projects (or other types of interactions) the perceived value of an organization to others will be adjusted.

This performance monitoring mechanism works in a similar way to the reputation mechanism that can be found in social networks or social tools like forums [36], but includes several perspectives for evaluation that can correspond to different types of interactions. The profile of the organizations, partially described in Table II, is updated based on its performance and on the perception the other organizations have of this performance.

Different evaluation possibilities are foreseen, depending on the type of interaction, but the type of behaviours of an organization that can be rated are listed in Table III. These characteristics are part of the organization profile.

Evaluations can also take the form of a recommendation, not related to any interaction registered in the platform but related to previous knowledge. This is helpful for example when a new organization registers in the platform.

2.5. Network coordination mechanisms

The network coordination mechanisms try to guarantee that the reputation of the organizations (perception/evaluation) converges over-time to the real value of the organization. In order to achieve this convergence the systems needs to cope with different forms of evaluation, made by different organizations and triggered by different interactions. These mechanisms are required to:

- Harmonize evaluations triggered by different interactions, with recommendations and also self-evaluations.
- Harmonize over-time evaluations and natural differences in the performance of organizations.
- Harmonize evaluations done by different organizations, with possible different perspectives over good performance.

Two approaches were considered to address these objectives:

- Use of adequate forecast models for time series [37], for example moving averages or exponential smoothing.
- Use of estimation models [38], for example Kalman filtering or particle filters.

Although estimation models are in principle more adequate, the majority were developed with physical systems in mind and considering the existence of Gaussian noise, meaning we would have to assume that the variation in the performance evaluation follows a Gaussian distribution which clear is not the case.

Forecasting models have been used, which are in fact estimation models where the behaviour of the time series is unknown, and bring a certain degree of risk or uncertainty. These models rely on time series, i.e. historical data as a series of values obtained over time, that are used to predict future values. This approach will allow predicting the performance of an organization on its next interaction, which can be seen as what is expected by the others (perception) on its performance.

To compute the reputation of an organization exponential smoothing is used. This method, a particular class of moving average widely used in the financial markets, has the characteristics of being applicable to any discrete set of values.

Since the evaluation mechanisms are using 4 criteria, one of those multi-dimensional, several interactions (including evaluations, recommendations and self-evaluations) will result in a set of historical values for each of these criteria. Considering that the sequence of raw values is $\{x_{it}\}$ and the result of applying the exponential smoothing method is [39], representing the best estimate on the performance of the organization *i* on criteria *j* in its next interaction (reputation). This estimate is determined by

Equation 11

$$r_{it} = \alpha x_{it} + (1 - \alpha) s_{t-1}$$

where α é is the smoothing coefficient ($0 < \alpha < 1$). Values of α closer to 0 have smaller smoothing effect (i.e. recent evaluations have higher impact), contrary to values of α closer to 1 that have a higher smoothing effect (i.e. reputation is less sensitive to the latest evaluations).

In order to harmonize different evaluation types and their impact in the reputation (e.g. a recommendation or self-evaluation as a smaller impact, to prevent cross-recommendation practices in the network) different coefficients are used according to the evaluation type. Typically, the value of α for a recommendation is smaller, closer to half the value of α for an evaluation.

2.6.Experimental Scenarios

In order to analyse the correct implementation and behaviour of the different mechanism some experimental scenarios were defined and run. To satisfy both objectives (test and validation), and since some of the mechanisms require a series of interactions between the participating organizations, an experimental simulation model was developed to simulate the use of the RDNet platform over long periods of time. The experimental simulation approach is extensively used in the social sciences [40] [41] and allows for a fast and inexpensive solution to test the platform with a reasonable number of interactions over a long period of time.

Simulation model

The simulation model used in the study is depicted in Fig. 2. This model is based on the entities organization, idea, consortium, funding, opportunity, and in a set of classes that in every period generate organization evaluations, new ideas, new opportunities. Although not addressed in this paper, the RDNet platform also includes mechanisms that help in identifying funding for ideas: opportunities to turn ideas into projects. The simulation uses the same mechanisms and database used by the RDNet platform.

The simulation starts by populating the database with organizations (including the organization profile, in terms of interests, competencies and reputation), ideas and funding.

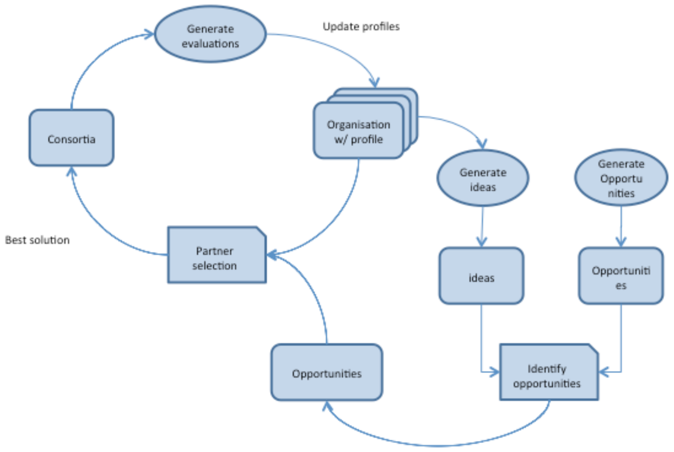


Figure 6: Experimental simulation model

The experimental simulation model generates opportunities, identifies consortia, collects organization performance, updates organization profiles, iterates over several periods. The simulation process is the following.

Simulation setup:

- Create organizations (100 at start)
- Generate funding (uniform distribution – 3)
- Generates ideas per entity (uniform distribution – 3)

This creates the initial context for the simulation. Afterwards, the simulation starts, emulating reality using the following sequence of steps:

- Run funding interest mechanism (identifies organization interested in the areas)
- Generate new ideas in organizations with interest (probability of 0.2)
- Run opportunity matchmaking mechanism (identify opportunities for projects)
- Decide on opportunities (probability of 0.25)
- Define project requirements
- Setup consortia
- Run partner selection mechanisms
- Accept/reject invitation (probability of 0.75)
- Go back to a. if consortium incomplete
- Organization evaluation and profile update
- Go back to 1.

This simulation was implement in Phyton, the same language used to implement the mechanisms in the RDNet platform. Besides the classes needed to control the simulation and collect data for the results, other classes were also developed to simulate the behaviour of the organizations when faced with a decision (e.g. accept invitation for a project?). These decisions were implemented using Bayesian models. Other classes were also developed to randomly populate the database with information form organizations, funding opportunities, ideas, etc.

2.1.Results and analysis

Several simulation runs were performed in order to validate the mechanisms and fine-tune the parameters used. All simulation runs used the same initial scenario, with the parameters presented in Table III.

Table III: Simulation Parameters

Parameter	Value (initial)
Number of organizations	111
Capabilities per organization (average)	5
Connections per organization (average)	8
New funding opportunities per cycle	3
New organizations per cycle	5
New ideas per cycle (per organization)	2
Number of cycles	20

Running several simulations with the same initial conditions, and tweaking with the mechanism parameters allows observing the variations in the evolution of the platform and in the network created by the different organizations. By changing the weights given to the different compatibility criteria, is also possible to understand the behaviour of the partner selection mechanism in the selection of the "best" consortium, and the impact of changes in the smoothing factor on the organization profile.

Whilst the correct parameters in the opportunity and match making mechanisms is related with the sensitivity and specificity of the mechanisms (i.e. capacity to identify true opportunities and to eliminate bogus opportunities), and in the coordination mechanisms is related with getting the right perception of the organizations in the platform, the weights to use for the different compatibility criteria in the partner selection mechanism are of paramount importance because otherwise the best solutions may be neglected. From the simulation results, it is clear that users that understand the problem and are able to set the correct weight for the multi-criteria optimization problem will obtain the best results.

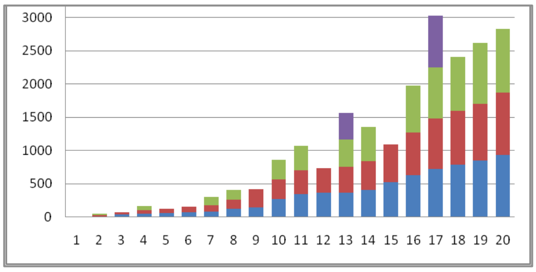


Figure 7: Number of feasible solutions per project (number of feasible solutions per project over time)

Another interesting result form the simulation is depicted in Figure 7. From this graph it is possible to see that the number of feasible solutions (i.e. number of alternative consortia to exploit an opportunity) increases as the organizations create connections between them and the “social” network in the platform becomes denser (i.e. a better knowledge of possible partners exists).

This simulation study was also very helpful to understand how the individual mechanisms behave. As an example, analysing the partner selection mechanism for a project that requires three different scientific-technical capabilities, the first stage selects all organizations that satisfy those areas: in this example organizations with IDs 7, 130, 28, 156, 189, 196, 201, 50, 60, 61, 79). The next stage selects the most efficient organizations per capability area: in this example per are the sets would be the following (7, 130), (189, 201), and (61, 79). The next step starts by determining the solution space (all possible combinations) for the consortium e determine the value of the different solutions. Tab. IV presents partially the results obtained in this example. The choice of the best solution is based on its value, meaning that in this particular example the “best” consortium would be the one composed of entities (7, 189, 50). These results were obtained with weight 1 for every compatibility criteria.

Table IV: Feasible Solutions

Consortium	Value
(7, 189, 50)	1.04
(7, 189, 61)	1.004
..	..
(7, 201, 79)	0.97
..	..
(130, 201, 79)	0.95

3. Plug’n’produce manufacturing systems

The manufacturing enterprises of the 21st century are in an environment in which market demand is frequently changing, new technologies are continuously emerging, and competition is global. Manufacturing strategies should therefore shift to support global competitiveness, new product innovation and customization, and rapid market responsiveness.

The next generation manufacturing systems will thus be more strongly time-oriented (or highly responsive), while still focusing on cost and quality. Such manufacturing systems will need to satisfy a number of fundamental requirements, including [42]: Full integration of heterogeneous software and hardware systems within an enterprise, a virtual enterprise, or across a supply chain; Open system architecture to accommodate new subsystems (software, hardware, *peopleware*) or dismantle existing subsystems “on the fly”; Efficient and effective communication and cooperation among different elements (units, lines, cells, equipment) within an enterprise and among enterprises; Embodiment of human factors into manufacturing systems; Quick response to external order changes and unexpected disturbances from both internal and external manufacturing environments; Fault tolerance both at the system level and at the subsystem level so as to detect and recover from system failures and minimize their impacts on the overall performance.

The new manufacturing paradigm of the 21st century contains many challenges that emerge in complex system formation, being characterized by strong couplings between the different operational and enterprise issues. Traditional approaches for the organization of manufacturing systems is the hierarchical approach, which is top-down and strictly defines the system modules and their functionalities. Some possible alternative approaches to address these challenges are presented in the next sections.

Modern Industries have a continuous need to satisfy their markets at better costs in order to keep competitive. This simple fact creates the continuous need for new products, new production lines and new control methodologies. The XPRESS (FleXible PRoduction Experts for reconfigurable aSSembly technology) project [43], a cooperative European project involving industry and academia, studied this issue in order to define a new flexible production concept. This concept, based on specialized intelligent process units, called *manufactrons*, is able to integrate a complete process chain, and includes support for production configuration, multi-variant production lines and 100% quality monitoring. The concept was demonstrated for the automotive, aeronautics and electrical component industries, but it can be transferred to nearly all production processes. This concept will be further explored in section III.

3.1.Reconfigurable manufacturing systems

Reconfigurability has been an issue in computing and robotics for many years. In general, reconfigurability is the ability to repeatedly change and rearrange the components of a system in a cost-effective way. Koren *et al.* [39] define a Reconfigurable Manufacturing Systems (RMS) as being "[..] designed at the outset for rapid change in structure, as well as in hardware and software components, in order to quickly adjust production capacity and functionality [..] in response to sudden changes in market or in regulatory requirements". Merhabi *et al.* [4] complement this definition with the notion that "reconfiguration allows adding, removing or modifying specific process capabilities, controls, software, or machine structure to adjust production capacity in response to changing market demands or technologies [..] provides customised flexibility [..] so that it can be improved, upgraded and reconfigured, rather than replaced".

RMS are seen as a cost-effective response to market changes, that tries to combine the high throughput of dedicated production with the flexibility of flexible manufacturing systems (FMS), and is also able to react to changes quickly and efficiently. For this to be accomplished, the system and its machines have to be adapted for an adjustable structure that enables system scalability in response to market demands and system/machine adaptability to new products. An RMS is composed of reconfigurable machines and open architecture reconfigurable control systems to produce variety of parts with family relationships. Structure may be adjusted at the system level (e.g., adding/removing machines) and at the machine level (changing machine hardware, control software or parameters).

3.2. Industrial applications of agent systems

Duffie and Piper [44] were one of the first to discuss and introduce a non hierarchical control approach, using agents to represent physical resources, parts and human operators, and implementing scheduling oriented to the parts. Yet another manufacturing system (YAMS), introduced by Parunak et al. [45], applies a contract net technique to a hierarchical model of manufacturing system, including agents to represent the shop floor. The autonomous agents at Rock Island Arsenal (AARIA) [46] control a production system with the goal to fulfil incoming tasks in due time, focusing on the dynamic scheduling, dynamic reconfiguration and in the control of manufacturing systems that fulfil the delivering dates. The manufacturing resources, processes and operations are encapsulated as agents using an autonomous agent approach.

Some relevant approaches have been introduced in this domain. The product resource order staff architecture (PROSA), proposed by Brussel et al. [5], is a holonic reference architecture for manufacturing systems, which uses holons to represent products, resources, orders and logical activities. Gonçalves et al. [47] presented an approach based on co-operating agents to the reengineering production facilities. The approach focus on several aspects related to enterprise dynamic reconfiguration due to product redesign or changing demand, and on optimizing the production process or removing errors that might have emerged.

In spite of all the research described above, only few industrial/laboratorial applications were developed and reported in the literature. Bussmann and Schild [48], as part of the *Production 2000+* project, use agent technology to design a flexible and robust production system for large series manufacturing that meet rapidly changing operations in a factory plant of DaimlerChrysler, producing cylinder heads for four-cylinder diesel engines. This agent-oriented collaborative control system, called FactoryBroker, proved to be useful to control widely distributed and heterogeneous devices in environments that are prone to disruptions and where hard real-time constraints are crucial.

Cooperative Engineering concerns the application of Concurrent Engineering techniques to the design and development of products and of their manufacturing systems by a network of companies coming together exclusively for that purpose. Gonçalves et al. [49] present an implementation of a framework for Cooperative Engineering based on a general framework of distributed hybrid systems and MAS.

More examples of agent-based approaches to manufacturing systems can be found in [50] , [51] and [52].

3.3.Reconfigurability in the Network Factory

This section presents a realization of a networked factory based on a multi-agent systems framework to implement the concept of re-configurable factory. Its contributions and limitations are discussed, along with the roadmap for future improvements.

The goal of XPRESS was to realize an Intelligent Manufacturing System (IMS) and to establish a breakthrough for the factory of the future, with a new flexible assembly and manufacturing concept based on the generic idea of “specialized intelligent process units” (referred to as *manufactrons* in the context of XPRESS) integrated in cross-sectorial learning networks for customized production and flexible system organization. This knowledge-based concept integrates the complete process hierarchy, from the production planning to the assembly, the quality assurance of the produced/assembled products and the reusability of process units. Different functionalities within a factory are encapsulated in specialized intelligent process units called “*manufactrons*”. By doing so, the single *manufactron* is able to perform the assigned tasks optimally within linked networks by considering their knowledge. The mechanisms of self-learning, self-organization, knowledge acquisition (experiments) as well as the use of shared communication opportunities, which are required for performing successfully, are stored in every *manufactrons*.

Intelligent Process Units

A *manufactron* is a self-contained entity, which encapsulates expertise and functionalities, and that interacts with its environment by the exchange of standardized synchronous messages. Being self-contained, it is expected that a typical *manufactron* can be included to a networked factory by just plugging an additional device (the *manufactron*) into the factory’s network. Therefore, the *manufactron* has to be realized as an independent component (comprising software and hardware) rather than a distributed set of parts, where a lot of different parts of the component are to be integrated into different systems of the factory – Enterprise Resource Planning (ERP), Manufacturing Execution Systems (MES), or different kinds of Programmable Logic Controller (PLC) systems [53].

The *manufactron* shall not only realize a simple functionality, but shall also provide expertise on this functionality to the outer world. This allows the outer world to state a task to be fulfilled to the *manufactron* without the need to know about every small detail associated with these tasks. The encapsulation of expertise is therefore the solution to demands stated by multi-variant production (higher levels do not have to concern about small details) and flexibility in terms of production resources (a task is not depending on a very special welding machine, but can be understood by every welding machine).

The *manufactrons* can be seen as autonomous agents, able to decide the best way to reach their given goals, but not when to do it. The task execution is triggered from outside as defined by a *manufactron* from a specific category, named “workflow manager”, responsible for overlooking the factory level with dedicated knowledge expertise [54]. This results in a *manufactronic* hierarchy (Fig. 1): (Field level) “Production *manufactrons*” (executing basic manufacturing tasks) and “Super *manufactrons*” (coordinating groups of Production *manufactrons*); (Factory level) “Workflow managers” (controlling the production flow of an item) conforming the manufacturing execution system up to production planning; (Planning level) “Configuration *manufactrons*” responsible for finding an optimum production configuration and for the creation of workflow managers for different product variants or for varying production conditions.

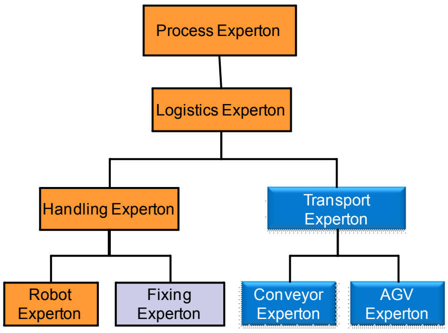


Figure 8: *Manufactronic* hierarchy (class hierarchy)

The lowest level of the *manufactronic* hierarchy is the single *manufactron*. In this context a distinction is made between “Production *manufactron*” (PM), “Handling *manufactron*” (HM), “Transport *manufactron*” (TM) and “Super *manufactron*” (SM). The last ones demonstrate a kind of cell-representation and can be considered as compound of single sub-*manufactrons*.

Communication

Communication between different systems is a major challenge in industrial environments. Most communication channels are particularly tailored to different systems and are often proprietary. Hence, integration of equipment requires additional engineering and makes it difficult the simple replacement of systems. On the other hand, if standard connections are used, the process slows down in most cases and finally just covers a subset of the necessary functionalities [53]. A generic understandable task description, describing the production tasks to be performed by a particular machine for a certain class of products can be a solution for this problem. The basic approach of the manufactron communication scheme is a synchronous exchange of documents. For that, only three types of documents exist: Task

Description Documents (TDD); Quality Result Documents (QRD); and Manufactron Self Description (MSD). This approach leaded to the development of an uniform and standardized communication protocol for the manufactronic framework. For that purpose, a XML based approach has been chosen, which guarantee a very flexible and extensible system, being at the same time powerful enough to handle all data and signals to be transported between system components.

TDDs provide input information for the *manufactron*, including all information needed to execute a task. This includes the information about what is to be done, the task goals as well as specific boundary conditions for task performing. The TDD is a XML-based document compliant with a pre-specified schema and follows a hierarchical structure. Considering that a large set of different *manufactrons* could be used in a factory shop-floor, a multi-level approach was followed, where the top-level specification of the TDD is only containing some data, which is the same for all different requirements. This top-level document then may be extended with different components that add more detailed specifications. These components are dependent of the *manufactron* type itself, which are the recipient of the TDD.

QRDs are released by the *manufactrons* after they received a TDD and performed the task. QRDs not only contain quality data information but can include any type of data which is the result of performing a task. For example, the QRD of performing a welding job could be the quality of each welding spot. As already mentioned, the TDD/QRD mechanism is a synchronous communication protocol. Therefore, each TDD request will lead to a QRD answer. After receiving a TDD, the *manufactron* will be blocked (no other TDDs will be received) until the operation ends with releasing a QRD. The QRD is also based in a XML schema.

This document enables manufactrons to describe themselves in relation to the system and specify which tasks they can execute, the data inputs they require for the task execution, as well as which product and process data they are able to provide. The MSD is stored and managed by the manufactron and therefore in the production equipment itself, thus being an enabler of “plug’n’produce” processes [54]. Each manufactron or other entity in the manufactronic factory can request the MSD from a manufactron. The following main information is included in an MSD: Information on the capabilities of the manufactron; Information on the quality result items generated by the manufactron after the execution of a task or simulation; Information on the TDD the manufactron expects. The result of a request of the MSD could be the information on the status or the current configuration of the manufactron.

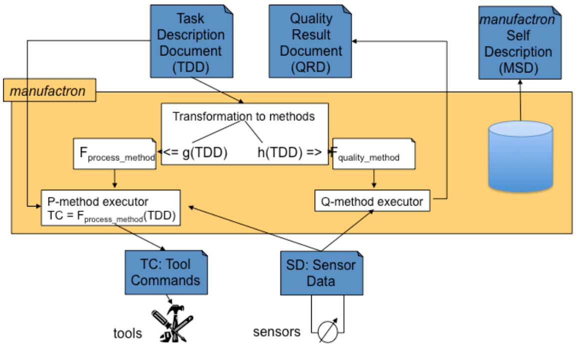


Figure 9: ***Manufactron*** **components**

Manufactronic Networks

The *manufactrons* are hierarchized into three categories according to their function: (Configuration *manufactrons*) responsible for finding an optimum production configuration and for the creation of a workflow manager template that can be instantiated to produce the product variant; (Workflow manager) controls the production flow of an item according to the workflow manager template; (Production *manufactrons*) responsible for executing basic manufacturing tasks and/or for coordinating groups of production manufactrons.

A major challenge of the approach is the interaction of the different components of the whole system. The communication scheme between components of the different layers (ERP, shop floor and cell level) and also within the layers must be powerful, flexible and extensible. The concept of *manufactronic* network comprises the Production Configuration System (PCS), the Workflow Execution System (WES), and the lower level *manufactrons*: Super *manufactron*, Production *manufactron* and Handling *manufactron*.

The PCS is divided in three components: production simulation system (PSS), production execution system (PES), and finally production quality system (PQS). The PSS performs simulation tasks, using different workflows with various production *manufactrons* and configurations. On the other hand, the PES is responsible for receiving and selecting the best configuration from production jobs issued by external ordering systems, such as SAP or Baan. Regarding PQS, this component is responsible for storing and retrieving the quality results in XML formatted files denominated quality result documents (QRDs), which are generated at the end of the production cycle and contain the complete quality information of the entire production process and the product itself.

The WES, instantiated by the PCS during the simulation phase or production phase, consists of a workflow manager (WFM) and a quality manager (QM). This component, the WES, is the mediator between the PCS and all the other production *manufactrons* (PMs), handling *manufactrons* (HMs) or super *manufactrons* (SMs). Each started instance of WFM or QM is responsible for the control and organization of the *manufactrons* related to the process. This allows the WES to suspend or to persist the *manufactrons*, if no activity is to be performed. It is the responsibility of every *manufactron* to communicate with lower or higher level *manufactrons* (SMs or WES “manufactron”). As far as the communication goes, it is via the exchange of XML data between the components and the system. The system’s communication is synchronous, therefore each TDD sent to a *manufactron* must result in a QRD. In case that the operation is not performed, a QRD containing an error message must be sent to upper level.

A production system implemented via a *manufactronic* network, in which several production equipment, and therefore *manufactrons* are considered to execute a process step, the Production Configuration System (PCS) collects the different specifications and generates a TDD. This file can then be understood by all *manufactrons* that are considered for the process. The structure of MSD and TDD documents is defined in such way that the integration and transformation can take place as easy and unambiguous as possible. An overview of the *manufactronic* architecture with the communication between layers is given in Fig. 3. During production, the Workflow Execution System (WES) sends the TDD to a particular manufactron (production equipment). Ideally, this happens simultaneously with the loading of the work piece. Due to the fact that it possesses all the necessary information, the manufactron should now be able to execute the process step successfully. The task description is a high-level document and should not be mistaken for a batch sheet or recipe: in most cases the task description is less extensive but at the same time more flexible than a pure batch sheet specification. At the end of the process step the product and quality data are returned to the WES simultaneously with the physical unloading of the work piece. The shape of the QRD sent to the WES is also predetermined by the MSD in order to ease the analysis of the resulting quality.

The radical innovations of the “*Manufactronic* Networked Factory” are knowledge and responsibility segregation, trans-sectoral process learning in specialist knowledge networks. The concept is built on coordinated teams of specialized autonomous objects (*manufactrons*), each knowing how to do a certain process optimally. They have the intelligence to choose the best-known production parameters for a given task. Assembly units composed of *manufactrons* can flexibly perform varying types of complex tasks, whereas today this is limited to a few pre-defined tasks. By sharing the specific knowledge of each *manufactron* in a network, other *manufactrons* are able to learn from each other in one production line, but also between different lines as well as different production units. This architecture allows continuous process improvement, and therefore the system is able to anticipate and to respond to rapidly changing consumer needs, producing high-quality products in adequate quantities while reducing costs.

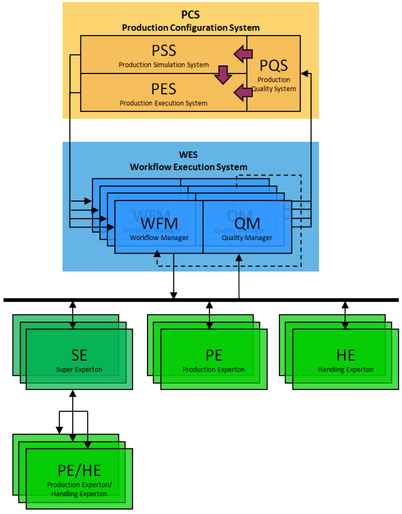


Figure 10: ***Manufactronic*** **Network**

3.4.Multi-Agent Implementation of the Networked Factory

As explained in the aforementioned sections, one of the steps forward on the reconfigurability in networked factories is the encapsulation of the equipment with software, extending it with communication capabilities and intelligent functionalities, such as negotiation. This kind of approach will allow not only the inter-equipment communication and collaboration, but also the communication between the shop-floor equipment and any software component, assuming it is also encapsulated with the same technology. This will leverage a much more flexible and effective way of equipment configuration, paving the way for the Network Factory implementation, and therefore, the shop-floor reconfigurability.

This way, a simple MAS was developed to mimic the pertinent behaviours and interactions between the most important Manufactronic components, and thus, analyse and predict the problems that might occur in a real industrial environment, at a collaborative and cooperative level. As can be seen from Fig. 3, there are three different levels of abstraction present in the Manufactronic Network, but only the first and the last ones were considered for the MAS modelling. This selection lies on the fact that only problems on the shop-floor reconfiguration will be analysed, not considering if the production is running well or not (monitoring and controlling), but instead, take into account the negotiation and collaborative abilities to verify if the requirements for fast shop-floor reconfiguration are met, in the presence of a new product variant.

Therefore, *Configuration Manufactron* and *Production Manufactron* Agents were developed, and as explained in Section III, the first one is responsible to find the optimum production configuration according to some product requirements, and latter one is intended to execute the basic manufacturing tasks. Hence, in terms of information flow, whenever a *Production Manufactron Agent* enters into the network, it should be able to generate a MSD, and send it to the already existing *Configuration Manufactron Agents*, so they can know how the shop-floor can be configured using the available equipment according to some product requirements. The first step towards the production process is related with the information sent to a certain *Configuration Manufactron Agent* about the product specifications, and the generation of the corresponding TDD to subsequently send it to the available *Production Manufactrons Agent* with the matching capabilities, for shop-floor operation. Furthermore, when the *Production Manufactrons Agents* finish their operation on the production process, the next step is the generation of the QRD that is then sent to the *Configuration Manufactron Agent* to update and report the information about the equipment's production performance. This quality feedback will drastically influence the selection of the available *Production Manufactrons* in the optimum production configuration, benefiting the equipment with better performances, tending, this way, to choose the most reliable and effective ones.

As previously mentioned, one of the MAS purposes is to study the problems associated with collaborative activities like the ones described earlier, when the *Configuration Manufactron Agent* delegates TDDs to *Production Manufactron Agents* to act accordingly, and subsequent feedback to report the process quality by means of QRD. However, most of the collaborative abilities can lead to a conflict situation, mainly when two different entities are trying to establish a partnership with the same third party. In the context of Network Factory, this can occur when there are several instances of *Configuration Manufactrons* that can include in their optimum production configuration the same *Production Manufactron* to operate on the shop-floor level, if this search is made concurrently.

One of the techniques associated for conflict resolution is the market-based negotiation. This concept can be simply explained as the increase of a resource cost until only one "costumer" is willing to pay for the achieved price. For the implementation of this technique, *Utility*, *Cost* and *Threshold* functions were built to measure the overall usefulness of using a certain *Production Manufactron* on the production configuration. The first one measures how distant an equipment operation is from the ideal product specification, the second one returns a value of how much an equipment execution can cost (not its actual running cost, but only a measure representative for this problem) based on QRDs information – as much worse the equipment performance is, the higher is the cost associated to it, and the latter one is how much an agent is willing to pay, based on the utility previously calculated – if the utility is high, the threshold value will also be, and vice-versa. Hence, when the same *Production Manufactron Agent* is the most suitable one for different *Configuration Manufactron Agents*, the cost of *Production Manufactron Agent*'s execution will be increase, until only one *Configuration Manufactron Agent* remains with the threshold value above the cost.

3.5.Results from the multi-agent implementation

The strategies presented on the previous sections regarding MAS, along with the agent paradigm and well structured communication processes (MSD, TDD and QRD), proved to be an effective and reliable approach, since some of the problems that arise from equipment collaboration were studied and successfully solved using the market-based negotiation approach. The modelled MAS represents a short step forward, but not less important, towards a flexible and extensible production reconfiguration, taking into account the complex industrial dynamics and heterogeneous environments.

One of the most important advantages of the MAS characteristics is undoubtedly the decentralized approach that verifies the fault tolerant property, in case of sudden equipment failure. The networked factory will maintain its communication and collaboration activities, avoiding stopping the production process due to component non-dependency issues, minimizing costs and maximizing the network reliability. Another important concept presented in this paper is the task-driven communication, in which equipment execution on shop-floor level are specified in XML-based format, and used to delegate responsibilities for operation according to precise specifications (TDD), and receive a valuable feedback on the equipment quality execution (QRD). Comparing with manual reconfigurability, which in turn reveals to be not cost effective, this concept is an important step forward regarding the automatic reconfiguration of equipment for shop-floor operation.

3.6.Limitations and future extensions of the approach

The main goal of the work presented in this paper is to provide methods, that can be either fully automated or an aid to the planning engineer, that select which *manufactrons* to use for a specific job (new product or variant); this will answer the question, which is the best configuration for this task?

From the modules that build the configuration *manufactron*, the Production Simulation System (PSS) is the responsible for the creation of new configurations to answer a specific Job description. The assignment problem is a special type of linear programming problem where resources are being assigned to perform tasks [20]. There is a simple algorithm to efficiently evaluate the solution. This algorithm is known as the Hungarian Method and is able to retrieve the best set of *manufactrons* for a set of tasks. However, these approaches are not helpful in the present context mainly due to the fact that the data made available by the *manufactron* (each *manufactron* provides a self description document with its typical production capabilities, times and quality levels) does not take into account the impact of working in tandem with other *manufactrons*. This is the main reason to include a simulation tool on the decision process. To be effective, this tool has to be able to analyse several hundreds of different line configurations. A specific data development analysis model referred to as Charnes, Cooper and Rhodes (CCR) [28] model is a fractional programming technique that evaluates the relative efficiency of homogeneous decision making units, in our case, the relative efficiency of *manufactrons*. The general efficiency measure, which will be referred as the cross-reference comparison, is presented in Equation 12.

Equation 12

$$\mathrm{E_{ks}} = \frac{\sum_{\mathrm{y}} \mathrm{O_{sy}} \mathrm{v_{ky}}}{\sum_{\mathrm{x}} \mathrm{I_{sx}} \mathrm{u_{kx}}}$$

where: $\boldsymbol{O}_{sy}$ are the output measures $\boldsymbol{y}$ of the *manufactron* $\boldsymbol{s}$; $\boldsymbol{v}_{ky}$ are the weights of the ”target” *manufactron* $\boldsymbol{k}$ to output $\boldsymbol{y}$; $\boldsymbol{I}_{sx}$ are the input measures $\boldsymbol{x}$ of the *manufactron* $\boldsymbol{s}$; $\boldsymbol{u}_{kx}$ are the weights of the ”target” *manufactron* $\boldsymbol{k}$ to input $\boldsymbol{x}$; $\boldsymbol{E}_{ks}$ is the cross-efficiency of *manufactron* $\boldsymbol{s}$, using the weights of ”target” *manufactron* $\boldsymbol{k}$.

An optimal value $\boldsymbol{E}^{*}_{kk}$ for the cross-reference comparison is obtained by maximizing Equation 13:

Equation 13

$$\mathrm{E^{*}_{kk}} = \frac{\sum_{\mathrm{k}} \mathrm{O_{ky}} \mathrm{v_{ky}}}{\sum_{\mathrm{k}} \mathrm{I_{kx}} \mathrm{u_{kx}}}$$

subjet to:

$$\mathrm{E_{ks}} = \frac{\sum_{\mathrm{y}} \mathrm{O_{sy}} \mathrm{v_{ky}}}{\sum_{\mathrm{x}} \mathrm{I_{sx}} \mathrm{u_{kx}}} \leq 1 \; \forall \, s$$

$$v_{ky} \geq 0, u_{kx} \geq 0, and \sum_{x} I_{sx} v_{kx} = 1$$

If $\boldsymbol{E}^*_{kk}$ is equal to 1 then there is no other *manufactron* which is better than *manufactron* ***k*** for its optimal weights. Solving this optimization to all the *manufactrons*, then it is possible to select the ones that are not optimal ($\boldsymbol{E}^*_{kk} < 1$) and remove them from the solution space.

The cross reference comparison leads to Pareto optimal solutions but it is not a sufficient condition, because it eliminates solutions which are strictly better than them with respect to, at least, one objective but it cannot guarantee that it eliminates a solution **A** when another feasible solution is, at least, as good as **A** with respect to some objectives and strictly better than **A** with respect to, at least, one objective.

4. Networked vehicle systems

A system might be defined as a group of components that work together for a specified purpose. This is a very simple but accurate definition. Being purposeful action a basic characteristic of any system, a number of functions must be implemented in order to achieve these purposes. This means that a system is a group of components that works together and have functions designed to execute specific tasks [55].

The idea of a system composed of a group of systems seems appropriate to capture the essential aspects of operation of networked systems with mixed initiative interactions. The observation is that the components in the network are part of a system, within which new properties arise, some of them planned, some of them emergent, and eventually leading to unpredictable behaviours. Moreover, since communication is not necessarily available, or instantaneous, the current state of the system – a network of systems with evolving structure – is not always accessible.

In a system of systems, a significant part of the "system" is embodied not as physical devices, such as sensors, actuators or communication networks, but as software that may be mobile, in the sense of migrating from one processing unit to another, as part of the evolution of the system.

For example, in coastal and harbour surveillance missions the mission environment evolves in multiple temporal and spatial scales as the result of complex interactions. Sensors are required to take measurements with adequate temporal and spatial resolutions, and the measurements may have to be communicated in real-time to adapt the sampling strategies (both temporally and spatially) to the observations. In summary, distributed sensing with mobile nodes has to be complemented with communications and real-time decision-making. This is a good example where the definition of networked vehicle system is applicable.

4.1. Unmanned vehicle systems

Networking is one of the major trends for unmanned vehicle systems; it is also one of the enabling technologies for distributed cooperation in unmanned vehicle systems. In the reminder of this section we use the designation "network vehicle systems" to describe systems where vehicles, sensors and operators interact through (inter-operated) communication networks.

Networked vehicle systems offer new possibilities to the operation of unmanned vehicles [56]. For example, in networked vehicle systems, information and commands are exchanged among multiple vehicles, sensors and operators, and the roles, relative positions, and dependencies of those vehicles and systems change during operations. These capabilities are essential for operations where the temporal and spatial coordination of vehicles is required, such as in environmental field studies and in surveillance missions. However, we are still far from realizing the potential of networked vehicle systems. Consider the case of an environmental disaster spanning a wide geographical area. With the current technologies, tools and models, it is simply not possible to inter-operate vehicles, sensors and communication networks from different vendors/institutions: although there are (multiple) standards for inter-operability, the capability to use multiple devices in a "plug-and-survey" approach is still not available.

Wireless sensor networks [57] are a major technological trend that is already impacting environmental field studies [58, 59]. The developments on miniaturization and power consumption are to accelerate this trend towards massive deployments thus enabling studies with unprecedented spatial and temporal resolution. A promising technological push comes from the inter-operation of vehicle systems with sensor networks [60]. This combines the coverage of sensor networks with fixed nodes, with the level of adaptation and detailed resolution provided by sensors mounted on vehicles.

Researchers and technology developers are devoting significant efforts to the development of concepts of operation for networked vehicle systems. Surprisingly, or not, the role of human operators is receiving significant attention in the development of concepts of operation for networked vehicle systems. In fact, this is the reason why researchers and technology developers have introduced the concept of mixed initiative interactions where planning procedures and execution control must allow intervention by experienced human operators. In part this is because essential experience and operational insight of these operators cannot be reflected in mathematical models, so the operators must approve or modify the plan and the execution [61]. Also, it is impossible to design vehicle and team controllers that can respond satisfactorily to every possible contingency. In unforeseen situations, these controllers may ask the human operators for advice.

Recent technological advances led to the creation of very capable unmanned systems built using low cost hardware. This allows the application of these technologies to scenarios where multiple unmanned systems can be employed simultaneously like patrolling, adaptive sensing, search and rescue, etc. However, human operators have turned into an increasingly scarcer and more expensive resource whose exploitation shall be optimized.

As we have seen previously the idea of a system of systems seems appropriate to capture the essential aspects of operation of networked vehicle systems. The challenges this approach poses, in the multiple disciplines as for example robotics, control, computer and communication, entail a shift in the focus of existing methodologies: from prescribing and commanding the behaviour of isolated systems to prescribing and commanding the behaviour of networked systems. These advances can only be achieved by adopting an inherently inter-disciplinary approach, bringing together researchers from traditionally separate communities to work on problems at the forefront of science and technology. Systems Engineering has an instrumental role in this approach.

4.2. Human operators and level of autonomy

In this chapter, we describe a conceptual framework for inclusion of the operator in a scenario were mixed initiative interactions are favoured, the human operator is in the control loop of the autonomous system(s), and the application of this framework to a Command and Control (C2) interface. The objective is to identify the best possible arrangement for a decentralized team of operators controlling multiple Unmanned Aerial Vehicles (UAVs) in order to distribute and reduce the workload. To achieve this objective the operator is advise on the best action and the C2 layout is automatically reconfigured. The operator can have different levels of situation awareness, at different stages of the mission. The system will help operators to dynamically configure an optimal view of the mission state from a set of predefined console layout profiles.

An adaption of the Level Of Autonomy (LOA) matrix, presented in Table V, will be used as the framework for the inclusion of the human operators in mixed initiative scenarios. The LOA-Level of Autonomy matrix [62] is based on Sheridan's 10-level of autonomy scale [63] and simplified to present only eight levels of autonomy. The two dimensions of the matrix are the eight levels (matrix rows) crossed with four functional categories (matrix columns). The second dimension presented in this matrix is the division of each task into four functional steps. These tasks present human decision-making processes as a set of OODA

(Observe, Orient, Decide, and Act) cycles as prescribed by Boyd [64]. The OODA loop, originally developed for strategic military requirements, was adapted for business and public sector operational continuity planning, for example into to the "Deming Cycle" also know as the "Plan-Do-Check-Act" (PDCA) cycle [65].

The framework present in Table VI is used to categorize the operator skills using the LOAs he is certified to respond to, the operator Console Profile he is trained with and the number of vehicles he can handle safely at a certain LOA.

To exemplify the framework's execution we will evaluate a mission scenario where the operators have to find a target and follow it. There will be two operators and five UAVs in this scenario.

Table V: Partial LOA matrix (originally published in [62])

Level	Observe	Orient	Decide	Act
8	The computer gathers, filters, and prioritizes data without displaying any information to the human.	The computer predicts, interprets, and integrates data into a result which is not displayed to the human.	The computer performs ranking tasks. The computer performs final ranking, but does not display results to the human.	Computer executes automatically and does not allow any human interaction.
7	The computer gathers, filters, and prioritizes data without displaying any information to the human. Though, a "program functioning" flag is displayed.	The computer analyses, predicts, interprets, and integrates data into a result which is only displayed to the human if result fits programmed context.	The computer performs ranking tasks. The computer performs final ranking and displays a reduced set of ranked options. Without displaying "why".	Computer executes automatically and only informs the human if required by context. It allows for override ability after execution. Human is for shadow contingencies.
		…		
		…		
1	Human is the only source for gathering and monitoring (defined as filtering and prioritizing) all data.	Human is responsible for analysing all data, making predictions and interpretation of the data.	The automate does not assist in or perform ranking tasks. Human must do it all.	Human alone can execute decision.

Table VI: Fields used to infer about the operators skills in the framework

Certified Type of LOA	Certified Consoles Profiles	Number of Vehicles
The LOA the operator is certified to operate at.	Set of operational Consoles the operator is certified to use by preference order (per LOA).	Operator fan-out of vehicles (for one LOA)

Currently existing UAVs offer little adaptability in terms of automation: operators can command the UAV to fly autonomously, following a pre-defined flight path, or they can control it manually. For this example we will use 2 LOAs for the operators, and another one of full autonomy used in handover and in emergency situations. The operators LOAs to be used are further sub-divided into a high level control LOA and low level control LOA in this scenario.

All three LOAs used are described as follows:

- **Operational Mode 1 –** Tele-Operation or Direct Control – LOA=(3,2,2,2);
- **Operational Mode 2 –** Survey – LOA=(6,6,7,6);
- **Operational Mode 3 –** Full Autonomy – LOA=(8,8,8,8).

The matrix from Table V can be related with the different types of console profiles. Different console profiles can be associated to different combinations of the four functional categories (OODA) – operational modes. For the presented framework we have a direct relation of LOA and CP. The formal representation for CP-LOA tuple is:

$$\text{CP-LOA}=(\{Obs_1\ldots Obs_n\},\{Ori_1\ldots Ori_n\},\{Dec_1\ldots Dec_n\},\{Act_1\ldots Act_n\})$$

The elements on the tuple are represented as sets so we can group the OODA functional categories. This way it is possible to have one CP capable of handling different Operational Modes.

We will use two CPs (CP1=({3},{2},{2},{2}) and CP2=({6-7},{6-7},{6-7},{6-7})) to handle this mission example as follow:

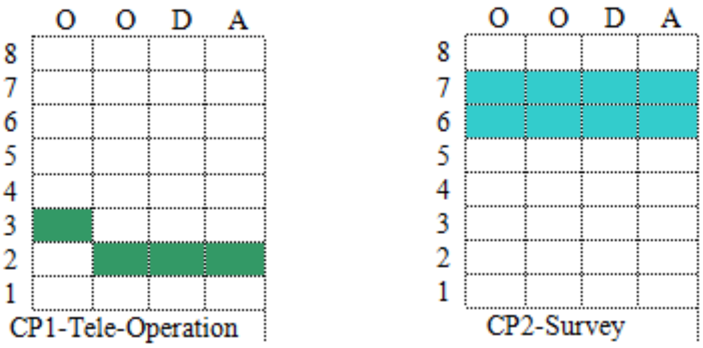


Figure 11: Two Console Profiles used in mission (For Low and High Level Control)

For this mission example we will have two operators with the following Skills (Table VII): Operator 1 can handle 3 UAVs in high level control and 1 UAV in low level control. Operator 2 can handle 4 UAVs in high-level control

Table VII: Skills Table

	Certified Type of LOA	Certified CPs (Consoles Profiles)	Number of Vehicles
Operator 1	(3,2,2,2)	{CP1}	1
	(6,6,7,6)	{CP2}	3
Operator 2	(6,6,7,6)	[CP2}	4

Figure 12 is an illustrative example of this framework in action. The state of the system before any of the operators finds the target is the beginning step (step 1). Initially, all the UAVs are in survey mode – mode 2 of the LOA definition. Both of the operators are using CP2 to control the UAVs: define survey areas and look at the payload data (video).

In step 2, Operator 1 finds the target. The target must be followed using direct control. To solve the excessive workload of Operator 1 (according to Table VII operator 1 can handle only 1 UAV in Operational Mode 1 – Tele-operation – and operator 2 is not certified for Operational Mode 1), the system (mission supervisor) will try to assign this UAV in mode 1 - Tele-Operation - to some operator. The only operator capable of handling mode 1 is operator 1. Since operator 1 is capable of handling only one UAV in this mode, the mission supervisor will advise operator 1 to hand-over the other 2 UAVs to operator 2. Here starts step 3 with the handover process: operator 1 releases the two controlled UAVs by setting them at mode 3 (Full Autonomy).

Finally, in step 5, Operator 2, that has accepted the hand-hover, takes over these UAVs that are in mode 2 and the operator 1 can now handle mode 1 (Tele-Operation) and follow the target. In this step the Mission Supervisor advises operator 1 to use CP1-Tele-operation to respond mode 1 LOA, which requires full attention to the vehicle, according to his skills.

4.3. Command and control in muti-UV systems

The concepts of operation for multi-UAV teams differ from single UAVs in the sense that in the former there exist common objectives like maintaining a common knowledge database [66] and redundant execution of crucial actions [67].

In our C2 framework, UAVs can be tasked either individually by an operator or they can be tasked by a software agent that acts as an operator (Team Supervisor). The team supervisor divides work among the vehicles according to a multi-UAV mission specification and simple task-allocation algorithms. If the control over the UAV is not overridden, they carry out planned behaviour until they are faced with failures, or there are any other unpredicted situations in which they contact the ground station and require human intervention.

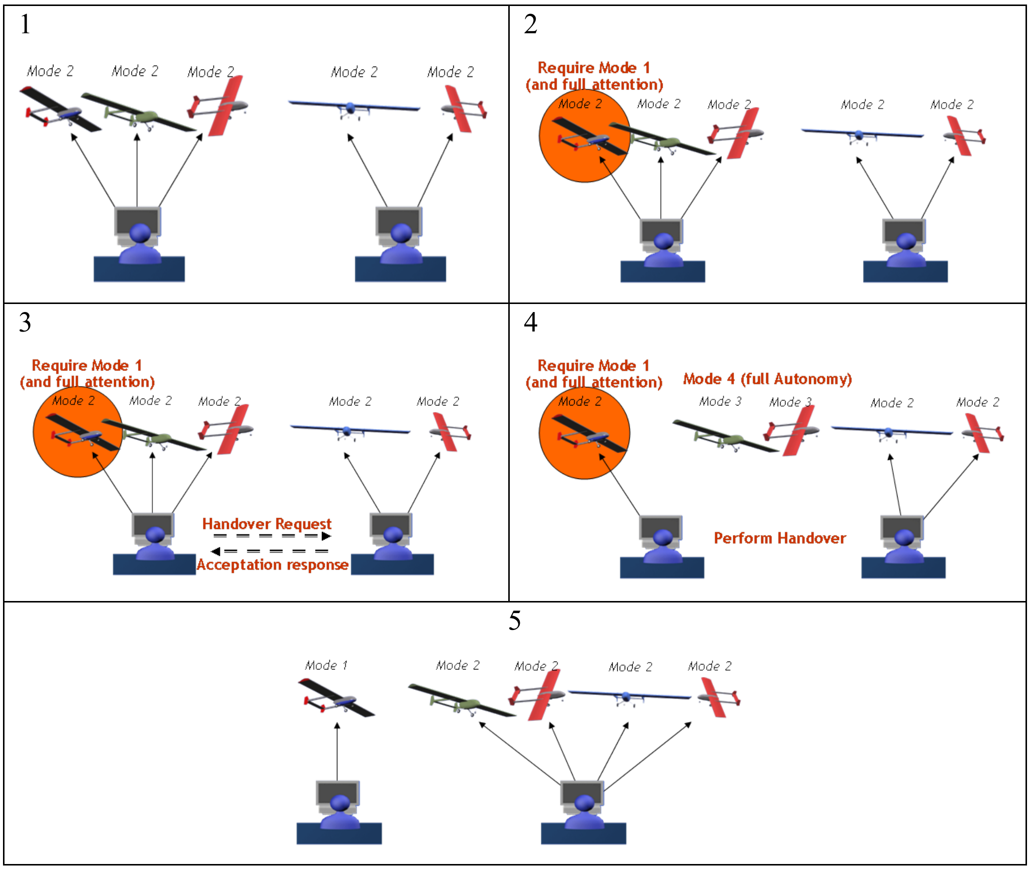


Figure 12: Example of mission workload distribution

To provide system-level control of multiple vehicles, we use a software agent that holds a multi-UAV mission specification. This mission specification is currently a list of individual plans that need to be executed by UAVs. Tasks are divided among UAVs in a way that workload is shared among capable vehicles. Some tasks however also require the intervention of human operators for correct execution, so the availability of operators must be taken into account by the team supervisor while tasking the network.

