



HAL
open science

(DET2009) Evolvable Systems : an approach to Self-X Production

Mauro Onori, Daniel Semere, Bengt Lindberg

► **To cite this version:**

Mauro Onori, Daniel Semere, Bengt Lindberg. (DET2009) Evolvable Systems : an approach to Self-X Production. *International Journal of Computer Integrated Manufacturing*, 2011, 24 (05), pp.506-516. 10.1080/0951192X.2011.566282 . hal-00688941

HAL Id: hal-00688941

<https://hal.science/hal-00688941>

Submitted on 19 Apr 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



(DET2009) Evolvable Systems : an approach to Self-X Production

Journal:	<i>International Journal of Computer Integrated Manufacturing</i>
Manuscript ID:	TCIM-2010-IJCIM-0020.R2
Manuscript Type:	Special Issue Paper
Date Submitted by the Author:	21-Feb-2011
Complete List of Authors:	Onori, Mauro; IIP, Production Engineering Semere, Daniel; IIP, Production Engineering Lindberg, Bengt; IIP, Production Engineering
Keywords:	AUTOMATED ASSEMBLY, AGENT BASED SYSTEMS, ADAPTIVE CONTROL, MULTI-AGENT SYSTEMS, EVOLUTIONARY ALGORITHMS
Keywords (user):	

SCHOLARONE™
Manuscripts

EVOLVABLE SYSTEMS : AN APPROACH TO SELF-X PRODUCTION

Mauro Onori, Daniel Semere, Bengt Lindberg
Royal Institute of Technology,
Inst. of Industrial production,
Stockholm, Sweden

ABSTRACT

Current major road mapping efforts, such as ManuFuture, FutMan and EUPASS, have all clearly underlined that true industrial sustainability will require far higher levels of systems' autonomy and adaptability. In accordance with these recommendations, the Evolvable Production Systems (EPS) has aimed at developing such technological solutions and support mechanisms. Since its inception in 2002 (Onori 2002) as a next generation of production systems, the concept is being further developed and tested to emerge as a production system paradigm. The essence of evolvability resides not only in the ability of system components to adapt to the changing conditions of operation, but also to assist in the evolution of these components in time such that processes may become self-X, x standing for one more desirable properties of a system subjected to a variable operation condition such as self-evolvable, self-reconfigurable, self-tuning, self-diagnosing, etc. Characteristically, Evolvable systems have distributed control, and are composed of intelligent modules integrated. To assist the development and life cycle issues, comprehensive methodological framework is being developed. A concerted effort is being exerted through European research projects in collaboration with European manufacturers, technology/equipment suppliers, and universities. After briefly stating the fundamental concepts of EPS, this paper presents current developments and applications.

Key words: Evolvable Systems, Modularity, Distributed Control

1. Introduction

The major problems of manufacturing companies are all relate to uncertainty. First of all, it is very difficult for companies to predict the type and range of products that will have to be developed. The second uncertainty regards the production volumes and lifespan reached by these future products.

Evolvable production systems (EPS), as a next generation of production systems, was first introduced for assembly in 2002 (Onori 2002) and has, since then, been developed and tested to emerge as a production system paradigm (see EUPASS, A3 projects, as given by Onori (2009)) and the results exhibited at international fairs (Hanover 2008).

Fig.1.- First EPS/EUPASS system, Hanover Fair 2008

The essence of evolvability resides not only in the ability of system components to adapt to the changing conditions of operation, but also to assist in the evolution of these components in time such that processes may become self-X, x standing for one more desirable properties of a system subjected to a variable operation condition such as reconfigurable, tuning, diagnosing, etc.

According to the results attained by many roadmaps [ManuFuture (2005), ManVis (2003), FutMan (2006), KPMG (2002), EURON (2005)], one of the most important objectives to be met by European industry is sustainability, which is multi-faceted: including economical, social and ecological aspects. The obvious conclusion to this holistic problem is that future manufacturing solutions will have to deal with very complex scenarios (Jovane et al, 2009) and focus on unpredictable scenarios rather than forecasted ones.

Evolvable Assembly Systems [(Onori 2002),(Barata et al 2007)] represents one of the paradigms proposed as an opportunity to solve such threats. NOTE: Holonic Manufacturing Systems (Van Brussel et al 1998) and Reconfigurable Manufacturing Systems (Elmaraghy 2006) do not represent evolvability, as will be described later.

Evolvable systems have characteristically distributed control, are composed of intelligent modules and are open in architecture. The technical and architectural aspects of the evolvable system development are supported by a comprehensive methodological framework. Evolvability being a system concept, it is envisaged to address every aspect of an assembly system throughout its life cycle, i.e., design and development, operation and evolution. The work has been, and continues to

be, implemented through large European research projects led by the authors (EUPASS, A3, IDEAS, POPJIM). Furthermore, integration of legacy subsystems and modules have been addressed in the methodology.

It has (to date) resulted in several demonstrators and offered methodologies and architectures in support. This paper presents current developments and applications.

2. Background Approaches

Evolvable systems are meant to address this by providing systems solutions that outlast several generations of product variants. For these reasons, in the process of development of a new product (or even a variant), companies need to optimise:

- cost
- time-to-market/ Time-to-Volume

In the case of a product-specific traditional assembly system, we have high initial costs and long assembly system development time (Onori et al 2008), thus shortening the actual production lead times. Furthermore, each time a new product is required, investments tend to rise dramatically since the assembly system has to be adapted or exchanged.

Fig. 2.0- Adaptability decreases with complexity.

The overwhelming reaction to these problems has been to attempt to develop extremely flexible assembly machines. In the late eighties and early nineties, the general trend in precision assembly was to develop Flexible Assembly Systems (FAS) and Flexible Automatic Assembly (FAA) cells or Hyper Flexible Automatic Assembly, as defined by Onori (1997).

The goal was to have general flexibility, but the actual assembly processes were not studied in depth, therefore resulting in unstable / non-robust or badly adapted solutions. They were fairly adequate to many different product types, but failed to be very performing in any domain. The high cost of such installations was another heavy problem, especially for smaller companies.

Flexibility, instead of the actual assembly process, has been the core issue of most of these developments. As shown in the Figure 2.0, the lower a component is positioned in the hierarchical structure, the more flexibility is necessary to ensure a certain flexibility to the whole system; this means even a system with low flexibility needs a very flexible control: this may imply that FAA solutions also failed because they never provided flexible/agile control systems!

The next attempt at finding a solution was Re-configurable Assembly Systems (RAS). In this case the focus was on the principle that innovative product design must not be limited by assembly process constraints.

The following scheme depicts the process that leads to the Assembly System design in a product-specific RMS assembly system approach.

Fig 3.0- Assembly System design in RMS

In RMS (Van Brussel 1998), the Assembly System design starts from the “New Product Requirements”. The product to be assembled is analysed in order to find all the “Assembly System Requirements”, which means that the driver of the whole process is the Product. In such an approach there are no links between the Product Design process and the successive steps: it leaves the maximum freedom to the designers, but this is not always the right strategy for the company success. Holonic Manufacturing Systems (Elmaraghy (2006) then described a method by which systems could be sub-divided into entities with selected functionality (holons), and subsequently introduced distributed control approaches, as proven by Maturana (2004). However, apart from the distributed control approach, HMS does NOT represent evolvability as the holons are standard manufacturing system components (not process-oriented), and it does not describe any evolutionary qualities.

3.0 EPS Concept.

Basically, the real objectives should not have been flexibility or reconfigurability. These are characteristics, not objectives. The real objective is overall system evolvability, which EPS targets with the following four points:

1. Optimised functionality: the assembly equipment is kept as simple as possible by deriving small, dedicated, process-oriented modules. These may be interconnected to form cells or systems.
2. Optimised orchestration: the control system needs to be the most agile aspects. This is achieved by adopting a multi-agent based, distributed control approach with embedded controllers.
3. Adaptability: the modularity allows for stepwise upgradeability and economic flexibility (it is cheaper & simpler to change a module than modify a system). The actual system may also adapt to minor changes via its control system, which, being skill-based, allows for emergent behaviour to be exploited.
4. Robustness: the equipment is dedicated, small, and includes an own processor. Some modules (robots) may even be reconfigurable. The control system is goal-oriented, and the system is process-oriented. This results in a dedicated system based on an adaptable concept with advanced interfaces.

Fundamentally, EPS suggests that true agility/flexibility can only be achieved if the lowest building blocks of a system are those that exhibit the highest rate of adaptability/evolvability. As the clustering of components increases in complexity, so does the agility/flexibility decrease. Hence, in order to build truly agile systems, one must begin by considering the control architecture.

According to the EPS Paradigm (Onori et al, 2010), each system should consist of several skill-based and process-oriented units: these elements should be very task-specific in order to accomplish only a simple action. It is possible, in fact, to consider every complex task as the union of several simple actions. Therefore each “shared element” can be used to accomplish the same action in different assembly cycles or in different product generations. Obviously, due to its modularity and unit-by-unit development, such a system is accessible even for small-medium sized companies that can spread the investment over a given timespan. Moreover, as will be detailed later, that the EPS process-oriented approach to the Ontology (developed in the EUPASS project (Onori et al, 2010) allows to define the Assembly System requirements for a whole **Class of Products** instead of a single product: the defined assembly processes are common to an entire set of products.

A fundamental condition is the **plugability** of all components.

3.1 Basic definitions

One of the major issue brought up at different conferences and workshops is whether the “Evolvable” term has been defined in a more topical manner: i.e. in relation to the topic of Production. As a matter of fact, the initial impression may be that EPS could be compared to natural evolution. However, the EPS paradigm actually applies evolvability at different levels. The EPS group studies begun with an analysis of the evolutionary computing (Dimopoulos et al 2000), such that one could try to identify analogies between natural systems and EPS, and possibly go as far as to see whether the computational aspects behind EPS could find support in their work. Therefore, the initial work focussed around the genotype/phenotype discussions (Fleming et al 2002). For example, within EPS one could, it was initially supposed, that, from a low-level perspective:

Genotype: lower-level, parameters (genes?) → function, structure, instance (parameters)
Phenotype: upper-level, visible part → observable behavior

Having defined this rudimentary relation (very preliminary), the group could conclude that:

“An EPS Module is a self-composed entity with a given functionality and with well-defined interfaces, that interacts with other modules”

This, however, brought about a certain dilemma, since EPS modules may actually be represented by two different perspectives/views: structural view and behavioral view. Therefore, if we related this back to the genotype/phenotype proposal, the research group began to wonder how these two views would be controlled. By definition,

Module{taxonomy(genotype)/functionality, associations, interactions} .