As stated before, this framework was employed in an existing C2 software framework: Neptus. Neptus has an underlining architecture that provides the means for creating the various consoles used in different CP's. This section introduces Neptus and gives an example of such consoles.

Neptus is a distributed C2 framework for operations with networked vehicles, systems, and human operators. Neptus supports all the phases of a mission's life cycle: planning, simulation, execution, and post-mission analysis. Neptus supports concurrent operations. Vehicles, operators, and operator consoles come and go. Operators are able to plan and supervise missions concurrently [68].

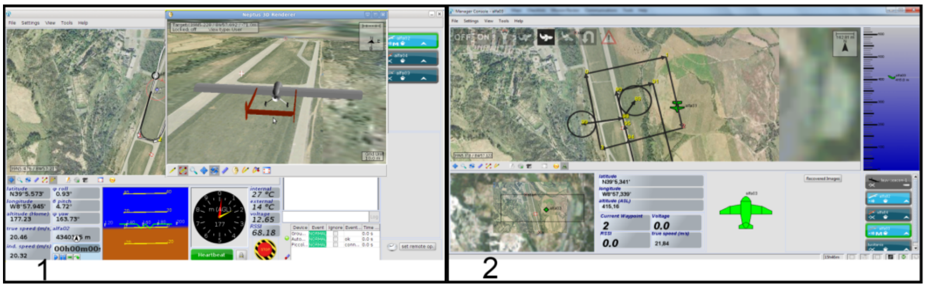


Figure 13: Tele-Operation (low level control - 1) and Supervision (high level control - 2)

The supervisory control console, as seen in Figure 3-2, was developed based on a Real-Time Strategy (RTS) paradigm with the intent of applying the concepts, learned by this type of games, on how to efficiently control and supervise groups of units of various dimensions and with varying capabilities. This approach, while not being new, has allowed the implementation of a console which supports high LOA levels CP-LOA= ({6-7},{6-7}, {6-7}, {6-7}) while, at the same time, enables the supervision of UAV teams with a low workload rating value for the operators.

Another layer will be implemented over the present framework, to extract the viability of mission execution. It is possible to reach combinations of plan state manoeuvres that overload the response of the operator team. This approach will use automata for the model to tackle this issue. By studding the plan loaded in each UAV and applying a transformation that combines all UAV plan states, the plan state change events probabilities, and the operator team recourses into a discrete automata, we can infer about the probability of reaching a failure state. The failure state can be considered to be a state where operator resources do not correspond to the mission state demands. In the last analysis, we can know the probability of reaching one mission state before the Mission Team Supervisor has to process the resource allocation. This information can be used to optimize the resource allocation process and also to help avoiding some mission states by desing during the mission-planning phase (e.g. find and avoid states that require full autonomy LOA=(8,8,8,8) manoeuvres).

4.4. Future trends in networked vehicle systems

The last decades have witnessed unprecedented technological developments in computing, communications, navigation, control, composite materials and power systems, which have led to the design and deployment of the first generations of unmanned aerial vehicles (UAV) and unmanned aerial systems (UAS). These vehicles have already seen action in many scenarios and proved their value.

As the operational capacity of UAS continues to grow, these systems can include multiple UAVs operating as a team, furthermore solidifying their employment in military and civilian scenarios. With the aid of these systems it is possible to remove the human element from “dirty, dull, and dangerous” situations and relocate it to a less operational and more supervisory role. However, with the rise of their operational capacity so rose the complexity of tasks they could perform.

Unmanned vehicle systems are currently being employed in the field for very distinct purposes. For instance, considering just individual UAVs, these can be used for precision sensing, aerial imagery, surveillance, etc. The full potential of these systems, however, requires the management of multiple networked vehicles operating as a whole, sharing their workload and knowledge about the environment.

The concepts of operation for multi-UAV teams differ from single UAVs in the sense that in the former there exist common objectives like maintaining a common knowledge database and redundant execution of crucial actions [69]. Moreover, operators are required to quickly perceive the entire system state, so that they can re-organize themselves in the face of unpredicted situations [70]. All this while taking into account the different levels of attention all the vehicles demand. In order to decrease the number of operators’ necessary on a multi-UAV deployment, we use mixed-initiative interaction for controlling the network at a system-level [71] [70].

In ongoing research on the DARPA mixed initiative control of automata teams program [72], the concept of a 'playbook interface' to allow a human to express his or her intent to multiple unmanned vehicles and sophisticated planning and control software to stipulate or constrain the methods that the automated agents use to achieve that intent. A preliminary example implementing this approach in an unmanned combat air vehicle (UCAV) domain is already existing [73].

Part II: Systems and models

"Any system consists of contrary and dissimilar elements, which unite under one optimum and return to the common purpose"

Pythagorean Kallicratides

In this part, we will start by introducing carefully, but rather informally, the basic concepts of system, system models and a few illustrative applications of these concepts coming from the areas of manufacturing and robotics. As we go along, the application of systemic thinking on the motivational case studies will be used to identify new fundamental notions and concepts, which are then, used to take the first steps towards the definition of a changeability framework for discrete event systems.

5. System and System of Systems

"The whole is greater than the sum of its parts."

Aristotle

Systems may be real and tangible or just concepts. They are made out of parts arranged in some way. A fundamental idea is that a system is characterized by some degree of order, i.e., there are some discernable configurations or patterns, which lead to the notion of structure and architecture, and the system in some way, through actions, activities or processes, is capable of doing "things".

5.1.Concept of System (in science and engineering)

System is a primitive concept whose exact definition is difficult and its understanding is best be left to intuition. Nonetheless, several definitions can be found in the literature:

- "*A set or arrangement of elements that are related and whose behaviour satisfies customer/ operational needs, and provides for the life cycle sustainment of the products.*" (IEEE Standard for Application and Management of the Systems Engineering Process)
- "*An aggregation or assemblage of things so combined by nature or man as to form an integral or complex whole.*" (Encyclopedia Americana).
- "*A regularly interacting or interdependent group of items forming a unified whole.*" (Webster's Dictionary).

Many and diverse disciplines have their own definition of this concept, but all highlight the same set common features: interacting "components" and intent to perform a "function".

The framework of Systems Theory [74], widely used in engineering, places the concept of system in an environment where input-output relations (models) are used to model the actual system. There is no obligation for a system to be associated with physical objects and natural laws. For example, system theory has provided very convenient frameworks for describing economic mechanisms or modelling population dynamics. This framework is illustrated in Figure 14.

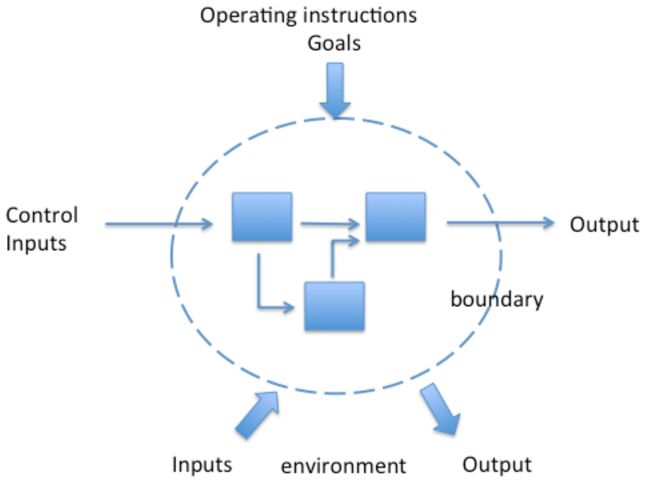


Figure 14: Topology, variables and environment (system)

Science and engineering are primarily concerned with quantitative analysis of systems, and with design, control, and explicit measurement of system performance. Qualitative definitions like the ones given above are inadequate and, for these purposes, the model of an actual system, seen as a device that simply duplicates the behaviour of the system, is needed. A system is a real "object" (e.g., a robot, a factory, a human body) and a model is an "abstraction" (a set of mathematical equations). This simple modelling process is depicted in Figure 15.

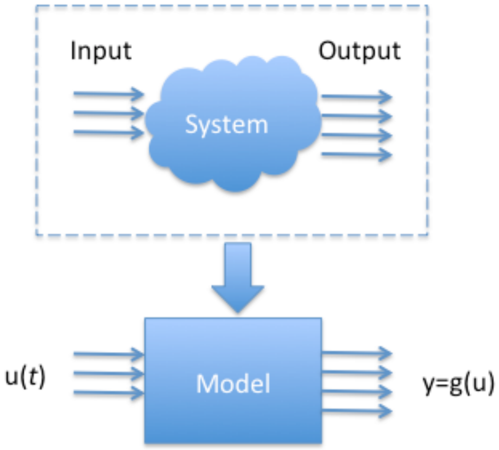


Figure 15: Modelling process

In engineering a system is represented as a tree, the System Breakdown Structure (SBS), where the hierarchy of products and processes that comprise the system architecture, and their relation, is represented (Figure 16).

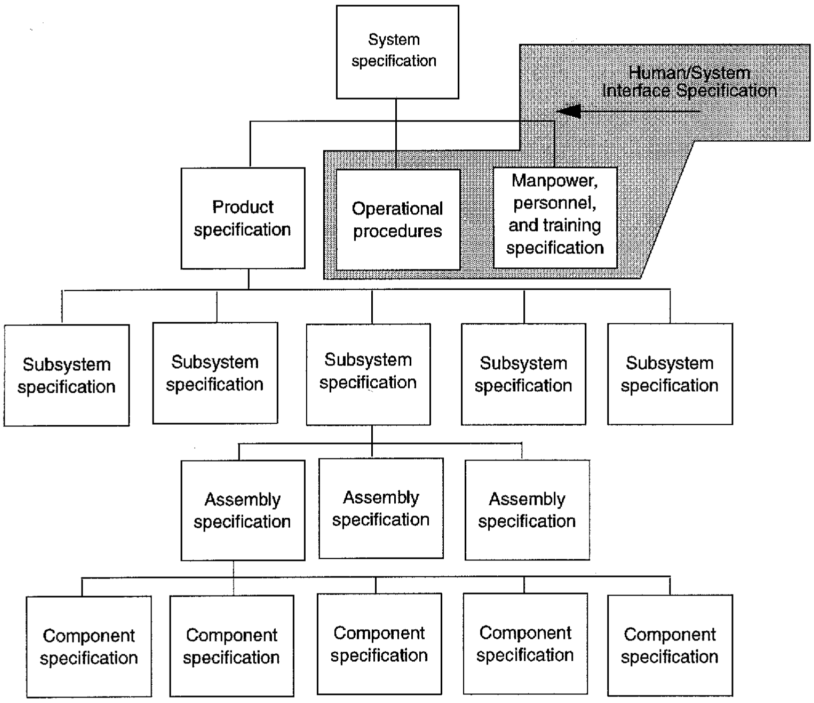


Figure 16: System specification hierarchy (source [75])

Research on a systems framework for general systems has been on going for several decades [74, 76-80]. Such developments have been predominantly influenced by the standard engineering paradigm and in many cases fails to cope with problems coming from other domains such as those of the business processes, data systems, biological systems, and emerging complex systems paradigms.

However, the systems approach is a compelling approach to address complex problems and issues. Ackoff [81] suggested three ways problems could be addressed: resolved, dissolved or solved (Figure 17). In general, when dealing with complex problems, most people resolve them, dealing with the symptoms in absence of full knowledge. This a pragmatic approach, guarantees short term satisfaction and sometimes results in more knowledge about the problem. Looking at these complex problems from the systems perspective it is possible to find the best solution. Understanding and balancing the interacting components and coupled processes of complex systems in their environment will give the best results.

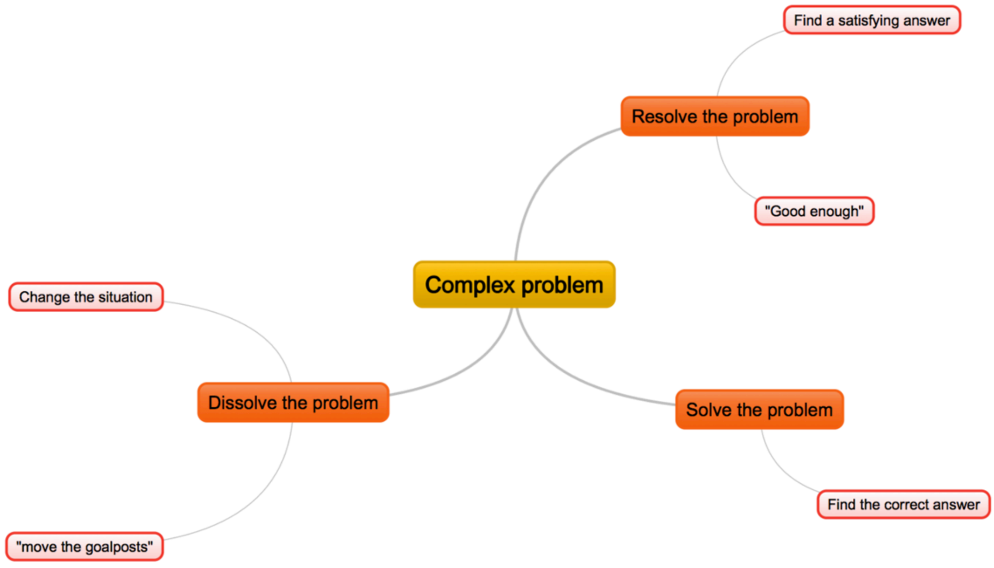


Figure 17: Addressing complex problems and issues

This system approach, which understands the part only in the context of the whole, interacting and adapting to its environment, has become widely used in various scientific disciplines, including the social and life sciences, management and organization sciences, and of course engineering. This *Systems Approach* [82] helped to apply engineering practices to systems that, besides hardware and software, include people (sociotechnical systems) offering approaches to understand and addressing complex systems involving human activity. The systems approach or *Systems Thinking* [83] looks at wholes, and at parts of wholes in the context of their respective whole, as open systems, interacting with other systems in their environment.

5.2. System Life cycle

In Systems Engineering, the system life cycle [84] defines the approach to address a system or proposed system covering all phases of its existence, including system conception, design and development, production and/or construction, distribution, operation, maintenance and support, retirement, phase-out and disposal. The system life cycle includes the complete system or product evolution initiated by a perceived customer need through the disposal of its products.

According to [75, 82, 84], the typical system life cycle includes stages of development, operation and disposal. These phases, depicted in Figure 18, include: System or concept definition; Preliminary system design; Detailed design and development; Fabrication, assembly, integration, and test (FAIT); Production; Customer support (including maintenance, refurbishment and upgrade); Phase-out and disposal.

Figure 18: System life cycle

This concept looks at a system at an entity that evolves with time. Evolving the system involves repairing or correcting malfunctions, providing added capabilities and extending its utility and lifetime. This involves developing the life cycle processes that are needed for system components to satisfy total life cycle needs and requirements.

In nature, systems are able to adapt to changing situations, environments, climate, etc. provided these changes are neither to rapid nor to extreme. Human systems also display a degree of adaptability and ability to evolve. Systems that are adaptive by design are able to track changing situations, operate in changing environments and situations and hence capable to offer sustainable utility and increased longevity. Life cycle cost is related with the total investment in product development, manufacturing, test, distribution, operation, support, training, and disposal. But design for sustainability [85] [86] [87] includes not only the economical aspect, but also the environmental and social aspects (Figure 19).

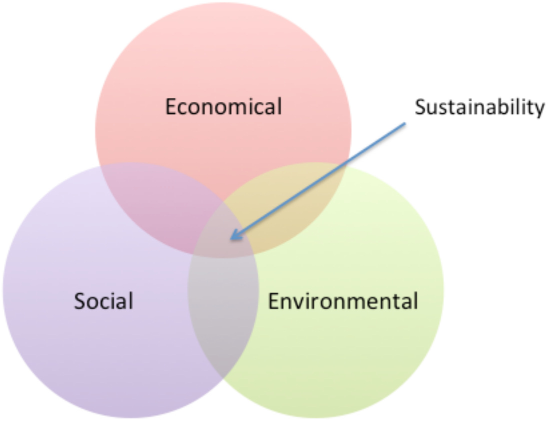


Figure 19: Three pillars of sustainability

Extending system life cycle and contribute to economic and environmental sustainability, due to the reduction in the need for new systems. Modularity in systems and the ability to re-use products and components between systems, eventually requiring some servicing or upgrading, builds on self-aware and knowledge-based components that need to be able to collect and manage information regarding their capabilities (and their evolution over time); maintenance, upgrade or refurbishment operations over it lifetime; and information of use and ware over time.

Although the issue of sustainability as been extensibility in what concerns interoperability [88] and enterprise integration [89], life cycle sustainability has not been adequately addressed. Methodologies for life-cycle management and assessment and strategies for re-use, re-configuring and upgrading systems and components within the whole system need to be developed to achieve this life cycle sustainability. This methodology for the design of systems must integrate reconfigure, dismantling, recycling and value-chain extension processes into the classical design methods. Thinking of a system as a *System of Systems*, or a collection of smart products and components that exhibit intelligent capabilities on the component level, allows components to maintain a representation that can be used for modelling and forecasting purposes. This will realize the need for sustainable systems through facilitating re-use of existing configurations while enabling the adaptation to new arising requirements in response to new needs or in response to performance degradation or upgrade opportunities throughout the system's life-cycle.

5.3. The emergence of System of Systems

Problems arising in domains such as business, industry and robotics, has it was presented in previous sections, are usually characterized by an aggregate of systems that leads to the creation of new forms of systems.

These new forms may either result in a "simple" composite of systems or, as it is usually the case, demonstrate additional features that add complexity. Recently, the concept of "System of Systems" (SoS) [90-92] has emerged in many fields of applications. These are large-scale systems, integrating many independent autonomous systems, frequently of large dimensions themselves, brought together to satisfy a global goal under certain rules of engagement. SoS are linked to problems of complex nature, where these complex multi-systems are very interdependent but exhibit features that go beyond standard system composition. The individual systems themselves have a variable degree of autonomy, subject to a central goal and common rules. When an aggregate of systems demonstrates additional features, which go beyond standard composition of systems, it is referred to as a "System of Systems".

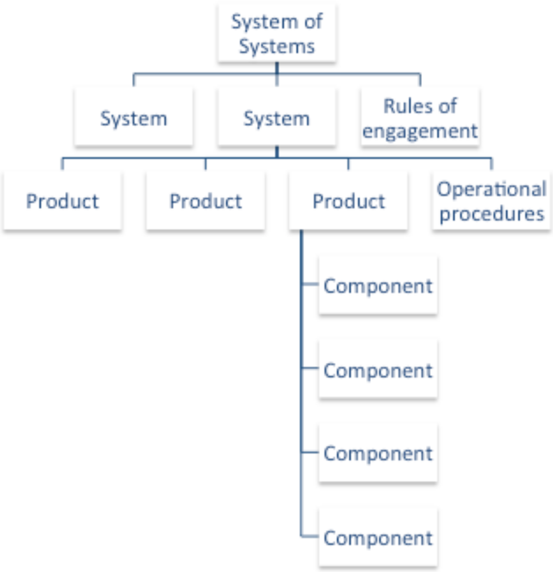


Figure 20: System of Systems hierarchy

According to [93-95], System of Systems exhibit dominant new features like an evolving structure or the form of the organization, which requires a new approach to the analysis and a global approach to their synthesis. Problem domains characterized as System of systems exhibit features such as [94]:

- Operational Independence of Elements.
- Managerial Independence of Elements.
- Evolutionary Development.
- Emergent Behaviour.
- Geographical Distribution of Elements.
- Inter-disciplinary Study.
- Heterogeneity of Systems.
- Networks of Systems.

Considering that the evolutionary development is one of the distinct features, it is important to remember that this evolutionary development must be though to take into account the complete life cycle, and not only isolated phases (like development or operation). Enhancing the sustainability (economic, environmental and social utility) of such system of systems throughout its entire life cycle is a challenging task.

Several abstract definitions of the a System of Systems have been published in the literature, and a discussion on these can be found in [94, 96]. A definition that, although generic in nature, reflects its key features and may be a solid based for a formal definition is the following:

Definition 1: System of Systems [96]

SoS are large-scale integrated systems which are heterogeneous and independently operable on their own, but are networked together for a common goal. A SoS is a “super system” comprised of other elements which themselves are independent complex operational systems and interact among themselves to achieve a common goal. Each element of a SoS achieves well-substantiated goals even if they are detached from the rest of the SoS.

Using this abstract definition as baseline, the system definition previously presented (Figure 14) can be extended to account for the SoS features. In this extended representation (depicted in Figure 21) a system is considered as an agent/actor (an autonomous entity) having its own operational instructions and goals, with modelling and supervisory capabilities integrated. If such a system is embedded in a larger system, relations with other systems may be defined in many ways, namely composition or interaction/play:

- Composition (product or parallel) is an operation that defines the joint behaviour of a set of systems that operate concurrently, for example via an interconnection topology of the automata that model the systems.
- Play is a configuration that defines how every system, entering as an autonomous agent with its own individual operational instructions and goals, interacts with each other.

In this sense, a composite system may be viewed as a system with a single goal, where the individual systems gave up their individual goals. In a play, the distinguishable feature of the SoS, each individual systems retains its individual goals and participates in the “composition” as an intelligent agent with relative autonomy and plays its role as a an actor in the overall play.

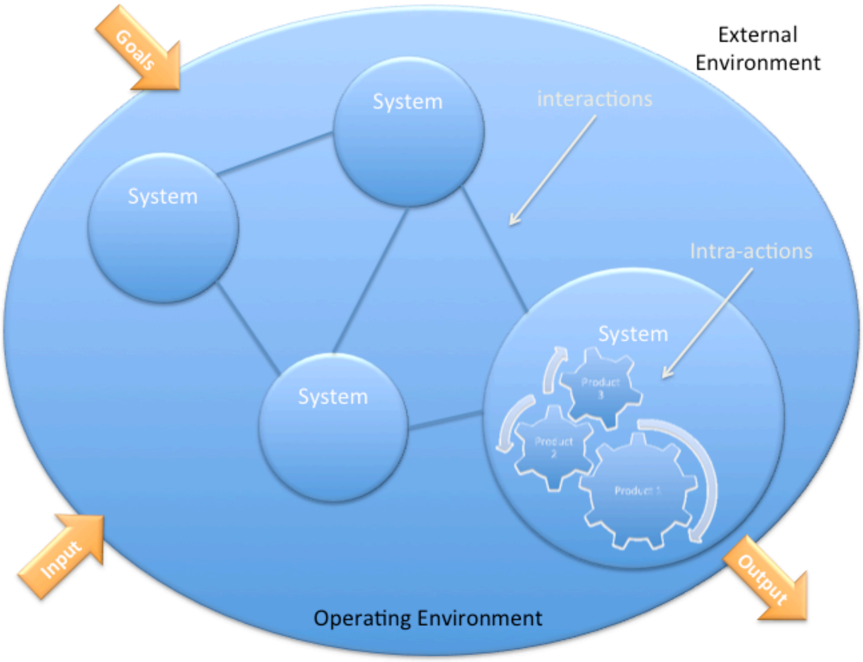


Figure 21: Topology, variables and environment (System of systems)

The notion of “system play” is identified as a crucial element to explain the “super system” nature of the SoS concept. The definition of “system play” [97] and playbook [73] allow for an extension of the standard notion of system by including, amongst others, the notion of agent/actor (independent systems), scenario (set of rules defining operations and interactions) and possible plays. This will be further explored in the next section.

5.4.Synthesis of System of systems

In Part I of this thesis, three illustrative case studies of networked systems, coming from different application domains (business, manufacturing and robotics), were discussed and analysed. These three case studies are clear examples of System of systems, and the result of their analysis will now be used to identify the main common themes and set the baseline for the synthesis of a changeability framework. Looking at the motivational examples of networked systems presented in the previous section, it is easy to identify their main features and justify why they must be modelled as System of Systems. Table VIII summarizes this synthesis, by highlighting for each of the three case studies their fulfilment of the features that characterize a SoS.

Table VIII: Compliance with SoS features

	Business	Manufacturing	Robotics
Operational Independence of Elements	Yes	Yes	Yes
Managerial Independence of Elements	Yes	No	-
Evolutionary Development	Yes	Yes	Yes
Emergent Behaviour	Yes	Yes	Yes
Geographical Distribution of Elements	Yes	Yes	Yes
Interdisciplinary Study	Yes	Yes	Yes
Heterogeneity of Systems	Yes	Yes	Yes
Networks of Systems	Yes	Yes	Yes

Table IX presents the how the different entities described in the three examples relate to the SoS hierarchy depicted in Figure 20.

Table IX: Correspondence with the SoS hierarchy

	Business	Manufacturing	Robotics
SoS	RD Net	Supply Chain	Networked Vehicle System
Team	Consortium	Production Plant	Team
System	Partner	Production Line	Vehicle System
Product	Department	Production Cell	Vehicle
Component	Researcher	Machine	Payload

If we focus on the manufacturing domain, the highest structuring level is the supply chain that can be interpreted as geographical separated sites (production units) connected by material and information flows. The lowest level, the level above the physical processes, is the single machine or workstation. These are the elements responsible for executing the value adding operations including work piece and tool handling. Often several resources are arranged into production cells that typically perform most of the necessary operations to finish a work piece or an assembly including quality assurance. The operations are executed partly by machines and partly by workers. If the processes are more or less automatically interlinked, the terms production line (or assembly line) is commonly used. Production cells and workstations can be merged into production plant: a working area with the same conditions regarding floor load, height, climate and light and the provision with energy and media (ICT). They usually need one or more building that also contain technical and staff rooms. A production plant can be responsible for more than one product segment and serves as a node of a production network or a supply chain.

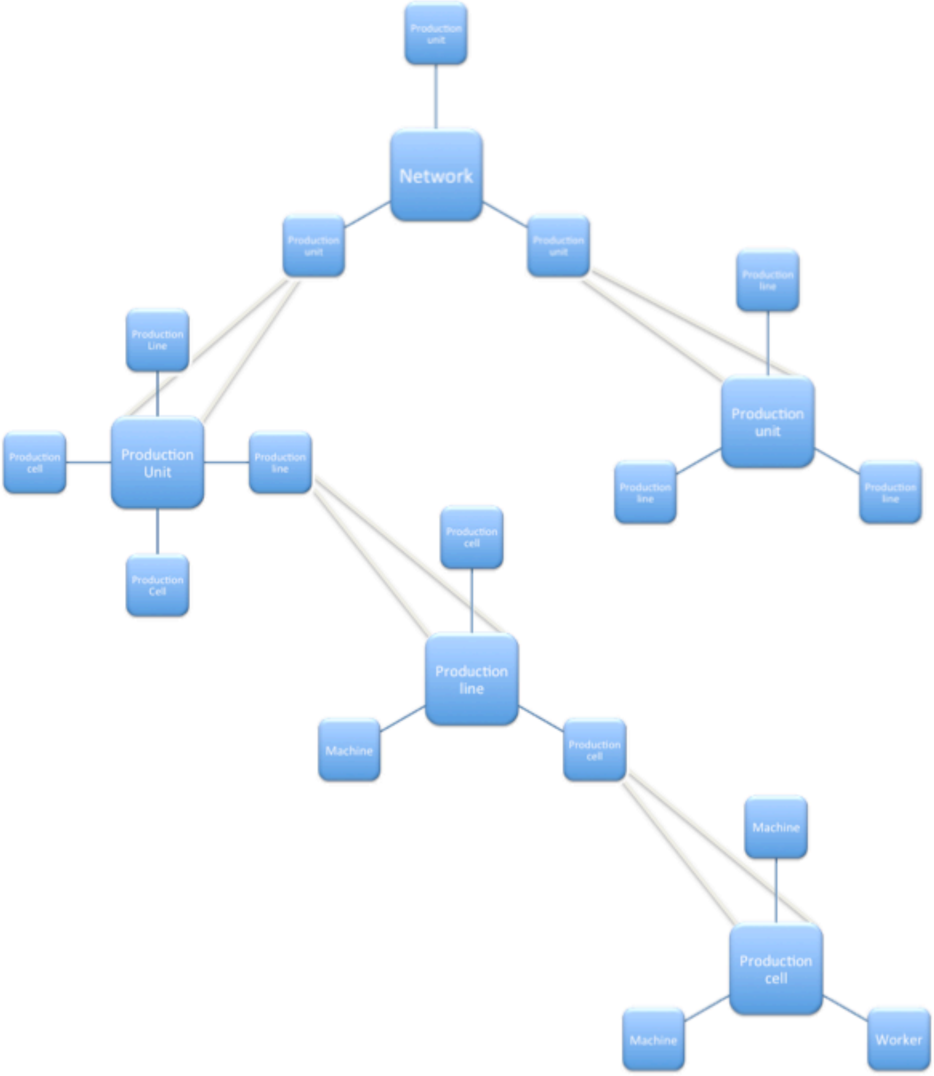


Figure 22: System of Systems view (manufacturing domain)

System and System of Systems must be able to respond to ever changing requirements (changing goals, operational needs or environmental changes, etc.) in order to maintain their utility. A production line should be able to adapt to volume variability and even to product changes. At the same time, a production cell should able to respond to a reduction in the performance of one of its components (machine or worker) due to temporary or permanent limitations.

The capability to adapt the interactions between systems and products, their roles and functions and even their functionalities (i.e. the capability to continuously evolve to respond to both exogenous and endogenous changes) is crucial to guarantee life cycle sustainability – understood as economical, environmental and social utility of the SoS.

These and other issues related with ability of the System of Systems to respond to various types of changes occurring at different levels of their architecture – which will be designated by *Changeability* – will be discussed in the next section.

6. System Changeability

"Traditionally flexibility is interpreted as the ability of a system to change its behaviour without changing its configuration. Conversely reconfigurability is interpreted as the ability to change the behaviour of a system by changing its configuration. These definitions however can be used only if the boundary of the system is clearly defined."

T. Tolio [98]

This quote clearly identifies the necessity to define the boundaries for flexibility and reconfigurability. Depending on the defined boundaries, the ability to change can be considered as either reconfigurability or flexibility [99]. Therefore, it is better to use the more general term changeability [100] [101], which encompasses both characteristics. Changeability can be defined as the characteristic to accomplish early and foresighted adjustments of the SoS and its processes, at all needed levels, in response to a modification in the operational needs and/or rules of engagement.

6.1. The need for changeability in manufacturing

"[..] the era of mass production is being replaced by the era of market niches. The key [..] is a short development cycle yielding low cost, high quality goods in sufficient quantity to meet demand. This makes flexibility an increasingly important attribute to manufacturing."

G. Chryssolouris [102]

This quote, coming from the manufacturing domain, illustrates the mutation in the life cycle of systems. A typical situation in the past would be a steady volume increase after release of the product, and quite long stable phases followed by a decline. Today, product volumes reach the first peak much faster, starts decreasing and then reach a second peak (promotion activities or a face lift in the product), followed by a sudden reduction in the produced volume due to the announcement future release of a new product. One of the results of this dynamics is the change in the system life cycle characteristic and the increasing divergence of the life cycles of the associated products, processes and equipment (Figure 23).

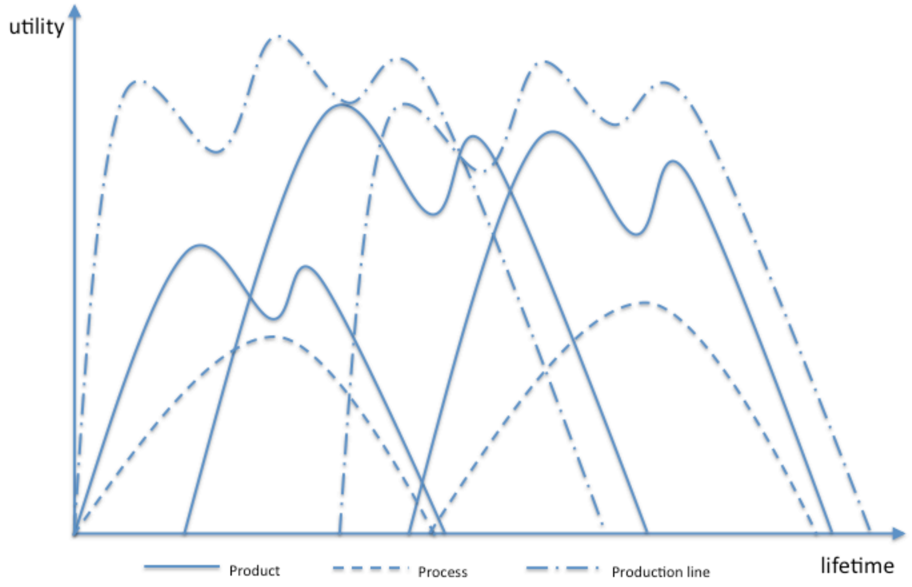


Figure 23: Diverging life cycles [103]

Manufacturing companies are subject to constant changes in their operational environment, which are influenced by innovation, government, the economic and the environment. Being capable to respond to these changes demands for reconfigurability, flexibility, adaptability and agility. Maximising performance is no longer about maximising profitability, but needs to take into account the flexibility, adaptability and agility (i.e. the changeability) to guarantee the sustainability throughout the entire lifecycle. This as made the hierarchy of company objectives evolve over time [104] as depicted in Figure 24.

More and more manufacturing companies are now operating in global supply chains. Not only in the automotive industry, OEMs and their suppliers, but also medium-sized enterprises that work in international markets of specialised products. In this context not only the life cycles have changed: the significant increase on the number of product models and variants, manufacturing operations at different sites (many of which performing outsourced activities) and the cooperation in networks, lead to an increase in the complexity of production processes, leading to a fundamental change in the characteristics of manufacturing systems. Over the years, to address the different challenges, manufacturing system have undergone several major steps of evolution [105] (Figure 25).

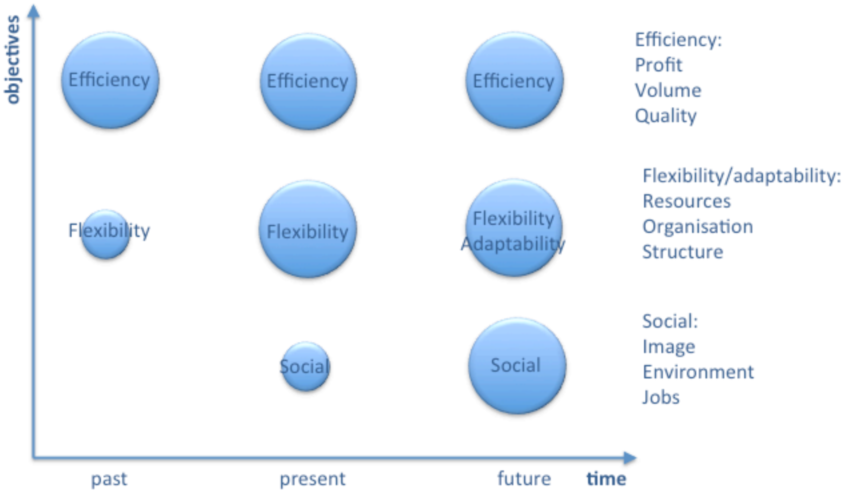


Figure 24: Changing objectives over time

The functional manufacturing system, with highly flexible resources and know how, designed for specific technologies but quite adaptable to product and volume changes. However it suffered from long delivery time and high inventory. The need for faster delivery times, created by competition and an increasing orientation towards customers, was addressed with the segmented manufacturing system.

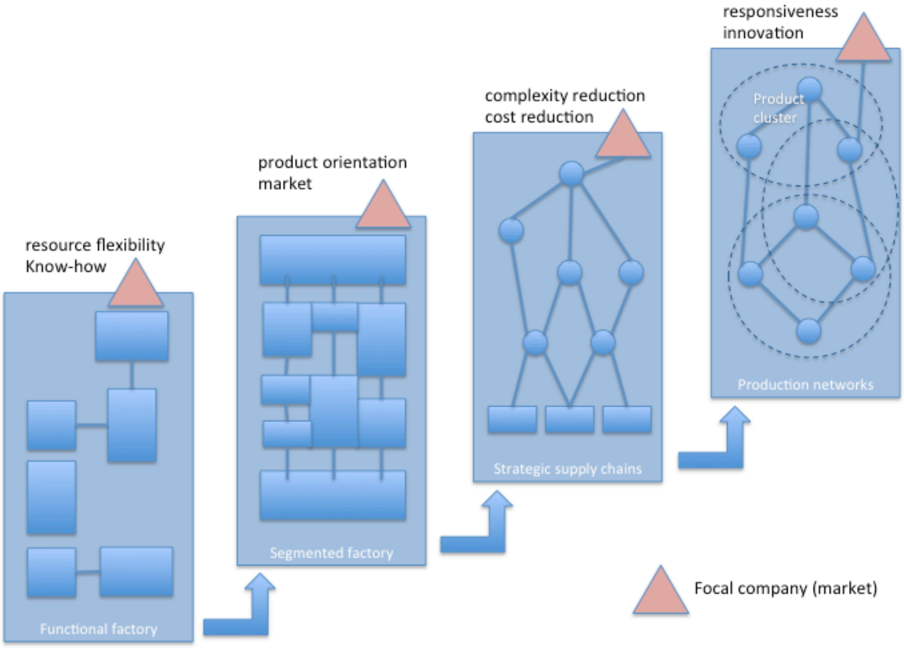


Figure 25: Evolution of manufacturing systems

Manufacturing and assembly activities were organised in cells, lines or segments. Today, production networks with temporary cooperation, mostly dedicated to the product life of a product family, are the dominant paradigm. In this network partners are not organised hierarchically but for the customer only one company is visible. The next generation of manufacturing systems is described as adaptive, transformable, high performing and intelligent. The European Technology Platform *Manufuture* underlines this vision in their Strategic Research Agenda [106].

One clear example of this evolution has been the different control architectures use over time in manufacturing systems. In this context, the term control includes the whole loop that allows a process or a system to be controlled, from sensors to actuators. This has been represented by Baker [107] in a block-diagram model of manufacturing control (Figure 26).

The difficulty of a single central controller to deal with the production system complexity (e.g., uncertainty of demand and resource availability, lag between events and relevant information processing) while at the same time reacting in real-time to events, has lead to distribute decision capabilities amongst different entities in the system, and to the emergence of non-centralized control systems. Distribution of control corresponds to the partitioning of a global control process based on some criterion (e.g., functional or task-oriented) into several decision processes that are assigned to sub-systems, able to support the global decision process.

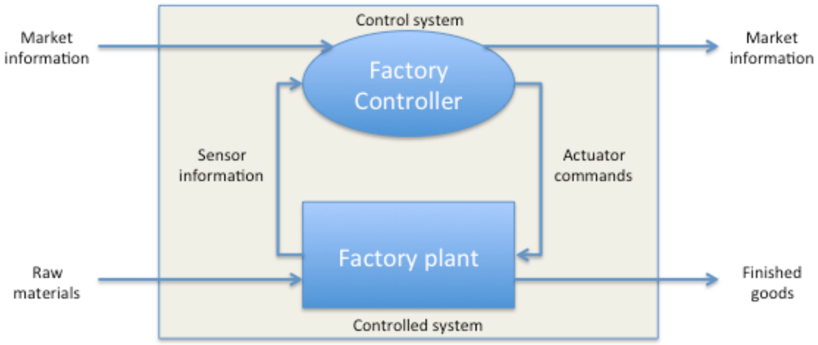


Figure 26: Block-diagram of manufacturing control

In the 1970s, the first kind of control distribution was fully hierarchical and based on the Computer Integrated Manufacturing (CIM) paradigm [108]. Splitting the global control problem into hierarchically dependent sub-problems with decreasing time ranges (i.e., strategic, tactic and operational, such as planning, scheduling and supervising) assigned to hierarchically dependent decisional entities allowed sufficient long-term optimization to be maintained (i.e., global optimality), while supporting less short- term optimization (e.g., agility, reactivity). This traditional CIM-based approach is known to provide near-optimal solutions [109] when some hard assumptions are met, for example, the long-term availability and reliability of the supply and demand, the optimal behaviour and high reliability of production systems, low product diversity, and the observability and controllability of all the possible internal variables.

Since the 1990s, due to the pressing requirement of local reactivity, other kinds of architectures based upon the distribution of control decision have also been considered. In hierarchical control, the time spent to inform the correct controller within the hierarchy (bottom-up), and then to decide and to apply the decision (top-down) generates lags and instabilities. The new approach enables sub-systems with decision capabilities to work together so as to react quickly instead of requesting control decisions from upper decision levels, which was generating response time lags. In this new approach to distribution, interaction processes other than coordination appear, mainly, negotiation and cooperation [52]. However, this approach raises new problems, for example, the need to guarantee deadlock avoidance mechanisms in negotiation and, more generally, the need to guarantee a sufficient level of performance. The relationship among such cooperating decision systems can be qualified as fully heterarchical.

Heterarchy is a concept that is simple to formalize by graph theory. A directed graph composed of nodes representing decision entities and arcs representing master–slave interaction of a decision entity (master) with another entity (slave) is called influence graph. In such a graph, if each node is simultaneously master and slave, then no hierarchy can be identified and thus the graph is strongly connected. This property defines a heterarchy and is consistent with the concept of heterarchy proposed by McCulloch [110].

Hierarchy is a vertical distribution of control, while heterarchy is a horizontal distribution of control. In fully heterarchical control systems (one-level heterarchy, as depicted in Figure 27), long-term optimization is hard to obtain and to verify while short-term optimization is easy to achieve. This is due to the difficulty in guarantying that a sufficient level of system performance can be achieved.

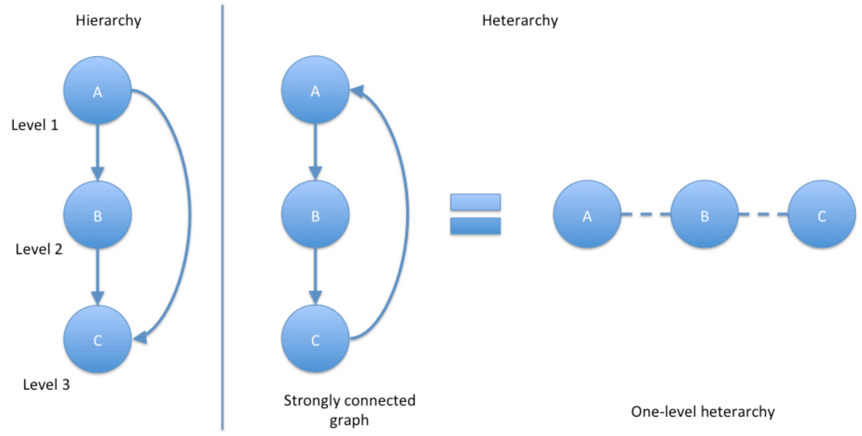


Figure 27: Hierarchy and heterarchy

Since the end of 1990s, a new paradigm has emerged: the holonic paradigm [5]. The integration of both hierarchical and heterarchical mechanisms into a distributed control system is the core feature of the holonic paradigm, allowing users to benefit from the advantages of both approaches.

Figure 28 summarizes the different architectures that can be used to distribute control decisions, from centralized control systems to non-centralized control systems. These different architectures are achieved based upon the design choice of the relationships to use amongst decision entities. The architecture typology proposed by Dilts *et al* [108] defines the three classes that are represented in the diagram. The use of fully hierarchical relationships leads to Class I control architectures, and the use of fully heterarchical relationships leads to Class III control architectures. Class II control architectures, being semi-heterarchical, fall in the middle, integrating both hierarchical or heterarchical relationships. A typical Class II control architecture is a Class III control system with a supervisory level.

Turbulent and fast changing environments, inherent characteristics of the environments in which SoS are embedded, require SoS to be quickly adaptable to changing conditions and uncertainties. In the manufacturing systems domain, the majority of the systems are designed for flexibility [6]. But flexibility makes manufacturing systems customizable and responsive only to pre-designed change drivers and within a narrow corridor of change. On the other hand the term flexibility is very general and must be differentiated according to the SoS level. Changeability is more encompassing and consequently “changeable SoS” is more appropriate to describe the required characteristic of the system.

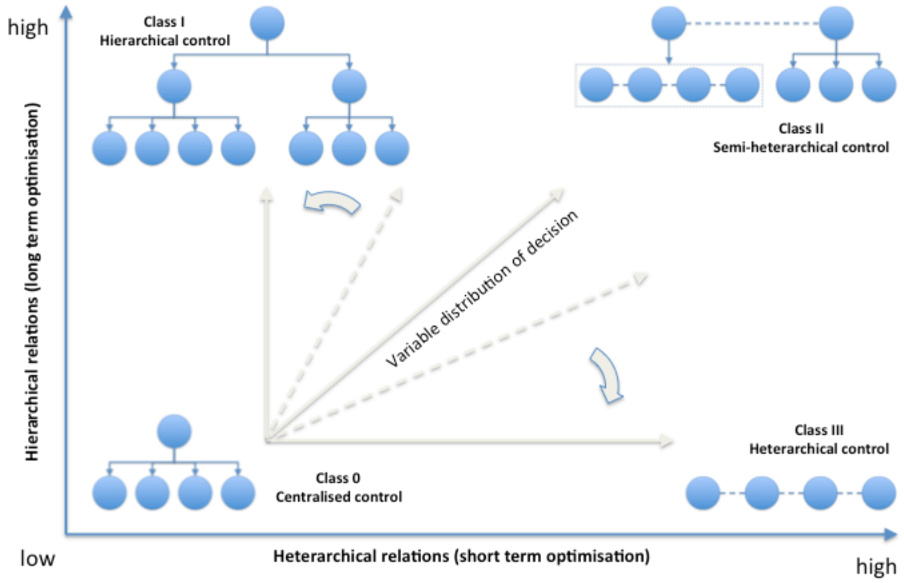


Figure 28: Distribution of decision capabilities in different control architectures

These notions and examples, borough from the manufacturing systems domain can be easily extended to any SoS domain. This will be explored in the following sections.

6.2. Concept and classes

We can define change as the transition over time of a system to a different state. If a system remains the same at time *i* and time *i+1* then it is unchanged. In the present context, we considerer change not only as a state change between time instants, but as a change in the state space or configuration of the system. Ross et al. [111] define change as the capability required from a system in order to be:

- Capable of adapting to changes in mission and requirements.
- Expandable/scalable, and designed to accommodate growth in capability.
- Able to reliably function given changes in threats and in the environment.
- Effectively/affordably sustainable over their lifecycle.
- Developed using products designed for use in various platforms/systems.
- Easily modified to leverage new technologies.

This ability of a SoS to change, its *changeability*, in response to a modification in the operational needs and/or rules of engagement, depends on the level in the SoS hierarchy at which the change is made and on the level of purpose where the change occurred.

Definition 2: Changeability

Changeability can be defined as the needed characteristics to accomplish early, foresighted and efficient adjustments of the structures and processes on certain levels of the SoS in response to change impulses (changing operational needs or purpose).