This results in that the given ontology may be associated to the “gene”, or software agent:

Ontology → agent < > gene

To clarify this somewhat, a distinction was also made between the far more simple taxonomic representation of modules, and their ontology (which represents the genetic parameters of EPS modules):

$$\begin{array}{l} \text{Genotype} + \text{Environment} + \text{Random variation} = \text{Phenotype} \\ \text{Taxonomy} + \text{Condition} + \text{Variation} \qquad \qquad = \text{Functionality} + \text{Associations} + \text{Interactions} \end{array}$$

In this description, the genotype is fairly fixed over time whilst the phenotype is adaptable. This means that the Module adapts whilst the ontology and agents evolve

This is an important conclusion as it defines system characteristics. In EPS the modules may adapt to a given range of changing conditions. They do not evolve in the true sense of the evolutionary approach; however, as the agents and associated ontology do adapt, there does exist a clear and definite distinction between EPS and Adaptable systems! This may be summarised as follows:

Flexibility: the ability of a system to operate within a variable functionality range. This range is often constrained by an additional term such as "Capacity/Capability/Variant/geometric/static/dynamic/etc". Such terms are added as the flexibility itself remains difficult to quantify. At present, over 90 constraining terms are in use. EPS does not use this term.

Adaptability: the ability of a system module/component to adapt its process functionality to a limited range of changing external conditions (related to process & module-feed rates, axes performance, etc.). This adaptability is exercised within a limited parameter range and will only affect the other local parameters (of module).

In other words this is the ability of an interacting group of modules to make structural changes in order to adapt. This is a Genotype transformation: the ability to modify its behavior in order to cope with external change. The final, phenotype transformation is achieved through the evolution of the agents controlling the module.

Evolvability: the ability of a system (coalition of modules) to gradually modify/update how these modules interact to achieve new goals. This occurs only within coalitions/systems. As will be detailed later, this is, basically, the evolution of social behaviour of the society of modules.

Note: Implies the ability to accommodate adaptability at modular level such that the interactions attain the new, wanted goals (gene transformation). This also implies that the Knowledge model update is autonomous.

EPS Group definition of an evolvable system entity:

"Any entity/object is evolvable from one state to another, within a dynamically changing environment, when its constituents may adapt to these changes whilst maintaining a fully functional interaction with one another. This new resulting interaction represents the evolution."

Please recall the previous explanation that components(constituents) adapt, while the group (coalition) evolves.

Constraints:

- Modularity is essential and must be linked to the dynamically changing environment that it works within.
- The interfaces are fundamental and must be stringently defined and must include social functionality (how clusters of modules interact, not only 1-to-1 behaviour).
- The evolvability must be accomplished autonomously. Any human interference limits the development to simply adaptability (one may not plan next step for the system!!see Twente paper). The modules, which actually only adapt, must have adaptive feedback.

Note that not all products/systems need to be evolvable! The goal dictates the choice: only products/systems that have to deal with unstructured, continuously changing demands are suitable.

Once the modules are clustered into coalitions, and begin to function as a full system, the evolutionary analogy attains a societal level. That is, the individuals (modules) will ADAPT to a change in the society's behaviour (EVOLUTION). That is, in human societies the actual human being adapts to new cultures (for example), whilst a society as a whole evolves to attain the goals dictated by new conditions (political, economic, social, etc.).

Fig.4.0- The Societal Evolutionary aspect of EPS

Hypothesis 1:

"The rate of evolvability is VERY dependent on the complexity of the object and on whether it adopts a group or singular behavioural pattern".

As we are dealing with man-made elements (modules), the natural evolution analogy arrives at a limit. Basically, as we are dealing with a social system and not a parasite or single-cell entity, the societal issues become predominant once we create a cluster.

So the analogy to detailed natural evolution has a limit for EPS. EPS consists, to date, of simple, dedicated entities acting within a context of limited formalised information (cannot model ALL possible conditions).

Consequently, the next step is to study the *societal evolution* in order to understand the conditions that will possibly determine the evolutionary patterns of the groups/clusters/coalitions.

Note: remember that this entails NO individual evolution, only group/cluster behavioural evolution; i.e. - how modules evolve in their interactions, as modules themselves only adapt.

Therefore, we limit the EPS analogy to natural evolution because our modules adapt, not evolve—they are non-biological, man-made, hence are indirectly influenced by human intervention (and limited). The real evolution we want to attain is at system level (continuous, autonomous adaptation to new requirements), which is closer to societal evolution: how the adaptable modules modify/update their inter-related functionalities, autonomously to meet new system goals.

Hypothesis 2:

“The most (only at present) evolvable module in a modern production system is the human operator: he/she may re-program themselves and self-adapt tools such that they may continue to collaborate with the other machines/operators, in a changing environment.”