Changeability serves as an umbrella term and encompasses different types of change according to the levels of the SoS. In the industrial context, ElMaraghy and Wiendahl define changeability as a property of a manufacturing system that enables an economical, timely and proactive adaptation of all factory components and processes at all factory levels [112]. Considering that the term changeability can be used at different levels of a SoS, a hierarchy emerges that allows the definition of five types of changeability. This hierarchy formulation, mimics the "classes of factory changeability" proposed by H-P Wiendahl [113] and ElMaraghy [114].

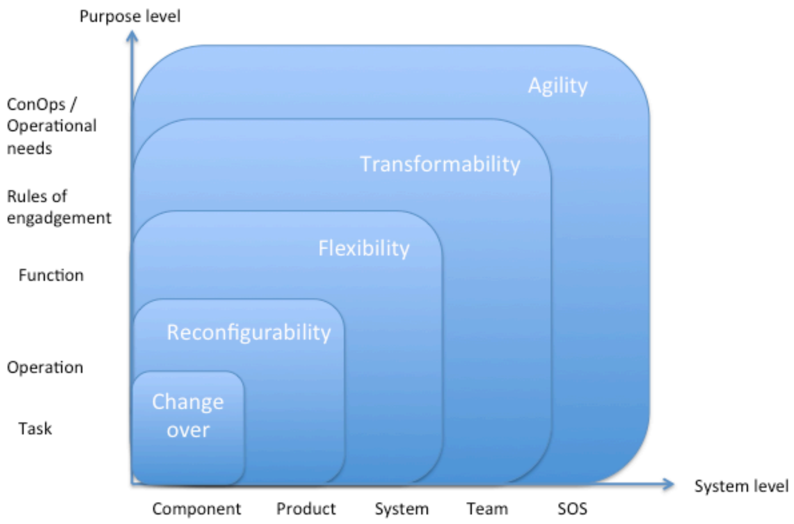


Figure 29: Classes of SoS changeability

These five classes of changeability represented in this diagram (Figure 29), assume that any class at a higher level subsumes the classes below it and can be described as follows:

- Changeover ability designates the ability to change the way a component performs a particular operation, thus enhancing/reducing/changing SoS function.
- Reconfigurability refers the operational ability of a product to execute, with minimal effort and delay, a different task, thus enhancing/reducing/changing SoS function.
- Flexibility describes the tactical ability of a system to start performing a slightly different function, by changing some of its product tasks and/or component operations.
- Transformability indicates the tactical ability to switch to a different set of rules of engagement. This requires a change in the internal structure of its organizational elements changing the roles different systems play inside a compound.
- Agility means the strategic ability of the SoS to respond to new operational needs, by changing organizational elements and including the necessary additional physical and organizational elements.

Definition 3: System of Systems changeability

A sustainable SoS must have the capacity to cross the changeability boundaries in response to changes in the operational requirements or in the rules of engagement, including the capability to increase/decrease its capacity and re-focus its purpose, throughout its life cycle.

6.3. Elements of changeability

Having defined the SoS levels and changeability classes, the next step is to identify the elements (either physical or logical) of a SoS involved in changeability. It is necessary to identify what are the agents involved in the change, the main change drivers, and to define the necessary and appropriate actions at the appropriate time. It is important to define the systems and components, which are changeable, their appropriate degree of changeability and the necessary mechanisms to achieve the required change.

Change agents, or change drivers, are the triggers for the impulse of change in a SoS. These agents can be either endogenous or exogenous (internal or external) to the system and are dependent on the domain of application of the SoS. For example, in the manufacturing domain change drivers can be the fluctuation of demand over time, a new company strategy (e.g. to sell or buy a product line), equipment breakdown, etc. Although dependent on the application domain, change agents are always related to other two important characteristics: the change objectives (or effects, depending if it is an internal or external impulse) and the change strategy (defensive or more tactical to respond the need of the foreseeable future?).

Having identified the change drivers, the change objectives and to respond strategy, it is possible to define the change objects (systems and/or components). The next step is to select the change focus (internal or external change), change depth (starting with the level of the SoS on which the changeability has to be ensured) and then the expected change frequency and the time allowed for each change (path). These three aspects defined the mechanism of change, which along with the change enablers defines the way the system will respond to the change impulse.

Additionally, the necessary and allowed effort (equipment, manpower, knowledge and time) should be measured (the cost of a change). This *performance measurement* (a set of key performance indicators) is needed to measure the impact of the implemented changeability with respect to the performance of the SoS. This process is depicted in Figure 30.

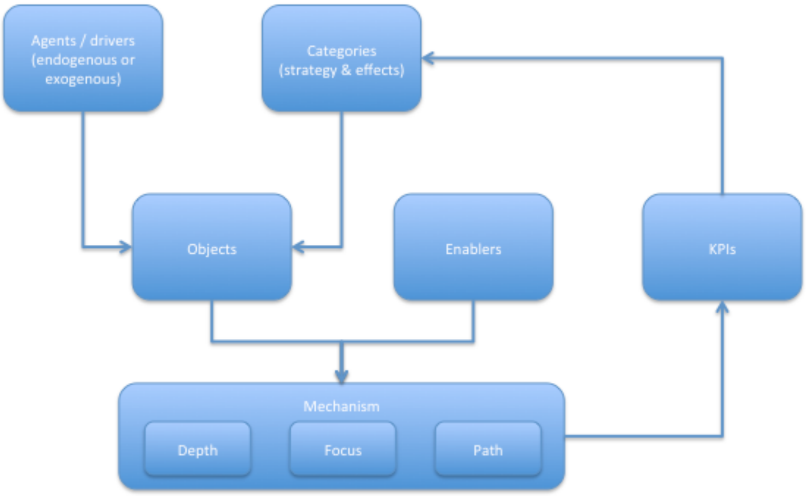


Figure 30: Changeability process

The mechanism used to respond to a certain change impulse has always a cost. The cost of responding to a specific change impulse (e.g. the cost of changing the system configuration) is dependent on the chosen mechanism. Several mechanisms can provide the same outcome, but have different costs (Figure 31).

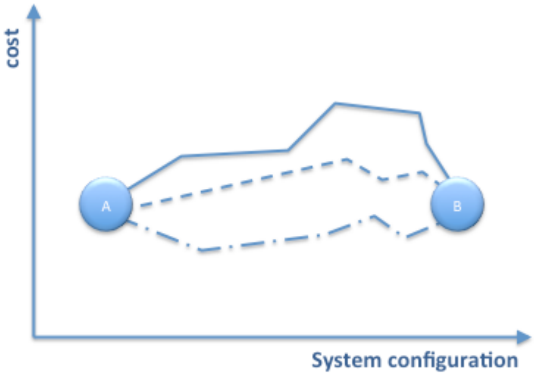


Figure 31: Change mechanisms

As such, quantification of this cost of change is a necessary step to develop a concrete specification for changeability. During exploration of a possible solution, a number of system designs and concepts are considered and assessed in terms of cost and benefit (i.e., utility) throughout the life cycle. A reasonable approach to comparing a large number of systems simultaneously is through a trade space [115].

Regarding the categories of change, and following what has been done in the manufacturing systems domain [102], we will concentrate on three objectives of changeability. Although they were originally intended to describe the flexibility of manufacturing systems, these objectives can be adapted to become applicable as changeability objectives for products, systems and the whole SoS. Figure 32 gives an overview of these aspects.

- Rules flexibility enables a system to work under a variety of rules of engagement with the same configuration.
- Function flexibility refers to the ability to provide a set of functions by changing some of its product tasks and/or component operations.
- Task and capacity flexibility allows a system to vary the performance in executing different tasks to accommodate changes in operational needs, while remaining sustainable.

These aspects involve not only technology but also organisation and human skills as the necessary enablers for all objectives to be achieved.

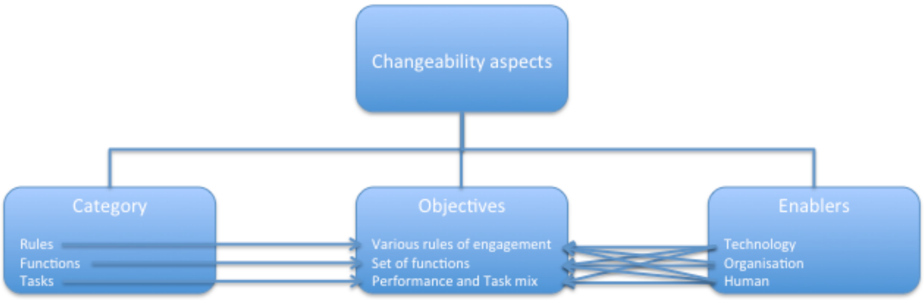


Figure 32: Changeability aspects of SoS

A system that is designed to be changeable must have certain features or properties, the changeability enablers, which enable the physical and logical objects of a system to change their capability towards a predefined objective in a predefined time. These enablers are not to be confused with the flexibility types or its objectives.

6.4. Changeability cycle and control

The goal of system design activities is not to achieve the transformability of all systems and agility of the SoS at all cost. Having defined the levels, objects and enablers of changeability, the question arises as to which degree of changeability is appropriate to guarantee sustainability of the system throughout its life cycle. We cannot expect to design a system with absolute changeability and this means that the changeability requirement has to be defined and then compared it with the actual conformance and aim for continuous adaptation.

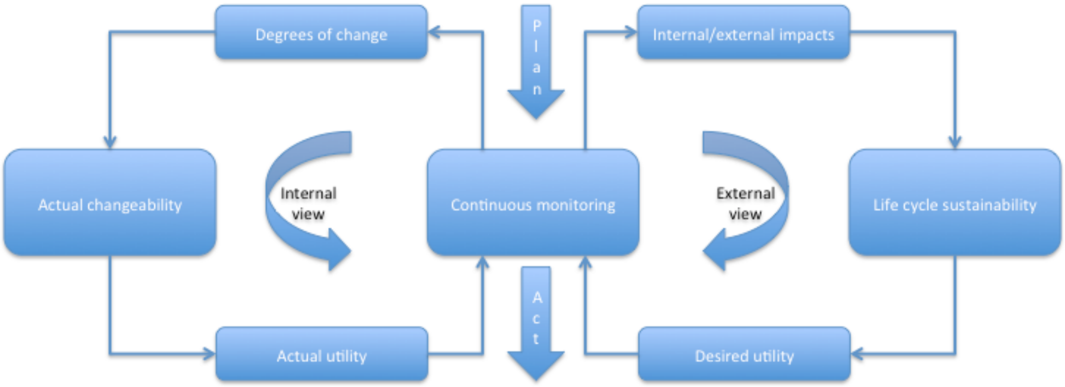


Figure 33: Cycle of changeability

Triggered by external and internal impacts (see Figure 33) target utility has to be set. This refers adaptations in the scope (e.g. operational, tactical, and strategic), the level (e.g. factory, segment, cell, and workplace) and the object (e.g. product, process, volume, mix) of the SoS. The result is the desirable sustainability. On the other hand the existing system has certain degrees of freedom to change, hence the actual changeability offers a potential for changeability. The process of change, and its dynamics, can be viewed as having both closed-loop and open-loop components. Figure 34 shows that change at each of the levels shown in Table IX can be viewed as being the result of two types of decisions:

- Proactive (Open-Loop) – using (reliable) models that estimate the changes that need to be made at the given level to achieve planned results.
- Reactive (Closed-Loop) – using a control loop to make continual incremental changes that try to minimize the difference between planned and actual results.

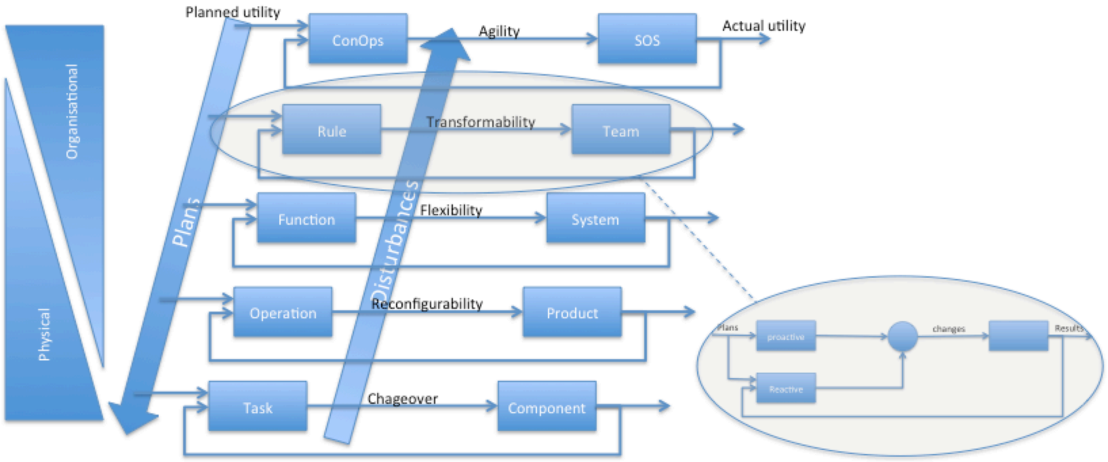


Figure 34: Changes at the different SoS levels

The proactive decision loop is the result of planning, generally top-down and targeting to accomplish life cycle utility. These are the result of planned drivers of change (known agents). Conversely, disturbances are unpredictable, difficult-to model factors that must be reacted to. They can be considered to be unplanned drivers of change, and they generally occur at the lower levels shown and propagate to higher levels. The change cycles previously presented aims for a continuous adaption between the operational needs and the actual performance during the whole life cycle. Figure 35 illustrates the life cycle phases in relation with changeability.

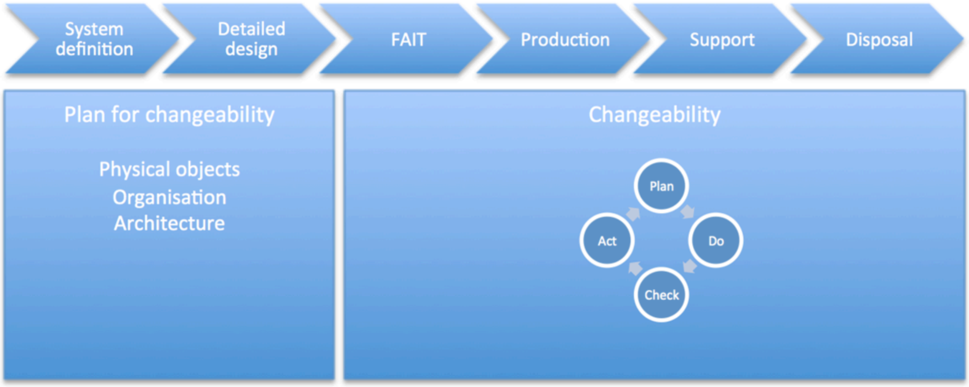


Figure 35: Changeability throughout the life cycle

Designing and maintaining systems in a dynamic environment requires rethinking how systems provide value over time. Developing either changeable or classically robust systems are approaches to promoting value sustainment. Designing systems that have the ability to change allows for maintaining value delivery over the system lifecycle, in spite of changes in the operational context.

7. Discrete Event Systems theory

This chapter will present an introductory review of current discrete event system theory and frameworks, by describing the mains concepts and results related with this class of systems. It will conclude by identifying what are the extensions needed to this framework to be able to deal with changeability in system of systems.

7.1. Discrete event systems

In engineering we are primarily concerned with technics for the design, control and analysis of system performance based on well-defined quantitative measures. Usually this is done using models, i.e. abstract representations, instead of the actual system. A model can be though of as an artefact that duplicates the behaviour of the real systems under a certain number of assumptions or, more precisely, it is a set o mathematical formulations that describes the system behaviour.

The most simple of these models is the input-output model. We start by identifying the set of measurable variables, associated with the system that can be measured over time. From these measurable variables, we select a subset and assume these can be varied over time. With this we defined a set of time functions that are called the input variables:

Equation 14

$$\{\boldsymbol{u_1(t)}, .., \boldsymbol{u_p(t)}\}, \boldsymbol{t_0} \leq \boldsymbol{t} \leq \boldsymbol{t_f}$$

We then select another set of variables, the output variables, that we can measure directly while varying $u_1(t), u_2(t), .., u_p(t)$:

Equation 15

$$\{y_1(t),..,y_m(t)\}, t_0 \leq t \leq t_f$$

This last set may be considered as the response of the system to the stimulus provide by the input variables. There may be other measurable variables from the system that are neither input nor output, and we refer to these variables as suppressed output variables.

This model can be represent by the following equation:

$$y = g(u) = [g_1(u_1,\dots,u_p),..g_m(u_1,\dots,u_p)\,]^T$$

This is the simplest modeling process possible and its depicted in Figure 36. The system is something “real” and the model is the corresponding abstraction, i.e. a set of mathematical equations that mimics the behavior of the system (response measured in terms of output variables to stimulus provided by input variables). Often the model does not replicate the true behavior of the system, only partially covers it and under a certain number of assumptions.

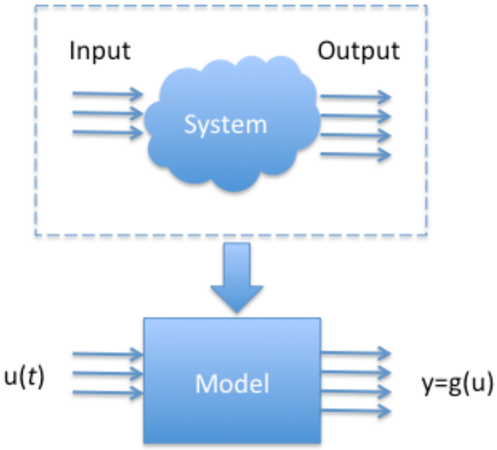


Figure 36: Simple modelling process

Important concepts in system and control theory are state, state space, control and feedback. These will be presented briefly in the next pages. For further details refer to [76] and [80].

The state of a system at time instant t describes its behaviour at that instant in a measurable way. Returning to the input-output model we can define this notion of *state* more precisely:

Definition 4: State

> The state of a system at time t_0 is the information required at $t_0$such that the output $y(t), \forall\, t \geq t_0$ is uniquely determined from the information and from $u(t), t \geq t_0$.

Like the input $u(t)$ and the output $y(t)$, the state is also usually a vector $x(t)$, and the components of this vector $x_1(t), .., x_{1n}(t)$ are called the state variables.

With this new notion of state, we can enhance the model of a system. In addition to selecting the input and output variables we also identify the state variables, and a set of relationship involving the input $u(t)$, the output $y(t)$ and the state $x(t)$. These relationships, the *state equations*, are referred to as the dynamics of the system.

Definition 5: State equations

> The set of equations required to specify the state $x(t), \forall\, t \geq t_0$ given $x(t_0)$ and the function $u(t), t \geq t_0$, are called state equations.

Definition 6: State space

> The state space of a system, denoted by X, is the set of all possible values that the state may assume.

A basic *state space model* consists of a set of equations describing the evolution of state variables over time as a result of a given set of input functions.

Equation 16

$$\dot{\boldsymbol{x}}(\boldsymbol{t}) = \boldsymbol{f}(\boldsymbol{x}(\boldsymbol{t}), \boldsymbol{u}(\boldsymbol{t}), \boldsymbol{t}), \qquad \boldsymbol{x}(\boldsymbol{t_0}) = \boldsymbol{x_0}$$

$$y(t) = g(x(t), u(t), t)$$

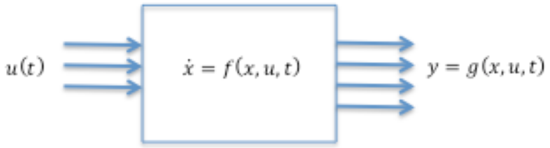


Figure 37: State space modelling process

In the input-output modelling process, also known as *black-box* approach, what we know about the system is only captured by the output response $g(u)$ to the input stimulus $u(t)$ but the internal structure of the system is unknown (or unspecified). The state space modelling process contains additional information captured by the state equations (the dynamics of the system).

The definition of a system contains the idea of performing a particular function. In order for such a function to be performed, knowing what the system will do based on a certain input is not enough. We need to be able to *control* the system by selecting the right input to achieve some desired behaviour. The input of the system is in these cases is viewed as a control signal aimed at achieving a desired behaviour. This desired behaviour is represented by a reference signal $r(t)$ and the control input to the system as

Equation 17

$$u(t) = \gamma(r(t), t)$$

This relationship is referred to as the control law or simply control.

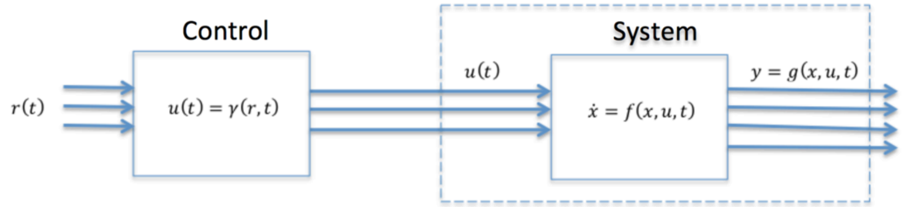


Figure 38: System with control input

In order to achieve and maintain the desired behaviour, it possible to use the available output information to continuously adjust the control input. This is the concept of *feedback*. To include this concept in our model we need to extend the control law (Equation 17) to include along with the reference $r(t)$ the observed output $y(t)$, or more generally the state $x(t)$.

Equation 18

$$u(t) = \gamma(r(t), x(t), t)$$

A system that includes feedback in the control law is referred to as a *closed-loop* system, as opposed to the *open-loop* system when no information on the state is included in the control law.

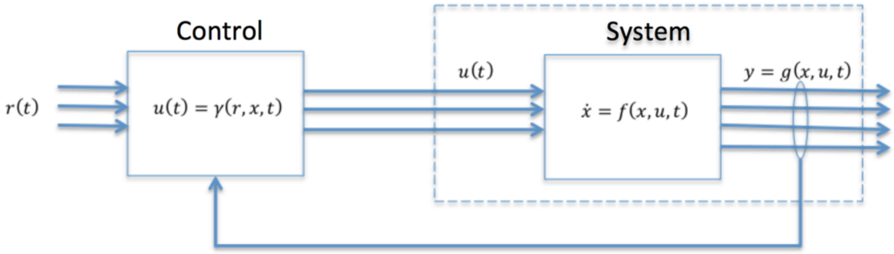


Figure 39: Closed loop system

Systems have been studied over the years involving quantities such as pressure, temperature, speed, and acceleration. These are continuous variables that change in time, and such systems are referred to as Continuous-Variable Dynamic Systems (CVDS). The modelling and analysis of this type of systems is mostly based on the theory and techniques related to differential and difference equations.

Many systems of interest have their state space described by a discrete set (e.g. $\{0,1,2,3,..\}$) and state transitions are observed only at discrete point in time (associated with *events*) and these systems are referred as *Discrete Event Systems*.

An event is a primitive concept easy to understand. In this context to important characteristics of an event are that we should consider that is occurring instantaneously (i.e. takes no time) and is causing transitions from one state to another.

Discrete Event Dynamic Systems (DEDS), or just simply Discrete Event Systems (DES), are systems with a discrete state space and changes in the state can only be the result of asynchronous occurring instantaneous events over time. Sample paths of DES are typically piecewise constant functions of time. Conventional differential equations are not suitable for describing such "discontinuous" behaviour. These samples paths can be viewed as a sequence of states, or as sequence of states with corresponding time instants at which state transitions take place. This distinction gives rise to the classes of untimed and timed models respectively. Timed models can include stochastic elements.

In contrast with a CVDS, a DES satisfies the following two properties (a) the state space is a discrete set and (b) the state transition mechanism is event-drive. Its main characteristics are a discrete state space, denote by X, and a discrete event set, denote by E.

Many systems that are found in our everyday life, particularly technological systems, have in fact discrete state spaces (e.g. computers, communication systems, manufacturing of products, warehouses, software, and traffic systems).

The higher-level behaviour of complex systems with underlying continuous variable dynamics are simplified and often modelled as a DES for the purpose of supervisory control, monitoring, and diagnostics. But in some cases such complex systems must be explicitly modelled so as to capture the interaction of event-driven and time-driven dynamics, giving rise to what are called Hybrid Systems. Figure 40 presents this taxonomy describing the relations between the major system classes.

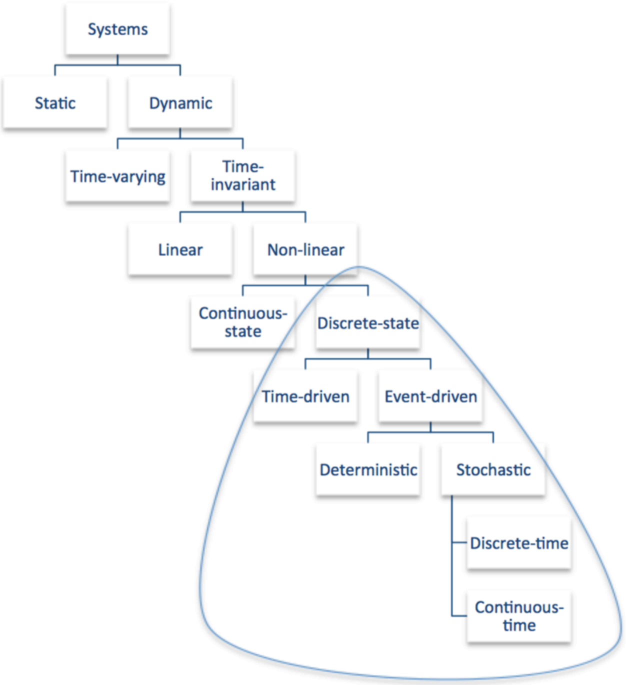


Figure 40: System taxonomy (major classes)

7.2. Models of Discrete Event Systems

Languages, timed languages and stochastic timed languages are three levels of abstraction at which DES are modelled and studied: untimed (or logical), timed and stochastic.

When studying the state evolution of a DES the first concern is with the sequence of states and corresponding events causing the state transitions. At this stage, the time the systems enters and how long it remains in a certain state is not the main concern. Thus we can assume the behaviour of the DES is described as a sequence of events $e_1\, e_2 \ldots e_n$, which provides the order the various events occurred over time but it does not provide any information on the time instants of the occurrences. This is the untimed or logical level of abstraction, and we consider the behaviour of the system is modelled by a *language* (the set of events is the *alphabet* and the sequence of events the *words*).

Automata can be considered as the most basic class of DES models to represent languages. They are intuitive, easy to use, susceptible to composition and tractable to analysis (in the finite state case). But on the other hand, they lack structure and may lead to quite large state spaces. Petri nets [116] have more structure than automata but in general do not posses the same analytical power. Alternative modelling formalisms are for example process algebra [117] and process calculus [118].

The theories of languages and automata [119] is one of the formal approaches to the study the logical behaviour of DES. Any DES has an underlying event set E, which can be considered the alphabet of a language and the sequence of events the words in that language.

Definition 7: Language

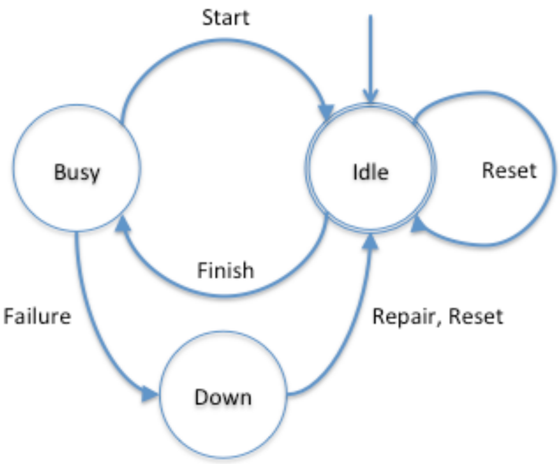
A language defined over an event set E is a set of finite-length strings formed from events in E.

The operation involved in building strings, and thus languages, from a set of events E is concatenation. E^* is the set of all finite strings of elements of E, including the empty string ε. The * operation is called *Kleeene-closure*. The common set of operations such as union, intersection, difference, and complement with respect to E^* are applicable to languages since languages are sets. Additional, languages also support the operations *concatenation*, *prefix-closure*, *Kleene-closure*, and *post-language*. Another type of operations frequently performed on languages is the projection, from a set of events to a smaller set of events. The projection operation takes a string formed from the large set and erases events in it that do not belong to the smaller languages. For further details on languages and their operations refer to [119].

Although languages are a formal approach to describe the behaviour of a DES, simple representations of the languages are not easy to specify and process. The modelling formalism of automata can be used as a framework for representing and manipulating languages.

Automata

An automaton is a device capable of representing a language according to well-defined rules. The simplest representation of an automaton is a directed graph or a state transition diagram.



$E = \{Start, Finish, Failure, Repair, Reset\}$

$X = \{Busy, Idle, Down\}$

Initial state is $Idle$ (in arrow)

Marked states set is $\{Idle\}$ (double circle)

Figure 41: Simple processor model

From the previous simple example we can infer a few observations regarding an automaton: (1) an event may occur without causing a state transition; (2) two distinct events may occur at a giving state causing exact the same state transition; (3) an automaton has an initial state and (4) an automaton has a set of marked states. A formal definition of deterministic automata follows.

Definition 8: Deterministic automaton

A Deterministic Automaton G is a six-tuple $G = (X, E, f, \Gamma, x_0, X_m)$ where X is the set of states, E is the finite set of events associated with G, $f: X \times \mathrm{E} \rightarrow \mathrm{X}$ is the state transition function (f is a partial function in its domain), $\Gamma: X \rightarrow E^2$ is the active event function (or feasible event function), x_0 is the initial state, and $X_m \subseteq \mathrm{X}$ is the set of marked states.

The automaton is said to be deterministic because f is a function from $X \times \mathrm{E}$ to X, meaning that there can be no two transitions with the same event label out of the same state. In contrast, the transition structure of a nondeterministic automaton is defined by means of a function from $X \times \mathrm{E}$ to 2^X. In this case there can be multiple transitions with the same event label out of a state.

The link between languages and automaton is easily to do by inspection of the state transition diagram of an automaton. Consider all the directed paths that can be followed, starting in the initial state, in the state transition diagram, and consider among these all of the paths that end in a marked state. This leads to the definition of the languages *generated* and *marked* by an automaton.

Definition 9:Languages generated and marked

The language generated by $G = (X, E, f, \Gamma, x_0, X_m)$ is

$$\mathcal{L}(G) := \{s \in E^* : f(x_0, s) \; is \; defined\}$$

The language marked by $G = (X, E, f, \Gamma, x_0, X_m)$ is

$$\mathcal{L}_m(G) := \{s \in E^* : f(x_0, s) \in X_m\}$$

In the definition of deterministic automaton, the initial state is a single state, all transitions have event labels and, as we just seen, the transition function is deterministic. But for modelling and analysis purposes it might be necessary to relax these requirements. For example, we might not know in advance the exact initial state of the system (might be one out of two possibilities) and we might not be able to say with certainty what the effect of an event might be (either by pure ignorance or because some states were merged). In other cases, we might have transitions in the internal state caused by events that are not "observable" by an outside observer (e.g. imagine there is no sensor to record this state transition), in which cases we would include the empty string as label (ε-transitions). The generalization of the notion of automaton is motivated by these observations.

Definition 10: Nondeterministic automaton

A Nondeterministic Automaton G_{nd} is a six-tuple $G = (X, E \cup \{\varepsilon\}, f_{nd}, \Gamma, X_0, X_m)$ where X is the set of states, E is the finite set of events associated with G, $f_{nd}: X \times \mathrm{E} \cup \{\varepsilon\} \rightarrow 2^X$ is the state transition function $f_{nd}(x, e) \subseteq X$ whenever it is defined, $\Gamma: X \rightarrow E^2$ is the active event function (or feasible event function), X_0 is the set of initial states, and $X_m \subseteq \mathrm{X}$ is the set of marked states.

Two other variants of the definition of automaton are useful in systems modelling: the Moore automaton and the Mealy automaton (named after E.F. Moore and G.H. Mealy who defined them). The differences for the previous definition is simple and applies to both deterministic and nondeterministic:

- Moore automata are automata with (state) outputs, meaning there is an output function that assigns an output value to each state.
- Mealy automata are input/output automata, meaning that transitions are labelled with input event and output event. The interpretation of a transition e_i/e_o from

state x to state y is the following – when the system is in state x and event e_i occurs then it will make the transition to state y and will emit the output event e_o.

Since any language can be marked by an automaton, automata are a practical tool to manipulate languages in analysis or controller synthesis problems. Although in some cases we can end up with a practical problem: it may require an infinite number of states. A language is said to be regular if it can be marked by a finite-state automaton, making these problems the more amenable to be treated with these tools.

Operations on automata

To be able to analyse DES modelled as automata we need a set of operations on a single automaton in order to modify appropriately its state transition diagram. Operations to combine or compose, two or more automata, so that models of a complete system can be built from model of the individual system components are also needed. These operations will be only briefly introduced here. For a more comprehensive discussion refer to [80] and [119].

Unary operations change the state transition diagram of an automaton, leaving the event set E unchanged. The operations *accessible part*, *coaccessible part*, *trim*, and *complement* will be presented.

From the definition of languages generated and marked by an automaton G, we can see that if we delete from G all the states that are not accessible or reachable from x_0 by some string in $\mathcal{L}(G)$ it will not affect the languages generated and marked by G. The operation called *accessible part*, denoted by $Ac(G)$ were Ac is interpreted as taking the accessible part of G, deletes all the states of G that are not accessible from x_0.

A state x of G is said to be coaccessible to X_m, or coaccessible, if there is a path in the state transition diagram of G from x to a marked state. The operation *coaccessible part*, denoted by $CoAc(G)$ were $CoAc$ is interpreted as taking the coaccessible part, deletes all the states of G that are not coaccessbile.

An automaton that is both accessible and coaccessible is said to be *trim*. The *trim* operation is defined as $Trim(G) := CoAc[Ac(G)] = Ac[CoAc(G)]$.

Supposing we have a trim deterministic automaton $G = (X, E, f, \Gamma, x_0, X_m)$ that marks the language $\mathcal{L}_m \subseteq E^*$. The *complement operation*, denoted by $Comp(G)$, will build the automaton G^{comp} that will mark the language $E^* \setminus \mathcal{L}$.

The operations Ac, $CoAc$ and $Trim$ are defined and performed similarly in the case of nondeterministic automata.

Composition operations, applied over two or more automata, model the forms of joint behaviour of a set of automata that operate concurrently. The operations *product* (also called completely synchronous composition) and *parallel composition* (also called synchronous composition) will be presented for the case of two deterministic automata. Generalization for a set of automata using the associative properties is straightforward as is for the case on nondeterministic automata.

Lets assume the following two automata are accessible: $G_1 = (X_1, E_1, f_1, \Gamma_1, x_{01}, X_{m1})$ and $G_2 = (X_2, E_2, f_2, \Gamma_2, x_{02}, X_{m2})$. No assumptions are made on the event sets E_1 and E_2.

The *product* of G_1 and G_2 is the automaton

$$G_1 \times G_2 := Ac(X_1 \times X_2, E_1 \cup E_2, f, \Gamma_{1\times 2}, (x_{01}, x_{02}), X_{m1} \times X_{m2})$$

where

$$f\big((x_1, x_2), e\big) := \begin{cases} \big(f_1(x_1, e), f_2(x_2, e)\big) & if\ e \in \Gamma_1(x_1) \cap \Gamma_2(x_2) \\ undefined & otherwise \end{cases}$$

and thus $\Gamma_{1\times 2}(x_1, x_2) = \Gamma_1(x_1) \cap \Gamma_2(x_2)$.

In the product the transitions of the two automata must always be synchronized on a common event (an event in $E_1 \cap E_2$). This operation represents the lock-step interconnection of G_1 and G_2, where an event only occurs if and only if it occurs in both automata. The product operation displays the commutative and the associative properties.

Composition by product is restrictive as it only allows state transition in common events. In general, when modelling systems composed of interacting components, the event set of each component includes both private (related to its own internal behaviour) and common events (shared with other automata) that capture the coupling between components. Parallel composition is the standard way to build a complete model from models of individual events.

The *parallel composition* of G_1 and G_2 is the automaton

$$G_1 \parallel G_2 := Ac(X_1 \times X_2, E_1 \cup E_2, f, \Gamma_{1\parallel 2}, (x_{01}, x_{02}), X_{m1} \times X_{m2})$$

where

$$f\big((x_1, x_2), e\big) := \begin{cases} \big(f_1(x_1, e), f_2(x_2, e)\big) & if\ e \in \Gamma_1(x_1) \cap \Gamma_2(x_2) \\ (f_1(x_1, e), x_2) & if\ e \in \Gamma_1(x_1) \setminus E_2 \\ \big(x_1, f_2(x_2, e)\big) & if\ e \in \Gamma_2(x_2) \setminus E_1 \\ undefined & otherwise \end{cases}$$

and thus $\Gamma_{1\parallel 2}(x_1, x_2) =_1 [\Gamma_1(x_1) \cap \Gamma_2(x_2)] \cup [\Gamma_1(x_1) \setminus E_2] \cup [\Gamma_2(x_2) \setminus E_1]$.

In parallel composition a common event can only be executed if both automata execute it simultaneously, and thus the two automata are synchronized on the common events. The private events are not subject to such constraints and can execute whenever possible. The parallel composition displays the commutative and the associative properties.

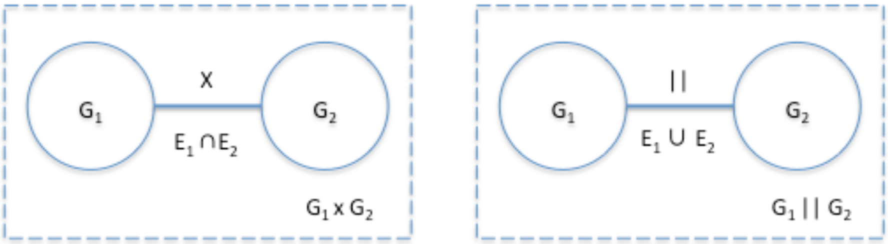


Figure 42: Interconnection of two automata (product and parallel composition)

These two operations, product and parallel composition, are two types of interconnection between system components. As we saw from the definition of the operations, the main difference between the two is the way they handle private events, i.e., events that are not contained by E_1 and E_2 simultaneously.

Observer automata

Nondeterministic automata are different from deterministic automata because the codomain of f is 2^X, the power set of the state space, and also by allowing ε-transitions.

But how do deterministic and nondeterministic automata compare in terms of language representation? We can always transform a nondeterministic automaton into a language-equivalent deterministic one, i.e., one that generates and marks the same languages as the original nondeterministic automata. The resulting equivalent deterministic automaton is called the observer corresponding to the nondeterministic automaton, following the concept of observer in systems theory: the equivalent deterministic automaton correlates to the estimate of the state of the nondeterministic automaton. The observer of G_{nd} is denoted by $Obs(G_{nd})$ or often G_{obs}.

The important properties of the observer automaton are that (1) $Obs(G_{nd})$ is a deterministic automaton, (2) $\mathcal{L}\big(Obs(G_{nd})\big) = \mathcal{L}(G_{nd})$ and (3) $\mathcal{L}_m\big(Obs(G_{nd})\big) = \mathcal{L}_m(G_{nd})$. Observer automata are an important tool in the study of partially-observed DES.

7.3. Analysis of Discrete Event Systems

DES modelled as finite-state automata are very tractable in it comes to answering various questions about the behaviour of the system. The analysis problems that are most times addressed in it comes to DES are the following: (i) safety and blocking properties of deterministic automata with all event observable, (ii) partially-observed systems where some events are unobservable, and (iii) event diagnosis problems with the goal to detect the occurrence of certain unobservable events.

Safety and blocking properties

Safety properties are concerned with the reachability of certain undesired states in the automaton. An automaton model of a system is generally built in two steps: (i) the automaton of the individual components are built and (ii) the complete system model is obtained by composition of the different components. The safety issues are mostly at the level of the complete system. Using the unary operations previously introduced, the safety questions are easily answered. For example, to determine if a given state y is reachable from another given state x, the accessible operation can be used with x as the initial state and looking for y in the result.

Blocking properties are concerned with the coaccessiblity of certain states to the set of marked states. The coaccessible operation can be used to determine if a given accessible G is blocking or not. If any state is deleted then G is blocking, otherwise it is nonblocking. The same operation can be used to identify deadlock states and livelock cycles.

Partially-observed DES

Nondeterministic automaton have some events, modelled as ε-transitions, that occur in the system modeled as an automaton but are not seen, or observed, by an outside observer of the system behavior. There are a few reasons for this lack of observability, from the absence of a sensor to record the occurrence of the event to the fact that the event takes place in a remote location and there is no communication (at least not in real time) of its occurrence.

As an alternative to label these events as ε-transitions and obtain a nondeterministic automaton, we can define specific labels for these events but qualify them as unobservable. In this case, our model of the system will be a deterministic automaton whose event set E is partitioned in two disjoint subsets: E_o the observable events set and $E_{uo}$the unobservable events set. Such a system is called *partially-observed*.

The same approach used before can be used to build an observer for a deterministic automaton $G = (X, E, f, x_0, x_m)$ with unobservable events where $E = E_o \cup E_{uo}$, by simply treating all events in E_{uo} as if they were ε. The observer will in this case have the event set E_o and we will need to define the *unobservable reach* of each state $x \in X$, which is $UR(x) = \{y \in X: (\exists t \in E_{uo}^*)[(f(x, t) = y)]\}$.

The state estimation of partially-observed DES using observers is possible, but some cautions must be taken. In some cases the current state of $Obs(G)$ may be a singleton, meaning that at this time the state of G is know precisely by an outside observer. But uncertainty may arise if unobservable events are possible in the future. Another situation is related to the fact that different strings in $\mathcal{L}(G)$ may have the same projection in $Obs(G)$, meaning that different strings may lead to the same state.

Unobservable events are a tool to capture nondeterminism at modelling time. This approach enables us to deal with uncertainty in systems analysis, and observer automata can be used to analyse the resulting system model. Two possible uses of unobservable events to deal with uncertainty are (i) the use of unobservable events in place of uncertainty in the behaviour of the system (e.g. which state transition will be triggered by a certain event?) and (ii) the use of mask functions to deal with events that are neither observable nor unobservable (e.g. a sensor that cannot distinguish between two or more events).

Event diagnosis

When system models contain unobservable events it is important to determine if certain unobservable events, e.g. events that model faults of system components, could have occurred or must have occurred, it is a problem of event diagnosis. Knowing that one of these events has occurred is very important in monitoring the performance of the system. If we continue to observe the system behaviour we can reduce the uncertainty about events executed by the system in the past. We can implement this inference about the past if we modify the construction of the observer and explicitly include tracking of unobservable events of interest. This modified observer is called a *diagnoser automaton*.

The diagnoser automata are able to keep track of the system behaviour and diagnose the prior occurrence of certain unobservable events. The diagnoser automaton built from G is denoted $Diag(G)$. The diagnosers are similar to observers with a difference that labels (Yes/No) are attached to the states of G in the states of $Diag(G)$ signaling if the event of interest has occurred or not. $Diag(G)$ has an event set E_o, it is a deterministic automaton and it generates the language $\mathcal{L}\left(Diag(G)\right) = P[\mathcal{L}(G)]$.

Event diagnosis in systems where sensor readings (observations) are not centralized but are distributed over a set of sites, each site must monitor and diagnose the system based on its own observations. In same cases sensors may or may not be able to communicate among each other to exchange raw or processed data about the system behaviour. In decentralized systems event diagnosis is performed by local diagnosers running at each site based on the system model and on its local observations. Local diagnosers do not communicate among each other while monitoring the system. The goal in *decentralized diagnosis* is that each occurrence of the unobservable event of interest be diagnosed by at least one local diagnose.

7.4. Supervisory control

Consider a discrete event system modelled at the logical level of abstraction by automaton G with a state space that does not need to be finite and an event set E. Automaton G models the *uncontrolled* behaviour of the system. Assuming this behaviour might not be satisfactory then it must by modified by feedback control[6] in order to achieve a set of pre-defined *specifications*: modifying the behaviour is understood as restricting the behaviour to a subset of $\mathcal{L}(G)$.

This modification, or restriction, of the behavior of G is implemented by the introduction of a *supervisor*, denoted by S. This approach for supervisory control is depicted in Figure 43 and illustrates the fact that the plant G is separate from the controller (or supervisor) S.

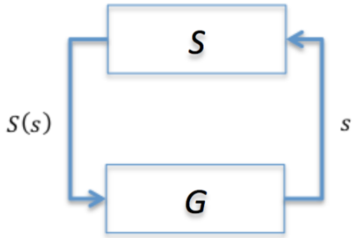


Figure 43: Feedback control loop of supervisory control

In this context specifications are defined as follows. A language $\mathcal{L}(G)$ contains strings that are not acceptable for some reason (e.g. violate safety of blocking conditions, allow for some behaviour we want to avoid, etc.). By specifications are defined by considering a sublanguage of $\mathcal{L}(G)$, that represents the legal or admissible behavior for the controlled system. A controlled behaviour that stays inside the legal language is called *safe*. In some cases, these specifications are defined as a range $\mathcal{L}_r \subset \mathcal{L}_a \subset \mathcal{L}(G)$ of sublanguages of $\mathcal{L}(G)$, where the objective is to restrict the behavior of the system to the range delimited by $\mathcal{L}_r$ and $\mathcal{L}_a$ ($\mathcal{L}_a$ is the maximum admissible behavior and $\mathcal{L}_r$ is the minimum required behavior). Additional requirements can be imposed in the specifications (e.g. guarantee that blocking is avoided).

Considering the how S can modify the behaviour of G, a general control paradigm for this interaction is used [120]. S observes some or all the events that G executes, and tells G which events from the active event set are allowed. This means that S has the ability to disable some (not necessarily all) of the feasible events of G. The feedback control applied by S over G is dynamic in the sense that the decision about which events will be disable is

---

[6] Similar to the feedback control previously introduced.

allowed to change (i.e. every time the system visits a certain state the set of allowed events might by different). To conclude, S is limited in terms of observing the events executed by G (*observable events* in E) and limited in terms of disabling feasible events of G (*controllable events* in E).

Control under partial controllability

The control problem considers a discrete event system modelled by a pair of languages $\mathcal{L}$ and $\mathcal{L}_m$, where $\mathcal{L}$ is the set of all strings the system can generate and $\mathcal{L}_m \subseteq \mathcal{L}$ is the marked language that represents the completion of specific operations. $\mathcal{L}$ and $\mathcal{L}_m$ are the languages generated by the automaton $G = (X, E, f, \Gamma, x_0, X_m)$, where E is the event set and X the state space (needs not be finite).

To *control* this system we will add a supervisor S to interact with G as depicted in Figure 43. Consider E partitioned into two disjoint subsets E_c and E_{uc} ($E = E_c \cup E_{uc}$), where E_c is the set controllable events (i.e. can be disabled by the supervisor) and E_{uc} is the set of uncontrollable events. For now we assume that all events in E executed by G are observed by the supervisor S.

The transition function of G can be controlled by S by dynamically enabling/disabling the controllable events of G. A supervisor S is a function from the language generated by G to the power set of E:

Equation 19

$$S: \mathcal{L}(G) \to 2^E$$

For each $s \in \mathcal{L}(G)$ generated so far by G (under the control of S), $S(s) \cap \Gamma\big(f(x_0, s)\big)$ is the set of enabled events G can execute at its current state. G cannot execute an event that is in its current active event set if that event is not also contained in the supervisor enabled events. By definition, a supervisor is not allowed to disable a feasible event that is part of the uncontrollable event set. A supervisor S is *admissible* if for all $s \in \mathcal{L}(G)$ the following holds

Equation 20

$$E_{uc} \cap \Gamma\big(f(x_0, s)\big) \subseteq S(s)$$

$S(s)$ is called the *control action* at s and S the *control policy*. This feedback control loop is an instance of dynamic feedback: the domain of S is $\mathcal{L}(G)$ and not X, thus the control action may change on different visits to the same state.