A preliminary conclusion may be that:

- At individual level, EPS modules are less evolvable than humans.
- At social level, however, EPS modules are more evolvable as they are very task-specific and even expendable (faster rate of adaptation and possible exclusion if obsolete).

4.0 Enabling Research Domains

One of the most important consequences of this approach relates to its control system: modules are agents, knowing their proper capabilities and possible forms of cooperation with other agents. The principles of emergence can be applied, working with module skills and forming higher capabilities out of it.

Therefore, when a system is created according to the EPS principles, the resulting capability of the sum of the modules will not be so easily predicted. When a multitude of small entities is brought together, new and unexpected capabilities surface from such coalitions, as given by Fogel (2006): this is called *emergent behaviour* and is best treated, amongst current paradigms, by EPS (Onori et al, 2010).

The main issue to be addressed in this section is describing the areas in which EPS control systems are getting inspiration to solve the requirements for adaptability at fine granularity. Numerous scientific domains investigating phenomena which EPS also exhibit have emerged in the last few years, which can provide helpful tools and valuable theoretical background to cope with the complexity of manufacturing systems. A more detailed definition may be found in Holland (1998).

Complexity Theory

Complexity Theory looks for simple causes leading to complex behaviors (Delic et al. 2006). Complex systems are spatially and/or temporally extended non-linear systems with many strongly-coupled degrees of freedom. They are composed of numerous in themselves often simple elements and are characterized by collective properties. Similarly, EPS consist of numerous equipment modules which are connected to each other and have multi-lateral interactions. Each of them has some degrees of freedom, which are constraint by other system parts. Together, the modules form a system with the desired global behavior.

Artificial Life

Taking natural life and its characteristics as an example, scientists attempt to create life-like behaviors with the capability of evolution on computers and other “artificial” media. EPS are very similar to artificial living systems. They have a modifiable structure, will exhibit some kind of self-organization, can adapt to their environment, and react to stimuli. They are capable of evolving according to the circumstances, namely in terms of equipment states, and can incorporate newly available technology. As any living organism, they will include efforts to keep themselves in a constant well-functioning state through self-surveillance and self-management – at least to a certain degree.

Autonomic Computing

Although at another level than the other areas described above, Autonomic Computing is a fundamental concept for EPS. The vision of Autonomic Computing (Delic et al 2006) refers to the tendency of computers to become ubiquitous. Forming large networks and having complex and multiple interactions, they become increasingly difficult to manage. As a consequence, software will be designed to take care of itself. User interaction will be minimized and reprogramming avoided. Note that the more modules of fine granularity include computational power, the more is necessary to find new ways of coordination and automatic plugability, which is exactly what EPS want to address.

Agents

Depending on the context, an agent can be a human person, an association, an animal, or a piece of software, eventually connected to some hardware. The fundamental characteristics are identity, intelligence and the ability to act and react in order to persecute goals. Agents have at least a certain degree of autonomy and can compete or collaborate with others.

There are numerous successful experiences with agent-based systems in industry (Kephart et al 2003), (Parunak 2000). Rockwell Automation even develops agent-based systems where the agents run inside the PLC itself (Monostori et al 2006) instead of on separate computers.

Self-Organization

Reasons for implementing self-organization in EPS are to minimize and facilitate user interaction, i.e. to hide complexity and increase system autonomy. Building and configuring a system composed of numerous entities with multi-lateral interactions is a highly complex task; the more autonomy the system has, the easier it gets for the user. Production systems tend to have many components of diverse nature which interact in many coupled ways. Agents need the capacity of organizing their collaboration themselves, in different forms and compositions, according to the needs, without passing through a central coordination point.

Emergence

Complex systems most often consist of at least two different levels: the macro-level, considering the system as a whole, and the micro-level, considering the system from the point of view of the local components. Local components behave according to local rules and based on preferably local knowledge; a representation of the entire system or knowledge about the global system functionality is neither provided by a central authority nor reachable for the components themselves. They communicate, interact with each other and exchange information with the environment. From the interaction in this local world emerge global phenomena (Ueda 2001), which are more than a straight-forward composition of the local components' behaviors and capabilities.

Typically, there is a two-way interdependence: not only is the global behavior dependent on the local parts, but their behavior is also influenced by the system as a whole. Emergent phenomena are scalable, robust, and fault-tolerant, i.e. insensitive to small perturbations and local errors as well as component failure, thanks to redundancy. They exhibit graceful degradation, meaning that there is no total break-down because of minor local errors.

Emergent Behaviour

Of course the lower the level of the device considered, the higher is the emergent behaviour (fine granularity = high emergence).

Evolvable Assembly Systems (EPS) exhibits the following characteristics:

- It is a fully "reconfigurable" system platform that exhibits an emergent behaviour.
- Mechatronically integratable assembly units.
- The reconfigurable system has to be composed of process-oriented components (gives granularity of emergent behaviour).
- A system that can automatically determine its functionality based on the components' skills.
- The change in paradigm is, partly, that we no longer invest in the programming & coding, but, rather, in how to establish and exploit relations.
- Maintenance, documentation and the ability to store information in support of operational stability.

5.0 Methodology

The EPS Methodology provides the references architecture, enablers, and modelling formalisms. In the following section brief description of the methodology is given. This work was led and conducted by the authors research groups within the EUPASS and IDEAS projects (Maffei & Onori, 2011).

5.1 Reference Architecture, RA

The EPS Reference Architecture (EAS_RA) describes the essential features of an Evolvable System which means the reference architecture specifies the necessary features that a system should have to be an evolvable system. The reference architecture is composed of three main elements: Principles, Technical Positions and Templates.

Principles : EPS has two fundamental principles which lay foundation and guide the development process of an evolvable system. These principles can be considered as description of the core ideas of the evolvable system paradigm.

Principle 1: the most innovative product design can only be achieved if no assembly process constraints are posed. The ensuing, fully independent, process selection procedure may then result in an optimal assembly system methodology.

Principle 2: Systems under a dynamic condition need to be evolvable, i.e., they need to have an inherent capability of evolution to address the new or changing set of requirements.

These principles have been described in detail in (Semere et. al., 2007),

Technical Positions: EPS design and implementation decisions and objectives set at a technical / technological level that describes the ontology, exploited protocols, standards or specifications for use with each major architectural component.

Templates and Partial Models: Reusable diagrams, graphs, objectives and knowledge and rules that address the distribution of system functions and how they relate topologically. Templates use models to show relationships and between components as specified by the Technical Positions and pertinent knowledge units.

5.2 Modelling Formalisms

Modelling formalisms refers to the ontology and the graphical tools used to build models in the reference architecture and the enabling models as described in section 5.3.

The EPS formalised concepts (ontology) and definitions are represented using a set of descriptive tools such as :

- Definitions of the most important concepts: module, process, product, EPS module, skills, EPS assembly system (which is a composition of modules), etc
- Diagrams (UML, etc) where the interactions between the concepts defined are shown. This enables to show how the EPS architecture generates assembly systems. The interaction may show the global system behaviour
- Formalisation of concepts.

The domain ontology indeed captures the concepts in the system with their specifications (consensual semantic) i.e., what the concepts are and how they are related to each other in the domain. However, it does not capture the logic behind the relationships and the how's in the synthesis and functionality of the system.

Fig. 5 Partial view of the EPS Ontology.

5.3 Enablers

This part of the methodology provides, the necessary models, tools and methods for the development and evolution of an evolvable system.

The enabling models includes, the development process model, the business model and the knowledge model. These models are constructed using the formalisms described above and most notably the EPS ontology

The figure below depicts a simplified overview of the EPS methodology, which comprises ontologies and knowledge models.

The traditional top-down system design is feasible only in cases where the emergent behaviour is fully describable (Marik et al 2005). If emergent behaviour has to be investigated even at design and development stages, then a heterarchical or a network approach are the options.

Fig 6- The simplified EPS Methodology

Concepts that are formalised in the EPS Ontology are used to capture the stakeholders understanding of their own domain. The EPS knowledge model is thus a structured and formalised collection of such knowledge capturing representations of the domains. The main objective of the EPS knowledge model is to provide an environment that supports the development and operation of evolvable systems.

Fig. 7- Basic aspects of the EPS Reference Architecture

The domain knowledge captured using the EPS ontology and the EPS knowledge templates are the two entities used to develop the knowledge model. The knowledge models are used among other things how each module in a system should address for a new set of conditions.

The elements in the EPS knowledge model consists of the following knowledge domains:

1. The enterprise knowledge domain - globalizes knowledge of the system and represents the business, organizational and global knowledge models captured. Enterprise knowledge enables environment recognition and maintenance of organizational associations.
2. The product knowledge domain – captures the knowledge related to product specification and design to assembly tasks.
3. The execution knowledge domain: capturing knowledge elements related to communications, planning and scheduling

The learning knowledge domain: containing knowledge elements that are used to incorporate case based reasoning. The EPS RA is ultimately be viewed from different perspectives addressing the different concerns of the stakeholders. The stakeholders include:

- those who build the system (structure and communication views),
- those who use it (functional view),
- those who are concerned with control (control view).

The architecture should, therefore, address the concerns of every stakeholder, which gives rise to a multiple set of viewpoints concerning the requirements, expected outputs and controlling mechanisms. These may come to include:

- Functional Views, which address the concerns of the users of the system, and describe the functionality of the system, the process flows, quality, logistical issues.
- Communication Views: which address the concerns of how the data and information is to be represented, interpreted, recorded and transmitted.
- Control views: which address the concerns of the control system.
- Structure Views, which address the concerns of the system integrators, and maintenance.

These represent the background work that needs to be carried out in order to develop and establish an adequate EAS_RA. The actual implementation of this architecture, when considering the EPS modules and how they are to be integrated and formed into a system, will also be subject to a set of "views".

Fig.8 - The preliminary EPS approach

6.0 Applications

Initial evaluations were carried out in the test cell shown below (fig.9.0a). More industrially viable evaluations are currently being deployed within a new system being developed at KTH (fig.9.0b).