Considering a system modelled by G and an admissible S, the resulting close-loop system is denoted by $S\ /\ G$ (S controlling G). This controlled system is a discrete event system and we can characterize its generated and marked languages, which are simply subsets of $\mathcal{L}(G)$ and $\mathcal{L}_m(G)$ containing the strings that remain feasible in the presence of S.

Definition 11:Languages generated and marked by S / G

The language generated by $S\ /\ G$ is defined recursively as follows:

1. $\varepsilon \in \mathcal{L}(S/G)$

2. $\left[\left(s \in \mathcal{L}(S/G)\right)\ and\ \left(s\sigma \in \mathcal{L}(G)\right)\ and\ \left(\sigma \in S(s)\right)\right] \Leftrightarrow [s\sigma \in \mathcal{L}(S/G)]$

The language marked by $S\ /\ G$ is defined as follows:

$$\mathcal{L}_m(S\ /\ G) := \mathcal{L}(S\ /\ G) \cap \mathcal{L}_m(G)$$

The notion of blocking previously defined for automata is also relevant for $S\ /\ G$ since this discrete event system has an associated generated and a marked language.

Definition 12: Blocking in controlled system

The DES $S\ /\ G$ is blocking if

$$\mathcal{L}(S/G) \neq \overline{\mathcal{L}_m(S/G)}$$

and non blocking when

$$\mathcal{L}(S/G) = \overline{\mathcal{L}_m(S/G)}$$

The blocking properties of $S\ /\ G$ are the result of S and of the structure of G. Thus the supervisor S controlling G is blocking if $S\ /\ G$ is blocking. Since marked strings represent completed operations (by design at modeling), a blocking supervisor results in a controlled system that cannot complete all operations.

Several results exist that can help us to deal with the presence of uncontrollable events. We will first look at the generated language $\mathcal{L}(S/G)$ and later into the marked language $\mathcal{L}_m(S/G)$ in order to address blocking issues.

Definition 13: Controllability theorem

Consider DES $G = (X, E, f, x_0)$ where $E_{uc} \subseteq E$ is the set of uncontrollable events. Let $K \subseteq \mathcal{L}(G)$, where $K \neq 0$. Then there exists a supervisor S such that $\mathcal{L}(S/G) = \overline{K}$ if and only if

$$\overline{K}E_{uc} \cap \mathcal{L}(G) \subseteq \overline{K}$$

This condition on K is called the controllability condition.

If the controllability condition is satisfied, then the supervisor that achieves exactly the required behaviour, $\overline{K}$ is

$$S(s) = \left[E_{uc} \cap \Gamma\big(f(x_0, s)\big)\right] \cup \{\sigma \in E_c : s\sigma \in \overline{K}\}$$

This controllability condition in the controllability theorem (CT) is a central concept in supervisory control: "*if you cannot prevent it, then it must be legal*". A general definition for controllability follows.

Definition 14: Controllability

Let K and $M = \overline{M}$ be languages over event set E. Let E_{uc} be a designated subset o E. K is said to be controllable with respect to M and E_{uc} if

$$\overline{K}E_{uc} \cap M \subseteq \overline{K}$$

This condition on K is called the controllability condition.

This result is relevant for the realization of supervisors. Assuming a language $K \subseteq \mathcal{L}(G)$ is controllable then from the controllability theorem we know that a supervisor S exists such that $\mathcal{L}(S/G) = \overline{K}$. In order to build an automaton realization of S we need to build an automaton that marks the language $\overline{K}$.

We now turn into supervisory control problems concerned both with $\mathcal{L}(S\ /\ G)$ and $\mathcal{L}_m(S\ /\ G)$. In these cases the specification on the controlled system is a sublanguage of $\mathcal{L}_m(G)$ and will require that the supervisor S is nonblocking, i.e., $\overline{\mathcal{L}_m(S\ /\ G)} = \mathcal{L}(S\ /\ G)$. The controllability theorem is extended to deal with these cases.

Definition 15: Nonblocking controllability theorem

Consider DES $G = (X, E, f, x_0, X_m)$ where $E_{uc} \subseteq E$ is the set of uncontrollable events. Consider the language $K \subseteq \mathcal{L}_m(G)$, where $K \neq \emptyset$. There exists a nonblocking supervisor S for G such that

$$\mathcal{L}_m(S/G) = K \text{ and } \mathcal{L}(S/G) = \overline{K}$$

if and only if the following two conditions hold:

1. Controllability: $\overline{K}E_{uc} \cap \mathcal{L}(G) \subseteq \overline{K}$
2. Closure: $K = \overline{K} \cap \mathcal{L}_m(G)$

As in the case of the controllability theorem, if the conditions in the nonblocking controllability theorem hold them the nonblocking supervisor is

$$S(s) = \left[E_{uc} \cap \Gamma\big(f(x_0, s)\big)\right] \cup \{\sigma \in E_c : s\sigma \in \overline{K}\}$$

This means the nonblocking supervisor can be realized the same way as previously presented. The only difference is that in this case it must also respect the $\mathcal{L}_m(G)$-closure condition $K = \overline{K} \cap \mathcal{L}_m(G)$.

Control under partial observation

We now consider the case where the supervisor does not observe all the events G executes. In this case the event set is partitioned into two disjoint subsets E_o and E_{uo} ($E = E_o \cup E_{uo}$), where E_o is the set observable events (i.e. can be seen by the supervisor) and E_{uo} is the set of unobservable events. Causes for this limited observability are the limitations of sensors present in the system and the distributed nature of some systems. The feedback control loop under partial observation is illustrated in Figure 44 and includes a natural projection P between G and the supervisor. In these cases we will denote the (partial-observation) supervisor as S_P to reflect the presence of P.

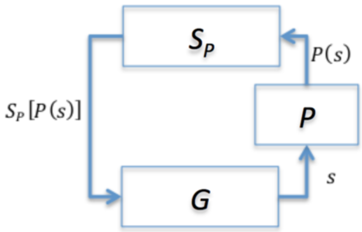


Figure 44: Feedback control loop of supervisory control under partial observation

The presence of P turns $\mathrm{S_P}$ unable to distinguish between two strings $\mathrm{s_1}$ and $\mathrm{s_2}$ that have the same projection. For such $\mathrm{s_1, s_2} \in \mathcal{L}(\mathrm{G})$ where $\mathrm{P(s_1) = P(s_2)}$ the supervisor will necessarily issue the same control action $\mathrm{S_P[P(s_1)]}$.

A partial-observation supervisor $\mathrm{S_P}$ is a function

Equation 21

$$S_P \colon P[\mathcal{L}(G)] \rightarrow 2^E$$

and S_P is called a P-supervisor. In a P-supervisor the control action can only change after an observable event as occurred, i.e. when $P(s)$ changes. No specific assumptions about the relation between the controllability and observability properties of an event: an unobservable event could be controllable; a uncontrollable event could be observable; etc.

As in the case of supervisors, P-supervisors have to be admissible, i.e., they should not disable feasible uncontrollable events. A P-supervisor S_P is *admissible* if for all $t = t\sigma \in P[\mathcal{L}(G)]$ the following holds

Equation 22

$$E_{uc} \cap {}^{\circ}\mathrm{F}^{\circ}\mathrm{C} \bigcup_{s \in L_t} \left[\Gamma\big(f(x_0, s)\big)\right] \subseteq S_P(t)$$

The closed-loop behaviour of $S_P \;/\; G$ is defined the same way as to the case of full observation.

Definition 16: Languages generated and marked by $S_p \;/\; G$

The language generated by $S_P \;/\; G$ is defined recursively as follows:

1. $\varepsilon \in \mathcal{L}(S_P/G)$

2. $\big[\big(s \in \mathcal{L}(S_P/G)\big)\ and\ \big(s\sigma \in \mathcal{L}(G)\big)\ and\ (\sigma \in S_P[P(s)])\big] \Leftrightarrow [s\sigma \in \mathcal{L}(S_P/G)]$

The language marked by $S_P \;/\; G$ is defined as follows:

$$\mathcal{L}_m(S_P \;/\; G) := \mathcal{L}(S_P \;/\; G) \cap \mathcal{L}_m(G)$$

From this definition it is clear that the languages $\mathcal{L}(S_P \;/\; G)$ and $\mathcal{L}_m(S_P \;/\; G)$ are defined over E and not E_o, corresponding to the closed loop behaviour of G before the effect of projection P.

Several results exist that can help us to deal with the presence of unobservable events in addition to the presence of uncontrollable events. The generalization of the controllability theorem and of the nonblocking controllability theorem to control under partial observation will require an additional condition besides controllability and $\mathcal{L}_m(G)$-closure: *observability*.

Definition 17: Observability

Let K and $M = \overline{M}$ be languages over event set E. Let E_c be a designated subset of E. Let E_o be another designated subset of E with P as the corresponding natural projection from E_o to E_o^*. K is said to be observable with respect to M, E_o and E_c if for all $s \in \overline{K}$ and for all $\sigma \in E_c$,

$$(s\sigma \in \overline{K})\ and\ (s\sigma \in M) \Rightarrow P^{-1}[P(s)]\sigma \cap \overline{K} = \emptyset$$

This condition on K is called the controllability condition.

This observability condition can be phrased as: "*if you cannot differentiate between two strings, then these two strings should require the same control action*".

With this additional concept we can now generalize the controllability and the nonblocking controllability theorems to control under observation.

Definition 18: Controllability and observability theorem

Consider DES $G = (X, E, f, x_0, X_m)$ where $E_{uc} \subseteq E$ is the set of uncontrollable events and $E_o \subseteq E$ is the set of observable events. Let P be the natural projection from E_o to E_o^*. Consider also the language $K \subseteq \mathcal{L}_m(G)$, where $K \neq \emptyset$. There exists a nonblocking P-supervisor S_P for G such that

$$\mathcal{L}_m(S_P/G) = K \text{ and } \mathcal{L}(S_P/G) = \overline{K}$$

if and only if the following three conditions hold:

1. K is controllable with respect to $\mathcal{L}(G)$ and E_{uc}
2. K is controllable with respect to $\mathcal{L}(G)$, E_o and E_c
3. K is $\mathcal{L}_m(G)$-closed

As in the case of the controllability and nonblocking controllability theorems, the controllability and observability theorem is constructive. If the conditions for controllability, observability and $\mathcal{L}_m(G)$-closure are satisfied, then a supervisor that will achieve the required behavior exists. For $s \in \mathcal{L}(G)$ with $P(s) = t$

$$S_P(t) = E_{uc} \cup \{\sigma \in E_c : P^{-1}[P(s)]\sigma \cap \overline{K} \neq \emptyset\}$$

highlighting the separation of estimation from control. This shows that the estimation policy is independent from the control policy.

Modular specifications

A common challenge in practical applications is the fact that the state space of a system can grow exponentially in the number of its components if coupled by parallel composition. Additionally specifications are often complex and involve the conjunction of individual specifications. Exploiting the modularity of the system and the structure of the specifications is an important way to deal with these issues.

Assuming we have a discrete event system G that needs to be controlled and that the safety specifications are in the form of language $\mathcal{L}_a$, where $\mathcal{L}_a$ can be decomposed as the intersection of two prefix-closed languages $\mathcal{L}_a = \mathcal{L}_{a1} \cap \mathcal{L}_{a2}$. If we have previously synthesized supervisor S_1 to handle $\mathcal{L}_{a1}$ and supervisor S_2 to handle $\mathcal{L}_{a2}$, then the modular architecture in presented in Figure 45 can be applied to control system G according to the safety specifications given by $\mathcal{L}_a$.

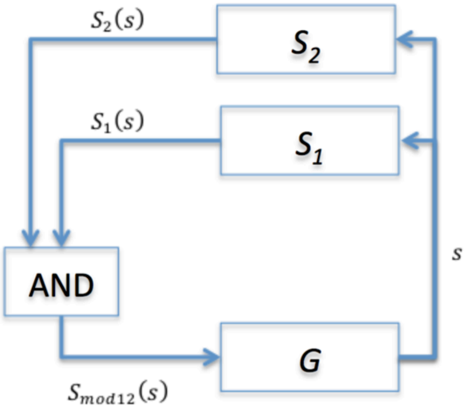


Figure 45: Control architecture with two supervisors

A discussion on forms of modular control that exploit the modularity of the system can be found in [121].

Decentralized control

Decentralized supervisory control is based on the idea of local supervisors (agents) simultaneously controlling a discrete event system G with each supervisor having access to local information and local controls (Figure 46). This is similar to modular supervision except that we have added the additional constraint of partial information and partial control:

individual supervisors may be partial-observation supervisors and their respective sets of observable and controllable events may be different. The distributed nature of the system makes that supervisors at different "sites" in the (distributed) system see the effect of different sets of sensors (some overlapping) and may control different sets of actuators (again, possibly overlapping).

The overall control task, as embodied in some constraint language $K \subset \mathcal{L}(G)$, often splits into subtasks for which *local* supervisors are simple to realize. The control actions of the individual supervisors, $S_i(s)$ are based on their own local observations of the system behavior (denoted by projections P_i). The control action on G is the fusion, according to a specific rule, of the individual $S_i(s)$. This makes decentralized control attractive. However, the question is if such local supervisors acting concurrently achieve the desired control objective, and if so, whether they achieve it in an *optimal* way [122].

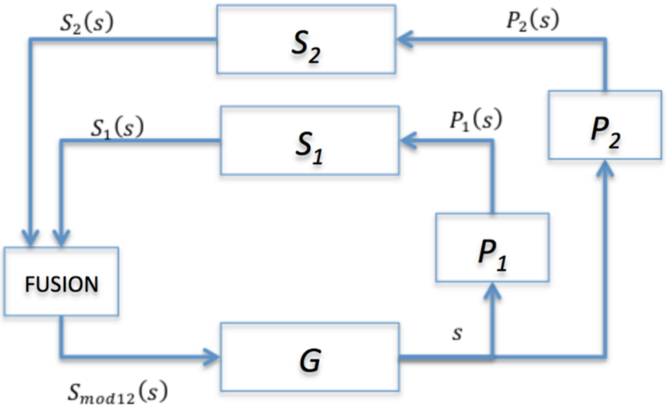


Figure 46: Decentralized control architecture

In modular control, the control action on G, since each supervisor is responsible for one (local) specification, is the intersection of the sets of events enabled by each supervisor (the overall specification is the intersection of all the local specifications). The situation is different in the case of decentralized control.

Considering the case of a single specification $K \subset \mathcal{L}_m(G)$ if safety and nonblocking are to be addressed, or $K \subset \mathcal{L}(G)$ for the case of safety only, the individual supervisors work as a team in order to jointly achieve K ($\overline{K}$). An important result in decentralized control is that different fusion rules have different properties in terms of the class of controlled languages that they can achieve [121] so we should not limit fusion to intersection. The fusion rule is related to the specific control *architecture*, namely, to a special case of the generic architecture of Figure 46. For now we will focus on two of the simplest fusion rules and their corresponding architectures: *conjunctive* and *disjunctive* architectures.

In the conjunctive architecture the fusion rule is intersection of enabled events and the corresponding control policy $S_{dec}^{conj}: \mathcal{L}(G) \to 2^E$ acting on G is

Equation 23

$$S_{dec}^{conj}(s) = \bigcap_{i=1}^{n} S_i(s)$$

In the disjunctive architecture the fusion rule is union of enabled events and the corresponding control policy $S_{dec}^{disj}: \mathcal{L}(G) \to 2^E$ acting on G is

Equation 24

$$S_{dec}^{disj}(s) = \bigcup_{i=1}^{n} S_i(s)$$

The resulting controlled behaviour in both architectures is described by the languages $\mathcal{L}(S_{dec} \ / \ G)$ and $\mathcal{L}_m(S_{dec} \ / \ G)$, with the appropriate superscript for S_{dec}. The global behavior is given by $\mathcal{L}(S_{dec} \ / \ G)$ in opposition to the local behaviors $P_i[\mathcal{L}(S_{dec} \ / \ G)]$ seen by the individual supervisors. At the global level this results in the closed loop behavior $\overline{K} = \mathcal{L}(G) \cap P^{-1}(K)$.

These two fusion rules are decentralized in the sense the control action is performed at each actuator, that is, at each controllable event. Associated with G are the four usual sets E_c (controllable events), E_{uc} (uncontrollable events), E_o (observable events), and E_{uo} (unobservable events). Considering the decentralized supervisors G_i we have

$$E_{c,i} \subseteq E, \bigcup_{i=1}^{n} E_{c,i} = E_c$$

$$E_{o,i} \subseteq E, \bigcup_{i=1}^{n} E_{o,i} = E_o$$

$$P_i: E^* \to E_{o,i}^*$$

The key feature of the control policies used in the context of the conjunctive architecture to generate the desired language K is that each supervisor enables an event when it needs to enable this event in order to allow some string in K, even if enabling this event may also allow a string not in K to occur. This possibility is “disregarded” by this supervisor. This situation is referred to as the supervisor being permissive when it is in doubt, i.e., when it is in a state where a control conflict occurs.

Definition 19: CP-coobservability

Let K and $M = \overline{M}$ be languages over event set E. Let $E_{c,i}$ and $E_{o,i}$ be sets of controllable and observable events, respectively, for $i = 1,..,n$. Let $P_i$be the natural projection corresponding to $E_{o,i}$, with $P_i: E^* \rightarrow E_{o,i}^*$.

K is said to be CP-coobservable with respect to M, $E_{o,i}$, and $E_{c,i}$, $i = 1,..,n$, if for all $s \in \overline{K}$ and for all $\sigma \in E_c = \bigcup_{i=1}^{n} E_{c,i}$,

$(s\sigma \notin \overline{K})$ and $(s\sigma \in M) \Rightarrow$

there exists $i \in \{1,..,n\}$ such that $P_i^{-1}[P_i(s)]\sigma \cap \overline{K}$ and $s\sigma \in E_{c,i}$

CP-coobservability can be understood as if there is an event σ needs to be disabled, then at least one of the supervisors that can control σ has to know that it must disable σ, that is, from this supervisor's viewpoint, disabling σ does not prevent any string in K. As a consequence of CP-coobservability, each supervisor can follow the permissive policy when it is uncertain about whether it should disable or not an event. This notion leads to the decentralized version of the Controllability and Observability Theorem (COT) presented in previously.

Definition 20: Controllability and Coobservability Theorem – Conjunctive case

Consider system $G = (X, E, f, \Gamma, x_0, X_m)$ where $E_{uc} \subseteq E$ is the set of uncontrollable events, $E_c = E \setminus E_{uc}$ is the set of controllable events, and $E_o \subseteq E$ is the set of observable events.

For each site $i = 1,..,n$ consider the set of controllable events $E_{c,i}$ and the set of observable events $E_{o,i}$; overall, $\bigcup_{i=1}^{n} E_{c,i} = E_c$ and $\bigcup_{i=1}^{n} E_{o,i} = E_o$. Let $P_i$be the natural projection from E^* to $E_{o,i}^*$, $i = 1,..,n$. Consider also the language $K \subseteq \mathcal{L}_m(G)$, where $K = \emptyset$. There exists a nonblocking decentralized supervisor S_{dec}^{conj} for G such that

$\mathcal{L}_m(S_{dec}^{conj} / G) = K$ and $\mathcal{L}(S_{dec}^{conj} / G) = \overline{K}$

if and only if the three following conditions hold:

1. K is controllable with respect to $\mathcal{L}(G)$ and E_{uc};

2. K is CP-coobservable with respect to $\mathcal{L}(G)$, $E_{o,i}$ and $E_{c,i}$, $i = 1,..,n$;

3. K is $\mathcal{L}_m(G)$-closed.

In a disjunctive architecture an event is globally enabled if it is enabled by at least one local supervisor. In this case, the default control action when a supervisor is in a situation of control conflict regarding a controllable event should be to disable the event, that is, to be *antipermissive*. If another supervisor that also controls this event is sure about its enablement, then it could alone ensure that the event is globally enabled since the fusion rule is union. Otherwise, the combination "disjunctive and permissive" would not take advantage of the fact that the supervisors can share the work on common controllable events and thus it would be overly restrictive in terms of the class of languages achievable under control. The dual notion of CP-coobservability in the case of the disjunctive architecture, termed DA-coobservability, corresponds to the Disjunctive architecture with the Antipermissive policy.

Definition 21: DA-coobservability

Let K and $M = \overline{M}$ be languages over event set E. Let $E_{c,i}$ and $E_{o,i}$ be sets of controllable and observable events, respectively, for $i = 1, .., n$. Let P_i be the natural projection corresponding to $E_{o,i}$, with $P_i: E^* \rightarrow E_{o,i}^*$.

K is said to be DA-coobservable with respect to M, $E_{o,i}$, and $E_{c,i}$, $i = 1, .., n$, if for all $s \in \overline{K}$ and for all $\sigma \in E_c = \bigcup_{i=1}^{n} E_{c,i}$,

$(s\sigma \in \overline{K}) \Rightarrow$

there exists $i \in \{1, .., n\}$ such that $\left(P_i^{-1}[P_i(s)] \cap \overline{K}\right)\sigma \cap M \subseteq \overline{K}$ and $\sigma \in E_{c,i}$

DA-coobservability is stated in terms of events that need to be enabled in order to achieve K, in contrast with CP-coobservability which is stated in terms of events that need to be disabled. In DA-coobservability can be understood as if there is an event σ that needs to be enabled, then at least one of the supervisors that can control σ has to know that it must enable σ. From this supervisor's viewpoint, enabling σ does not allow any string in $M \setminus \overline{K}$. Using this rule each supervisor can follow the antipermissive policy when it is uncertain about whether it should enable or not an event. This notion leads to the "Controllability and Coobservability Theorem – Disjunctive Architecture".

Definition 22: Controllability and Coobservability Theorem – Disjunctive case

Consider system $G = (X, E, f, \Gamma, x_0, X_m)$ where $E_{uc} \subseteq E$ is the set of uncontrollable events, $E_c = E \setminus E_{uc}$ is the set of controllable events, and $E_o \subseteq E$ is the set of observable events.

For each site $i = 1, .., n$ consider the set of controllable events $E_{c,i}$ and the set of observable events $E_{o,i}$; overall, $\bigcup_{i=1}^{n} E_{c,i} = E_c$ and $\bigcup_{i=1}^{n} E_{o,i} = E_o$. Let P_i be the natural projection from E^* to $E_{o,i}^*, i = 1, .., n$. Consider also the language $K \subseteq \mathcal{L}_m(G)$, where $K = \emptyset$. There exists a nonblocking decentralized supervisor S_{dec}^{disj} for G such that

$\mathcal{L}_m(S_{dec}^{disj} / G) = K$ and $\mathcal{L}(S_{dec}^{disj} / G) = \overline{K}$

if and only if the three following conditions hold:

1. K is controllable with respect to $\mathcal{L}(G)$ and E_{uc};
2. K is DA-coobservable with respect to $\mathcal{L}(G)$, $E_{o,i}$ and $E_{c,i}, i = 1, .., n$;
3. K is $\mathcal{L}_m(G)$-closed.

The complementarity of CP- and DA-coobservability can be exploited in the cases where the controllable events leading to a violation of CP-coobservability do not lead to a violation of DA-coobservability, and vice versa. This is possible when the control architecture includes both types of fusion rules, conjunction and disjunction, resulting is what is called the combined architecture [123] (this combination is possible since the fusion rule is decentralized and can be performed at each actuator individually).

7.5. Extensions for changeability

This chapter has reviewed current state of the art in terms of discrete event systems theory. This review shows that the concepts of *System of Systems* and *Changeability* introduced in chapters 5 and 6, respectively, are not fully covered by current discrete event systems theory and existing frameworks. Existing results on controllability, observability and decentralised control have to extend to encompass these additional concepts. Moreover, the notion of life cycle and sustainability of the system throughout its entire life cycle also have to be introduced.

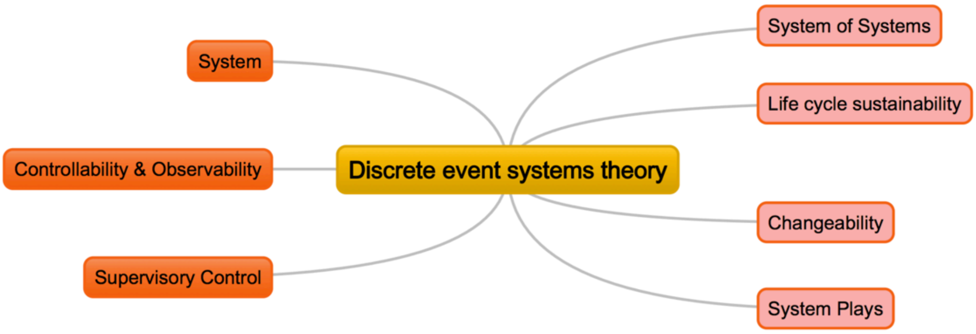


Figure 47: Discrete event system theory extensions

This will be the focus of the next chapter: what extensions in current theory are needed in order to enhance the life cycle sustainability of systems?

8. Event driven framework for Changeability

This chapter will lay the foundations for an event driven framework for changeability. It will start by a set of definitions followed by the introduction of the changeability function and changeability playbook. After introduction changeability in the context of supervisory control, this chapter will end with a first definition of the changeability framework.

8.1. System of systems, playbook and plays

The section will use the discrete event theory framework presented in the preceding chapter to describe the concepts introduced earlier. The concept of System of System, playbook and play discussed in chapter 5 can now be formalised.

A generic definition (Definition 1) of System of Systems that reflects its key features was presented. In this definition it is stated that SoS are large-scale integrated systems that are heterogeneous and independently operable on their own, but are networked together for a common goal. As previously discussed, a system is considered as an agent/actor (an autonomous entity) having its own operational instructions and goals, with modelling and supervisory capabilities integrated. If such a system is embedded in a larger system (SoS), relations with other systems may be defined in many ways, namely composition or interaction/play.

The concepts of system play and playbook (a collection of system plays), based on the notions of “system plays” [97] and “playbook” [73, 124], will be used to extend the current concept of supervisory control and will be a crucial element to introduce the concept of SoS into current discrete event systems theory.

Looking at the example of human interaction with complex automation, where a myriad of challenges so that a satisfactory, safe and effective mix of human and machine roles results, can be helpful to understand how supervision and delegation can be designed in SoS. On the one hand, the "technological imperative" [125] argues for ever-increasing delegation of roles and performance duties to automation in order to reduce the costs (in terms of workload, training, person-hours, boredom and, in some cases, physical safety) of human operators. On the other hand, there are now well-understood drawbacks [126] to the over-use of automation, especially when that automation operates in a less than-perfect manner and/or is implemented in such a fashion that its use will be "clumsy" for the humans that must engage it.

Before proceeding with a formal definition of the system play and playbook concepts, a number of key basic concepts must be defined (adapted from [95]):

- Independent systems (or actors) - independent integrated systems, with own goals and capabilities to react to changes in their environment.
- Scenario: set of operational needs and rules defining the operation and interactions between independent agents acting within a set of constraints.
- Initiation Events: events that trigger the execution of a play; these can either be exogenous (generated by the external environment or context) or endogenous (generated by the agents in response to their realisation of their state in the running of the current play).
- Acting: execution of a play under the stimuli of initiating events.
- Scene Sequence: Observation of acting by an external observer.
- Director: This is an external to the play agent (or collection of agents) that may set objectives and generate initiation events and define games.

A play is a configuration that defines how independent systems, entering as autonomous agents with their own individual operational instructions and goals, interact with each other. Each individual system retains its individual goals and participates in the "composition" as an intelligent agent with relative autonomy and plays its role as an "actor" in the overall play.

Definition 23: System play

A system play (or play) π is five-tuple $\pi = (\gamma, o, \rho, E \cup \{\varepsilon\}, \lambda)$ where γ is a set of *independent systems* which interact under a given *scenario*, defined by a finite set of *operational needs* o and a *rules of interaction* function $\rho: \gamma \times o \rightarrow 2^{\gamma}$, initiated by events from the finite set $E \cup \{\varepsilon\}$ and executed in an environment defined by an active event function (or feasible event function) $\lambda: \gamma \rightarrow E$.

Different rules of interaction, which along with the operational needs constitute the scenario, amongst independent systems define the role of each system in the play.

The playbook concept is based on the metaphor of a sports team's book of acceptable plays. We will use the play and playbook concepts to represent a "delegation" approach to system-system interactions: allows a system[7] to task or delegate authority to another system with much of the same flexibility with which a human supervisor or team captain can delegate objectives, methods, constraints and even detailed instructions to subordinates.

Definition 24: SoS Playbook

A SoS playbook B contains plays for different purposes and the rules to switch between them is a triple $\mathrm{B} = (\Pi, \mathrm{O}, \mathrm{P})$, where Π is a collection plays, O is a set of different purposes and P is a function $\mathrm{P}: \Pi \times \mathrm{O} \rightarrow \Pi$ to switch between plays.

We are now ready to provide a definition of System of Systems.

Definition 25: System of Systems

A System of Systems Σ is a six-tuple $\Sigma = (\Gamma, \mathrm{B}, \Delta, \Omega, \pi_0, \Pi)$ where Γ is the set of independent systems, B is a playbook, Δ is the director, Ω is the collection of scene sequences, π_0 is the initial play and Π the set of desired plays.

The set of independent systems Γ considered in the definition of SoS is different from the set of independent systems γ from the definition of play: whilst Γ is the set of all systems that are part of the SoS, γ is the set of systems involved in the play.

This last definition introduces two new elements, the *director* and the *scene sequences*. The scene sequences are the equivalent of the *observes* concept that have been previously introduced in section 7. The *director* will be introduced and formalised in the next section.

8.2. Changeability function

Changeability is a function of efficiency, adaptability and automation/autonomy of the systems and its capacity to respond to modifications in the operational needs and/or rules of engagement. Changeability can be achieved by the SoS ability and potential to realize fast

---

[7] In this context a system can either be fully autonomous, supervised by humans or even a human.

adjustments within adaptability corridors, at multiple levels with low effort. However in some cases the SoS must be able to cross the boundaries of the changeability corridors, including the ability to increase/decrease the SoS capacity and functionality (Figure 48). For example, reducing the level of autonomy of a certain system might be a suitable measure. SoS change may take place at the physical (hard) level or at the logical or organisational (soft) level [112], but physical change almost always requires logical change at the software level.

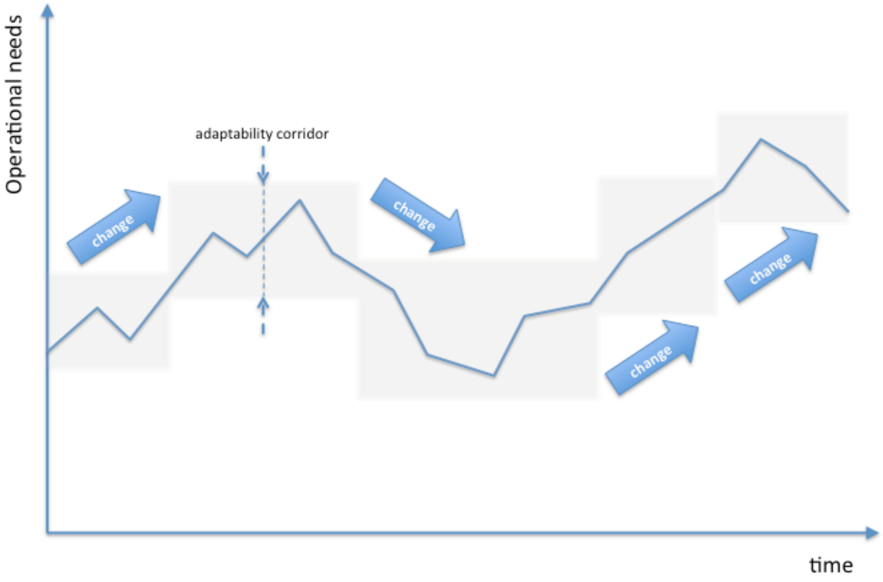


Figure 48: Switching between adaptability corridors by change

The change enablers, previously identified in chapter 6, have to possess characteristics to enable change both at the physical and logical levels. Examples of change enablers include scalability, modularity or compatibility.

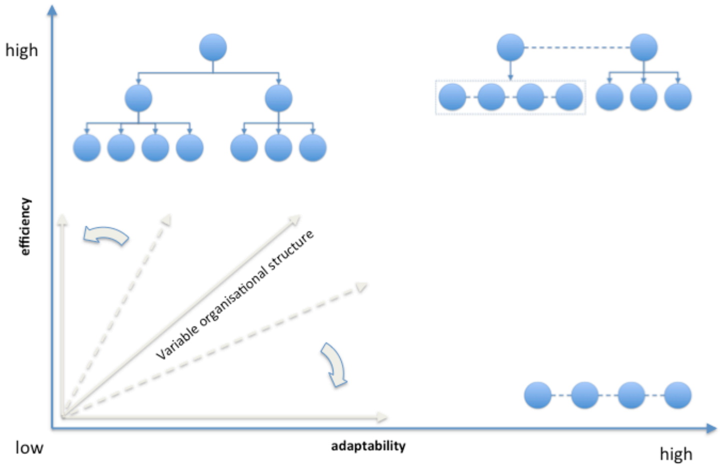


Figure 49: Changing the organizational structure

Each element contributes to the change of the SoS to a different degree, to guarantee the SoS configuration can deliver the required performance under the current requirements. We will concentrate the following discussion on change enablers related with logical levels, i.e., related with the organisational structure or the configuration of the systems (and not with their physical structure).

The changeability function is nothing else than the *director* (Δ) introduced in the definition of SoS. This function is external to the play and can be implemented by an agent or collection of agents that set objectives, call plays from the playbook, generate initiation events and ultimately define the rules of the game.

Definition 26: Changeability function (director)

The Changeability function Δ is a function $\Delta\colon \mathrm{O} \times \mathrm{B} \rightarrow \mathrm{B}$, where O is the set of system purposes and B is the playbook, that based on the collection of scene sequences Ω decides if it needs to change play and if so decides on which play to call.

The changeability function analyses the behaviour of the system and in case the current operational needs fall outside the adaptability corridor of the current play decides on a new play to call (Figure 50).

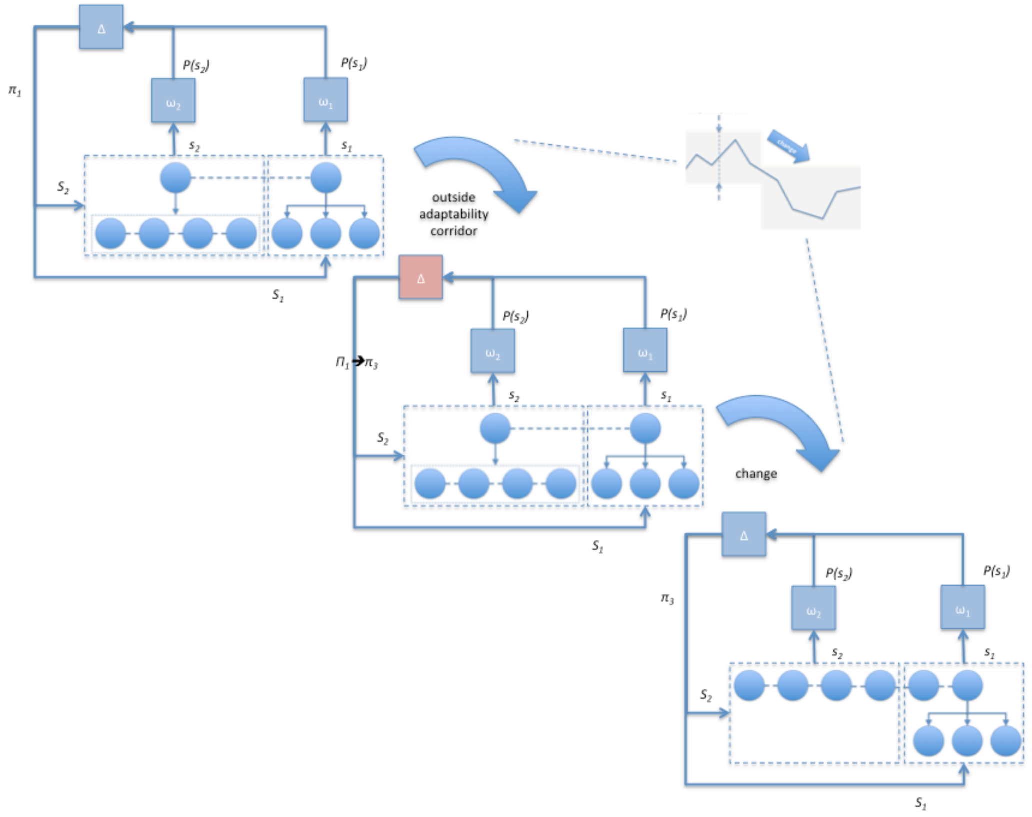


Figure 50: Changeability function

To be able to adapt to the continuously changing operational needs and/or rules of engagement, using the SoS's enablers, a two-stage control loop-based approach for the shaping the configuration of the SoS is proposed. Figure 51 illustrates the proposed control loop at the two main levels of change.

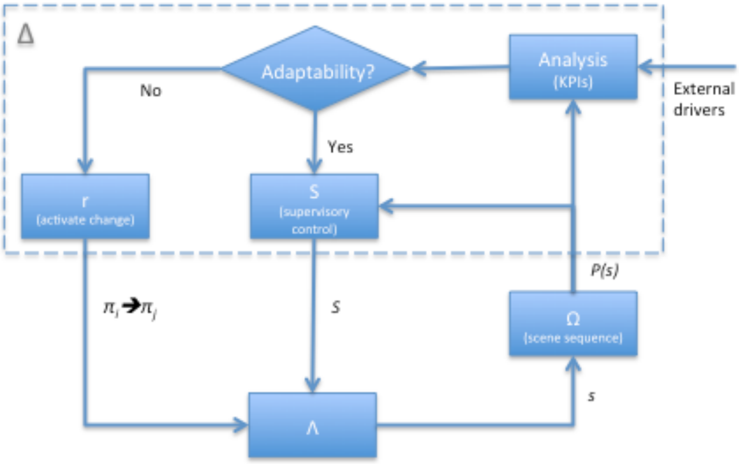


Figure 51: Changeability control loop

The starting point is an operating SoS that is continuously analysed and evaluated with respect to its change drivers and enablers. After comparing the target and the actual performance a potential need of change is identified. If the system inherent adaptability is sufficient to meet the required operational needs, the control loop runs through the process ''supervisory control'' and adjustments to the SoS are executed within its respective pre-defined corridor of adaptability (possible changes in the configuration of some system, etc.). If the actual current changeability of the system lies within the desired/required ability zone, it means that sufficient adaptability is presently available within the current configuration, and can be utilized without any further reconfiguration/change. If needed changes are made at this level with built-in functionality and capabilities. The objective is to satisfy operational requirements with minimal changes within the current adaptability corridor.

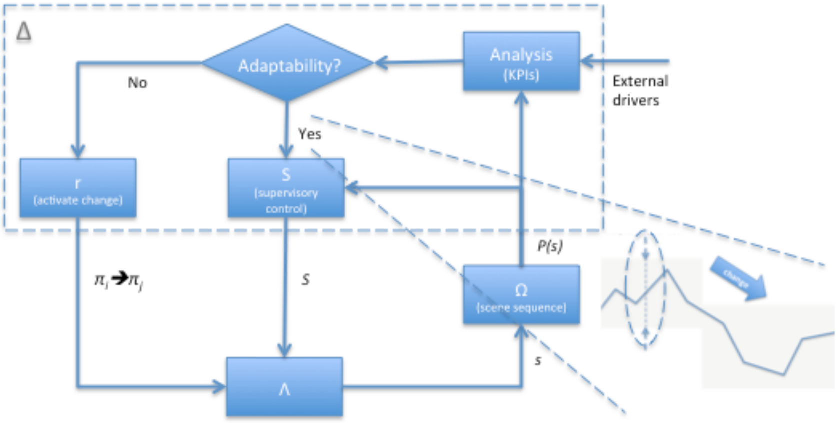


Figure 52: Changeability control loop (within adaptability corridor)

If the SoS inherent adaptability is not sufficient, the process ‘‘activate change’’ is run through and a new play is selected from the playbook. Then, planned measures for reconfiguring the SoS will be implemented either on systems, products, organisational, and/or soft levels to meet the required change

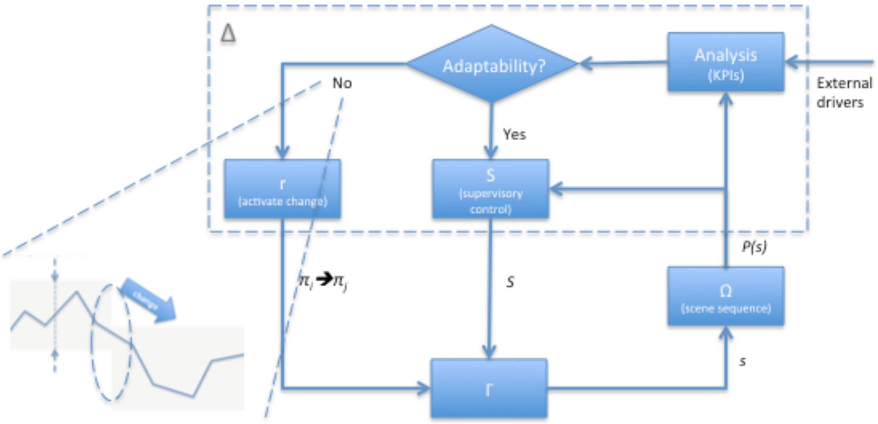


Figure 53: Changeability control loop (outside adaptability corridor)

These measures can be translated into parameters for the economic, ecological and socio-technical sustainability of the SoS. Therefore, the sustainability set points (Sustainability domain) have to be translated into concrete configuration measures for the purposes of reconfiguring the SoS (changeability domain) so as to make it possible to reach target sustainability values. For example, in the manufacturing systems domain, the use of less automation (results in energy savings, increased flexibility and higher number of humans) is an example of reducing automation and demonstrates an important characteristic of changeable manufacturing systems – scalability – in contrast with flexible manufacturing systems that are quite rigid outside of its built-in capacity and functionality (its corridor).

Using this change control loop model, the SoS change activities can be synchronized according to operational requirements by continuously comparing the changing requirements with the existing SoS capabilities. At this level, an extra control loop component needs to be included; a loop to switch between two domains: changeability and sustainability. The current values of the different used sustainability key figures would be compared with their desired values. Examples for sustainability key figures are: from a social Perspective, the labor turnover rate; from an ecologic perspective the carbon footprint or from an economic perspective, the return on investment.

8.3. Playbook changeability

The goal of the playbook is to enable the same degrees of flexibility in commanding and delegating tasks to systems that a human supervisor has to knowledgeable collaborators. Supervisors (or team captains) interacting with human subordinates decide how much and what instruction and constraints to impose on their subordinates (as a function of different factors such as time and capacity available, skills and confidence in the subordinate, and specific constraints of current scenario). Supervisors can provide very high level and minimal instruction about the objectives and methods (can “call a play”), leaving most of the decision making and execution responsibilities to the subordinate – at expense of less certainty about exactly what methods will be used. Alternatively, they can provide more detailed instruction about specific methods to be used (i.e., subtasks to be performed or avoided, specific resources to be used or not used, etc.) with a reduction in the uncertainty of how the task will be performed – at the expense of additional time spent in the tasking process.

Although the playbook is one of the components of the SoS it should not be considered static. The playbook should dynamically evolve in time to accommodate changes in the SoS corresponding to the different stages of its life cycle. Moreover, knowledge gained in the operation and evolution in the capabilities of the independent systems (amongst other reasons) open the possibility to include new plays and possibly make existing plays obsolete. This knowledge can be explored using the information collected in run-time and using methods like regression [127] to develop new plays.

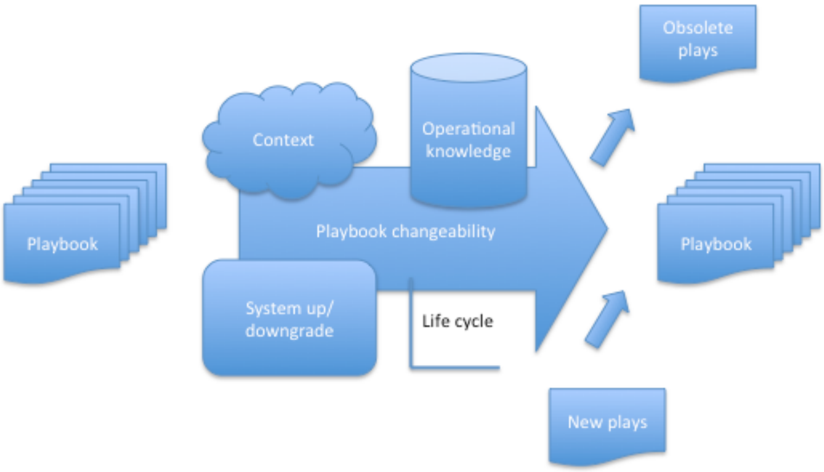


Figure 54: Playbook changeability

Recent techniques for big data analytics [128], data mining and different machine learning approaches [129] can be used for modelling and exploring new plays (or marking existing ones as obsolete) and to implement the playbook changeability.

8.4. Changeability framework

This section will lay the foundations for the future definition of a Changeability framework for Discrete Event System. This will be done by formalising the main changeability concepts in the discrete event framework and by highlighting the main results presented in chapter 7 that are applicable in this context.

In Definition 25 a system of systems was defined by a six-tuple Σ, where Γ is the set of independent systems. Each of these independents systems can be modelled as a nondeterministic automaton G_{nd} (Definition 10).

As introduced in chapter 7, decentralized supervisory control is based on the idea of local supervisors (agents) simultaneously controlling a discrete event system, represented by an automaton G, with each supervisor having access to local information and local controls. This implies partial information and partial control since individual supervisors may be partial-observation supervisors and their respective sets of observable and controllable events may be different.

This can be directly applicable to a system of systems, based on the idea of supervisors controlling each of the independent systems from the set Γ, with each supervisor having access to local information and local controls of the individual system. The distributed nature of the system of system makes that supervisors of the different systems, see the effect of different sets of sensors (with possible some overlapping in the results) and may control different sets of actuators (again, possibly overlapping in the effect over the overall system of systems).

The overall control task, as embodied in some constraint language $K \subset \mathcal{L}(\Sigma)$, often splits into subtasks for which *local* supervisors are simple to realize. The control actions of the individual supervisors, $S_i(s)$ are based on their own local observations (following for example the approach proposed in [130] and [131]) of the system behaviour (denoted by projections P_i). The control action on Σ is the fusion, according to a specific rule, of the individual $S_i(s)$. This rule can be derived from the *rules of interaction* function ρ that is defined for the current system play (π).