Fig. 9.0a- The Evaluation Test Case Setup

Fig. 9.0b- Industrial Test Case Setup

This setup (fig.8.0b) is being finalised at KTH on the basis of the FESTO mini-assembly system that was recently tested within the IDEAS project. The FESTO demonstrator was built at their premises in Esslingen and exploited reconfigurability and distributed control. This control approach is now being transferred to commercial control boards (ELREST and validated within the ABB and MASMEC cell shown above. This is a real industrial scenario as the solution is for Electrolux Home Products Italy SpA and Centro Ricerche FIAT through the IDEAS FP7 project. The layout given is being setup for two industrial products: piston ring assembly in an engine block and washing machine handles (self-configuring & reconfiguring).

The products are standard, off-the-shelf products. The project has now gone as far as to demonstrate that legacy equipment may be modified to EPS modules through a Mechatronic Architecture [Onori (2009)]. An intelligent interface was developed out of an old Schunck gripper, as shown below, and the EUPASS project has now developed a full Evolvable Assembly System in Windisch, Switzerland (Fig 10).

Fig. 10- The Intermodular Receptacle

At present the EPS paradigm is only just starting to take a practical form, and the control solution, ontologies, and methodologies only partially describe the most recent developments, some of which have been presented at the Hannover Fair 2008, Productronica 2008 (see www.Eas-Env.Org), and in a FESTO application that deploys some of the EPS ideas.

Fig. 11- The EUPASS System at FHNW, Switzerland

7. CONCLUSIONS

EPS, as with other similar approaches, offers great opportunities for attaining true agility and cost-effective, stepwise automation. The technologies for achieving this are available and there are several partners willing to partake in this endeavour; however, it is vital to point out that EPS does imply that the manner in which we develop and create projects for the development of assembly systems are radically changed, assuming a more synthesis-based approach.

The experience of this group is that there exists far too strong a resistance to such change in certain academic and industrial circles, and that the creation of new approaches and even new service sectors, all based on EPS and similar approaches, will have to prove their validity through even more elaborate industrial scenarios. Nevertheless, it must be said that at the end of the day it is not the most efficient or technologically advanced solution that may represent the future way of developing assembly systems, but the one that best paves the way to large-scale exploitation.

In order to stimulate the further development and update of the paradigm and its applications, a collaborative web space has been developed, in which the architecture, standards, equipment modules and other details are detailed and made accessible: www.Eas-Env.Org. This EPS Environment web space is currently being expanded to include work from related projects (Onori et al 2008).

8. REFERENCES

- Barata, J., Onori, M. and Frei, R. (2006b) In *ISIE'06 -IEEE International Symposium on Industrial Electronics* IEEE, Montreal - Canada.
- Barata, J., Onori, M., Frei, R., Leitão, P. (2007); "Evolvable Production Systems – Enabling Research Domains"; Proceedings of the 2nd International Conference on Changeable, Agile, Reconfigurable and Virtual Production (CARV 2007), Toronto, Canada on 22-24 July 2007
- Delic, K. A. & Dum, R. (2006) On the Emerging Future of Complexity Sciences. *ACM Ubiquity*, 7
- Dimopoulos, C. & Zalzal, A.M., 2000. Recent developments in evolutionary computation for manufacturing optimization: Problems, solutions, and comparisons. *IEEE transactions on evolutionary computation*, 4(2), 93–113.
- Elmaraghy, H. A. (2006) Flexible and Reconfigurable Manufacturing Systems Paradigms. *International Journal of Flexible Manufacturing Systems*, 17, 261-276.
- EURON , "Research Roadmap V1- EURON European Robotics Network"; Network of Excellence, Information Society Technologies Priority; EURON/Univ.of Karlsruhe; August 2005, EC
- Fleming, P.J. & Purshouse, R.C., 2002. Evolutionary algorithms in control systems engineering: a survey. *Control Engineering Practice*, 10(11), 1223-1241.
- Fogel, D.B., 2006. *Evolutionary computation: toward a new philosophy of machine intelligence*, Wiley-IEEE Press.
- FutMan, "The Future of Manufacturing in Europe 2015-2020-The Challenge for Sustainability"; Institute for Prospective Technological Studies, European Commission Joint Research Centre, EUR 20705 EN
- Holland, J. H. (1998) *Emergence - From Chaos to Order*, Oxford University Press, Oxford.
- Jovane, Francesco; Westkämper, Engelbert; Williams, David: The ManuFuture Road - Towards Competitive and Sustainable High-Adding-Value Manufacturing; Springer, Vol.10, 2009
- Kephart, J. O. & Chess, D. M. (2003) The Vision of Autonomic Computing. *IEEE Computer*, 0018-9162/03, 41-50.
- KPMG Special Services , "SMEs in Focus- Main results from the 2002 Observatory of European SMEs", submitted to the Enterprise Directorate-General of the European Commission by KPMG Special Services; Observatory of European SMEs, ISBN 92-894-4878-4