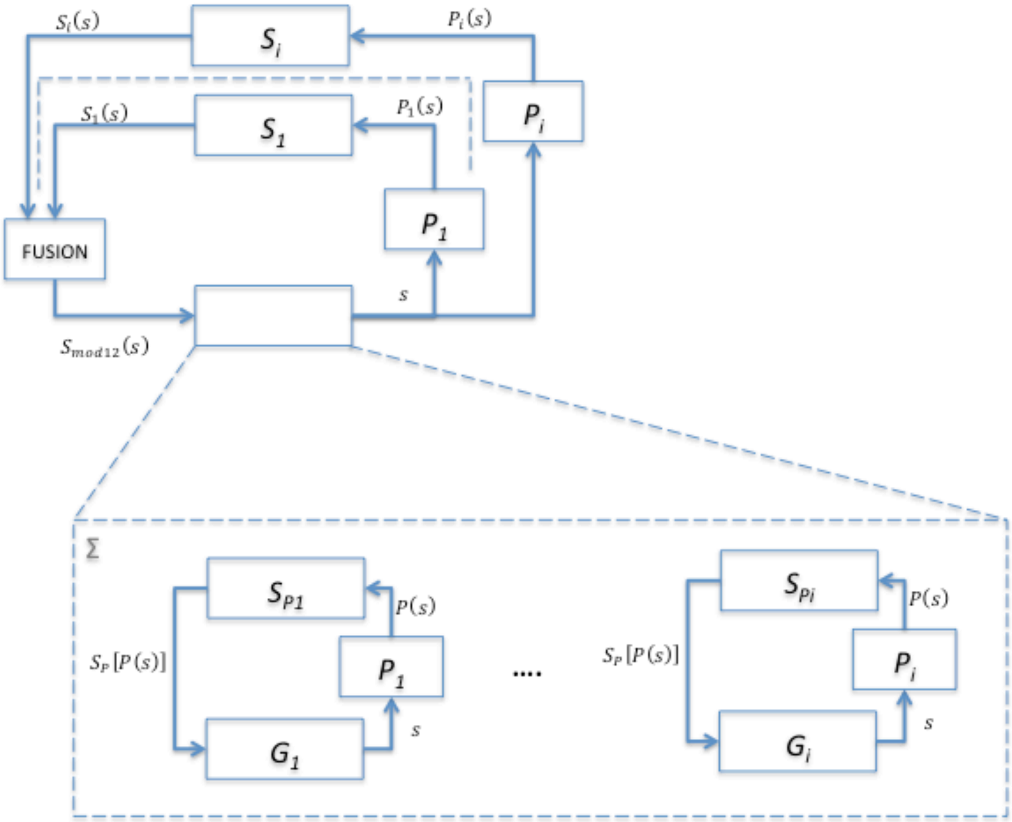


Figure 55: System of Systems control architecture

The building blocks of the changeability framework are represented in Figure 56. Although a first approach to the control architecture is presented, further developments are needed in order to transpose some of the results presented in chapter 7, like for example controllability and coobservability, to the context of system of systems and to the proposed framework.

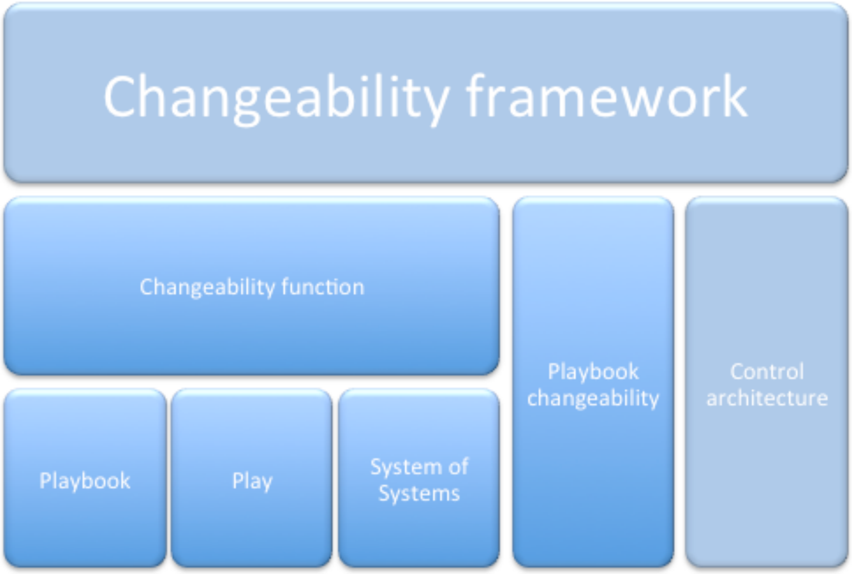


Figure 56: Changeability framework

Part III: Models and Applications

"To successfully respond to the myriad of changes that shake the world, transformation into a new style of management is required. The route to take is what I call profound knowledge – knowledge for leadership of transformation."

W Edwards Deming

In this part, we will introduce the context where the concepts of system and system of systems, changeability and life cycle sustainability are relevant. A few illustrative challenges and applications of these concepts coming from the areas of manufacturing systems and robotics will be explored. As we go along, motivation case studies will be used to demonstrate the new fundamental notions and concepts defined in the changeability framework for discrete event systems.

9. Industrial application scenarios

This chapter presents the context for the application of the event driven changeability framework, defined in the previous section in industrial application scenarios. It will start by defining the context for those scenarios, provided by two European projects, and will conclude with the introduction of several possible demonstration scenarios.

9.1. Introduction

European industry is very active in all manufacturing fields, making Europe one of the strongest outfitters and operators of factories. One of the reasons for this success is the high quality of the produced components and production systems. “Made in Europe” is a worldwide synonym for high-quality and high-end machinery as well as for effective and reliable technology, thanks to the knowledge and innovation in European component suppliers and system integrators, which are continuously developing new components, integrating them into production systems, and customize them for the specific needs of their clients.

According to Eurostat, in 2010 there were 98.1 thousand enterprises operating with machinery and equipment manufacturing as their main activity in the EU-27. Together they employed 2.84 million persons and generated EUR 150.0 billion of value added, amounting to a huge number of equipment and systems every year.

Industrial processing machinery and production systems cover a wide range of products destined to specific purposes in downstream manufacturing sectors and, as such, demand for these products (components and production systems) is closely linked to new products or product renovation in the downstream manufacturing sectors which is very dependent on the general economic developments.

In downstream sectors, customization and make to order needs lead to smaller lot sizes with higher variability of products, and to reduced product life cycles. At the same time, globalization brings in cost pressure from emerging economies forcing European industry to think over the costs of both, their products as well as their investments in equipment, factory planning, ramp-up and operation, whist maintaining high responsiveness and quality standards.

Facing these challenges requires highly flexible, intelligent and self-adaptive production systems and equipment, which can react to continuously changing demand (versatility), can be rapidly and smoothly brought into operation (ramp-up), and can extend equipment life cycle (re-use).

Achieving these goals will contribute to economic and environmental sustainability of production systems, by increasing its effectiveness and reducing the need for additional and or/new production equipment. To make machine and component self-adaptive to avoid or reduce the time need for the setup of machine configuration and parameterization, these need to be endowed with knowledge about the processes and themselves. Only machines, which are capable to understand their process, can support multiple products, fast ramp-up and re-use in production networks, where these smart components co-operate by based on standardized information exchange.

Smart components are key enabling facilitators towards intelligent manufacturing in future factories. But to do so, components within the manufacturing system need to be equipped with features such as capabilities for fast exchange, rapid setup, plug&produce, condition monitoring, analysis and diagnostics, etc.

However, manufacturing systems and its subcomponents fulfilling those goals require high investments. Factory planning for such flexible production plants is time intensive and modular, standard devices are costly. OEM's are therefore seeking for new solutions for covering the investment costs for their manufacturing systems. In addition to that, an improved sustainability of equipment is required in order to enable green production by e.g. energy and material reduction, decreased CO2 emission.

Equipment and line builders have to respond to this need by delivering equipment addressing future requirements of adaptive and flexible manufacturing. Furthermore, business models are required which provide an answer to the price pressure of their customers whist at the same time strengthen the position of the suppliers. One of the most important competitive factors is the deep process knowledge of the European equipment manufacturers, which makes their machines more efficient and enables them to provide advice services to line builders and factory operators on how to make best use of their processes. This knowledge allows them to rapidly find appropriate methods for their machines, when they are integrated and used in new production systems.

Since the Factories of the Future (FoF) was launched in 2008 as one of the PPPs by the European Commission, the initiative successfully attracted committed industrial and public players who enthusiastically responded to the challenges of tomorrow's manufacturing industry by bringing forth about 130 projects targeted on smart and intelligent solutions. We will focus on two of these projects, I-RAMP^3 and ReBorn, which focus on the issues of versatile production, fast ramp-up and re-use of equipment.

9.2.I-RAMP^3 and ReBorn research projects

I-RAMP3 and ReBORN are two European research projects funded by FP7 under the Factories of the Future PPP.

The vision of I-RAMP3 is to enable zero ramp-up time integration of additional capabilities in existing and new production networks by task-driven "on the fly" cooperation of plug&produce devices. To do so, I-RAMP3 proposes the transformation of production equipment into Network-enabled Device Structures (NETDEVs), which form the plug&produce building blocks of a heterogeneous production network. NETDEVS allow the flexible creation of production networks, which operate by intra-device and global optimization mechanisms. Furthermore, production in Europe needs to faster achieve the point of providing dedicated, but also versatile capabilities as well as exceptional quality (Figure 57). Both, individual (existing) processes and process interaction along the value chain need to be strengthened. As main obstacle (for interoperability, flexibility and quality), the gaps of knowledge availability between distinct process steps are identified.

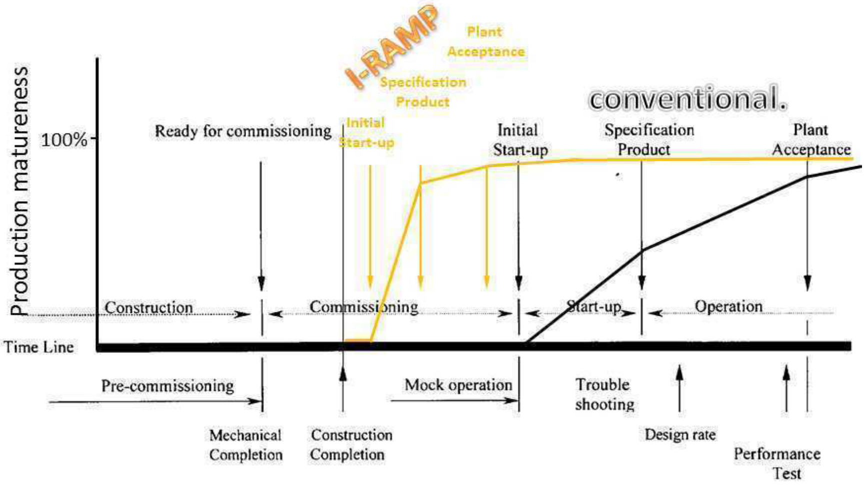


Figure 57: I-RAMP^3 addresses the need to faster commissioning and ramp-up

For that, intelligent and flexible production devices are needed which are able to interpret data coming from other processes in order to perform process optimization by the usage of proactive models. Sensors need to be faster deployed and provide complex data. Advanced data analysis and decision-making tools are needed in order to guarantee process optimization also for small batch series. Processes need to align with each other in order to find the best controls settings with respect to joint goals: products with zero defects.

Building on the same concepts, the vision of ReBorn is more encompassing and intends to demonstrate strategies and technologies that support a new paradigm for re-use of production equipment in old, renewed and new factories; maximizing the efficiency of this re-use and making the factory design process much easier and straight forward, shortening ramp-up times and increasing production efficiency and flexibility. This paradigm will give new life to decommissioned production systems and equipment, making it possible their "reborn" in new production lines. This new modular production equipment will be re-used between production systems but will require servicing and upgrading. In this scenario European machinery industry will move from an equipment-based business to a value added business, where equipment servicing and equipment knowledge are main business drivers.

The proposed paradigm builds on self-aware and knowledge-based equipment that need functionalities to collect and manage information regarding their capabilities (and their evolution over time); maintenance, upgrade or refurbishment operations over it lifetime; and information of use and ware over time.

This demonstration will be based on the implementation of versatile and modular, task-driven plug&produce devices, with built-in capabilities for self-assessment and optimal re-use, along with strategies for their re-use and models for factory layout design and adaptive configuration. ReBORN will contribute to demonstrate technologies for the realization of the knowledge-based and agile manufacturing enterprise of the future, with an innovative flexible and fast reconfigurable manufacturing solution based on the ideas of repair, upgrade and re-use of equipment, the (re-)design of factory layouts and flexible & adaptable production on shop floor.

Of special interest in this context is the re-configuration and upgrade of existing factory layouts and production lines as well as the modularization and re-use of equipment. The target of those approaches is to extend the lifetime of both, plants and factories as well as devices, machines and controls. Standardized and flexible interfaces need to be developed and established which allow for easy adaption of equipment to new requirements. Modular and extensible machines and devices for fast disassembly and (re-)assembly need to be addressed.

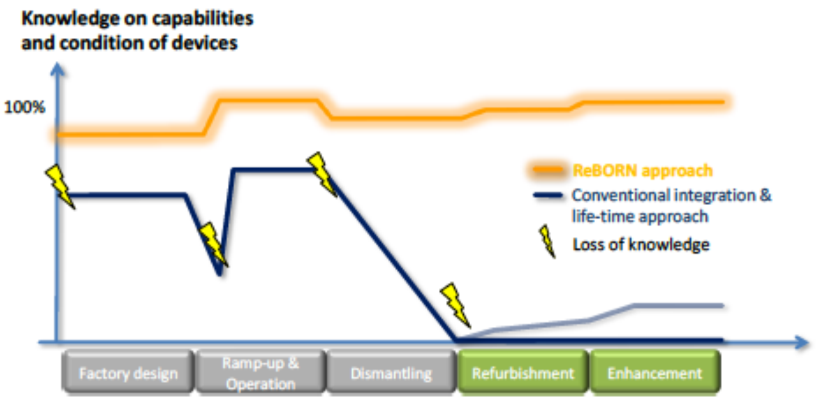


Figure 58: ReBORN enables for lifetime extension and for constant knowledge availability

An essential aspect of re-using manufacturing systems and its components is the knowledge on the respective conditions. Nowadays, re-use and re-tooling is often hampered by the fact, that information on wear, previous operating conditions, history of maintenance and service, exchanged sub-components, etc. is not available (Figure 58).

Planning of manufacturing systems with used components cannot be done reliably due to the unavailability of relevant knowledge on the components conditions, e.g. the remaining performance. For the same reason, production ramp-up is difficult and time-intensive. After dismantling, all knowledge on the components condition is lost. For that, the reliability, the remaining lifetime and the efforts and costs for subsequent component maintenance cannot be foreseen. For those reasons, people are hesitating to extend the lifetime of manufacturing plants beyond the planning horizon and avoid the re-use of machines and devices whenever possible.

To overcome those doubts and to establish real re-use and re-tooling, condition-relevant knowledge needs to be implemented in machines and components. Only machines, which are capable to monitor their own behaviour and the environmental influences on their conditions, can support their own re-use in a reliable way. Furthermore, a continuous condition monitoring and assessment needs to be implemented in order to react rapidly on upcoming system failures and to plan required component maintenance or substitution. Making the added value the extension of the lifetime of manufacturing systems and its components of European industries directly available in the equipment will create an extra economic basis for the equipment manufacturers.

The vision of ReBORN is to enable full economic sustainability of the production systems and innovative re-use of modular equipment. For that a Collaborative Communication Environment will be developed which accumulates knowledge for 360° life-cycle, broken into three main ideas of: Strategies for Repair, upgrade and re-use of equipment, the (Re-)Design of factory layouts and flexible & adaptable production on shop floor.

The basis for flexible and adaptable production are machines and devices with built-in intelligence for self- and condition monitoring which will also be applicable for existing hardware (upgrade). Methodologies for factory (re-)design will be developed and online intra-logistic and material handling optimization for most efficient production of even small lot sizes will be available. Finally, repair, upgrade and re-use of equipment covers lifecycle cost assessment and design models for refurbishment and enhancement of modular equipment for device re-use in old, renewed and new factories.

The project is envisaging solutions, which are suited for the needs of OEM's, system integrators and component suppliers with special attention on Small and Medium Sized Enterprises (SMEs), which is clearly depicted by the ReBORN consortium and its objectives.

The key element for enabling modular and flexile production as well as easy dismantling and re-use is equipment on shop floor which provides capabilities for self-description, condition monitoring, state assessment and refurbishment and enhancement planning. ReBORN will address these needs by the introduction of (VERSONs (Versatile, Flexible Lifecycle Extended Devices).

To make even better use of equipment even in the case of performance degradation the VERSON concept allows the "on the fly" creation of new, adapted and optimized capabilities whenever needed. A pool of current skills and information is formed by the total of equipment. Optimal structures are composed from this pool, which form an optimal network to fulfil a production task. An example is the re-allocation of a resistance spot welding gun with degraded peak current within a different task.

Relation between the ReBORN and I-RAMP^3

In this section the complementarity of the two projects regarding the motivation, approach and objectives will be demonstrated. I-RAMP^3 focuses on automating and shortening the ramp-up efforts in a production network, while ReBORN concentrates on the re-use, maintenance, refurbishment and life-long enhancement of equipment, structures and concepts. This imposes already almost completely different requirements, which have to be fulfilled by dedicated concepts respectively.

The aim of the I-RAMP$_3$ project is to decrease the time and efforts for the production ramp-up. This is done for three different ramp-up cases:

1. The initial ramp-up of a new production line;

2. The ramp-up of a production after component exchange or reorganization;

3. The ramp-up of a running production after planned or unplanned maintenance.

To do so, so-called NETDEVs are introduced. NETDEVs are logical entities, which are encapsulating a device, a complex sensor unit, or a group of components / sensors (e.g. robots and welding machine) into one logical unit. NETDEVs can be equipped with built-in intelligence by incorporating an extensible set of internal models on e.g. fast ramp-up, optimal process execution, and maintenance or quality assessment.

To reach these goals, a concept will be realized, which allows for the automatic adaption of production devices to the encountered environment according to a given task. This will be realized by encapsulating the equipment in a NETDEV shell, which allows it to understand a task and fulfill it in co-operation with the encountered production environment. The shell will contain process and optimization models, which allow the quick adaption to task and environmental changes.

I-RAMP^3 concentrates on the ramp-up cases mentioned above only. I-RAMP^3 does not consider the dismantling, refurbishment and re-use of equipment and other entities of the manufacturing system. Also, factory layout planning is not in focus of I-RAMP^3. Furthermore, use cases of the I-RAMP^{3}initial ramp-up phase are restricted to the use of new equipment. Initial ramp-up with old or renewed equipment is not addressed by I-RAMP^3. The core concept is the ability of new equipment to co-operate with other equipment with reduced human supervision in order to effectively reduce ramp-up efforts. This implies the creation of according communication schemes and co-operation models as reflected in the S/T objectives of I-RAMP^3.

The ReBORN goals are complementary to the goals of I-RAMP^3 by focusing on the life cycle extension and life-long enhancement of manufacturing components. ReBORN is closing the commissioning loop with new concepts for re-using of existing systems after use and/or dismantling by providing strategies for refurbishment and enhancement of entire manufacturing systems or single equipment (Figure 59). For that, also old or renewed equipment can be integrated by taking their condition and constraints into account.

This enables for a trusty reuse of old and renewed equipment and thus, improves significantly the re-use rate of equipment. Re-use is also foreseen for the factory planning processes. Existing factory designs can be used in a semi-automated (re-)planning process, e.g. in case of a change in the products. This will approach will reach significant reduction of planning time. In this context, the constraints of old or renewed equipment will be considered in the planning process. This also contributes to an improved re-use rate of equipment.

Figure 59: ReBORN approach for a closed loop of equipment lifetime

These complementary ReBORN goals are reflected by a different approach and concepts, which are realized in the ReBORN project. The key concept here is the implementation of the ability of production equipment to analyse its own state, to cope with eventual restrictions due to wear, to integrate new capabilities during life-cycle and to optimally deploy or refurbish used equipment. Such a concept is also foreseen for higher levels of production structures until the reuse of complete factories.

Differences and similarities between NETDEVs and VERSONs concepts

This section provides an overview of the I-RAMP[3] NETDEV and the ReBORN VERSON concepts. In the first sub-section, the common features of both concepts are illustrated. After that, a more detailed description of the features and capabilities are provided. This section ends with a summary providing a comprehensive comparison of both concepts.

NETDEVs and VERSONS are both logical entities, which are encapsulating a device, a complex sensor unit, or a group of components / sensors (e.g. robots and welding machine) with dedicated functional logic and with corresponding knowledge into one logical unit. The functional logic and the knowledge of those agents determine (based on the device core) what the entity can do. While NETDEVs allow for fast ramp-up of production systems by a flexible combination of different capabilities, VERSONs are optimized for easy re-use and lifetime extension. Both concepts can reside on the same equipment, each of them bringing different beneficial properties.

NETDEVs are logical entities, which are encapsulating a device, a complex sensor unit, or a group of components / sensors (e.g. robots and welding machine) into one logical unit. NETDEVs can be equipped with built-in intelligence by incorporating an extensible set of internal models on e.g. fast ramp-up, optimal process execution, maintenance, quality assessment.

To reach these goals, a concept will be realized, which allows for the automatic adaption of production devices to the encountered environment according to a given task. This will be realized by encapsulating the equipment in a so-called NETDEV shell, which allows it to understand a task and fulfil it in co-operation with the encountered production environment.

The shell will contain process and optimization models, which allow the quick adaption to task and environmental changes.

To make even better use of equipment with respect to quality and flexibility, the concept breaks the barriers between machines and processes and allows the "on the fly" creation of new, adapted and optimized capabilities whenever needed. A pool of skills and information is formed by the total sum of sensors, actuators, data and knowledge contained in the machines. Optimal structures are composed from this pool, which form an optimal network to fulfill a production task. NETDEVs do change their role, but remain in the same range of capabilities for their whole lifetime.

The orthogonal ReBORN goals are reflected by a different approach and complementary concepts, which are realized in the ReBORN project. The key concept here is the implementation of the ability of production equipment to analyze its own state, to cope with eventual restrictions due to wear, to integrate new capabilities during life-cycle and to optimally deploy or refurbish used equipment. Such a concept is also foreseen for higher levels of production structures up to the reuse of complete factories. The key element for enabling modular and flexile production as well as easy dismantling and re-use is equipment on the shop floor which provides capabilities for self-description, condition monitoring, self-state assessment, refurbishment and enhancement planning.

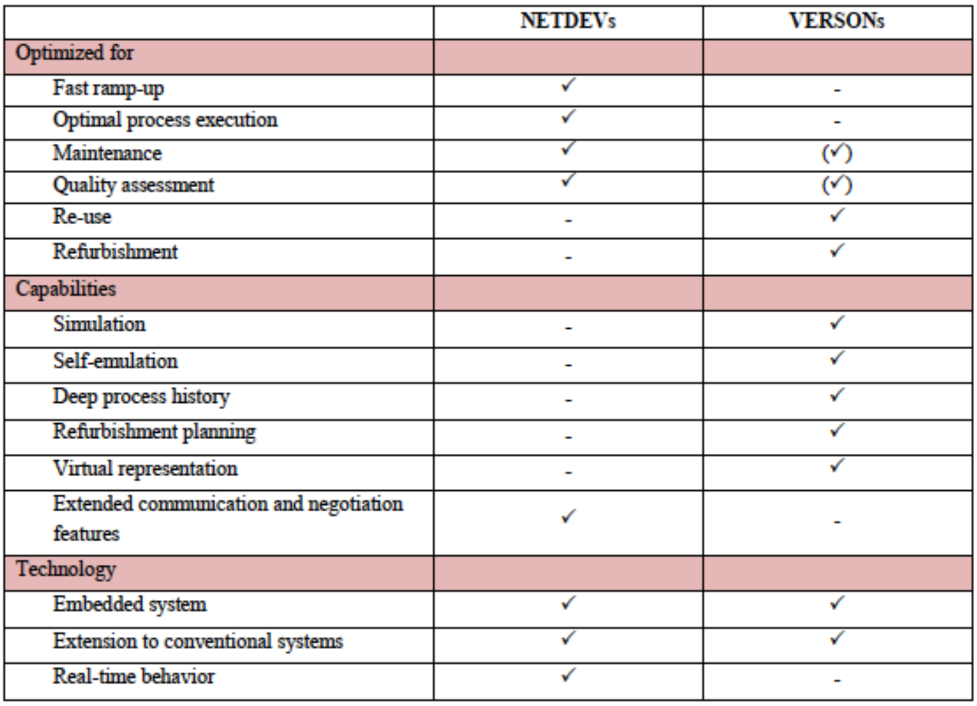
ReBORN will address these needs by the introduction of versatile, flexible and lifecycle extended Devices (VERSONs). VERSONs are agents, which can have a physical or virtual representation of production equipment. The virtual representation is mainly used for simulation purposes. In the physical representation, a VERSON wraps existing equipment and turns it into modular devices, which are always aware of their own state of capabilities. Based on their state information they will also be easily refurbished or turned into new devices with enhanced or new capabilities for re-use even for different production tasks, extending their life over several production life cycles.

These VERSONs shall have analytical capabilities to determine their own state, to find the best practice operational parameters; intelligence to derive lifetime prognosis, maintenance requirements, refurbishment plans; state-dependent cost model estimation related to task execution and maintenance; capabilities to describe and optimize themselves towards their environment by providing knowledge and models about their properties, abilities, constraints and reuse abilities.

In addition to that, VERSONs can acquire different new capabilities by incorporating new knowledge and models (eventually in combination with refurbishment or enhancement of the equipment).

Table X provides a comprehensive comparison of the NETDEVs and VERSONs in terms of their optimization goal, their capabilities and the technology used.

Table X: NETDEV Vs. VERSON



	NETDEVs	VERSONs
Optimized for		
Fast ramp-up	✓	-
Optimal process execution	✓	-
Maintenance	✓	(✓)
Quality assessment	✓	(✓)
Re-use	-	✓
Refurbishment	-	✓
Capabilities		
Simulation	-	✓
Self-emulation	-	✓
Deep process history	-	✓
Refurbishment planning	-	✓
Virtual representation	-	✓
Extended communication and negotiation features	✓	-
Technology		
Embedded system	✓	✓
Extension to conventional systems	✓	✓
Real-time behavior	✓	-

9.3.Challenges and S/T objectives

This section describes some of the scientific and technical objectives of the I-RAMP[3] and ReBORN projects related with life cycle sustainability of System of Systems.

Plug&Produce devices with built-in intelligence (NETDEVs agents) for fast exchange of components

Target of this S/T objective is to establish agent based plug&produce devices for smart factories, which can be exchanged and adapt with at least 50% less configuration and customization effort.

To do so, these NETDEVs shall describe and optimize themselves towards their environment by providing knowledge and models about their properties, abilities, constraints and re-use abilities (device self-description). Furthermore, they shall have the ability to (1) perform condition monitoring and maintain a device history, (2) interpret and execute tasks (process model), (3) optimize process and expose abilities (optimization model) and to (4) predict its maintenance requirements (maintenance model). The NETDEV concept will therefore allow component and system integrators suppliers to built-in expertise into their devices.

Relevance: This target will contribute to scalable extension of production networks and to the reconfiguration of system functionality by an agent-oriented approach, whenever components are brought into. Through their self-description and built-in models, they support the discovery and retrieval of abilities throughout the production network. Furthermore, condition monitoring and self-assessment support the re-use and maintenance of manufacturing equipment.

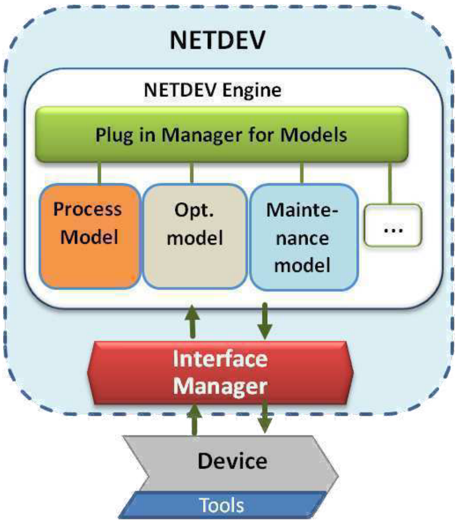


Figure 60: Building blocks of the NETDEVs agents

Versatile and modular plug&produce equipment (VERSONs) with built-in intelligence for flexible production, self-state monitoring and optimal re-use

Target of this S/T objective is to establish modular, agent-based, task-driven plug&produce devices for smart factories, which can be exchanged and adapted for new production goals and for new production structures. The devices are always aware of their own state of capabilities, which they offer to the production network. Based on their will also easily refurbished or turned into new devices with enhanced or new capabilities for re-use even for different production tasks, extending their life over several production life cycles. This versatility and the task-driven process execution of the devices, which we call VERSONs, will guarantee the reusability in new life cycles and allow 50% less configuration and customization effort.

Furthermore, they shall have the ability to (1) perform condition monitoring and maintain a device history, (2) interpret and execute tasks (process model), (3) optimize process and expose abilities (optimization model) and to (4) predict its maintenance requirements (maintenance model).

The VERSON concept will therefore allow component and system integrators suppliers to build-in their expertise on different levels expertise into their devices: flexible, task-driven process execution methods, process optimization, best practice, self-state estimation, maintenance requirements and efforts, refurbishment and enhancement measures and efforts. It will allow planners and line-builders to make maximum use and benefit of equipment within and across production life cycles.

Relevance: This target will contribute to scalable extension of production networks and to the reconfiguration of system functionality by an agent-oriented approach, whenever components are brought into. Through their self-description and built-in models, they support the discovery and retrieval of abilities throughout the production network. Furthermore, condition monitoring and self assessment support the re-use and maintenance of manufacturing equipment.

Intra-device and global optimization models for automated device configuration

Target of this S/T objective is develop optimization models, which are applied inside NETDEVs for local optimization as well as global optimization models, which can be applied across the entire process-chain. This will lead to a fully automated device configuration considering the entire process chain.

The intra-device optimization approach follows either a general cost minimization objective or can follows specific cost function based on a cost function document, which shall be provided via the communication framework. In the latter case NETDEVs need to control the process in a way to minimize the total cost, which is the sum of process and result cost.

A global optimization engine “Workflow Optimizer” will manage this process wide optimization. The optimization shall be commenced with the latest NETDEV in the workflow. This device will perform an optimization and then provide input to the predecessor until the first process is reached.

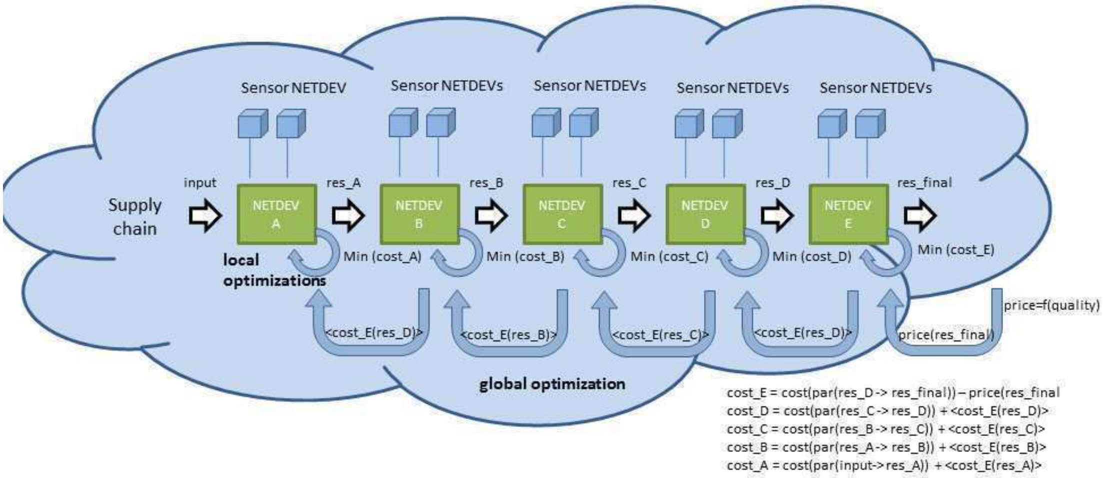


Figure 61: Intra-device and global optimization in process chains

Hereby, the S/T objective will achieve that devices can be automatically optimized based on built-in intelligence (intra-device optimization models) as well as towards process-chain wide optimization criteria. This will strongly reduce the expertise and time required for system configuration and customization by approx. 50%, and will lead to higher process efficiency in versatile production systems.

Relevance: This target will contribute to the development of configuration modules for single devices and the global process chain. The I-RAMP[3] optimization approach will strongly rely on built-in self-configuration skills of NETDEVS and thereby reducing the complexity for the system integrators and end users.

Strategies for re-use of production equipment in existing production systems

ReBORN targets to enable easy and quick integration of new and legacy equipment components into new and existing manufacturing systems through realizing the vision for Plug&Produce systems in both future and existing manufacturing environments. The aim of this S&T objective is to advise methods and strategies that enable the reuse and refurbishment of existing production equipment and devices into new and existing production systems based on a set of criteria and assessment methods.

Relevance: This target will contribute to the expected lifetime extension of modular equipment by providing methods for lifetime measurement and assessment. Furthermore, the reusability and adaptability of existing manufacturing systems will be enhanced by the development of strategies for the introduction of new products and product variant into existing production systems. Bases on the strategies created, vendors, system integrators and OEM's are enabled to provide innovative business models for their products and services.

Models for innovative factory lay-out design techniques and adaptive reconfiguration

Target of this S/T objective is to propose models for the design and adaptive reconfiguration of factory layouts, based on knowledge about production equipment properties, abilities, constraints and re-use abilities (device self-description) and distributed simulation and optimization tools. This will allow to decrease the ramp-up time of assembly lines by at least 50%, and to respond to rapidly changing consumer needs while saving costs.

These models take the whole production process into consideration. Moreover, production equipment is not only communicating with each other but are members of a co-ordinated team of specialized autonomous objects in learning networks (environment of intelligent collaboration) and are able to constantly self-describe their capabilities and state.

This distributed knowledge is used at the planning level, not only to support the fab planner in the design of new factory layouts, but also in adapting existing layouts to new conditions and/or new knowledge. Having constantly updated knowledge on the production resources capabilities and state creates the possibility to select in each situation the best candidates to integrate a certain factory layout or, for existing layouts, identify the best candidates to replace equipment already included in the layout but underperforming at the moment. These functionalities can be extended in order to cover not only the support of activities in the design phase, but also activities during ramp-up and even production, allowing for an semiautomatic adaptive reconfiguration of the factory layout.

Relevance: This objective will contribute to the demonstration of technologies for the realization of the knowledge-based and agile manufacturing enterprise of the future (MANUFUTURE 2020), with an innovative flexible and fast reconfigurable manufacturing solution based on the idea of autonomous/self-acting intelligent production units where on-demand knowledge-based production can be realized, and innovative tools and techniques for factory layout design and adaptive reconfiguration.

Design methodology for de-manufacturing, dismantling, recycling and value chain extension incorporating prior expert knowledge and experience

The target of this S/T objective is to develop a methodology for the design of manufacturing systems that integrates de-manufacturing, dismantling, recycling and value-chain extension processes into the classical design methods. This methodology is primarily based around the notion of virtual/physical systems which has been explored and developed in IDEAS and XPRESS projects. In this representation the manufacturing system is thought to be a system-of-systems, or a collection of smart devices that exhibit intelligent capabilities on the component level. This component-level intelligence allows the components to have a parallel virtual representation that could be used for modelling and prediction purposes. The design methodology will also be based on the distributed collaborative working and life-cycle knowledge about the production equipment and their components developed in TRANSPARENCY project. It shall be realized as "structured knowledge", which can be adapted easily also by non-IT-personnel, and shall have numerous interlinks with Structure of life-cycle knowledge. The methodology shall be requirement-driven and include the ability to execute virtual test cases.

Relevance: This target will realize the need for sustainable manufacturing systems through facilitating Re-use of existing factory layouts while enabling the adaptation to new arising requirements in response to new production needs (e.g. product variations and volume variations) or in response to performance degradation or upgrade opportunities throughout the system's life-cycle. The knowledge about the various performance indicators of the manufacturing equipment will be continuously captured and formalised in order to enable its reuse throughout the different phases of the system's life-cycle to help make informed decisions with regard to the usage of used and renovated equipment. This will be based on the knowledge capture framework developed in TRANSPARENCY project; however the framework will need to be extended beyond the machine tools sector, which was TRANSPARENCY's focus into manufacturing systems in general across various sectors.

Enabling commercial Manufacturing Execution Systems to optimize workflow during ramp-up

Target of this S/T objective is to enable the optimization of workflow during the ramp-up of new production systems and in case of fast changing production systems. This shall not be done by replacing existing Manufacturing Execution Systems and SPC solutions, which are performing well in stable production settings. Instead I-RAMP^3 will complement the capabilities of MES for ramp-up and re-configuration phases, when no sufficient production data is available to perform conventional workflow optimization.

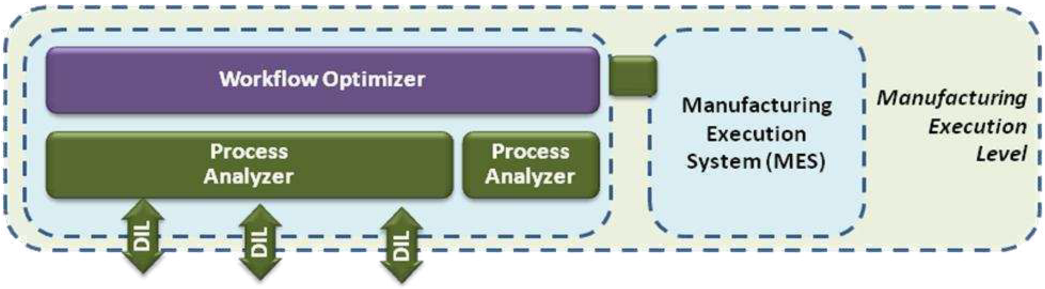


Figure 62: **Workflow optimizer and Process Analyzer to enhance MES**

This S/T objective shall be achieved by introducing Process Analysers and Workflow Optimizers between the NETDEVs at the MES layer. I-RAMP^3 will supply the Process Analysers with the capabilities to interpret the result description documents of the NETDEVs and to analyse them according to rules, which classify the results. The classification results will be forwarded to the Workflow Optimizer, where they are assessed by a rule-based knowledge system. The latter decides on re-configuration, modification of workflow and maintenance. The optimized workflow configuration is forwarded to the MES.

Hereby, the workflow can be optimized already in the early phase of the ramp-up as well as during the operation phase of highly volatile production systems. This will reduce the time to full production output by approx. 30%.

Relevance: This target will contribute to Workflow Optimizer as knowledge driven systems, which will increase the fault tolerance and self-configuration skills of systems in fast changing environments. This will allow interconnecting mitigation of manufacturing systems to modern architectures.

9.4. Smart Factory

In existing production the “smart factory” is still far from reality. Commissioning is a mainly “manual” process, where machine parameters have to be found, sensors have to be calibrated and communication between devices has to be established. Software tools and simulation exist to support this process. This continues after commissioning, when re-adjustment and reconfiguration measures have to be taken to make the whole production run smoothly and efficiently. The same holds true, when a production needs modification or a production device has to be replaced. On the other hand side, agent-based production concepts have been introduced in research, especially in the form of so-called “holonic manufacturing”, which constitute in principle a solution for smart factories. These concepts are on their way from abstract academic structures towards real production over a couple of research projects focused on flexible, re-configurable and adaptive production at different levels and with different focus. To realize highly flexible production systems, various concepts (Figure 63) have been developed.

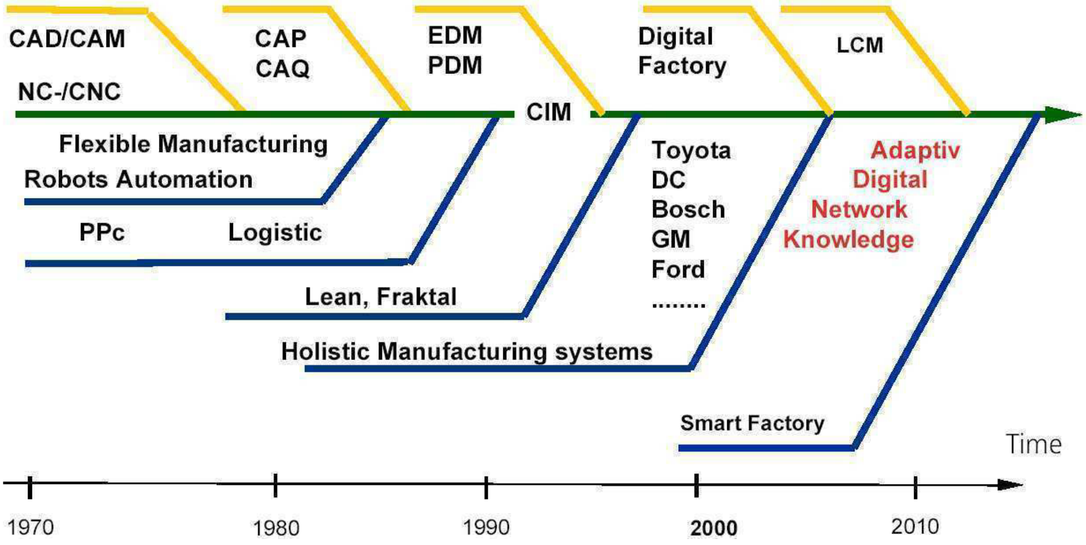


Figure 63: Timeline of manufacturing concepts (source Manufuture Workshop 2004)

After the classical Computer Integrated Manufacturing (CIM), since the early 1990s, scientific approaches of self-organisation have been investigated which are mostly based on multi-agent systems (MAS). The “Holonic manufacturing systems” (HMS) 3 are commonly seen as a further development of MAS. The FP6 IP project XPRESS addresses structural and technological questions in the latest conceptual framework, the “Smart Factory”, which fills and extends the HMS framework. There have been a couple of large projects to set up the framework of HMS, which are discussed shortly now.

In the Next Generation Manufacturing Systems (NGMS) project models to merge a bottom-up view of manufacturing flow with a top-down view of the globally distributed virtual enterprises were developed in order to create a global network of self-organizing, autonomous units, global network of self-organizing companies or supply chains. Most relevant to XPRESS was the effort to set up a Scalable Flexible Manufacturing (SFM) architecture, a framework for organizing resources of hardware (machine tools, robots, ...) and software (cell controllers, process planning, ...) in computer automated environments with an emphasis on autonomous de-centralized scheduling. In this approach, each unit in a factory is autonomous and manufacturing execution is the result of negotiations between the autonomous modules with a central "blackboard" containing order information and planning status information. Each resource makes a bid for the work and the best bid wins, leading to an autonomous distributed control. The resulting flexibility unfortunately has a high price: The resources must be all-rounders to be flexible enough and have overall production knowledge for a qualified bid. The dynamical system behaviour is no more predictable and may become unstable.

The GNOSIS project concentrated on configuration systems for design and manufacturing. One part of GNOSIS dealt with "soft products" and knowledge intensive engineering. In relation to XPRESS, the PROConfig Process Configuration Framework was created. It consists of a plan skeleton editor with graph-based description model and generic "plan skeletons" for designing multiple-variant production processes. Furthermore a virtual factory was proposed, which provides reactivity and efficiency by the optimal use of distributed manufacturing resources. These resources are connected to form virtual manufacturing processes which can be configured and operated as work cells based on product, process or production line principles according to changing demands from the market. The core idea is communicable models which provide both planning and co-ordination throughout the virtual factories. These GNOSIS concepts have been partly adopted by commercially available planning software. In the PLANARIA sub-project of GLOBEMAN21 autonomous working cells (ARC) driven by CAD data, flexible transfer systems were proposed. The projects discussed so far created organizational frameworks for flexible manufacturing on the organizational level. No major attention was given so far on structuring the autonomy and responsibility of the manufacturing units at all levels with respect to knowledge and expertise. The PABADIS project and the XPRESS project already mark the transition to the "smart factory", why they are discussed in more detail. In the organization structure of a company, three different levels can be distinguished: bureau level, factory level and field or shop-floor level (see figure 9). Respectively the HMS concept presented 3 types of holons: order, product and resource holons. The transition from traditional solutions to MAS and further on to the XPRESS concept is shown in Figure 64.

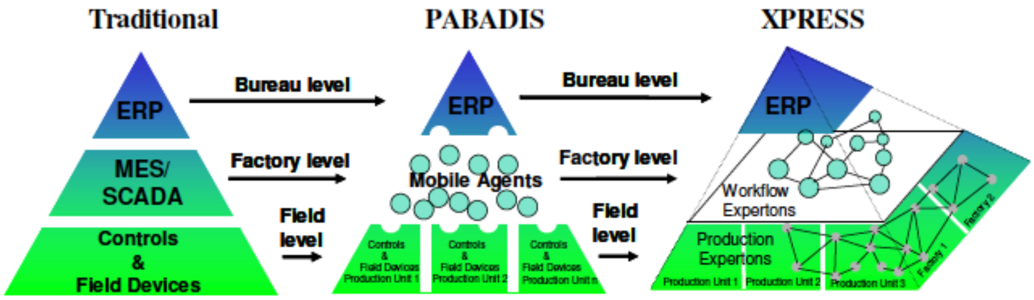


Figure 64: From rigid to reconfigurable systems

PABADIS demonstrated the advantages of mobile agents compared to classical Manufacturing Execution System (MES) and Supervisory Control and Acquisition (SCADA). Concerning the field level, only fundamental concepts were postulated. However, a flexible production is only possible if the integration of production units at the lowest level (machines, sensors...) are taken into account. This was resolved by the XPRESS project. In other projects like SIARAS and EUPASS the encapsulation of process knowledge in agent-based production equipment was the core property to make the equipment more versatile, adaptive and combinable. XPRESS incorporated this approach and extended it into a task-based production, where process equipment has expertise about a certain process domain and can execute any task of its domain based on the description of the task and can produce a quality result document. XPRESS developed a framework to wrap existing equipment with a so-called "manufactronic" shell, containing the required process intelligence and communication means. This manufactronically wrapped equipment is called a Manufactron and communicates with other Manufactrons via the exchange of so-called "Task Description Documents" and "Quality Result Documents". Manufactrons can form hierarchies to fulfil higher level tasks. This framework has made holonic production available for real process devices and let it produce benefit also in non-holonic environments. It has been taken up prominently in Airbus and Fiat production and in other smaller companies because of its reduced commissioning effort and its adaptability. This was the reason why the concept was proposed as a success story of the European 6th framework program and why it is used as the starting point of IRAMP[3]. However, there are three main drawbacks in the XPRESS results, which prevent immediate plug&produce capabilities of the XPRESS Manufactrons:

- Co-operating Manufactrons have to use exactly the same defined exchange documents in order to understand each other.

- Manufactrons cannot connect themselves, but must be connected by higher-level objects, defining the exchange.
- Manufactrons cannot optimize their process with respect to an overall optimum.

I-RAMP^3 and ReBORN intend to extend the XPRESS concept with properties described in the following sections.

Plug&Produce device with build-in intelligence for fast exchange of components and Plug/Produce Communication Framework for heterogeneous devices

Today competitive manufacturing domain is characterized by an increasing need for a high degree of automation on shop floor level, an increasing diversification following the trend to mass customization and increasing product requirements with respect to customer specific variants, small and medium lot sizes and shorter product life cycles, shorter manufacturing cycle times and higher throughput. Furthermore, the expectation of the customers for product quality is rapidly increasing towards an accepted failure rate of zero ppm. At present, mainly hierarchically oriented Manufacturing Execution Systems (MES) are trying to cope with these requirements. Today's modern shop floor IT systems are based increasingly on innovative communication architectures like service oriented architecture (SOA) that takes flexibility and maintainability into account. Current research approaches are going beyond this point and are e.g. focusing on a Software-as-a-Service (including a Cloud Computing approaches). Never the less these systems have still a tremendous effort for configuration and implementation. Semantic technologies for integration purposes are raising interest, as well.

A decentralized concept is intelligent manufacturing systems (IMS), where the components have to be capable of simultaneously addressing both knowledge processing about manufacturing capabilities and material requirements9. Multi-Agent Systems (MAS) are widely used to model IMS. Despite the efforts done in MAS research there is still the challenge to apply those approaches into real-world manufacturing environments.

Innovative aspects and progress beyond state of the art I-RAMP^3 will combine the widespread industrial best-practice (MES) and the more theoretical IMS approach and make use of the benefits of both systems.

I-RAMP^3 will implement autonomous distributed IMS system based on a holonic structure on top of existing machine architecture. The project is not aiming at substituting existing and accepted Commercial of- the-Shelf applications like MES, Statistical Process

Control and others but expanding their knowledge range to additional dimensions and therefore allowing a novel way of Plug&produce production. That new highly scalable paradigm of cross-process knowledge exchange of manufacturing Information will enable ground breaking changes in manufacturing behaviour.

Systematic approach: The I-RAMP[3] approach will be able to be implemented to existing machines ("Process n-1" and "n") and taking advantage from existing factory infrastructures ("enterprise service bus"). They will be converted into intelligent NETDEVs as explained in earlier chapters.