- Maffei,A.; Onori, M.; "Evolvable Production Systems"; Key Engineering Materials Vols.467-469 (2011) pp.1592-1597; Trans Tech Publications, Switzerland.
- Manufature Platform-Executive Summary, "Strategic Research Agenda-assuring the future of manufacturing in Europe"; December 2005, EC
- MANVIS, "Manufacturing Visions-Integrating Diverse Perspectives into Pan-European Foresight"; FP6 Support Action, NMP2-CT-2003-507139
- Mařík, V., Vrba, P., Hall, K. H. & Maturana, F. P. (2005) Rockwell automation agents for manufacturing. AAMAS. Utrecht, NL, ACM Press NY, USA.
- Maturana,F., Staron,R.,Hall,K., Tichý,P., Slechta,C.,Marik,V.,; "An Intelligent Agent Validation Architecture for Distributed Manufacturing Organizations. 81-90"; Proceedings of the BASYS 2004 conference, Austria, Vienna, 2004
- Monostori, L., Váncza, J. & Kumara, S. R. T. (2006) Agent-Based Systems for Manufacturing. CIRP Annals, 55.
- Onori, M. (2002a) In *ISR2002 - 33rd International Symposium on Robotics* Stockholm, pp. 617-621.
- Onori,M.,; "A Low Cost Robot Programming System for FAA Cells"; Proceedings of the 28th Int.Symposium on Robotics (ISR), Detroit, Michigan, USA, May 1997
- Onori M., Barata J. (2009) Evolvable Production Systems: mechatronic production equipment with process-based distributed control; Proceedings of the IFAC/SYROCO 2009 Conference, Gifu, Japan
- Onori, M.; "EUPASS Adaptive Assembly Roadmap 2015-deliverable 1.5f"; Project Report-Public, Document 1.5f, EUPASS-Evolvable Ultra Precision Assembly, NMP-2-CT-2004-507978; October 2008
- Onori, M.; Barata, J.; "Outlook Report on the Future of Assembly Automation"; Journal of Assenby Automation, Manuscript no. ID AA-09-038, Spring 2010.
- Parunak, H. V. D. (2000) Agents in Overalls: Experiences and Issues in the Development and Deployment of Industrial Agent-Based Systems. International Journal of Cooperative Information Systems.
- Semere, D., Barata, J. Onori, M. *Evolvable Systems: Developments and Advance.*, ISAM 2007 IEEE conference, Ann Arbor, Michigan, USA. July 2007.
- Ueda, K. (2001) Journal of Artificial Intelligence in Engineering, 15, 319-320.
- Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L. & Peeters, P. (1998) Reference Architecture for Holonic Manufacturing Systems: PROSA. Computers in Industry, 37, 255-274



Fig.1.- First EPS/EUPASS system, Hanover Fair 2008

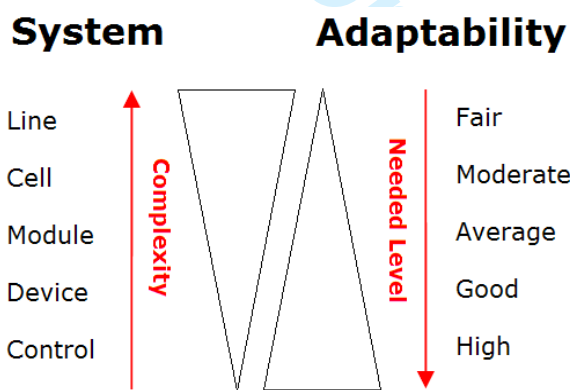


Fig. 2.0- Adaptability decreases with complexity.