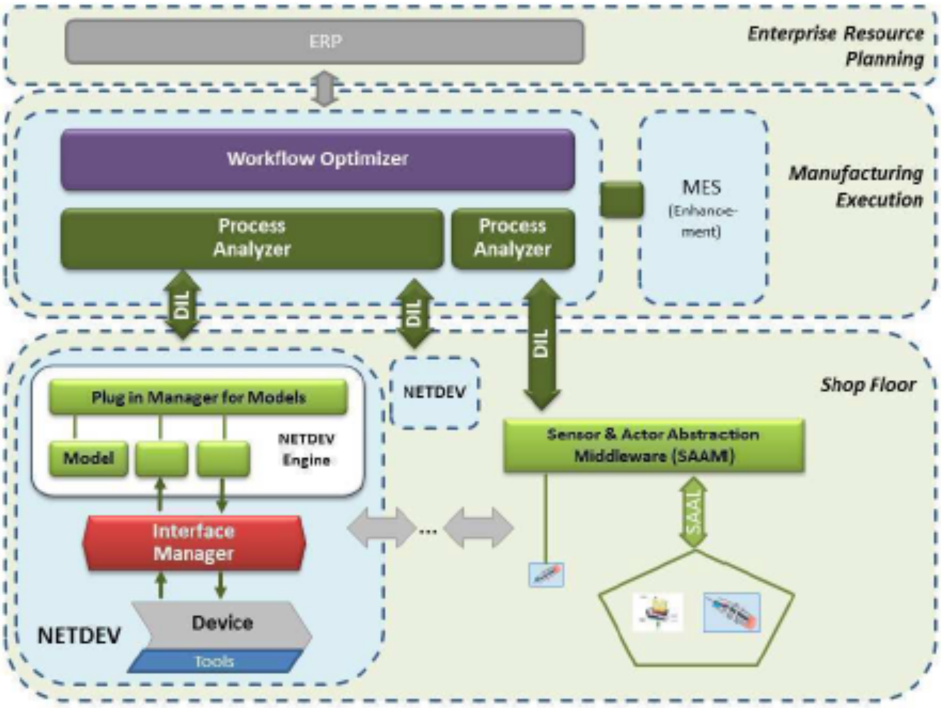


Figure 65: NETDEVs and inter-process communication

The concept foresees a two tier approach: The NETDEV shell will have two core components: The "Interface Manager" will care for providing data, knowledge and communication (technical focus; server), while the "NETDEV Engine" will use the models to control the activities of the NETDEV such as process execution based on task information or local and process chain optimizations for seamless plug&produce and (ideally) zero commissioning and ramp-up. The radical breakthrough will be the model-driven communication between the two tiers.

The "Interface Manager" will be adopted for the different machines. The I-RAMPÑ system will provide different generic templates, which will fasten the implementation. The Interface Manager will realize a number of connections towards the new "grid of sensors", towards the machine's PLC (which allows for acquiring further process data), to a possible machine host PC (red) and to a so called "interceptor", which interfaces the communication of the machine with the existing systems in the factory backbone10. The connections will be proprietary and of different formats (serial, OPC, ADS, SECS/GEM...). Integration of knowledge systems and HMI's: The NETDEV engine will be also the point to store knowledge about the implemented process, its dependencies and capabilities. For this, all NETDEV engine will share the same HMI (human machine interface), which will build upon semantic relationships (ontologies). With this HMI and the underlying I-RAMP^3 meta-model, a consistent representation of the machine structure (including sensor/ actuator definitions), the incorporated process (important parameters, key performance indicators (KPI), quality measures), its capabilities and requirements as also important product features and dependencies will be captured. The knowledge will be directly accessible on the shop floor level for machine and process optimization, diagnosis and maintenance. Further on, the knowledge will be integrated towards a coherent model of the whole production line/ shop floor. Concerning process interlinking, the project will take up research results from the European project XPRESS, which successfully implemented capability descriptions and a model of pre- and post-conditions.

Data collection and processing: Consequently, the NETDEV engine will also care for the storage and aggregation of sensor data acquired from "grid of sensors" and the machine's PLC and process signals acquired from the machine's PLC and host adapter. It will therefore use according definitions within the model. The data pre-processing will cover super/subsampling of multi-resolution signals, offset/ scale/ drift compensations and limit checking14.

The aggregation will work on time-based and triggered intervals and will also provide (auto-) correlation. The entities will buffer these data to storage containers, allowing the "NETDEV Engines" to retrieve the data in good time. Trigger signals will be communicated to interlinked managers.

Extensive integration capability: The "glue" between the Interface and all other I-RAMP^3 components will be the model-driven communication interface. This can be seen as an "IRAMP^3 language", which uses a well-defined syntax to represent a variety of complex data and signals, knowledge models, queries to predictive models and their responses, commands to external entities and further. The semantics of that language will lie in always relating to facts of the different process models and to concepts of the I-RAMP^3 meta-model. By this, the communication language and the model representation of an overarching process chain will become integral parts of the I-RAMP^3 concept. The transportation layer will be either by utilizing the underlying enterprise service bus or by using state-of-the-art service oriented communication.

Distributed approach: Using the communication language to seamlessly acquire data and knowledge from one or more Interface Managers, the "NETDEV Engines" can fully dedicate themselves to the task of plug&produce production and optimization. As proprietary interfacing, persistency, knowledge gathering, data pre-processing and aggregation is already done by the Interface Manager, the NETDEV Engines can focus on the pure control and decisions functionality. For optimization, both concepts of "feed-forward" (quality and corrective information is given to the next process steps) and "feed-back" (quality and corrective information is used to optimize the own process step) will be possible. By decoupling from any proprietary interfacing the NETDEV Engines can be quickly migrated to other process steps, if similar statistical or analytical functionality is required. This will protect investments in developing these models.

Intra-device and global optimization models for automated device configuration

In advanced factories data and knowledge become more and more important using systems for advanced process control to govern complex process sequences executed in production networks. The knowledge is represented by models and used to interpret data and find processing strategies. The most established method in practice is Statistical Process Control (SPC), which is mainly applied to monitor and supervise the measurements and findings. SPC is mainly used as an interactive tool for experts and might automatically trigger alarms.

Advanced Process Control (APC) is an approach in which measurement data is automatically used to adapt process parameters using a process model. APC is starting to penetrate production in chemical and semiconductor industries. Other industries are eager to follow that approach but need to take their specific manufacturing environment into account. Data mining has been successfully employed in the semiconductor industries due to their extremely data rich production environments. In APC we also find first approaches to overcome process boundaries by providing production data to other process steps.

Many modelling technologies have already been investigated for industrial use in production environments. These models include semantic technologies, numerical simulations, neural networks and process data mining.

Semantic technologies and functional representations are used to capture prior process and machine models on a logical level. They are designed to automatically answer process-related questions by combining data and their inherent relations into a semantic model. Numerical simulation is adjusting the distributions of the physical parameters to fit experimental findings. Methods such as FEM (Finite Element Method) and FDM (Finite Difference Method) are widely used. Due to the numerical complexity, they can only be used off-line for designing production processes. For on-line purposes, the process knowledge essential for control is concentrated in cognitive models such as non-linear support vector regression models or neural networks. Inputs for such networks can be derived from both simulation and production data. They are used to relate process signals to process state information and quality or to relate task parameters to process parameters. In cases where no prior knowledge is available, models are built on process data mining applying advanced statistical methods such as PCA (Principal Component Analysis) or PLS (Partial Least Squares regression) in linear or Kernel versions. Further approaches are addressing the systematic modelling of the interactions of the individual processes, which reflect the so far undefined side effects of a process on the material properties in relation to their effect on subsequent processes. Some work is dedicated to develop models to study the data of entire factories to discover problem areas instantly affecting any subsequent processes. The models are built via statistical data mining without taking prior knowledge into account.

I-RAMP^3 proposes supervisory controls named “NETDEV Engines” as one core component of the NETDEV shell (see Fig. 15), which are allowing the improvement of local process controls, based on a set of customizable models. At the same time, these NETDEV Engines shall align themselves with adjacent NETDEV Engines, to achieve a distributed but overall optimization of the production along the process chain.

The I-RAMP^3 approach will be able to hold a functional decomposition of different models by the means of a plugin architecture. This will allow combining a range of standard models in order to achieve a process-specific optimization of production control. The used models are encapsulated into the NETDEV Engine by the Plug-In Manager for Models. Virtually all data for the NETDEV Engine is provided by the Interface Manager thus the NETDEV Engine does not need to take care about data acquisition.

The set of models (process model, sensor model(s) and quality model) are serving for the translation of external information into the internal process state space and vice versa. This allows easy integration of sensor information from the process and the Grid of Sensors and allows also predicting quality outcome (quality model), based on multivariate and stochastic approaches. Additionally, this translation allows aligning with other NETDEV Engines along the process chain.

The parameter evolution and cost function will be used to drive the process towards the final process goal and will also care for the real-time adaptation of the NETDEV Engine in case of external disturbances and deviations of the process. The NETDEV Engine will integrate input from existing statistical process control (SPC) and Run-2-Run optimizations. The translation capabilities of the decomposition of models will also allow pre-process prognosis before executing the process: this will be done by investigating a set of hypothetical final process states and evaluating them by using the cost function and quality model. The visualization will use the same means to allow for process monitoring and diagnosis by human operators, displaying not only internal process state variables but translating them for the user into physical values, quality measures and relations to other processes.

Finally, the downstream optimization will take care of aligning the process goal with other NETDEV Engines. Before executing the process, a downstream optimization model is sent (via the model-driven communication) towards the subsequent NETDEV Engine including a projection of the own cost function. The downstream NETDEV Engine will use pre-process prognosis and its own cost function to align about the joint optimum of both processes. This selection will be communicated back and process execution will start.

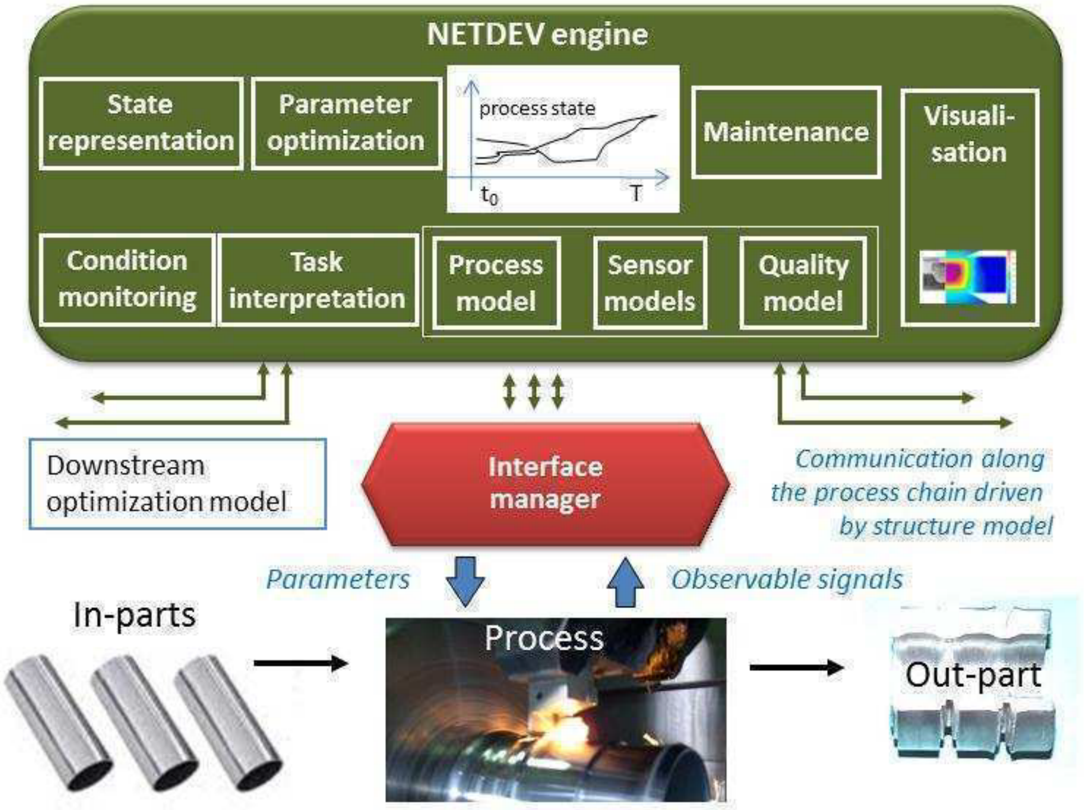


Figure 66: Proposed I-RAMP^3 NETDEV engine optimizing control approach

This architecture approach can be illustrated with an example: A machining process ("A") produces a defined geometry but an uneven stress distribution in the work piece. A subsequent heat treatment ("B") may then cause a geometry deformation to the work piece induced by the uneven stresses from machining. With the knowledge about the stress distribution ("downstream optimization model") an inhomogeneous heat treatment (executed by "B") could avoid deformation. Alternatively the machining process could produce a geometry, which reaches the final measures after heat treatment.

Enabling commercial MES to optimize workflow during ramp-up

Manufacturing Execution Systems (MES) and are applied in most industrial production systems and are often extended with Statistical Process Control (SPC) modules. These systems allow process optimization during the operation phase of production systems. Meier20 and Bergholz21 state that these IT systems play an important role in shortening the ramp-up of production systems, improving the process quality, improving the resource efficiency as well as the factory throughput. Furthermore, they lead to a higher flexibility of the production systems on changing requirements. According to Bergholz, the ramp-up of production systems can be improved, when IT systems are earlier available and can be used for error analysis in the entire system.

While existing systems allow analysing process chains and optimizing workflows for stable production settings, these systems cannot be applied during the commissioning and early ramp-up phase of production systems. The reason is, that the analysis of process data – and thereby the identification of sub-optimal configuration across the process chain - can only be conducted with sufficient production data at hand. Therefore, erroneous settings are recognized too late in the ramp-up process and workflow optimization aspects are analysed to late.

To allow the more effective use of existing MES and SPC systems in the commissioning and ramp-up phase and also to ensure their usability in frequently changing production environments (as envisaged by I-RAMPÑ), MES systems need to be enhanced with rich process data analysis capabilities and workflow optimization capabilities, which can be applied during the ramp-up phase. This means that:

- Workflow optimization shall be optimized based on limited gathered production data (during the early ramp-up phase).
- The rich process date from the Plug&Produce devices and the Plug&Produce sensors shall as well as the inherent logic inside these logical units shall be taken into account by the MES system.

I-RAMP[3] will overcome the limitations of the current state of the art by the implementation of a rule-based expert system for workflows (Workflow Optimizer), which is interfacing existing commercial MES and is building upon the Process Analyser entities.

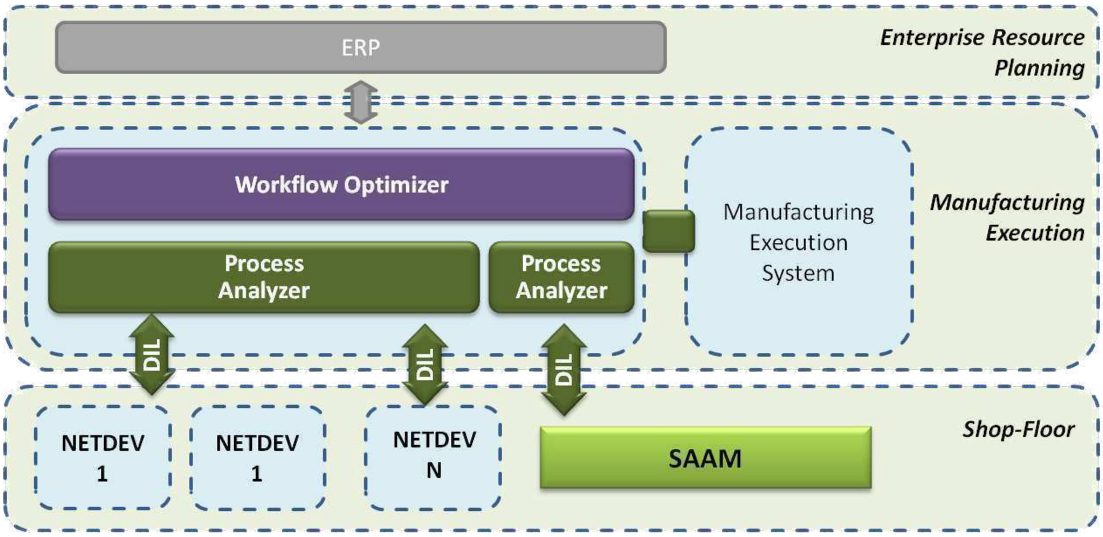


Figure 67: Three level-layered architecture

Since especially the I-RAMP[3] Plug&Produce environment operate in a highly data rich environment it is near at hand that the tasks of MES shall also use this rich data capabilities, which is included by the component suppliers in the NETDEV itself. The core elements to achieve this objective will be the Workflow Optimizer, which is responsible (1) interpreting the information coming from the Process Analysers and (2) exchanging tasks and parameter values with the MES and (3) optimizing the workflow using both.

The execution of the workflow optimization can be based on an orchestration vision (BPEL), choreographic vision (Rules), and use empirical knowledge about "successful" process steps stored in the factories repository - or a combination of all approaches23. In the first approach the engine launches every process following a deterministic flow of control. In a choreographic approach there is not a predefined flow of control but there are a set of rules that govern the selection of an optimal production strategy according to given constraints (quality, throughput, costs, availability of resources…). The Workflow Optimizer is responsible to evaluate the rules and launches the processes associated. The workflow optimization itself will be based on artificial intelligence techniques like decision trees, case based reasoning, neural networks, machine learning, modification of action planning techniques etc. to perform its task. The reaction on irregularities or sub-optimal performance of the processes is also performed by the workflow optimizer and triggered by process analysers.

Results of the Workflow Optimizer might be introduction of new devices/NETDEVs in production systems, re-organization of workflow (e.g. shift of task from one NETDEV to another) due to bottlenecks and optimization of workflow due to cost optimization. The intelligent workflows generated with this mechanism can interact with the MES system. Thus, the MES will be able to manage such information for applying the rules and decisions made in the workflows.

Versatile and modular plug&produce equipment with build-in intelligence for flexible production, self-state monitoring and optimal re-use

The re-use of shop-floor equipment for other production variants is today related with high efforts of adapting to new process goals on the machine component and software level. This drawback has already been addressed by task-driven process execution within the framework of intelligent manufacturing systems (IMS). IMS feature a decentralized concept, where the components have to be capable of simultaneously addressing both knowledge processing about manufacturing capabilities and material requirements2. This allows the manufacturing components to flexibly react to changing requirements and production conditions in an optimal way. Multi-Agent Systems (MAS) are widely used to model IMS and to enable component-based plug-and produce structures4 In very recent approaches these agents can even find process methods, which take the effect on subsequent processes into account in order contribute to an overall production optimum5. These capabilities already allow the integration of such production equipment in old, new or renewed production environments.

The central point in all of these approaches is the intelligence, which is built into the equipment, reflecting the process expertise of the equipment builder. The corresponding knowledge is represented in a manifold of ways such as mappings and semantic structures and it is accessed by corresponding technologies (graphs, functions, ontologies). Today's modern shop floor IT systems are based increasingly on innovative communication architectures like service oriented architecture (SOA) that takes flexibility and maintainability into account.

Current research approaches are going beyond this point and are e.g. focusing on a Software-as-a-Service (including a Cloud Computing approach) . Semantic technologies for integration purposes are raising interest, as well. All these approaches form a sound basis for the development of production entities, which can be re-used in a very versatile way. The challenge of REBORN is to extend the advanced agent approaches to self-awareness of the equipment; add knowledge and methods to derive maintenance needs, refurbishment opportunities and to estimate related effort; create adequate communication schemes; and merge the agent approaches with the MES technologies

The main progress beyond the state of the art will be the implementation of the intelligence to achieve self-awareness of production equipment, to communicate the recognized self-state to other devices and superior systems and the ability to optimize under capability variations of other co-operating devices. A major step-up will be reached by also implementing the knowledge on how to re-constitute a former device state or how to enhance the capabilities beyond the original ones.

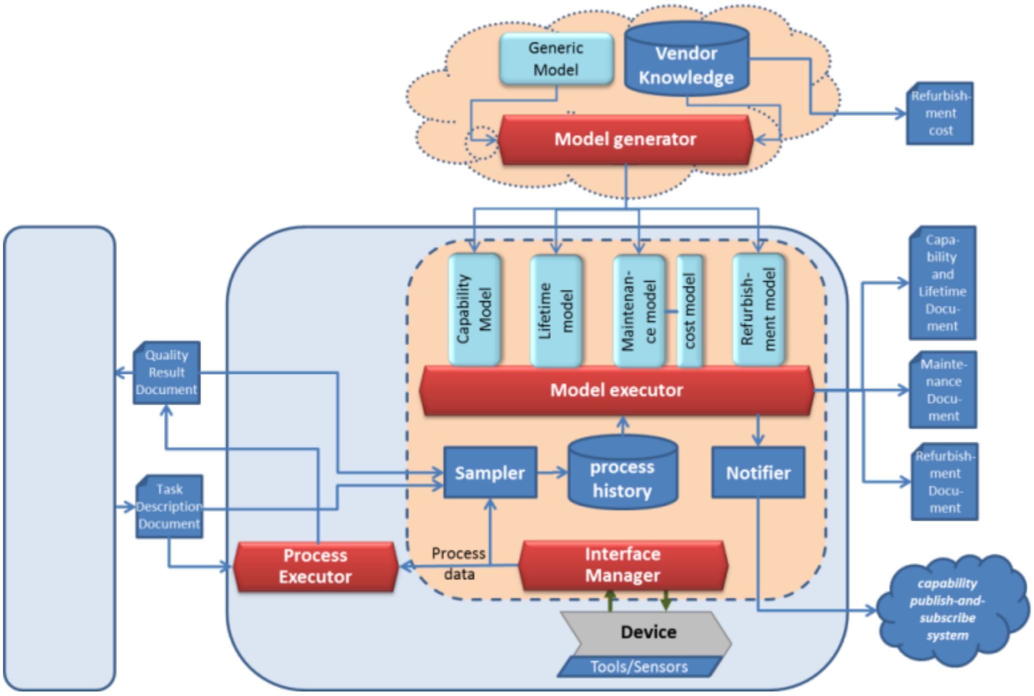


Figure 68: Building block of versatile, modular plug&produce equipment

REBORN will extend the widespread industrial best practice (MES) and the more theoretical IMS approaches in this sense and make use of the benefits of both. REBORN will implement autonomous distributed IMS system based on a holonic structure on top of existing machine architecture. The project is not aiming at substituting existing and accepted Commercial-of-the-Shelf applications like MES, Statistical Process Control, Model Predictive Control and others but expanding their knowledge range to additional dimensions (see challenges above) and therefore allowing a novel way of using equipment when composing and re-arranging production structures.

This break-through will be achieved by enhancing the equipment with new properties, based on the equipment-integrated expert knowledge about the effects of operation on the equipment. These new properties are as follows.

The VERSONS will permanently monitor their own capabilities with respect to the degree of task fulfilment, which can be achieved in the present state. They will also determine the present-state effort (cost) associated with the respective task execution to allow for overall optimization. This information will be derived from the processing history, which records the tasks to be fulfilled, the respective process execution and the achieved quality. The findings of this analysis will cover the current cost or effort related to a certain task and will be reflected in the self-description, where the occurrence of major changes is signalled to all other production entities. This is also connected to an estimate of a capability lifetime, which reflects the time to the next major change in capabilities. In this sense, the VERSON is making an evolution during its lifetime, which is made transparent to optimization, planning and maintenance services. A VERSON with wear-related restrictions of its capabilities can still be used for a different task class where the capabilities are sufficient. If no such redistribution of tasks can be found, a maintenance, replacement or refurbishment service can be invoked. But the VERSON evolution can also enhance the capabilities whenever the VERSON manufacturer supplies new process methods to the VERSON.

Systematic Approach: The REBORN methodology will allow enhancing existing machinery with the new properties (see above). These are created by developing new, corresponding equipment models (capability model, lifetime model, maintenance and refurbishment model with associated cost models), which describe the transformation from process history data to the estimates of the relevant information such as remaining capabilities, lifetime, and so on.

Generic models will be defined, which can be instantiated by the model generator with the data of dedicated process equipment. The models will build upon semantic relationships (ontologies) as well as functional relationships. On this basis, a model description language will be defined, which allows the formal representation of the models. A model processor will be developed, which interprets and executes the models (transformations) with the process history data in order to derive the desired information. As the models represent vendor specific IPR or must not be disclosed for other reasons, model documents would be encrypted or they might be represented as well by directly executable transformations, which would just be run by the model processor. The processor finally creates equipment state documents containing the information and releases a notification via the Notifier to a dedicated capability publish-subscribe system, when a major change in the capabilities occurred. The documents will be XML and a schema is developed for standardized exchange.

The "Interface Manager" will be adopted for the different machines. The REBORN system will provide different generic templates, which will fasten the implementation. The Interface Manager will realize a number of connections towards the machine's PLC (which allows for acquiring further process data), to a possible machine host PC (red) and to a so-called "interceptor", which interfaces the communication of the machine with the existing systems in the factory backbone. The connections will be proprietary and of different formats (serial, OPC, ADS, SECS/GEM...).

Data collection and processing: Consequently, the Sampler component will also care for the storage and aggregation of sensor data acquired from the machine's PLC and process signals acquired from the machine's PLC and host adapter and from an eventually existing "grid of sensors", where devices might be organized. The data pre-processing will cover super/subsampling of multi-resolution signals, offset/ scale/ drift compensations and limit checking. The aggregation will work on time-based and triggered intervals and will also provide (auto-) correlation. The entities will buffer these data to process history database.

The new models and processors will be integrated in the already existing XPRESS, EUPASS and TRANSPERENCY schemes, which deal on the shop floor level with device self-description, task- and model-driven process execution, real-time co-operation, document exchange communication schemes, cross-process optimization and live-cycle performance recording.

Strategies for re-use of production equipment in existing production systems

One focus of this S&T objective will be to identify new methods for bringing the concepts of Plug&Produce technology into legacy devices. The aim will be to allow existing legacy devices to be equipped with embedded intelligence and integrated into the wider future and existing Plug&Produce architectures.

This will be possible through the use of low cost processing devices that were developed and demonstrated in IDEAS project along with light-weight device-tailored wrapper agents that embody the device intelligence and software interface which were developed and demonstrated in XPRESS and IDEAS projects.

This will be achieved through the use of agent wrappers around existing equipment controllers, providing the necessary mechanisms to interface with them. These agent wrappers will be based on previous efforts developed and demonstrated in previous projects IDEAS and XPRESS. However since the focus in IDEAS and XPRESS was primarily on proof of concept specifically with new systems; the aim here will be to bring these concepts into existing legacy equipment and enable the realization of industrial adaptation through exploring standardisation avenues and industrial demonstration across different manufacturing sectors.

Another focus of this S&T objective will be on enabling continuous assessment of the status of the devices and equipment throughout the various stages of the system's lifecycle(s).

This life cycle assessment will be based on continuously collecting operational data about the devices and their different performance indicators, which will be easy due to the embedded intelligence capabilities that the Plug&Produce agent wrappers provide. Continuous knowledge capture and feedback will result in updating the initial prediction and reliability models that reside within the design framework and make them more accurate and representative of the current status of the system or device. The purpose behind this will be to ensure that all devices that make-up a manufacturing system are accurately represented in the design framework through their virtual/agent presence which will make maintenance, refurbishment, upgrade, reuse and disposal decisions more informed and optimised down to the device level.

The outcome of this objective will be that equipment which are no longer in use, can be easily reused in another manufacturing system with different requirements, due to the easy adaptable machine/machine interfaces and simplified user-machine interfaces. The knowledge separation and the flexible system reconfiguration capability of the intelligent device play an essential role to achieve up to 100% reusability of manufacturing equipment.

Moreover the optimised life-long assessment of the device status along with accurate simulation and prediction models will have the end result of optimum usage and utilization of manufacturing equipment throughout their lifecycle(s) and high reduction in their waste and disposal rates.

Current industry's state of the art lacks in formal methods of designing easy to integrate interfaces that enable modular equipment reuse. To overcome this, formal models for building easy to integrate machine-to-machine interfaces will be devised based on the multi-agent software paradigm. A specific focus will be given to the standardisation and industry validation of these concepts.

The knowledge framework will be based on the knowledge capture and reuse framework developed and demonstrated in TRANSPARENCY project. However TRANPARENCY's knowledge framework was demonstrated specifically for the machine tools domain, the aim here will be to industrially demonstrate this methodology within various settings across the different manufacturing sectors in Europe and seeking more robust ways to standardise the different knowledge engineering processes involved in order to bring them into maturity levels that suit standard industrial adoption.

Models for innovative factory lay-out design techniques and adaptive reconfiguration

Due to ever decreasing product life cycles and high external pressure to cut costs, the ramp-up of production lines must be significantly shortened and simplified. This is possible only if the simulated production scenarios mirror accurately the real conditions in terms of speed, performance, costs, availability, reusability and reliability will be automatically identified through simulation.

After the configuration and building phases, the ramp-up of new production lines needs adaptation efforts depending how good the simulated scenarios mirror the real production conditions on-site. This is currently not the case, the ramp-up phase can last up to 12 months e.g. in the automotive industry. Because current simulation software are only able to integrate specific characteristics of robots or joining components in a very limited way, e.g. via libraries, high effort is still necessary to adapt the production line which has been planned with such a simulation tool to the real conditions on site[12]. The goal is not only to minimize these adaptation efforts but also to support the fab planner in identifying the changes needed to achieve an optimum layout and set-up of the production line.

ReBORN will provide a high performance simulation tool with a dynamic self-learning environment and a modular structure. Each software module represents and simulates a specific production process under real conditions through the connection with a knowledge database including all the methods for the performance of this specific process. Tasks performed by humans can also be easily considered in the simulation through an advanced Human Machine Interface developed in ReBORN.

The reduction of the ramp up time up to 50% can only be achieved if the different simulated scenarios mirror the real production conditions on-site. The connection to a network of knowledge databases with actual process data (self-described knowledge about production equipment and processes) and with high performance data mining capability is the innovative breakthrough, which will allow mirroring the reality on-site. The effect of changing boundary conditions can easily be simulated and the production configuration can continuously be adapted by semi-automatic simulation and optimisation loops. As a consequence only minor adaptation efforts will be necessary on site to achieve the optimum set-up of the production line. The effort for optimisations during the ramp-up phase will be reduced drastically. Nowadays, this optimisation time takes a minimum of 6 months.

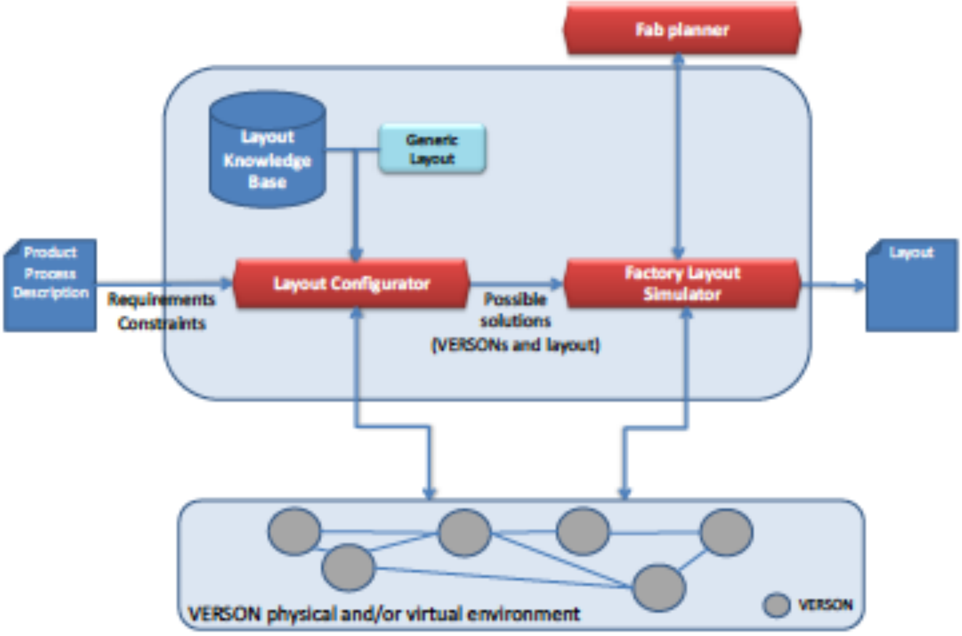


Figure 69: ReBORN high performance simulation tool

ReBORN will integrate and demonstrate the technologies that can reduce this time to a couple of weeks. Existing production lines can also be optimised by the simulation system in terms of their efficiency, availability and reliability by improving, replacing and reorganising the components based on changing product requirements and equipment capabilities[13]. To reduce the simulation effort, ReBORN will integrate a multi-variable optimization module to reduce the solution space and identify the best candidate solutions that will them be explored by simulation[14].

The equipment improvement is an on-going, automatic process possible due to the connection with knowledge databases. Also the adaptive reconfiguration, replacement and reorganization, of the factory layout is easily done by the new simulation tool. In this area an optimization potential up to 10% is expected. This capability to fully simulate and optimize the factory layout will also contribute to shorter reaction times on process disturbances by simulating different fall-back scenarios for e.g. failure or underperformance of an element in the production chain.

Design methodology for de-manufacturing, dismantling, recycling and value chain extension incorporating prior expert knowledge and experience

There are numerous design methodologies for the development of products, which can be implemented for production systems. Some of these methodologies are House of quality, axiomatic design or Design for X; however a common point of all these methodologies is the establishment of a sequencing of events and tasks. The application of these methodologies assumes that there are three factors that enable their implementation: detailed knowledge of the functional and non-functional requirements, applicability of this knowledge in the design of new machines and previous experience of the designer.

All these methodologies implement four basic phases designing a product[19]: Stakeholders requirements, Conceptual design, and Virtual testing and detailed design. In the first phase, manufacturing system builder should establish basic requirements (technical and economical) for developing the conceptual design. In the conceptual design phase, the system's basic structure and functional concepts are defined and stabilized. At the detailed design phase, involved components are dimensioned using several software systems including virtual testing models. Depending on the type of product and associated risks in the development, a number of prototypes may be necessary to achieve for an industrialized product.

The proposed methodology will enable efficient co-design environment that enables design and continuous evaluation of modular component based manufacturing systems in which the life-cycle cost assessment represents a central focus. The main objective is to create a methodology, which is knowledge-based and can be concurrently used by suppliers, system integrators and end-user that continuously integrates the knowledge about the state of various components and equipment in a formal way that makes design, upgrade, renovation, reuse and recycling decisions seamlessly integrated to allow highly informed trade-offs.

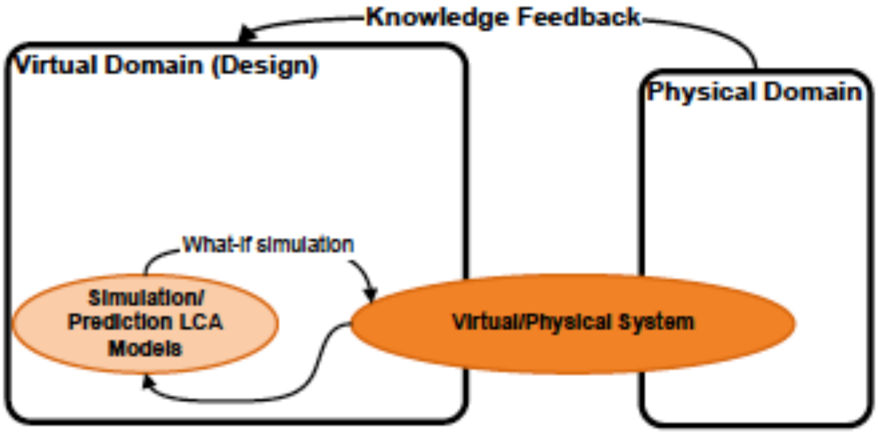


Figure 70: Knowledge feedback from physical to virtual domain

The methodology will range over the life-cycles of operating manufacturing systems, upgrade/ redesigning machine-tools and disposing and re-using components. Detailed provisions will be met to feedback knowledge from operating life-cycles to subsequent redesign cycles or to the design of subsequent generations of similar systems.

Further on, by implementing (i) virtual test-cases against predicted key performance indicators (KPIs) and fed-back knowledge and by (ii) executing sensitivity analyses, the methodology will stimulate iterative approaches on conceptual design to find best fitting design approaches and components. Besides, the methodology will be flexible enough to adapt to changes occurred externally by a redefinition of a component or its performance requirements. The methodology shall allow also the definition of new requirements and the redesign of components based on field data.

9.5. Industrial demonstration scenarios

The section below will explain the proposed industrial scenarios and pilot implementations in detail. These scenarios describe how I-RAMP[3] and ReBORN results will contribute to scientific and technological progress.

Modular plug&produce equipment for electrical industry

This demonstration scenario focuses on the extension and re-configuration of existing production systems during their life-time. To do so, an existing production system for electrical components at Technax will be used. This system consists of handling, transport and gripping units as well as of PLC's for machine control. Partner IEF Werner and Harms & Wende contribute with versatile components based on linear slides and welding control units respectively. Partner ISG will deliver universal communication interfaces for the controls.

The tests shall demonstrate the integration of new elements in the production system. Various test cases will be defined which shall focus on different features such as:

- The adaptability will be demonstrated by the introduction of new product variants in an existing line. It will be show how ReBORN technology is able to react on unexpected customer demand on new products or variants The flexibility shall be demonstrated to product various products and product variant on one production line.
- The modularity will be demonstrated by re-using, re-configuring and enhancement of equipment in case of changing product geometries or materials.
- The capability for process data acquisition, assessment and condition monitoring of devices will be demonstrated by long-lasting production of test products.

- The capability for fast ramp-up will also be demonstrated after the machine or single components have been exchanged.

The scenario will therefore illustrate the relevance and benefits of the ReBORN developments during the entire lifetime of production systems. The scenario will directly lead to the development of a versatile device for the IEF automation component and a Plug&Produce subassembly unit (Technax).

Factory (re-)planning with re-used equipment

The scope of this demonstrator scenario is to demonstrate the capabilities for factory planning and re-planning including old, renewed and new equipment, focussing on three major scenarios:

- Demonstration of the re-use of factory planning and simulation procedures: The target of this scenario is to show how the factory planning and simulation time can significantly be decreased by using already existing planning and simulation jobs as a basis. Furthermore, the demonstration shall also show how existing planning scenarios can be optimized using the ReBORN factory planning modules.
- Demonstration of the planning with old, renewed and new equipment: The target of this scenario is to demonstrate how factory planning for with old, renewed and new equipment can be done using the ReBORN approach. The explored KPI's related to the conditions and states of the versatile devices will be used as a data basis. Those KPI's will be taken into account during factory planning and re-planning.
- Demonstration of the online connection between planning tool and versatile devices: The final scenario targets on the online connection between the planning tool and the versatile devices.

The online connection allows for continuous condition monitoring and state surveillance. Based on the component's KPI's, re-planning jobs can be triggered automatically in case of expected component breakdowns or malfunctions.

Set-up and ramp-up of a new E-Vehicle assembly line

The first demonstration scenario targets the commissioning and ramp-up of a new real-life production system. This activity will be led by AWL as a system integrator, who will integrate components and sensors into an assembly system for E-Vehicles. The scenario will show the cooperation of a robot, clamping devices, a welding machine and a variety of sensors, which will all be equipped with a NETDEV shell. The robot will be responsible for handling and positioning but will also be used as a component for a higher-level sensing NETDEV by incorporating lower-level sensing NETDEVs.

The demonstrator will clearly illustrate how heterogeneous components and sensors – including the sensor-equipped robot of INOS, the welding controller of HWH and sensor packs provided by FEUP – will be implemented as NETDEVs and integrated in a real-life assembly system. To perform this task, AWL will apply the Plug&Produce Communication Framework as well as the Configuration & Optimization wizards and tools. The scenario will lead to a real-life assembly system with Plug&Produce capabilities (AWL).

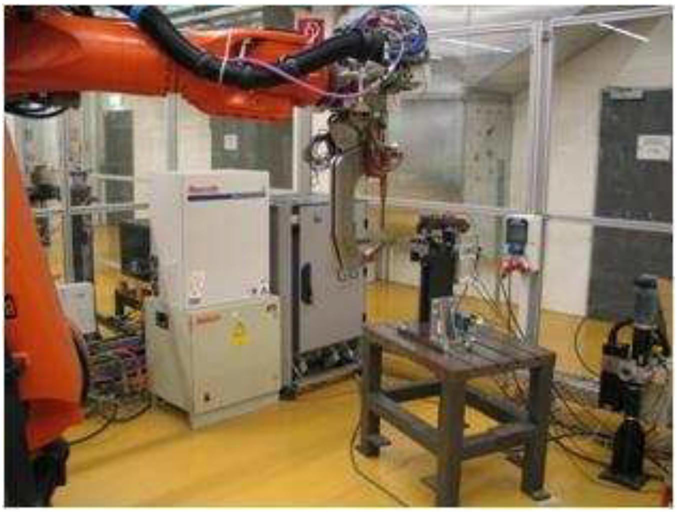


Figure 71: Demonstration robot cell

Component exchange in E-Vehicle subassembly unit

The second demonstration scenario focuses on the extension and re-configuration of existing production systems during their lifetime. To do so, an existing production system for electrical components for E-Vehicles at Technax will be used. Together with IEF as a component supplier for automation component like linear slides, the integration of new elements will be demonstrated.

The scenario will therefore illustrate the relevance and benefits of the I-RAMP^3 developments during the entire life-time of production systems. Based on the applied Plug&Produce Communication Framework as well as the Configuration & Optimization wizards and tools the device integration for an automation component of IEF will be conducted. The scenario will directly lead to the development of a NETDEV for the IEF automation component and a Plug&Produce subassembly unit (Technax).

Enhancing devices with re-use and predictive maintenance capabilities

The third scenario will demonstrate the capability of the I-RAMP^3 concept to include logic / optimization models into the NETDEVs and thereby allow an optimization of production systems based on component supplier's knowledge. This scenario will be performed on the same production system of Technax for electrical components.

The IEF NETDEVs for scenario 2 will be enhanced with analysis models for predictive maintenance and the re-use of components based on condition monitoring. This means that the usage (load, temperature, etc.) of the component will be continuously measured and analysed by models stored on the component itself. For the demonstration scenario models for the optimization of preventive maintenance intervals and re-use decision for components will be developed. This scenario will thereby illustrate the potential of the I-RAMP^3 concept to incorporate optimization models in their devices and thereby generating added value for their customers.

Figure 72: Assembly unit at Technax

9.6. Acknowledgements

The description of the I-RAMP3 and ReBORN projects vision, S/T objectives and industrial demonstration scenarios is based on the documentation from both projects and contributions from several projects partners. Consortium of both project include the following organisations: Harms & Wende, AWL-Techniek, Fraunhofer, GAMAX, TECHNAX, Hochschule Karlsruhe TW, Steinbeis-Europa-Zentrum, Critical Manufacturing, IEF-Werner, INOS HELLAS, FreedomGrow, FAGOR, Centro Ricerche Fiat, University of Oulu, University of Loughborough, ISG-Industrielle Steuerungstechnik GmbH, Polytechnic, University of Madrid, PARO AG, Z.E.C. AG and University of Porto.

10. Industrial case study

This chapter builds on the context of the industrial domain and its challenges, as detailed in the previous chapter in scope of the I-RAMP^3 and ReBORN projects, to present a first application of the changeability framework.

The case study and will focus on a scenario that targets the commissioning and ramp-up of a new real-life assembly system for E-Vehicles. A system integrator, who will integrate machines, components, actuators and sensors into the production system, is typically the leader of this activity. The scenario includes robots, clamping devices, welding machines and a variety of sensors. The case study will also include and extension and re-configuration of the existing system during its lifetime, with the replacement of existing and integration of new elements.

10.1. Introduction

To meet the challenges which come up with rapid changing product portfolios, smaller lot sizes and continuously evolving process technologies, manufacturing systems have to be easily upgradeable and versatile, in order to readily integrate new technologies and new functions, and able to respond to demands for increasing productivity through highly optimized production processes. This creates the need for novel manufacturing control systems able to cope with the increased complexity required to manage product and production variability and disturbances, effectively and efficiently, and to implement agility, flexibility and reactivity. In order to meet theses challenges, high efforts have been and are still done in research for reconfigurable and agile manufacturing systems. Significant improvements in re-configuration, performance or dependability have been achieved in the last years. However, the large-scale adoption in the industry is still missing, not only due to the limitations in current control systems [132] but also to the stepwise approach required in order to introduce new technology successfully [133].

The introduction a new model in the automotive industry generally takes from three to five years, from inception to assembly. Ideas for new models are developed to respond to unmet pubic needs and preferences. With the help of computer-aided design equipment, designers develop basic concept drawings that help them visualize the proposed vehicle's appearance. Aerodynamic engineers also review the models, studying air-flow parameters and doing feasibility studies on crash tests. Only after all models have been reviewed and accepted are tool designers permitted to begin building the tools that will manufacture the component parts of the new model.

Modern production facilities, such as assembly lines or complete production plants, are the result of integrating subsystems supplied by several manufacturers and turnkey suppliers. These vendors use many different models in order to plan, design, and improve their subsystems before delivery to the facility. However, a production facility assembled from well designed sub-systems does not in general provide the best system design and does not guarantee the vest value throughout the entire system life cycle. This life cycle can be split into a series of phases in which the processes of planning, designing, deploying and running (and eventually dismantling) a production system are developed. Each phase describes a sub-process, corresponding tasks and sub-tasks, and the role of every actor involved.

Starting from the very beginning, preliminary investigations are performed by the suppliers and the manufacturer. Based on these investigations, the so-called offer phase starts. In this phase a rough layout and a rough flow of material has to be established. The determination of cycle times, costs and delivery times follows. Afterwards the project plan should be prepared. Placement and acceptation of the order conditions the engineering phase start up. It consists of a detailed planning and an explicit flow of material, as well as geometrical simulations. This leads to the design phase. The tasks are to create a detailed design of the plant, programming of the controllers, and to make exact geometrical simulations. The subsequent phase is the realisation of the plant. In the first place the needed machines and tools have to be procured or produced. Next to all these preparations, the assembly of the technical resources is done, as planned in the preceding phases. In addition to that, the software for every controller is developed in this phase.

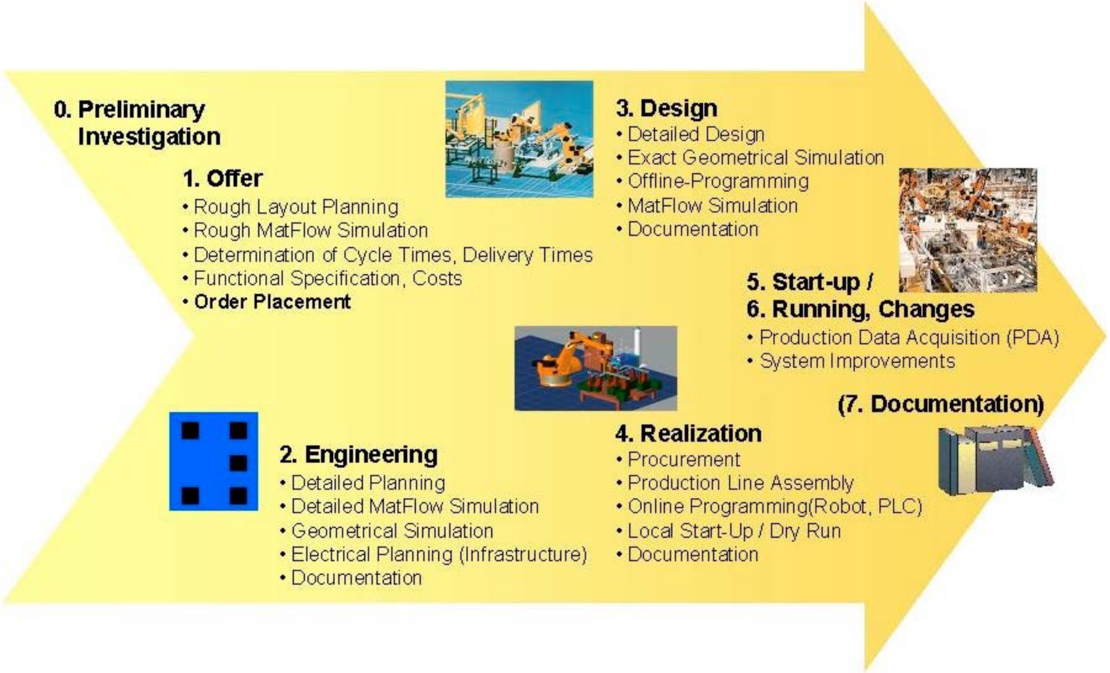


Figure 73: Phase in a production system life cycle

The sixth phase consists of performing the start-up, with the important so-called ramp-up, where system improvements and software optimisation are done. This is followed by the running phase of the plant with possible changes all along the this operational phase. This phase includes, in addition to a part of the tasks of the design phase, the quality management. The last phase, where the documentation of the whole project is made, is a very important phase nowadays, but it will become smaller or even be cut off in the future. This will be possible, because of moving the related parts of the whole documentation process to each task supported by additional meta information.

Within these seven phases the plant is customised to the system requirements determined by the customer. In each phase different roles are necessary to make the associated sub-process run. To classify the co-operative work in these phases, a Role-and-Phase-Model, formulated within the VIDOP[8] project [134] [135] [47] [49] and represented, shows what tasks are performed by what role with what resources. For each of these tasks the input and the expected output is defined. An example of activities, roles, input, output and resource per phase is presented in **Error! Reference source not found.**.

---

[8] “Vendor integrated decentralized optimization of production facilities”, Project reference G1RD-CT-2000-0030, funded by the European Community under the ‘Competitive and Sustainable Growth’ Programme (1998-2002).

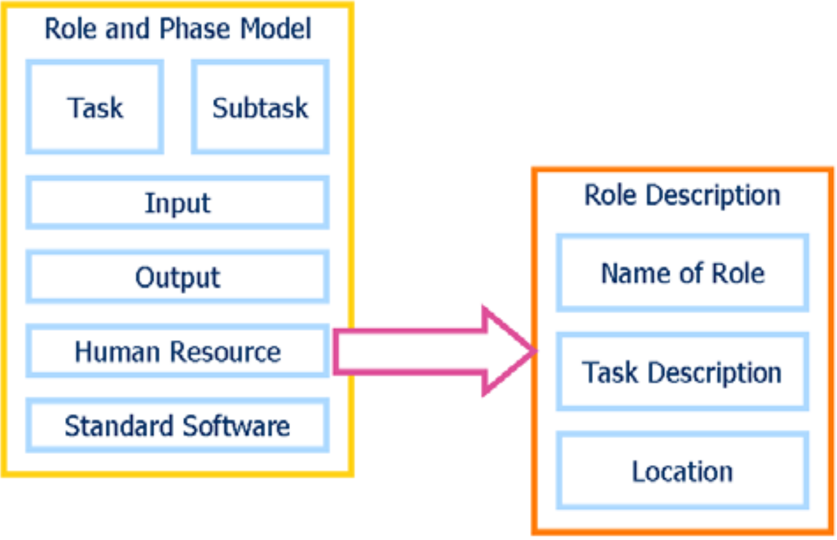


Figure 74: Role-and-Phase-model

Once a facility is in use, there are also several different events related with the performance and interactions of the components, the product being produced, and the operational conditions that will require further tuning and adjusting of the production system.

The case study will clearly illustrate how a production system composed of multiple heterogeneous equipment, components and sensors can be modelled as a system of systems in order to exhibit Plug&Produce capabilities and to extend its life cycle sustainability. This will therefore illustrate the relevance and benefits of the changeability framework in multiple phases of the system life cycle.

10.1. eVehicle assembly line

The assembly line represents the final step in the process of the vehicle production, for it is where the components supplied by multiple suppliers, including company-owned parts suppliers, are brought together for assembly in the final product.

The typical car is built from the ground up (and out). The frame forms the base on which the body rests and from which all subsequent assembly components follow. The frame is placed on the assembly line and clamped to the conveyer to prevent shifting as it moves down the line. From here the frame moves to component assembly areas where complete front and rear suspensions, rear axles and drive shafts, gearbox, steering box components, wheel drums, and braking systems are sequentially installed.

Table XI: Role-and-Phase-Model (example)

	Activity	Input	Output	Role	Technical resources
1	Assembly sequence planning	Product data version 1(drawings), Factory layout, Technical information (e.g. process data, materials)	Manufacturing bill of material	Process planner, (Assembly planner)	Digital mock-up tool, Production planner, CAD
2	Detailed planning	Product data version 2, Factory layout, Technical information(e.g. process data, material)	Manufacturing bill of material, Assembly sequence	Project engineer, Project planner, Process planner, Material flow simulation expert, Logistics and material flow expert, Technology expert, Assembly planner, Factory planner	Digital mock-up tool, Production planner, Office productivity software, Manufacturing process planner, 2D/3D-layout systems,
3	Detailed design	Product data version 4 (3D geometry), Factory layout, BOM, Assembly sequence	Optimized line and factory layout, Necessary standard machine tools	Resource design engineer, Factory planner	CAD, Manufacturing process planner, PDM, ERP
4	Online programming	Existing code, Cycle time, Assembly sequence, Sequence of optimization, Process in/out, data in/out	Software validation, Verification	Process planner, Work cell simulation expert	Human-machine interaction model, Off-line programmer for PLC
5	System improvement	Product data version 5, Factory layout, Cycle time, Throughput time, Buffer size	Modified cycle time, Modified throughput time, Modified (buffer) capacity, Modified resources	Project engineer, Process engineer, Process planner, Technology expert, Resource design engineer	CAD, PDM, Human-machine interaction model,, Work cell modelling tool
6	Detailed design	(New) product data version 7, Factory layout, (New) BOM, (New) assembly sequence	Optimized line and factory layout, Necessary standard machine tools	Resource design engineer, Factory planner	CAD, Manufacturing process planner
7	Final drawings design	Factory/Line Layout,Details of tools and machinery, Collection of drawings from all suppliers and sub-suppliers	Complete set of drawings, Simulation models according to OEM specification,(e.g. Data and Hardcopy)	Resource/product design engineer	CAD, Work cell modelling tool Robot instruction program

An off-line operation at this stage of production mates the vehicle's engine with its transmission. Workers use robotic arms to install these heavy components inside the engine compartment of the frame. In the first stages of production, robots weld the floor pieces together and assist workers in placing components such as the suspension onto the chassis. Because of the nature of these heavy component parts, articulating robots perform all of the lift and carry operations while assemblers using pneumatic wrenches bolt component pieces in place.

Generally, the floor is the largest body component to which a multitude of panels and braces will subsequently be either welded or bolted. As it moves down the assembly line, held in place by clamping fixtures, the shell of the vehicle is built. First, the left and right quarter panels are placed onto the floor pan, where they are stabilized with positioning fixtures and welded. The front and rear door pillars, roof, and body side panels are assembled in the same fashion. The shell of the vehicle (the "body in white") assembled in this section of the process lends itself to the use of robots because articulating arms can easily introduce various component braces and panels to the floor and perform a high number of weld operations in a time frame and with a degree of accuracy human workers could ever approach. Robots can pick and load heavy roof panels and place them precisely in the proper weld position within strict tolerance variations. During welding operations, parts are held securely in a jig while operations are performed. Figure 75 provides and example of a body in white assembly line.

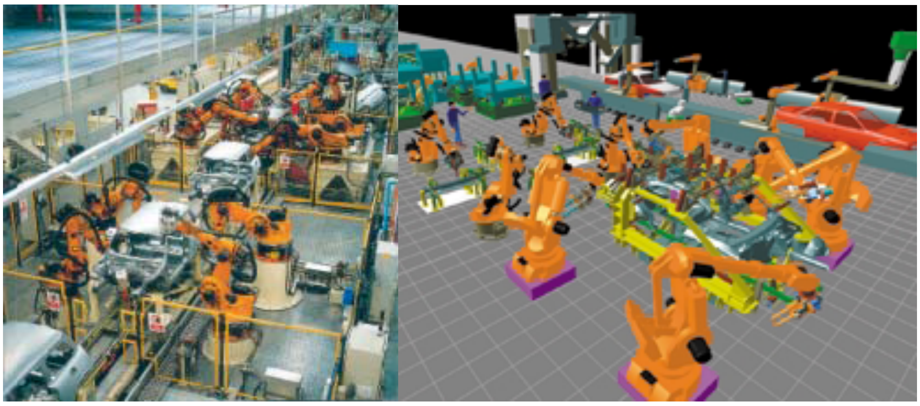


Figure 75: Body in white assembly line (real / digital)

Once the body in white is complete, it is attached to an overhead conveyor for the painting process. The multi-step painting process entails several steps like inspection, cleaning, undercoat dipping, drying, topcoat spraying, and baking.

As the body moves from the isolated painting area of the assembly line, subsequent body components including fully assembled doors, deck lids, hood panel, fenders, trunk lid, and bumper reinforcements are installed. Although robots help workers place these components onto the body shell, the workers provide the proper fit for most of the bolt-on functional parts using pneumatically assisted tools. After the painted body leaves this area it is ready for interior assembly to include all the accessories.

In the interior assembly area workers assemble all of the instrumentation and wiring systems, dash panels, interior lights, seats, door and trim panels, headliners, radios, speakers, all glass including the windshield, steering column and wheel, body weather-strips, vinyl tops, brake and gas pedals, carpeting, and front and rear bumper.

After passing through this section the vehicle typically goes through a water test to ensure the proper fit of door panels, glass, and weather-stripping.

Until this phase the process to build a gasoline car and an electric car is similar. In this final phase the electric motor (and drivetrain), the controller and batteries are assembled into the vehicle.

The complete assembly is extremely complex, but can be decomposed into multiple subsequent steps. For the purpose of this case study we will focus on a specify step in the assembly of the body in white: the step in which the roof (or the sunroof) are assembled/attached to the car body. The chosen manufacturing line is part of the e-Vehicle production system. Figure 76 shows a possible CAD layout for the roof assembly line and Figure 77 and image of the assembly line.

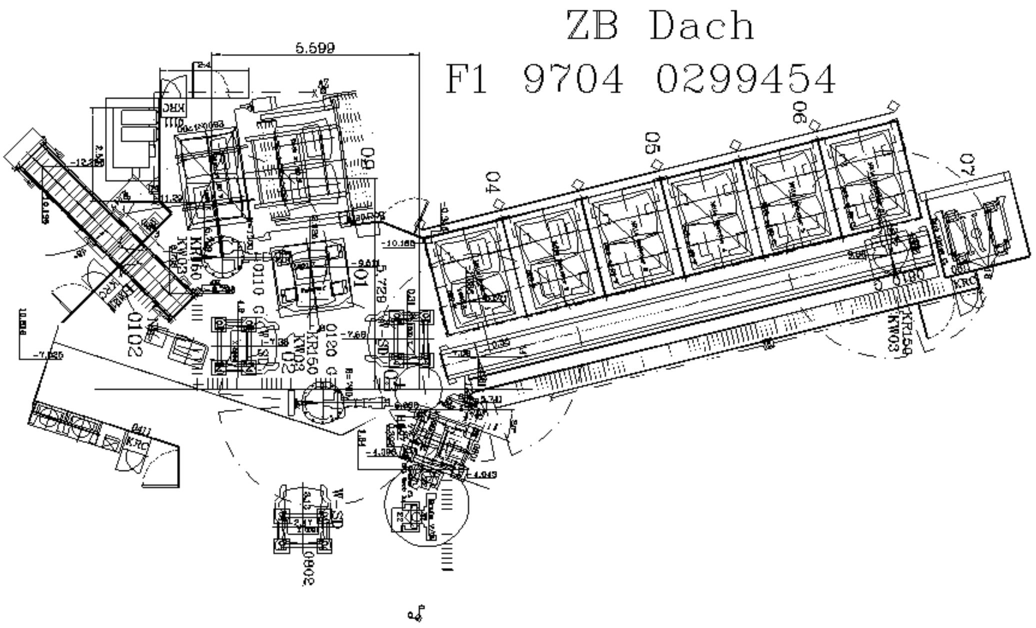


Figure 76: Layout of the roof assembly line

Figure 77: Roof assembly line

This manufacturing line is capable to produce two variants of the e-Vehicle: one with the conventional roof and another with a sun with a smaller roof radius. Figure 78 shows the parts that change for each of the type of vehicles in this scenario.

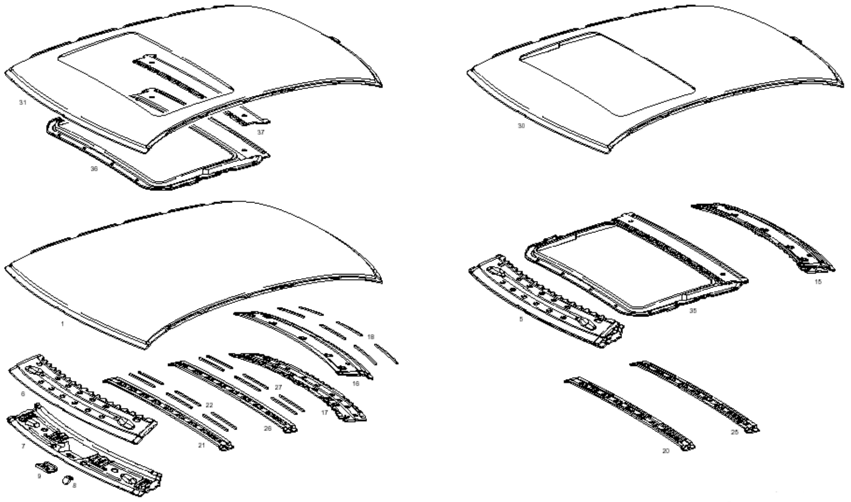


Figure 78: Parts affected in the virtual scenario

The manufacturing line includes two robots, one hemming[9] device, two conveyers and multiple other sensors and actuators. Figure 79 is a digital representation of the possible layout and Figure 80 the digital representation of the hemming device.

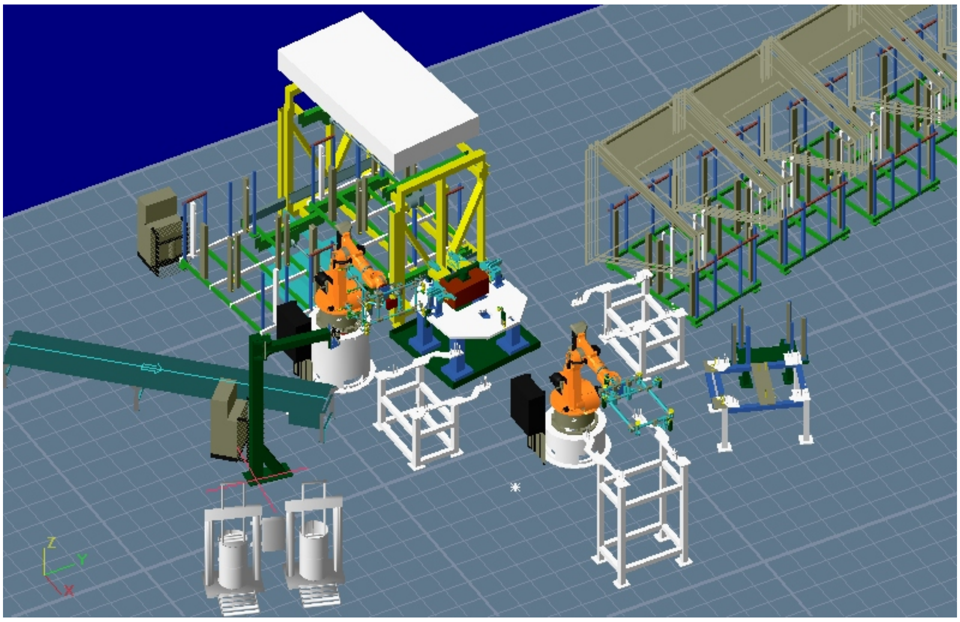


Figure 79: Model of the production line

---

[9] Hemming is a technology used by the automotive industry to join inner and outer closure panels together by bending/folding the flange of the outer panel over the inner one.

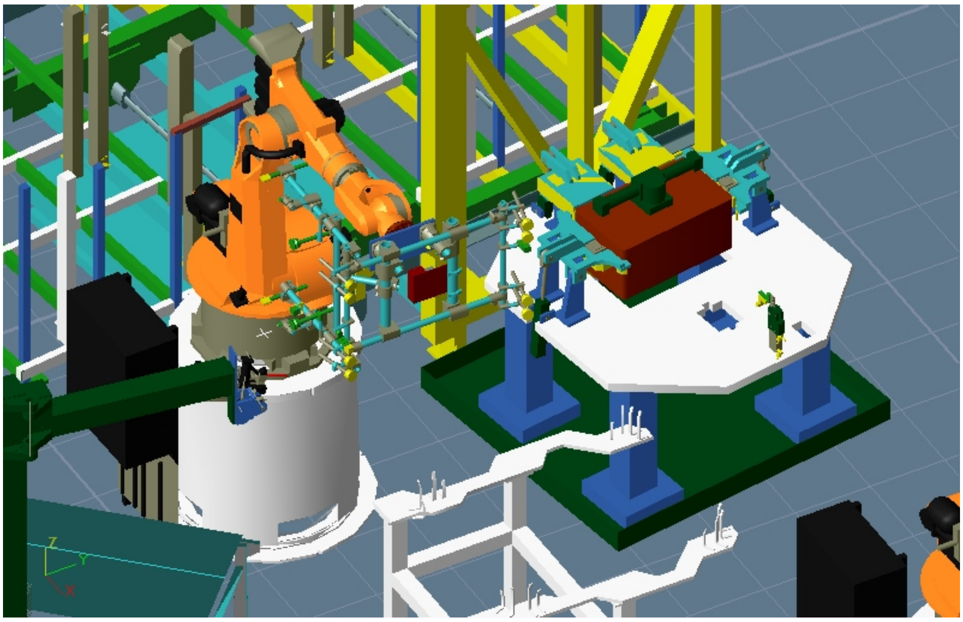


Figure 80: Hemming device

10.2. Shop floor layout and ramp-up

The first scenario involves the selection of the best equipment, the initial shop floor layout and the ramp-up of the production system. Following a Plug&Produce approach, the idea is that each production equipment is wrapped by an Intelligent Manufacturing Unit (as describe in 3.3) and each IMU describes themself in relation to the system by means of the a “Self Description” (SD). The SDs are responsible to specify which tasks the equipment can execute, the data they required for the task execution, as well as which product and process data they are able to provide.

In the case that several production equipment comes into consideration for executing a specific process step, it is necessary to collect the different specifications and choose the most adequate equipment to perform the task, attending to several performance criteria and other requirements for the manufacturing line.

After the selection of the best equipment, system configurations and interaction between the different elements are specified and validated. During ramp-up, control and other programs are loaded into the equipment (and respective controllers) these parameters are fine-tuned and the respective IMUs are configured.

In the current scenario, the implementation of the production line can be modelled as five IMUs: one for each of the robots (2x), one for the hemming device (1x) and one for each of the conveyers (2x). For the sake of tractability other IMUs that could have been include (for example for the grippers, fixtures or sensors) were not considered. In this case, implementation of the production line could follow the architecture presented in Figure 81.

Figure 81: Production line architecture

The corresponding control architecture, using the IMUs as shells that encapsulate production equipment, could be the one represented in Figure 82, with one observer responsible for estimating and evaluating the state of each of the independent production equipment.

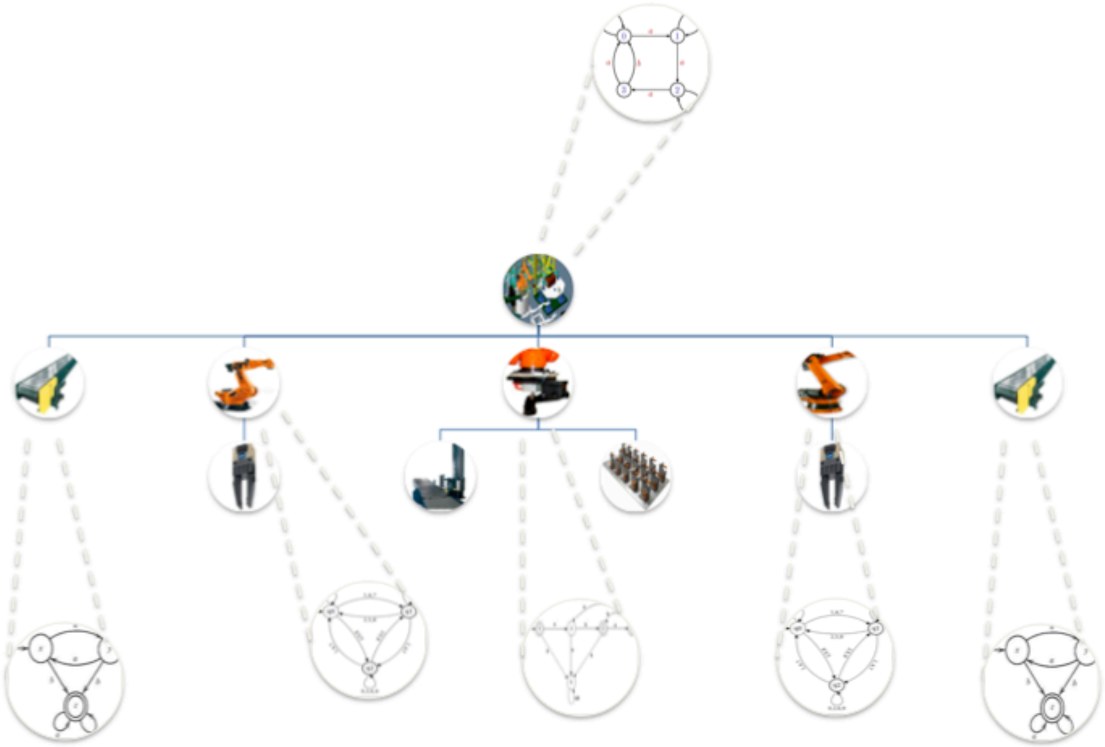


Figure 82: Production line control architecture

After ramp-up is concluded, the execution process at the shop floor starts. During production, the overall controller of the system passes task descriptions, generated in the configuration phase, to individual IMUs. Due to the fact that it possesses all the necessary information and knowledge, the IMU should be able to execute the process step successfully.

At the end of the process step the product and quality data are returned to the controller simultaneously with the physical unloading of the work piece. An observer uses the result documents to evaluate the process, and depending on the results, suggest changes or adaptions. During this phase, running, changes might be needed to respond to evolving requirements, equipment malfunctioning or other unexpected events. Changes can be easily absorbed by control architecture, both in terms of responding to equipment failure as in terms of need for additional flexibility (Figure 83).

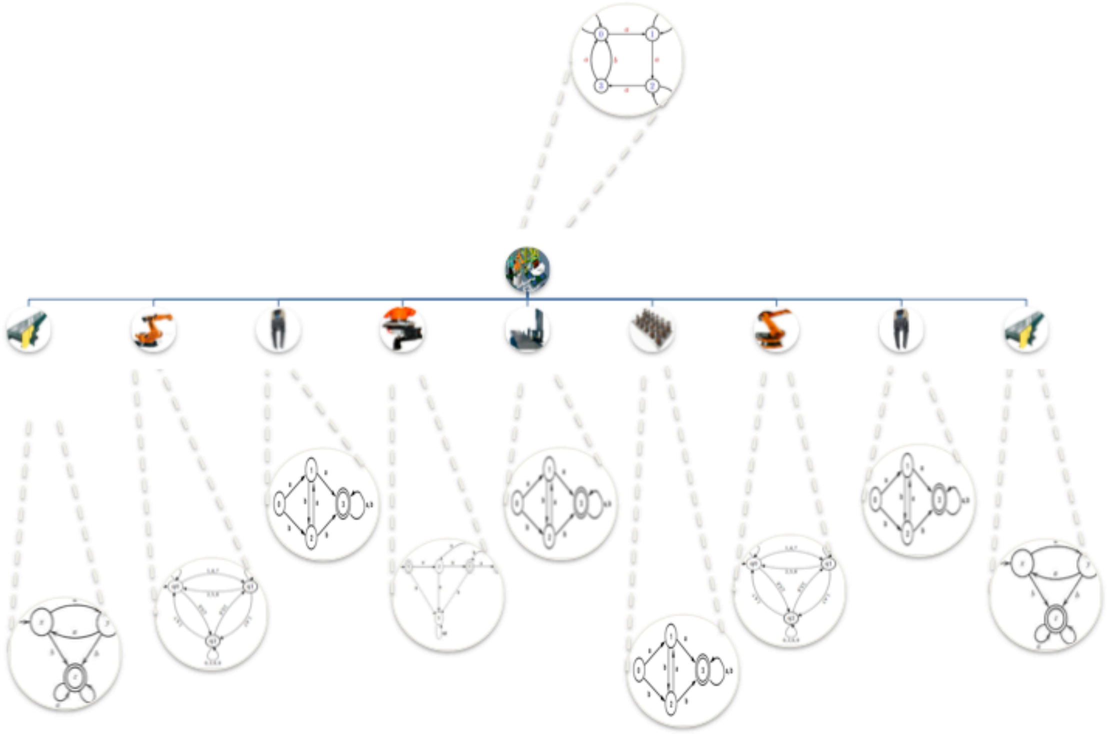


Figure 83: Alternative control architecture

Multiple control architectures (plays, as defined in 8.1) can be defined and analysed during the design phase of the manufacturing system. This way time to react to "change events" can be reduced and based on a selection of the most adequate play amongst the existing ones, i.e. to selection of the most adequate play from the playbook (see Figure 84).

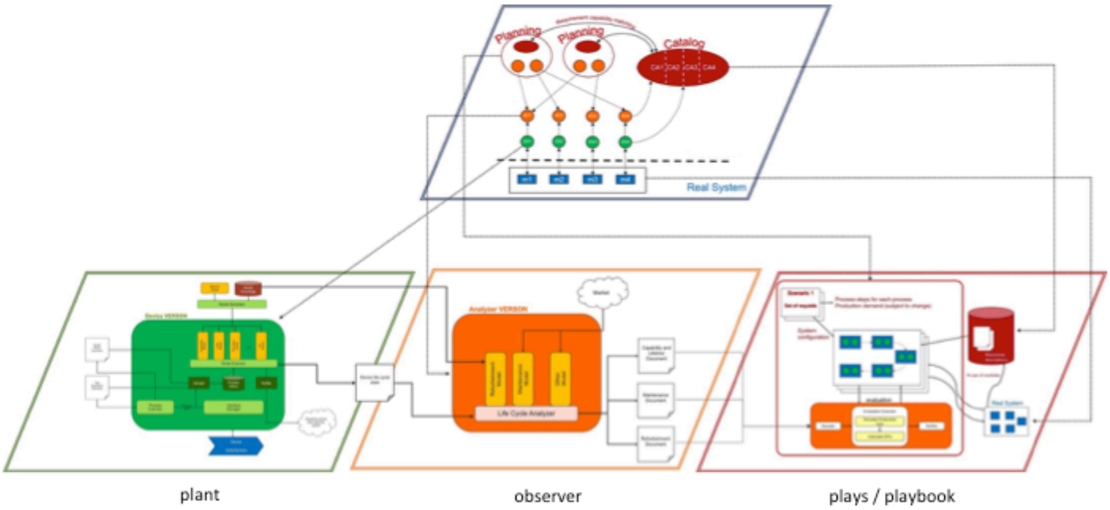


Figure 84: Overall architecture including playbook

During the running phase, new plays are included and obsolete plays marked as such. This guarantees the dynamics of the playbook and ensures its adequacy for the control of the system.

10.3. Assembly line (re-)planning with re-used equipment

After initial commissioning and ramp-up, and until dismantling, the assembly line is operated for a certain period, within a certain (variable) workload and in a certain location. Re-use of equipment means, that during this period the purpose (assembly of roofs), the context (workload) or the location, or any combination of those is changed, and the assembly line needs to be re-planned for use under the changed conditions. Another possible event that requires change and decision is the mal-function of equipment and the need to have it replaced. According to [136], three types of re-use[10], with incremental complexity, can be distinguished:

- The easiest and trivial case when there is an equipment mal-function and it needs to be replaced for a similar one.
- When changes in the conditions are only minor, the equipment may be used only with adapted parameters and configuration, without changes in the internal composition (changes within the adaptability corridor). In this case the only

---

[10] There is an additional type when only the change of location is needed. In this case no modification except transport is required and the equipment can be used as is, if the remaining functionality and lifetime is sufficient.

question to answer is how the parameters have to be adapted in order to meet the new requirements.

- The most challenging re-use case is met when the existing capabilities are insufficient with all feasible parameter setting and configurations (change outside the adaptability corridor). In this case the internal composition (hardware and/or software components) has to be modified. If these modifications are minor, this is still considered to be a kind of re-use.

In all three cases the information on the present wear state and the remaining capabilities is essential to decide on re-use.

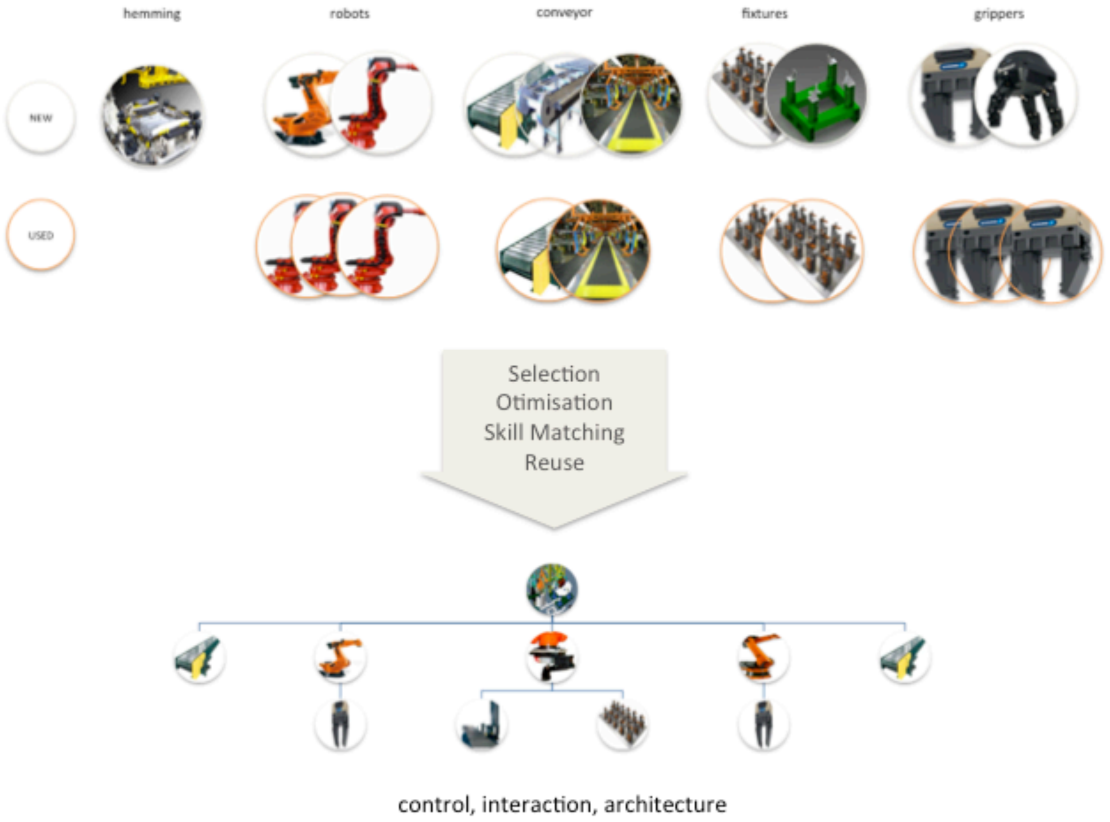


Figure 85: Factory planning with re-use of equipment

A pre-requisite for a re-use decision is the information if the capability range of the equipment allows the operation under the new conditions and for how long they can be sustained. Means for the adaptation of the parameters and configuration to the new conditions are required. The selection of the right equipment and the optimisation of their interactions for the desired purpose will follow a similar approach as the one for the previous scenario.

11. Robotics case study

This chapter presents the context for the application of the event driven changeability framework in a concrete robotics application scenario. It will start by defining the context for a scenario of a permanent Ocean observatory and will conclude with a first application of the changeability framework to the robotics domain.

11.1. Introduction

Unmanned vehicles have already proved invaluable in environmental field studies by providing levels of spatial-temporal sampling resolution that could have not been attained before. Recent trends show that the levels of spatial-temporal sampling resolutions attained with individual vehicles are feasible for wide areas through the operation of persistent vehicle networks. The possibility of persistent sampling over wide areas has the potential to revolutionize environmental field studies and for the deployment of permanent Ocean observation. The roles of unmanned vehicle systems in future Ocean observatories are discussed in the light of the recent technological developments and trends, along with the major challenges associated to this vision. The discussion is illustrated with examples of developments from the Underwater Systems and Technologies Laboratory from the University of Porto[11], Portugal.

The last decades have witnessed unprecedented technological developments in computing, communications, navigation, control, composite materials and power systems, which have led to the design and deployment of the first generation of unmanned vehicle systems. These vehicle systems have seen action at sea, in the air, on the ground, and even on

---

[11] http://lsts.fe.up.pt/

other planets, in particular Mars. Future generations of unmanned vehicle systems will reflect several current trends: increased levels of autonomy and self-awareness, lower cost, longer endurance, and networking capabilities. These trends will enable scientists and engineers to develop visions for future systems, and applications, that could have not been imagined before. Ocean observatories are one these applications.

Environmental field studies are becoming more and more demanding as scientists seek to understand environmental processes and how these affect, and are affected, by mankind. This is a challenging task. Our environment evolves in multiple temporal and spatial scales as the result of complex interactions that are far from being fully understood.

Environmental data collection is one of the difficulties associated to environmental field studies. Sensors are required to take measurements with adequate temporal and spatial resolutions, and the measurements may have to be communicated and processed in real-time to adapt the sampling strategies (both temporally and spatially) to the observations. In summary, distributed sensing with mobile nodes has to be complemented with communications and real-time decision-making. This is why network vehicle and sensor systems have the potential to revolutionize environmental studies.

There are several challenges associated to the vision underlying this revolution. The availability of affordable vehicle systems with inter-operable networking capabilities is still far in the future. The same happens with the capability to design and deploy networked vehicle systems in a systematic manner and within an appropriate scientific framework.

11.2. Overview of unmanned vehicles

The last decades have witnessed the increasing success of unmanned vehicle systems: AUV operating under in the Artic [137] [138]; UAV performing atmospheric research[12]; cars driving autonomously in the desert[13] or in the city, [139]; data collection in Mars[14], robots playing soccer [140], etc.. The key to this success comes from the obvious fact that these are unmanned vehicles: they can perform dirty, dull and dangerous tasks in all types of environments (ocean, air, land and space).

The operation of unmanned vehicles does not necessarily remove humans from the operation of the vehicle. In remotely operated (or piloted) vehicles there is a human operator, which may be located at some remote location, in charge of piloting the vehicle. This is done

---

[12] See research from the Dryden Flight Research Center http://www.nasa.gov/centers/dryden/news/FactSheets/FS-059-DFRC.html.Accessed July 2010

[13] DARPA Grand Challenge, http://www.darpa.mil/GRANDCHALLENGE/. Accessed July 2010

[14] MARS-ROVER, Mars exploration rover mission. http://marsrovers.nasa.gov/home/index.html. Accessed July 2010

with the help of a communication channel: sensor information is sent from the vehicle to the operator that, in turn, sends commands to the vehicle. The reliance on the operator and on the communications channel is the main limitation of this mode of operation. This is not compatible with the operation of vehicles in remote environments, such as the ocean or the space, where communications are typically difficult and a significant time delay may exist.

Autonomous vehicles are the (partial) answer to the limitations of remotely operated vehicles. Autonomous vehicles are capable of executing mission plans without the intervention of human operators (i.e. autonomously). There are several degrees of autonomy, some of which are not feasible with the current technologies [141]. For example, full autonomy is still not feasible today, and vehicles still lack the sensing and reasoning capabilities required for that purpose. This is partly why the concept of mixed initiative operation was introduced in the last decade [142]. In this concept, human operators are part of the planning and control loops of the vehicle. Informally this can be described as "supervised" autonomy. For example, the operator is capable of generating plans and uploading these plans to the vehicle for autonomous execution, or the operator is also able to override plan execution and re-task the vehicle to execute new plans.

Depending on the operational environment, key technical specifications for unmanned vehicles include endurance, size, payload, range, communication and navigation capabilities, and deployment mechanisms [143] [144]. Endurance is highly correlated with the limitations of current energy storage technologies.

There is no Moore's law[15] for unmanned vehicles. However, from the technological advancements in computation, power storage, sensor technologies and communications it is possible to infer some current trends for unmanned vehicles: miniaturization (more capabilities in less space), longer endurance and networking capabilities.

Space limitations preclude a thorough discussion of current capabilities and limitations of unmanned vehicle systems. However, these need to be fully understood before unmanned vehicles can be effectively deployed in field studies. Next we present examples of ocean and air going unmanned vehicles from USTL (Figure 86) to illustrate some of the key concepts discussed in this section.

The most recent AUV from USTL, the Light Autonomous Underwater Vehicle (LAUV) is a prototype of a low-cost submarine for oceanographic and environmental surveys[16] . It is a torpedo shaped vehicle made of composite materials (110x16 cm) with one propeller and 3

---

[15] http://www.intel.com/technology/mooreslaw/index.htm. Accessed July 2010

[16] Seascout system (http://whale.fe.up.pt/seascout/). Accessed July 2010

(or 4) control fins. The LAUV has an advanced miniaturized computer system running modular controllers on a real-time Linux kernel. It is configurable for multiple operation profiles and sensor configurations. In the standard configuration, it comes with a low-cost inertial motion unit, a depth sensor, a LBL system for navigation, GPS, GSM and WiFi. The maximum operating time is 8 hours.

Figure 86: Unmanned vehicle systems

Lusitânia is a UAV based on a remotely controlled model airframe equipped with one OS 91-FX, 15cc, 2.9HP, 2 stroke engine. Lusitânia is equipped with the Piccolo autopilot (Vaglienti et al. 2004), with a small video camera and with Telos motes[17] (with meteorological sensors optimized for use on a UAV platform). The camera can be remotely controlled, and provides the operator with a video feed in real-time. This is done through a 2.4GHz wireless transmission system with a range of 8Km [145]. Flights are limited to 80 minutes in duration.

ANTEX is a family of UAV platforms developed by the Portuguese Air Force Academy [146]. ANTEX-X03 is a 6 meter wingspan platform with a 220cc, 22HP, 2 stroke 3W engine for a payload weight not exceeding 30kg. ANTEX X02 is a 1:2 scale model of ANTEX-X03 with a 15cc, 2Hp, 4 stroke Saito100 engine, for a maximum payload takeoff weight of 7Kg. The ANTEX UAV family has a standard computational and sensor configuration. It is configured to fly with two different autopilots: Piccolo and Micro-Pilot[18]. The maximum

---

[17] A mote is an autonomous sensor capable of mesuaring parameters such as temperature, sound, vibration, pressure, motion or pollutants. A wireless sensor network (WSN) consists of spatially distributed motes that cooperatively monitor physical or environmental conditions.

[18] Micro-Pilot autopilot. http://www.micropilot.com/. Accessed July 2010

flight time ranges from 1 hour to 12 hours, depending on the platform and on its configuration.

11.3. Networked vehicle systems

Networking is one of the major trends for unmanned vehicle systems; it is also one of the enabling technologies for distributed cooperation (and computation). In the reminder of the paper we use *network vehicle systems* to describe systems where vehicles, sensors and operators interact through (inter-operated) communication networks.

Network vehicle systems offer new possibilities to the operation of unmanned vehicles [56]. For example, in network vehicle systems, information and commands are exchanged among multiple vehicles, sensors and operators, and their roles, relative positions, and dependencies of those vehicles and systems change during operations. These capabilities are essential for operations where the temporal and spatial coordination of vehicles is required, such as in environmental field studies. However, we are still far from realizing the potential of network vehicle systems. Consider the case of an environmental disaster spanning a wide geographical area. With the current technologies, tools and models, it is simply not possible to inter-operate vehicles, sensors and communication networks from different vendors/institutions: currently there are no standards for inter-operability.

Wireless sensor networks [57] are a major technological trend that is already impacting environmental field studies [58] [59]. The developments on miniaturization and power consumption will accelerate this trend towards massive deployments thus enabling studies with unprecedented spatial and temporal resolution. A promising technological push comes from the inter-operation of vehicle systems with sensor networks [60]. This combines the coverage of sensor networks with fixed nodes, with the level of adaptation and detailed resolution provided by sensors mounted on vehicles.

Researchers and technology developers are devoting significant efforts to the development of concepts of operation for network vehicle systems. Surprisingly, or not, the role of human operators is receiving significant attention in the development of concepts of operation for network vehicle systems. In fact, this is the reason why researchers and technology developers have introduced the concept of mixed initiative interactions where planning procedures and execution control must allow intervention by experienced human operators. In part this is because essential experience and operational insight of these operators cannot be reflected in mathematical models, so the operators must approve or

---

modify the plan and the execution [67] [142]. Also, it is impossible to design vehicle and team controllers that can respond satisfactorily to every possible contingency. In unforeseen situations, these controllers ask the human operators for direction.

The idea of a system of systems seems appropriate to capture the essential aspects of operation of network vehicle systems. The observation is that the components in the network are part of a system within which new properties arise, some of them planned, some of them emergent and of unplanned nature. In a system of systems, a significant part of the "system" is embodied not as physical devices, such as vehicles, sensors or communication networks, but as software applications, which may be mobile, in the sense of migrating from one computer to another one, as part of the evolution of the system. This poses challenges to robotics, control, computer and communication scientists. These challenges entail a shift in the focus of existing methodologies from prescribing and commanding the behaviour of isolated systems to prescribing and commanding the behaviour of networked systems. These advances can only be achieved by adopting an inherently interdisciplinary approach, bringing together researchers from traditionally separate communities to work on problems at the forefront of science and technology.

The USTL has a two-fold approach to these challenges: 1) a planning, command and control framework within which the interactions among heterogeneous vehicles, sensors and operators are standardized and mediated; and 2) a software tool set which implements the framework over inter-operated, and sometimes intermittent, communication networks. These are briefly described next.

Planning, command and control framework

The USTL has a layered approach to planning and execution control. This approach decomposes a complex design problem into a number of more manageable sub-problems that are addressed in separate layers, which can be verified in a modular fashion. This leads to the modular verification of the framework [147]. Using the concept of manoeuvre – a prototype of an action/motion description for a vehicle – as the atomic component of all execution concepts. We abstract each vehicle as a provider of manoeuvres and services. A simple protocol based on an abstract vehicle interface governs the interactions between the vehicle and an external controller. The external controller sends a manoeuvre command to the vehicle; the vehicle either accepts the command and executes the manoeuvre, or does not accept the command and sends an error message to the controller; if it accepts the command the vehicle sends a done message or an error message to the controller depending on whether the manoeuvre terminates successfully or fails. This protocol facilitates inter-operability with other platforms. Actually, the same protocol is used on-board each vehicle for autonomous execution control [145].

The control architecture consists of two main layers: multi-vehicle control and vehicle control. Each layer, in turn, is further decomposed into other layers. The vehicle control architecture is standard for all the vehicles. The multi-vehicle control structure is mission dependent. We use our vehicle abstractions in multi-vehicle controllers that may reside in some remote locations or in some other vehicles. This leads to different control configurations and strategies. The vehicle control architecture consists of four layers: low-level control, manoeuvre control, vehicle supervision and plan supervision. The *vehicle supervisor* controls all of the on-board activities and mediates the interactions between an external multi-vehicle controller or the internal mission supervisor and the manoeuvre controllers. This supervisor accepts manoeuvre commands (or commands to abort the current manoeuvre) and passes the manoeuvre parameters to the corresponding manoeuvre controller for execution, and signals back the completion or failure of the manoeuvre. The *plan supervisor* commands and controls the execution of the mission plan. It commands the vehicle supervisor to trigger the execution of a manoeuvre specification and waits for the acknowledgment of its completion, or for an error. When it receives the acknowledgement, the plan supervisor selects the next manoeuvre to be executed. The process is repeated until the plan is successfully terminated, or it fails. The plan also has provisions for mixed initiative control by allowing the operator to enable and disable some of the transitions.

The concept of manoeuvre plays a central role in this architecture: it facilitates the task of mission specification, since it is easily understood by a mission specialist; it is easily mapped onto self-contained controllers, since it encodes the control logic; and is a key element in modular design, since it defines clear interfaces to other control elements. We allow the operator to interact with the execution of some manoeuvres. There is a library of manoeuvres/manoeuvre controllers. Example manoeuvres include: Hover, FollowTrajectory, Surface, Goto, Rows and Tele-operation. The addition and deletion of manoeuvre to the library does not require changes to the control architecture [67].

Software tool set

Neptus/Seaware/DFO/Dune tool set, developed at USTL, to support the implementation of our planning, command and control framework.

Neptus is a distributed command, control, communications and intelligence framework for operations with networked vehicles, systems, and human operators [148] [68]. Neptus supports all the phases of a mission life cycle: world representation; planning; simulation; execution and post-mission analysis. Neptus supports concurrent operations: vehicles, operators, and operator consoles come and go; operators are able to plan and supervise missions concurrently. Additional consoles can be built and installed on the fly to display mission related data over a network. Neptus has a Console Builder (CB) application. This facilitates the addition of new vehicles with new sensor suites. Neptus implements a subset of the NATO standard STANAG 4586 (NSA 2007) for communications with unmanned air vehicles.

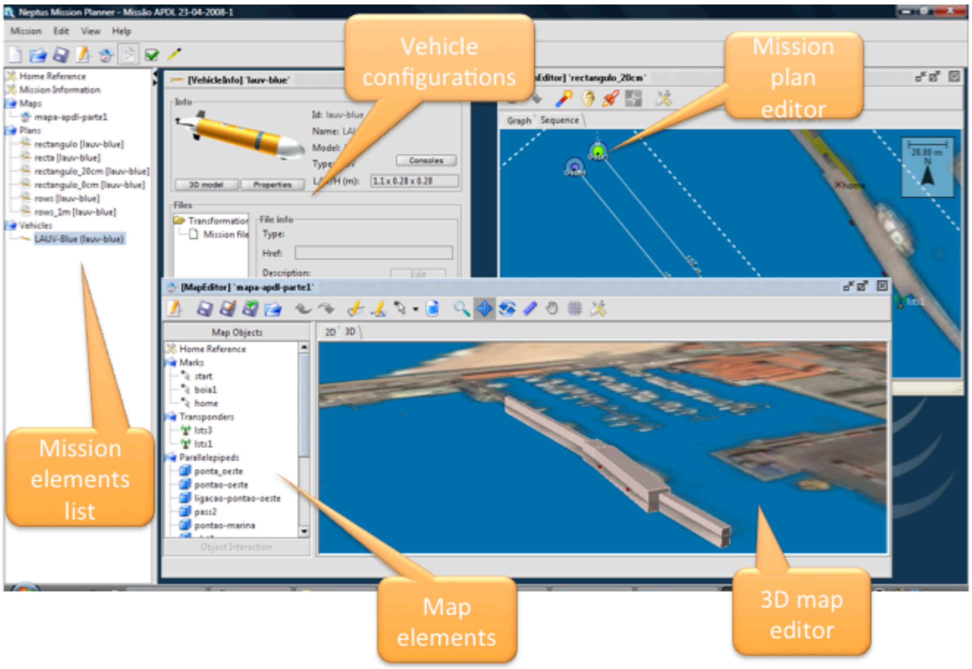


Figure 87: NEPTUS C4I

Seaware is a middleware framework that addresses the problem of communications in heterogeneous environments with diverse requirements [149]. Seaware adopts publish/subscribe based messaging, defined by anonymous message exchange between data subscriptions and publications, to provide an interface for applications to exchange data in a network through a set of transports, including Wi-Fi, RF and acoustic modems. Each application dynamically registers itself, specifying the topics it wishes to publish and subscribe, without the need to know in advance who its peers are or where they are located. There is a Seaware node per vehicle and per operator console (one per vehicle). Each vehicle node is characterized by a topic domain identifying the vehicle to allow for a set of messages to be exchanged with the corresponding operator console.

Dune supports the implementation of the vehicle control architecture in a predictable and efficient manner for real-time performance. At the core of Dune there is a platform abstraction layer, written in C++, enhancing portability among different CPU architectures (Intel x86 or compatible, Sun SPARC, Intel XScale/StrongARM and IBM PowerPC) and operating systems (Linux, Sun Solaris 10, Apple Mac OS X, FreeBSD, NetBSD, Microsoft

Windows 2000 or above and QNX 6.3). Dune can be extended in the native compiled programming language C++ or using an interpreted programming language such as Python or Lua.

Operations

Operational deployments are the opportunity to test and evaluate tools and technologies [149]. Figure 88 illustrates a possible operational scenario for a network vehicle system deployed in June 08. We deployed the LAUVs, Lusitânia, Isurus and Swordfish, in addition to a wireless sensor network consisting of Telos Motes. There were inter-operated over Wi-Fi and acoustic communications. There was one *Seaware* node per vehicle and per operator console (one per vehicle and one per sensor network). There was one operator in charge of the supervision of each vehicle. The operators subscribed to data provided by the wireless sensor networks. This was done transparently with the help of *Neptus*, which also helped with the visualisation of sensor measurements. The operator used *Neptus* for mission planning and evaluation prior to publishing the mission plan to the network. Each vehicle subscribed to commands sent by its operator (these included commands to load and execute mission plans). The operator also subscribed to data provided by the vehicle under his control. Moreover, at each console it was also possible to subscribe to data provided by other vehicles, or by the sensor network. This was done with the help of the *Neptus* visualization tools with layering capabilities. Coordination among vehicles was achieved through the coordination of mission plans (with the help of the operators). A web browser depicted the evolution of vehicles and of measured quantities. *Seaware* published real-time data to the Internet with the help of a GSM card.

Figure 88: Operational scenario

This operational scenario was used to understand the role of autonomous vehicles in emergency response to maritime incidents, and gain insights against which practitioners and technology developers can refine concepts of operation, and identify the main difficulties facing the operational deployment of autonomous vehicles in maritime incidents. The problem of integrating autonomous vehicles in the response plans to maritime incidents is discussed along with scenarios for their future utilization in [150]. Another example of a possible network centric operation would be the case of a maritime incident spanning a wide geographical area. With the current technologies, tools and models, it is simply not possible to bring together, in a systematic manner, vehicles sensors and communication networks from all over Europe to address this problem. In [151] we discuss what could be possibly done in a maritime incident with the tools and technologies from Porto University, considering for this purpose the case of an oil spill resulting from the collision of two ships in the Gulf of Biscay.

11.4. Future environmental studies

Networked vehicle systems have the potential to revolutionize environmental field studies. However, and despite the advances described in the previous sections, we are still far from being able to design and deploy network vehicle systems for environmental field studies in a systematic manner, and within an appropriate scientific framework. This requires a significant expansion of the basic tool sets from each area, and the introduction of new techniques that extend and complement the state of the art. Furthermore, these advances can only be achieved with an interdisciplinary approach, bringing together researchers from traditionally separate communities to work on problems at the forefront of science and technology. This is the reason why, in what follows, we only describe trends, without advocating specific concepts for environmental field studies.

Currently, there is a worldwide trend for the development of ocean observatories, like MARS-MBARI[19]; ESONET[20]; NEPTUNE[21] and others [152]. This is a good example of large-scale persistent data sampling, with adjustable sampling resolutions. Figure 4 depicts a simplistic illustration of an ocean observation system. It involves a wide range of mobile platforms including drifters, autonomous underwater vehicles and ships, fixed measurement assets such as moorings and radar, and remote measurements from satellites and aircraft.

---

[19] Monterey Bay Accelerated Research System (MARS) http://www.mbari.org/mars/. Accessed July 2010

[20] European Sea Floor Observatory Network (ESONET). http://www.oceanlab.abdn.ac.uk/research/esonet.php. Accessed July 2010

[21] Neptune Observatory. http://www.neptune.washington.edu/ Accessed July 2010

Moreover, the components of the ocean observatory system are reconfigurable to respond to observational opportunities and changing objectives.

Communications are a major challenge for ocean observatory systems. This is why these systems include intermittent inter-operated networks. Often deployed in mobile and extreme environments lacking continuous connectivity, many such networks have their own specialized protocols, and do not utilize the Internet Protocol IP. The DTN is one approach to address this problem[22]. The DTN is a network architecture and application interface structured around optionally-reliable asynchronous message forwarding, with limited expectations of end-to-end connectivity and node resources. For example, this enables vehicles to perform the role of data mules to move data between places that are not physically connected. Energy storage and transmission is another major challenge. Cabled observatories are been proposed to address this challenge. The cost of this approach is the motivation behind the development of other energy sources for ocean observatories.

In most concepts for ocean observatories sampling is achieved with the help of both fixed and mobile sensors. This aims at combining the best of both Eulerian and Lagrangian approaches to the problem of studying fluid properties [153]. The terms Eulerian and Lagrangian refer to the most common frames of reference used for studying these properties: Eulerian frames of reference are fixed in space and time; Lagrangian frames of reference move with the fluid. Moorings are the most common Eulerian platforms in oceanography. Unmanned underwater vehicles and drifters are Lagrangian platforms. This classification is not strict since moorings may move with the flow – in a limited fashion – and vehicles can move independently of the fluid flow.

The experience gained with experiments like the MB06[23] (Monterey Bay 2006) may help the community to understand the operation of ocean observatories. MB 06 took place over a two-month period from mid-July through mid-September 2006, and involved over a dozen different institutions, thirteen research vessels, over three dozen robot submarines, and many other fixed and drifting oceanographic instruments. The uneven seafloor and constantly changing currents in the Monterey Bay explain the scale of the experiments. These experiments examined coastal ocean processes from different perspectives, and at unprecedented different physical scales. These took place on a 24/7 base. The Collaborative Ocean Observatory Portal was developed to support the day-to-day participation of the large group of researchers with ties to geographically diverse institutions throughout North

---

[22] Delay-Tolerant Networking Research Group (DTNRG). http://www.ietf.org/proceedings/05mar/DTNRG.html. Accessed July 2010

[23] MBARI Monterey Bay 2006 Field Experiment MB06. http://www.mbari.org/mb2006/ (last visited January 2008. Accessed July 2010

America[24]. These investigators had to interact on a continuous basis to optimize data collection and analysis [154].

Persistent large-scale observation is not specific to the oceans. The oceans represent an extreme environment where technical challenges are exacerbated (e.g. GPS and radio communications do not work underwater), thus providing guidelines for deployments in other environments. In some environments, the deployment of new sensors and vehicles will complement existing sensing systems. For example, in most cities and coastal areas we can find environmental sensors, belonging to different organizations, which have been collecting data for years. This has a significant potential for environmental field studies. However, several difficulties must be faced before this data can be used. First, data has to be available. Second, it must be available on the right formats. Third, it has to be reliable (sensors have to be calibrated). These are not the only difficulties facing the networking of existing sensors. Security, levels of access, availability on a need to know basis, and models of operation represent other difficulties. This means that, in addition to the technical difficulties, there are some organizational, cultural and political difficulties. The technical difficulties are not insurmountable, and the cost of networking does not seem to be a major issue.

Networking existing sensors has the potential to add value to the existing infrastructure. This value can be further increased with network vehicle systems. Cities are one example where this idea can be easily applied. Different institutions (high-schools, universities, companies, municipalities, etc…) have been using environmental sensors on their daily activities. The Internet is now pervasive, and connecting these sensors to the Internet is not a major technical problem. In fact, permanent connectivity is not needed. City transportation vehicles can be instrumented with sensors for area coverage. The DTN technology allows the data collected along each route to be automatically stored on each vehicle and later forwarded to some Internet server at specific locations where short range (i.e. low cost) communications are available. Citizens can also contribute sensor measurements from either their mobile phones or from sensors connected to their home computers. This may lead to a sensing system of unprecedented dimensions and capability, which has applications not only in environmental field studies, but also on civil protection and on improving the quality of city life. The new sensing system will have certainly new properties, which cannot be fully anticipated now. This model can be easily replicated; it may be a first step towards the instrumentation of the Earth.

---

[24] Collaborative Ocean Observatory Portal. http://aosn.mbari.org/coop/. Accessed January 2008

Challenges

There are several obstacles in the road to the practical – as opposed to experimental – deployment of network vehicle systems. These are briefly discussed next.

Currently, there are no *legal frameworks* to encompass the operation of unmanned vehicles. In most countries the operation of air vehicles in controlled air space is severely restricted. Efforts are underway to address this problem in some European countries and in the United States. The operation of unmanned ocean going vehicles also presents legal challenges. Each deployment is the exception, and not the rule.

The lack of *standards for inter-operability* is preventing researchers to operate, in a transparent manner, vehicles from different vendors in a networked environment. The lack of standards is not unique to inter-operability. Currently there is no standardization in the area of underwater communications, to name just one example. There are several initiatives addressing these issues. NATO has been working on standards for inter-operability, namely the Stanag 4586 (NSA 2007) which has seen some acceptance in the UAV community – this is confirmed by the existence of commercial software products compliant with this standard. The Joint Architecture for Unmanned Systems (JAUS[25]) is receiving wide acceptance in the military, especially across the Atlantic in the United States. The NATO Undersea Research Center in La Spezia is developing the JANUS standard that will allow acoustic modems to co-exist, advertise their presence and potentially interoperate. A word of caution is needed here: the existence of standards does not imply standardization.

In general, commercial vehicles have not been developed as open systems. Moreover, the lack of standards for inter-operability is not conducive to open systems. Closed systems tend to raise vehicle and maintenance costs, and may be conducive to forms of market practice that are not necessarily in the benefit of the customer. This is especially critical in a field where technological obsolescence arises rapidly: vehicles and their components have to be upgraded periodically. The technological trends, namely those related to miniaturization and embedded systems, may contribute to change this state of affairs by contributing to the reduction of cost. The co-evolution of the Internet and of the personal computer changed dramatically the way society operates. Low cost open systems may prove fundamental to the dissemination of network vehicle systems.

This state of affairs should not prevent us from deploying unmanned vehicle systems. On the contrary, we are learning important lessons from our deployments [149]. These may

---

[25] http://www.jauswg.org. Accessed September 2007

prove invaluable for the development of legal frameworks, standards and concepts of operation.

11.5. Final remarks

The purpose of this chapter is to set the context of the development and deployment of network vehicle systems for environmental field studies over the next decades. The approach used to accomplish this goal was to present current developments in unmanned vehicle systems and in network vehicle systems before examining future trends and challenges for environmental field studies. Examples of developments from the Underwater Systems and Technologies Laboratory from Porto University illustrated the key points.

11.6. Acknowledgements

This chapter summarises the work conducted in the USTL overall several years and projects. It is based on the documentation from the projects and includes contributions from present and past researchers from the laboratory.

12. Conclusion and future work

This chapter summarises the main achievements presented in this thesis alongside with a highlight of the main contributions. A roadmap for future research is also discussed.

12.1. Accomplishments

This thesis presents a systems thinking approach for the definition of an event driven framework to enhance life sustainability in System of systems. Systems thinking combines analytical thinking with synthesis, starts by listing as many elements as possible (analysis) followed by the identification of repeating pattern across those elements (synthesis).

Starting with two observations as simple as "*The Times They Are a-Chagin'*" and "*Our world is a complex system of systems*" the argument that an extension to existing models and tools to deal with systems composed of interconnected elements capable of adapting themselves to an ever-changing environment are required is built. Building from an analysis of three different case studies coming from different domains – business, manufacturing and robotics – all possible elements that characterise these *system of systems* and what is required from them to adapt to *changes* in environment and in purposes are listed. Following the pre-selected methodology, the next step identified the repeating patterns and this lead to the identification of the main concepts that were missing in current discrete event systems theory: *system of systems* (and associated life cycle sustainability), *play* and *playbook*, and *changeability*.

Changeability is defined as the capability to accomplish early and efficient adjustment of the structures and processes at different levels of the system of systems. This requires not only the capability to foresee the need for change (and the required class of change needed) and models to defined the needed re-configurations in the system, but also highly flexible, intelligent and self-adaptive components, which continuously react to change, can be rapidly and smoothly brought into operation, and can enhance its utility throughout its life cycle, contributing this way to enhance the sustainability of the overall system.

Once these concepts were defined, the next task has been to include (formalise) them in the scope of the discrete event system theory. At this stage the event driven framework for changeability has been defined. This framework was then applied in two case studies to demonstrate its potential.

The application of the event driven framework for changeability started with the definition of the context of application. As previously mentioned, changeability requires systems composed of interacting smart components.

The selected industrial case study has been defined in the scope of two European projects – I-RAMP[3] and ReBORN – which are working in concepts related with plug'n'produce and smart components for manufacturing systems, involving variability in the production demand, fast ramp up times and re-use of production equipment. The event driven framework was applied in a case study that involves the design of a production line, involving new and re-used equipment, and the exchange of equipment in the production line during operation.

An additional case study, selected from the robotics domain, was used to further demonstrate the applicability of the event driven framework for changeability. The event driven framework was applied in a case study that involves the design of an ocean observatory, involving persistent operations in wide areas executed by teams of autonomous vehicles.

These case studies made possible to demonstrate the adequacy of the framework to the manufacturing and robotics domains and to the defined contexts. It was also possible to demonstrate that the framework can be applied in different phase of the life cycle and to realise the importance of *evolution* in the scope of such a framework.

12.2. Main contributions

This work has its focus on the framework of discrete event systems and follows a systems thinking approach to understand how current challenges require this framework to be extended. The main purpose of the identified extensions is to make the discrete event systems framework adequate to support system of systems in many domains, notably in the industrial, business and robotics domains. The main scientific research objectives achieved during the course of the this work have been the following:

- Cases where system of systems thinking is necessary in the three aforementioned domains were identified and described.
- The identified cases were used to synthesise a definition of system of systems amenable to be treated inside the discrete event systems framework.
- A set of issues (common to the three domains) that need the discrete event systems framework to be extended in order to be addressed were identified, notably *changeability*.
- The first steps towards the definition of an event driven framework for changeability, contributing to extend current discrete event systems framework, were defined, contributing to the enhance life cycle sustainability in system of systems.
- The applicability of these results was demonstrated in two cased studies: one from the industrial domain, applied in a case study defined within the scope of two European projects, and another from the robotics domain, applied within the scope of an ocean observatory based on multiple autonomous systems.

The main results of the work presented in this thesis might contribute to extend current discrete event systems theory and existing frameworks, by laying down the foundations for including in the theory support for the concepts of *System of Systems*, *Changeability* and *Life Cycle*. The main results of the work are the following:

- Support for System of Systems modelling in the discrete event systems framework.
- Formal definition of System of Systems, play and playbook.
- Contribution to the extension of the current framework with the Event driven changeability framework.
- Application and demonstration of the framework in two scenarios (industrial and robotics).

12.3. Future work

No research work is ever closed or 100% complete and this is no exception. The focus of the effort was on the synthesis of the building concepts for the changeability framework. This effort was anchored on case studies from the three selected application domains and on the analysis of current frameworks for discrete event systems.

During the course of this research work it was possible to define several concepts that might help to extend the existing theory. Besides further applications and validation of the proposed framework, it was also possible to identify several lines of research to pursue.

Theoretical results

The changeability framework needs to be further developed and some theoretical results from the discrete event system framework need to be further explored and extended for sustainability. For example, concepts like controllability and coobservability need to be formalised in the framework.

Line of Research #1: further extend theoretical concepts from the discrete event systems framework.

NETDEVs and VERSONs

The industrial application context introduced in chapter 9 builds on ideas, challenges and concepts introduced by the two European projects I-RAMP3 and ReBORN. Although chapter 10 already applies the results developed by this research work to an industrial scenario, this application does not give a full coverage of challenges and concepts presented in chapter 9.

Line of Research #2: formalise the concepts introduced by I-RAMP3 and ReBORN (and possibly other similar projects) using the proposed framework.

Playbook changeability

The playbook changeability was introduced and it is clear that the dynamics of the playbook is a "must have" requirement. Although the principles for the dynamics (or evolution) were discussed, its study and implementation were out of scope.

Online or Recursive learning, based on case studies or experimental data, to validate plays or develop new plays when new information is available would allow for the adaptation of the playbook over time to improve its accuracy in terms of efficiency. This will allow to account not only for changes in the operational needs, which occur over time, but will also accommodate for changes in the performance and response of the systems themselves. For example, industrial equipment operating today will have a different performance 2/3 years from now, due to components wear or upgrades.

Line of Research #3: build on results from data analytics, data mining and machine learning to further detail the playbook changeability concept and to include the concept of evolution in the framework.

Smart components and manufacturing systems

Smart components can be described as equipment or machines devices with communications capabilities, which incorporate functions of sensing and control in order to analyse a situation, make decisions based on the available data and modify their behaviour through feedback. The future of manufacturing will involve connecting these machines in order to create intelligent networks that communicate and control each other with reduced intervention required by human operators.

Line of Research #4: enhance the definition and implementation of smart components in industrial applications.

Cyberphysical systems

Coupling of the cyber and physical in manufacturing opens up many possibilities to improve the value, including productivity and sustainability, of manufacturing systems. This will allow for an increase in factory wide visibility, network automation, better energy management, proactive maintenance and connected supply chain.

Line of Research #5: validate the adequacy of the proposed framework for cyberphysical systems in general and cyberphysical systems in manufacturing in particular.

Modelling tools and case tools

There are a number of modelling and case tools, both commercial and open source, that support different frameworks, including discrete event systems.

Line of Research #6: implement the results into existing modelling and case tools (e.g. Modelio, Ptolemy, ..).

References

[1] J. A. Byrne. (1993, February 8, 1993) The virtual corporation: the company of the future will be the ultimate in adaptability. *International Business Week*. 36-40.

[2] P. M. Senge, *The Fifth Discipline: The art and practice of the learning organization*. New York: Doubleday, 1990.

[3] A. Toffler, *The Third Wave*: Morrow, 1980.

[4] M. G. Mehrabi, A. G. Ulsoy, Y. Koren, and P. Heytler, "Trends and perspectives in flexible and reconfigurable manufacturing systems," *Journal of Intelligent Manufacturing,* vol. 13, pp. 135-146, 2002/04/01 2002.

[5] H. Van Brussel, J. Wyns, P. Valckenaers, L. Bongaerts, and P. Peeters, "Reference architecture for holonic manufacturing systems: PROSA," *Computers in Industry,* vol. 37, pp. 255-274, 11// 1998.

[6] G. Morel, P. Valckenaers, J.-M. Faure, C. E. Pereira, and C. Diedrich, "Manufacturing plant control challenges and issues," *Control Engineering Practice,* vol. 15, pp. 1321-1331, 11// 2007.

[7] L. M. Sanchez and R. Nagi, "A review of agile manufacturing systems," *International Journal of Production Research,* vol. 39, pp. 3561-3600, 2001/01/01 2001.

[8] H. Yang and D. Xue, "Recent research on developing Web-based manufacturing systems: a review," *International Journal of Production Research,* vol. 41, pp. 3601-3629, 2003/01/01 2003.

[9] P. Korsten and C. Seider, "The world's 4 trillion dollar challenge: Using a system-of-systems approach to build a smarter planet," IBM Institute for Business Value, IBM Global Business Service Partner2010.

[10] O. G. Standard, "TOGAF Version 9.1," ed: The Open Group, 2011.

[11] D. o. Defense, "DoD Architecture Framework 2.02," ed: DoD, 2010.

[12] M. Fowler, *UML Distilled: A Brief Guide to the Standard Object Modeling Language* 3rd ed.: Addison-Wesley Professional, 2003.

[13] T. Weilkiens, *System Engineering with SysML/UML*: Morgan Kaufmann OMG Press, 2006.

[14] J. Barton and T. Haslett, "Analysis, synthesis, systems thinking and the scientific method: rediscovering the importance of open systems," *Systems Research & Behavioral Science,* vol. 24, pp. 143-155, 2007.

[15] C. C. Snow, R. E. Miles, and H. J. Coleman, Jr., "Managing 21st century network organizations," *Organizational Dynamics,* pp. 5-20, 1992.

[16] R. H. Coase, *The Firm, the Market, and the Law*: University of Chicago Press, 1988.

[17] A. Koestler, *The Ghost in the Machine*: Hutchinson, 1967.

[18] N. A. C. Mat, C. Yen, and H. Scheepers, "A framework for partner selection criteria in virtual enterprises for SMEs," in *Service Systems and Service Management, 2008 International Conference on*, 2008, pp. 1-7.

[19] T.-L. Tseng, C.-C. Huang, H.-W. Chu, and R. R. Gung, "Novel approach to multi-functional project team formation," *International Journal of Project Management,* vol. 22, pp. 147-159, 2// 2004.

[20] F. S. Hillier and G. J. Lieberman, *Introduction to Operations Research*, Eighth ed.: McGraw-Hill Primis, 2008.

[21] A. Zakarian and A. Kusiak, "Forming teams: an analytical approach," *IIE Transactions,* vol. 31, pp. 85-97, 1999/01/01 1999.

[22] J. Hauser and D. Clausing, "House of Quality," *Harvard Business Review,* May 1988.

[23] E. H. Forman and S. I. Gass, "The analytical hierarchy process—an exposition," *Operations Research,* vol. 49, pp. 469–487, 2011.

[24] T. J. Ross, *Fuzzy logic with engineering applications*: McGraw-Hill, 1995.

[25] Y. A. Hajidimitriou and A. C. Georgiou, "A goal programming model for partner selection decisions in international joint ventures," *European Journal of Operational Research,* vol. 138, pp. 649--662, 2002.

[26] J. P. Ignizio, *Introduction to linear goal programming*: Sage Publications, 1985.

[27] S. Talluri, R. C. Baker, and J. Sarkis, "A framework for designing efficient value chain networks," *International Journal of Production Economics,* vol. 62, pp. 133--144, 1999.

[28] A. Charnes, W. W. Cooper, and E. Rhodes, "Measuring the efficiency of decision making units," *European Journal of Operational Research* vol. 6, p. 429.444, 1978.

[29] D. Bertsimas and J. N. Tsitsiklis, *Introduction to linear optimization*: Athena Scientific, 1997.

[30] H. Huang and J. Chen, "A partner selection method based on risk evaluation in virtual enterprises," in *Services Systems and Services Management, 2005. Proceedings of ICSSSM '05. 2005 International Conference on*, 2005, pp. 608-612 Vol. 1.

[31] B. v. Dijk, "Case-Studies of High-Impact Projects in the ICT Area," Technomar GmbH 2002.

[32] E. H. Wagner, S. M. Greene, G. Hart, T. S. Field, S. Fletcher, A. M. Geiger*, et al.*, "Building a Research Consortium of Large Health Systems: The Cancer Research Network," *Journal of the National Cancer Institute Monographs,* 2005.

[33] A. Buchholz, S. Sandhu, T. Kupczyk, and A. Zerfass, "Anticipating Content Technologies Needs: Business Monitor & Know-How Transfer," Germany2004.

[34] G. Gonçalves, "Support fools for collaboration networks in R&D and Innovation projects - Final Technical Report," University of Porto, Faculty of Engineering, Internal Report2008.

[35] J. Von Neuman and Morgenstern, *Theory of Games and Economic Behavior*, 6th ed.: Princeton University Press, 1990.

[36] M. Eirinaki, M. D. Louta, and I. Varlamis, "A Trust-Aware System for Personalized User Recommendations in Social Networks," *Systems, Man, and Cybernetics: Systems, IEEE Transactions on,* vol. PP, pp. 1-1, 2013.

[37] J. S. Armstrong, *Principles of forecasting: a handbook for researchers and practitioners*. Massachusetts: Kluwer Academic Publishers, 2001.

[38] D. Simon, "Kalman filtering with state constraints: a survey of linear and nonlinear algorithms," *Control Theory & Applications, IET,* vol. 4, pp. 1303-1318, 2010.

[39] Y. Koren, U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy*, et al.*, "Reconfigurable Manufacturing Systems," *CIRP Annals - Manufacturing Technology,* vol. 48, pp. 527-540, // 1999.

[40] D. McFadzean, D. Stewart, and L. Tesfatsion, "A computational laboratory for evolutionary trade networks," *Evolutionary Computation, IEEE Transactions on,* vol. 5, pp. 546-560, 2001.

[41] N. Gilbert, A. Pyka, and P. Ahrweiler, "Innovation Networks – a simulation approach," *Journal of Artificial Societies and Social Simulation,* vol. 4, 2001.

[42] W. Shen, Q. Hao, H. Yoon, and D. Norrie, "Applications of agent-based systems in intelligent manufacturing: an update review," *Advanced Engineering Informatics,* vol. 20, pp. 415-431, 2006.

[43] F. Almeida, P. Dias, G. Gonçalves, M. Peschl, and M. Hoffmeister, "A proposition of a manufactronic network approach for intelligent and flexible manufacturing systems," *International Journal of Industrial Engineering Computations,* vol. 2, pp. 873-890, 2011.

[44] N. A. Duffie and R. S. Piper, "Non-hierarchical control of a flexible manufacturing cell," *Robotics and Computer-Integrated Manufacturing,* vol. 3, pp. 175-179, 1987.

[45] H. V. D. Parunak, J. F. White, P. W. Lozo, R. Judd, B. W. Irish, and J. Kindrick, "An architecture for heuristic factory control," presented at the American Control Conference, Seattle, WA, USA, 1986.

[46] H. V. D. Parunak, A. D. Baker, and S. J. Clark, "The AARIA Agent Architecture: From Manufacturing Requirements to Agent-Based System Design," presented at the

International Conference on Autonomous Agents (ICAA'98), Minneapolis, MN, USA, 1998.

[47] G. Gonçalves, J. Sousa, F. Pereira, P. Dias, and A. Santos, "A framework for e-cooperation business agents: An application to the (re)engineering of production facilities," presented at the International Conference on Advanced Production Management Systems, 2002.

[48] S. Bussmann and K. Schild, "An Agent-Based Approach to the Control of Flexible Production Systems," presented at the 8th IEEE International Conference on Emergent Technologies and Factory Automation (ETFA 2001), Antibes Juan-les-pins, France, 2001.

[49] G. Goncalves, P. Dias, A. Santos, J. Sousa, and F. Pereira, "An implementation of a framework for cooperative engineering," presented at the Proceedings of the 16th IFAC World Congress, Czech Republic, 2005.

[50] H. Dyke Parunak, "What can agents do in industry, and why? An overview of industrially-oriented R&D at CEC," in *Cooperative Information Agents II Learning, Mobility and Electronic Commerce for Information Discovery on the Internet.* vol. 1435, M. Klusch and G. Weiß, Eds., ed: Springer Berlin Heidelberg, 1998, pp. 1-18.

[51] B. Saint Germain, P. Valckenaers, P. Verstraete, Hadeli, and H. Van Brussel, "A multi-agent supply network control framework," *Control Engineering Practice,* vol. 15, pp. 1394-1402, 11// 2007.

[52] V. Mařík and J. Lažanský, "Industrial applications of agent technologies," *Control Engineering Practice,* vol. 15, pp. 1364-1380, 11// 2007.

[53] M. Hoffmeister and R. Wertz, "Adaptive manufacturing by encapsulation of functionality and expertise," *unpublished.*

[54] M. Peschl, N. Link, M. Hoffmeister, G. Gonçalves, and F. Almeida, "Designing and implementation of an intelligent manufacturing system," *Journal of Industrial Engineering and Management,* vol. 4, pp. 718-745, 2011.

[55] G. A. Goncalves and J. B. Sousa, "Quality assurance and validation issues in the development of networked vehicle systems," in *Automation Quality and Testing Robotics (AQTR), 2010 IEEE International Conference on*, 2010, pp. 1-7.

[56] A. Girard, J. Sousa, and K. Hedrick, "A Selection of Recent Advances in Networked Multi-Vehicle Systems," presented at the Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering
, 2005.

[57] D. Culler and H. Mulder, "Smart Sensors to Network the World," *Scientific American,* vol. 2906, pp. 84-9, 2004.

[58] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," presented at the Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, Atlanta, Georgia, USA, 2002.

[59] M. Welsh, G. Werner-Allen, K. Lorincz, O. Marcillo, J. Johnson, M. Ruiz*, et al.*, "Sensor networks for high-resolution monitoring of volcanic activity," presented at the

Proceedings of the twentieth ACM symposium on Operating systems principles, Brighton, United Kingdom, 2005.

[60] B. Sinopoli, C. Sharp, L. Schenato, S. Schaffert, and S. S. Sastry, "Distributed control applications within sensor networks," *Proceedings of the IEEE,* vol. 91, pp. 1235-1246, 2003.

[61] J. Borges de Sousa and G. Gonçalves, "Mixed initiative control of unmanned air and ocean going vehicles models, tools and experimentation," [Neuilly-sur-Seine Cedex, France].

[62] R. W. Proud, J. J. Hart, and R. B. Mrozinski, "Methods for determining the level of autonomy to design into a human spaceflight vehicle: a function specific approach," DTIC Document2003.

[63] T. B. Sheridan, *Telerobotics, automation, and human supervisory control.* Cambridge, MA, USA MIT Press, 1992.

[64] F. Osinga, *Science, Strategy and War: The Strategic Theory of John Boyd*: Routledge, 2006.

[65] W. E. Deming, *Out of the crisis*: MIT Press, 1986.

[66] J. Jariyasunant, E. Pereira, M. Zennaro, K. Hedrick, C. Kirsch, and R. Sengupta, "CSL: A Language to Specify and Re-Specify Mobile Sensor Network Behaviors," in *Real-Time and Embedded Technology and Applications 2009*, 2009.

[67] J. Sousa, T. Simsek, and P. Varaya, "Task planning and execution for UAV teams," in *Control and Decision Conference 2004*, 2004.

[68] P. S. Dias, J. Pinto, R. Gongalves, G. Gonçalves, J. B. Sousa, and F. L. Pereira, "Neptus, Command and Control Infrastructure for Heterogeneous Teams of Autonomous Vehicles," in *Robotics and Automation, 2007 IEEE International Conference on*, 2007, pp. 2768-2769.

[69] E. Pereira, R. Bencatel, J. Correia, G. Gonçalves, L. Félix, J. Morgado*, et al.*, "Unmanned Air Vehicles for coastal and environmental research," presented at the 10th International Coastal Symposium (ICS 2009), 2009.

[70] R. Gonçalves, S. Ferreira, J. Pinto, J. Sousa, and G. Gonçalves, "Authority sharing in mixed initiative control of multiple uninhabited aerial vehicles," in *9th international conference on Engineering psychology and cognitive ergonomics (EPCE'11)*, 2011, pp. 530-539.

[71] C. Fuchs, S. Ferreira, J. Sousa, and G. Gonçalves, "Adaptive Consoles for Supervisory Control of Multiple Unmanned Aerial Vehicles," in *Human-Computer Interaction. Interaction Modalities and Techniques*. vol. 8007, M. Kurosu, Ed., ed: Springer Berlin Heidelberg, 2013, pp. 678-687.

[72] G. Gonçalves, R. Gonçalves, S. Ferreira, J. Pinto, and J. Sousa, "Supervisory Control: Optimal Distribution of Workload Among Operators for Mixed Initiative Control of Multiple UAVs," in *Supervisory Control of Multiple Uninhabited Systems - Methodologies and Enabling Human-Robot Interface Technologies*. vol. RTO-TR-HFM-170 AC/323(HFM-170)TP/451, ed: NATO Science and Technology Organization, 2012.

[73] C. A. Miller, H. B. Funk, M. Dorneich, and S. D. Whitlow, "A playbook interface for mixed initiative control of multiple unmanned vehicle teams," in *Digital Avionics Systems Conference, 2002. Proceedings. The 21st*, 2002, pp. 7E4-1-7E4-13 vol.2.

[74] F. M. Callier and C. A. Desoer, *Linear System Theory*: Springer New York, 1991.

[75] "IEEE Standard for Application and Management of the Systems Engineering Process," ed, 1998.

[76] K. Ogata, *State Space Analysis of Control Systems*. Englewood Clifs: Prentice-Hall, 1967.

[77] S. P. Banks, *Control Systems Engineering*. Englewood Clifs: Prentice-Hall, 1986.

[78] C. Cassandras and P. J. G. Ramadge, "Special Edition on Discrete Event Systems," *Control Systems Magazine,* vol. 10, pp. 66-112, 1990.

[79] C. Cassandras and S. Lafortune, "Discrete Event Systems: The State of the Art and New Directions," in *Applied and Computational Control, Signals, and Circuits*. vol. I, B. Datta, Ed., ed Boston: Birkhäuser 1999.

[80] C. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, 2nd ed. ed.: Springer, 2008.

[81] R. L. Ackoff, *Creating the Corporate Future*. New York: Wiley, 1981.

[82] D. K. Hitchins, *Systems Engineering: a 21st century systems methodology*. England: John Wiley & Sons, 2007.

[83] B. Richmond, *An introduction to Systems Thinking*: ISEE Systems, 2004.

[84] J. E. Kasser, *A framework for understanding Systems Engineering*. England: The Right Requirements, 2007.

[85] J. Messina, E. Simmon, and M. Aronoff, "Environmental Regulations Impose New Product Lifecycle Information
Requirements," in *Complex Systems Concurrent Engineering: Collaboration, Technology Innovation and Sustainability*, G. Loureiro and R. Curran, Eds., ed: Springer, 2007.

[86] J. Blewitt, *Understanding Sustainable Development*: Earthscan, 2008.

[87] F. Jovane, E. Westkämper, and D. Williams, "Towards Competitive Sustainable Manufacturing," in *The ManuFuture Road*, ed: Springer Berlin Heidelberg, 2009, pp. 31-52.

[88] R. Jardim-Goncalves, C. Agostinho, and A. Steiger-Garcao, "A reference model for sustainable interoperability in networked enterprises: towards the foundation of EI science base," *International Journal of Computer Integrated Manufacturing,* vol. 25, pp. 855-873, 2012.

[89] R. Jardim-Goncalves, C. Agostinho, and A. Steiger-Garcao, "Sustainbable System Interoperability: a reference model for seamless networked business," presented at the International IEEE conference on Systems, Man and Cybernetics, 2010.

[90] M. Henrie and E. E. Delaney, "Towards a common system of systems vocabulary," in *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, 2005, pp. 2732-2737 Vol. 3.

[91] J. J. Simpson and C. H. Dagli, "System of systems: Power and paradox," in *System of Systems Engineering, 2008. SoSE '08. IEEE International Conference on*, 2008, pp. 1-5.

[92] J. J. Simpson and M. J. Simpson, "System of systems complexity identification and control," in *System of Systems Engineering, 2009. SoSE 2009. IEEE International Conference on*, 2009, pp. 1-6.

[93] N. Karcanias and A. G. Hessami, "Complexity and the notion of system of systems: part (I): general systems and complexity," presented at the World Automation Congress (WAC), 2010, 2010.

[94] S. J. Luskasik, "systems, system of systems, and the education of engineers," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing,* vol. 14, pp. 461-479, 1998.

[95] N. Karcanias and A. G. Hessami, "Complexity and the notion of system of systems: Part (II): defining the notion of system of systems," presented at the World Automation Congress (WAC), 2010, 2010.

[96] M. Jamshidi, "Introduction to System of Systems," in *System of Systems Engineering: Innovations for the Twenty-First Century*, M. Jamshidi, Ed., ed: John Wiley & Sons, 2008.

[97] N. Karcanias and A. G. Hessami, "System of Systems and Emergence Part 1: Principles and Framework," presented at the Emerging Trends in Engineering and Technology (ICETET), 2011 4th International Conference on, 2011.

[98] T. Tolio and A. Valente, "An Approach to Design the Flexibility Degree in Flexible Manufacturing Systems," presented at the 16th Int. Conf. on Flexible Automation and Intelligent Manufacturing, University of Limerik, Ireland, 2006.

[99] R. M. Setchi and N. Lagos, "Reconfigurability and reconfigurable manufacturing systems: state-of-the-art review," in *Industrial Informatics, 2004. INDIN '04. 2004 2nd IEEE International Conference on*, 2004, pp. 529-535.

[100] C. Löffler, E. Westkämper, and K. Unger, "Changeability in Structure Planning of Automotive Manufacturing," *Procedia CIRP,* vol. 3, pp. 167-172, // 2012.

[101] Seleim, A. Azab, and T. AlGeddawy, "Simulation Methods for Changeable Manufacturing," *Procedia CIRP,* vol. 3, pp. 179-184, // 2012.

[102] G. Chryssolouris, *Manufacturing Systems: Theory and Practice*: Springer New York, 2006.

[103] M. Schenk, S. Wirth, and E. Müller, *Factory Planning Manual: Situation-Driven Production Facility Planning*. Berlin: Springer-Verlag, 2010.

[104] G. Schuh, J. Arnoscht, and M. Völker, "Product Design Leverage on the Changeability of Production Systems," *Procedia CIRP,* vol. 3, pp. 305-310, // 2012.

[105] H.-P. Wiendahl and R. Hernández, "The Transformable Factory - Strategies, Methods and Examples," presented at the 1st International Conference on Agile, Reconfigurable Manufacturing, Ann Arbor, USA, 2001.

[106] Manufuture, "Manufuture Strategic Research Agenda," ed, 2006.

[107] A. D. Baker, "A survey of factory control algorithms that can be implemented in amulti-agent heterarchy: dispatching scheduling and pull.," *Journal of Manufacturing Systems,* vol. 4, pp. 297–320, 1998.

[108] D. M. Dilts, N. P. Boyd, and H. H. Whorms, "The evolution of control architectures for automated manufacturing systems," *Journal of Manufacturing Systems,* vol. 10, pp. 79-93, // 1991.

[109] N. A. Duffie and V. Prabhu, "Heterarchical control of highly distributed manufacturing systems," *International Journal of Computer Integrated Manufacturing,* vol. 4, pp. 270–281, 1996.

[110] W. S. McCulloch, "A heterarchy of values determined by the topology of nervous nets," *Bulletin of Mathematical Biophysics,* pp. 89–93, 1945.

[111] A. M. Ross, D. H. Rhodes, and D. E. Hastings, "Defining changeability: Reconciling flexibility, adaptability, scalability, modifiability, and robustness for maintaining system lifecycle value," *Systems Engineering,* vol. 11, pp. 246-262, 2008.

[112] H. A. ElMaraghy and H.-P. Wiendahl, "Changeability – An Introduction," in *Changeable and Reconfigurable Manufacturing Systems*, E. HA, Ed., ed Berlin: Springer, 2009, pp. pp. 3–24.

[113] H.-P. Wiendahl, H. A. ElMaraghy, P. Nyhuis, M. F. Zäh, H.-H. Wiendahl, N. Duffie*, et al.*, "Changeable Manufacturing - Classification, Design and Operation," *CIRP Annals - Manufacturing Technology,* vol. 56, pp. 783-809, 2007.

[114] H. A. ElMaraghy, "Flexible and Reconfigurable Manufacturing Systems Paradigms," *International Journal of Flexible Manufacturing Systems. Special Issue: Reconfigurable Manufacturing Systems,,* vol. 17, pp. 261-276, 2005.

[115] A. M. Ross and D. E. Hastings, "Assessing Changeability in Aerospace Systems Architecting and Design Using Dynamic Multi-Attribute Tradespace Exploration," presented at the American Institute of Aeronautics and Astronautics Space 2006, San Jose, CA, 2006.

[116] J. L. Peterson, *Petri Net Theory and the Modeling of Systems* Prentice Hall, 1981.

[117] J. Baeten, T. Basten, and M. Reniers, *Process Algebra: Equational Theories of Communicating Processes* Cambridge University Press, 2009.

[118] R. Milner, *Communicating and Mobile Systems: The Pi Calculus*: Cambridge University Press, 1999.

[119] J. E. Hopcroft and J. D. Ullman, *Introduction to Automata Theory, Languages and Computation* Addison-Wesley, 1979.

[120] P. J. G. Ramadge and W. M. Wonham, "The control of discrete event systems," *Proceedings of the IEEE,* vol. 77, pp. 81 - 98, January 1989.

[121] W. M. Wonham and P. J. Ramadge, "Modular supervisory control of discreteevent systems," *Mathematics of Control, Signals, and Systems,* vol. 1, pp. 13–30, 1988.

[122] K. Rudie and W. M. Wonham, "Think Globally, Act Locally: Decentralized Supervisory Control," *IEEE_J_AC,* vol. 37, pp. 1692 -1708, November 1992.

[123] K. Rudie, S. LaFortune, and F. Lin, "Minimal communication in a distributed discrete-event system," *Automatic Control, IEEE Transactions on,* vol. 48, pp. 957 - 975, June 2003.

[124] C. Miller, R. Goldman, H. Funk, P. Wu, and B. Pate, "A Playbook Approach to Variable Autonomy Control: Application for Control of Multiple, Heterogeneous Unmanned Air Vehicles," presented at the FORUM 60, the Annual Meeting of the American Helicopter Society, Baltimore, 2004.

[125] T. Sheridan, "Supervisory control," in *Handbook of human factors*, G. Salvendy, Ed., ed New York: John Wiley & Sons, 1987, pp. 1244-1268.

[126] R. Parasuraman and V. Riley, "Humans and automation: Use, misuse, disuse, abuse," *Human Factors,* vol. 39, pp. 230-253, 1997.

[127] I. Schwab and N. Link, "Reusable Knowledge from Symbolic Regression Classification.," presented at the Genetic and Evolutionary Computing, 2011.

[128] J. Lee, B. Bagheri, and H. Kao, "Recent Advances and Trends of Cyber-Physical Systems and Big Data Analytics in Industrial Informatics," in *Int. Conference on Industrial Informatics*, Brazil, 2014.

[129] A. Gupta, "Predictive modelling of turning operations using response surface methodology, artificial neural networks and support vector regression," *International Journal of Production Research,* vol. 48, pp. 763-778, 2010.

[130] M. Senn and N. Link, "Hidden State Observation for Adaptive Process Controls," presented at the Second International Conference on Adaptive and Self-adaptive Systems and Applications, 2010.

[131] M. Senn, J. Schäfer, J. Pollak, and N. Link, "A System-Oriented Approachfor the Optimal Control of Process Chains under Stochastic Influences," presented at the 9th International Conference of Numerical Analysis and Applied Mathematics, 2011.

[132] P. Leitão, "Agent-based distributed manufacturing control: A state-of-the-art survey," *Engineering Applications of Artificial Intelligence,* vol. 22, pp. 979-991, 10// 2009.

[133] C. Gerber, H. M. Hanisch, and S. Ebbinghaus, "From IEC 61131 to IEC 61499 for distributed systems: a case study," *EURASIP Journal of Embedded Systems,* 2008.

[134] J. Woerner, T. Laengle, and H. Woern, "Corporate Planning and Simulation of Plant Production Facilities in the Virtual World," *Proceedings of the 18th International Conference on CAD/CAM, Robotics and Factories of the Future,* pp. 109--116, 2002.

[135] K. Caskey, H. Rouibah, and H. J. Pels, "Managing Vendor Suplied Models of Production Facilities," in *E-Manufacturing: Business Paradigms and Supporting Technologies*, J. J. P. Ferreira, Ed., ed: Springer, 2004, pp. 221-228.

[136] M. Senn and N. Link, "A Universal Model for Hidden State Observation in Adaptive Process Controls," *International Journal On Advances in Intelligent Systems,* vol. 4, 2012.

[137] M. Deffenbaugh, H. Schmidt, and J. G. Bellingham, "Acoustic navigation for Arctic under-ice AUV missions," presented at the OCEANS '93, 1993.

[138] T. B. Curtin, J. G. Bellingham, J. Catipovic, and D. Webb, "Autonomous Ocean Sampling Networks," *Oceanography,* vol. 6, pp. 86-94, 1993.

[139] P. Varaiya, "Smart cars on smart roads: problems of control," *IEEE Transactions on Automatic Control,* vol. 38, pp. 195-207, 1993.

[140] M. Veloso, "Autonomous robot soccer teams," *The Bridge, National Academy of Engineering,* vol. 33, pp. 8-12., 2003.

[141] P. J. Antsaklis and K. M. Passino, *Introduction to intelligent control systems with high degrees of autonomy. An introduction to intelligent and autonomous control.* MA, USA: Kluwer Academic Publishers Norwell, 1993.

[142] J. B. Sousa and G. Gonçalves, "Mixed initiative control of unmanned air and ocean going vehicles: models, tools and experimentation," presented at the RTO AVT-146 Symposium - Platform Innovations and System Integration for Unmanned Air, Land and Sea Vehicles, 2007.

[143] J. G. Bellingham and K. Rajan, "Robotics in Remote and Hostile Environments," *Science,* vol. 318, pp. 1098-1102, 2007.

[144] T. B. Curtin, D. M. Crimmins, J. Curcio, M. Benjamin, and C. Roper, "Autonomous Underwater Vehicles: Trends and Transformations," *Marine Technology Society Journal,* vol. 39, pp. 65-75, 2005.

[145] P. Almeida, G. A. Gonçalves, and J. B. Sousa, "Multi-UAV Platform for Integration in Mixed-Initiative Coordinated Missions," presented at the 1st IFAC Workshop on Multi-vehicle Systems (MVS'06), 2006.

[146] A. Costa, M. Matos, and C. Silva, "Desenvolvimento de aeronaves não tripuladas na Força Aérea Portuguesa," *Mais Alto,* pp. 2-8, 2005.

[147] J. B. Sousa, K. H. Johansson, J. Silva, and A. Speranzon, "A verified hierarchical control architecture for coordinated multi-vehicle operations," *International Journal of Adaptive Control and Signal Processing,* vol. 21, pp. 159-188, 2006.

[148] P. S. Dias, S. L. Fraga, R. M. F. Gomes, G. M. Gonçalves, F. L. Pereira, J. Pinto*, et al.*, "Neptus - a framework to support multiple vehicle operation," in *Oceans 2005 - Europe*, 2005, pp. 963-968 Vol. 2.

[149] E. R. B. Marques, J. Pinto, S. Kragelund, P. S. Dias, L. Madureira, A. Sousa*, et al.*, "AUV Control and Communication using Underwater Acoustic Networks," in *OCEANS 2007 - Europe*, 2007, pp. 1-6.

[150] J. B. Sousa, G. Gonçalves, and F. L. Pereira, "Autonomous Vehicles in the Response to Maritime Incidents," presented at the NGCUV08, – Navigation, Guidance & Control of Underwater Vehicles Conference., 2008.

[151] G. A. Gonçalves, J. B. Sousa, L. Madureira, and A. Sousa, "Xtreme: an AUV system for network centric operations," presented at the Maritime Systems and Technology Conference, 2009.

[152] T. B. Curtin and J. G. Bellingham, "Autonomous ocean-sampling networks," *IEEE Journal of Oceanic Engineering,* vol. 26, pp. 421-423, 2001.

[153] M. J. Perry, I. C. Darling, and D. L. Rudnick, "Observing the Ocean with Autonomous and Lagrangian Platforms and Sensors (ALPS): the Role of ALPS in Sustained Ocean Observing Systems," *Oceanography (Special Issues Ocean Observations),* vol. 16, pp. 31-36, 2003.

[154] M. A. Godin, J. G. Bellingham, K. Rajan, N. E. Leonard, and C. Y., " A Collaborative Portal for Ocean Observatories," presented at the Ocean MTS, 2006.