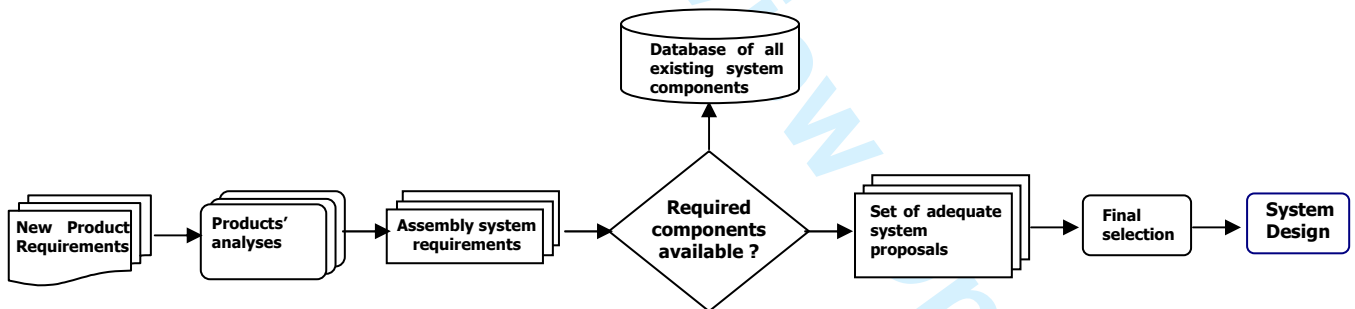


Fig 3.0- Assembly System design in RMS

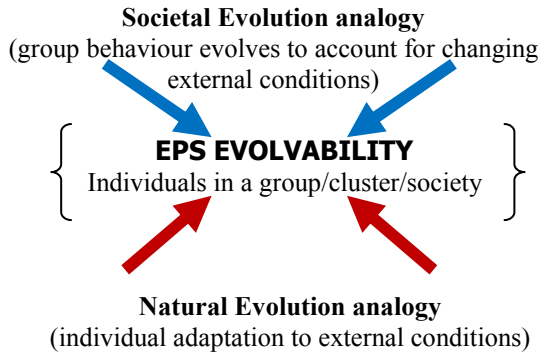


Fig.4.0- The Societal Evolutionary aspect of EPS

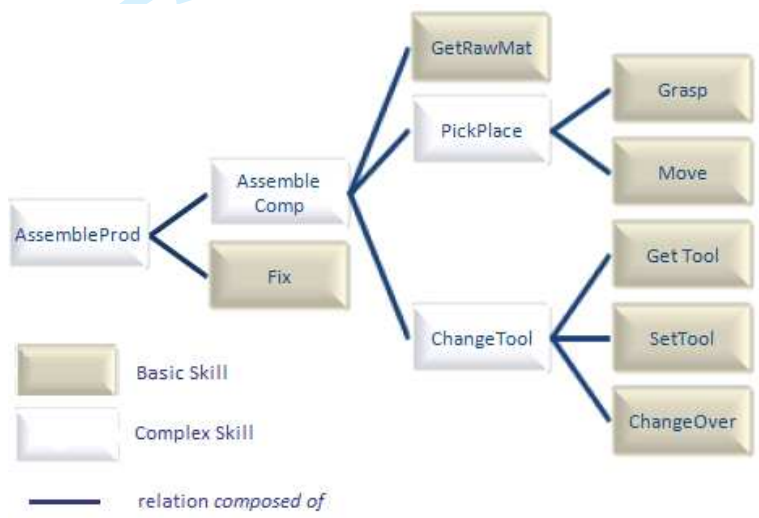


Fig. 5 Partial view of the EPS Ontology.

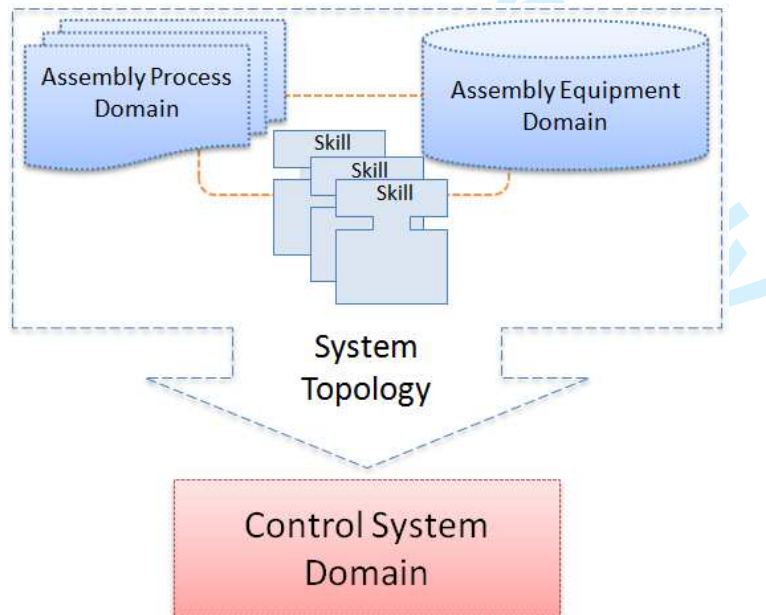


Fig 6- The simplified EPS Methodology

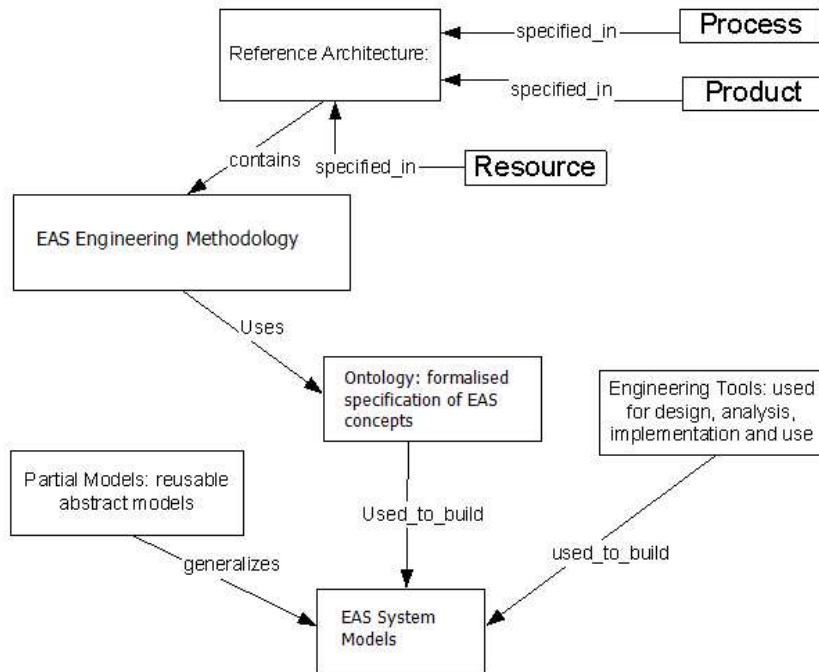


Fig. 7- Basic aspects of the EPS Reference Architecture

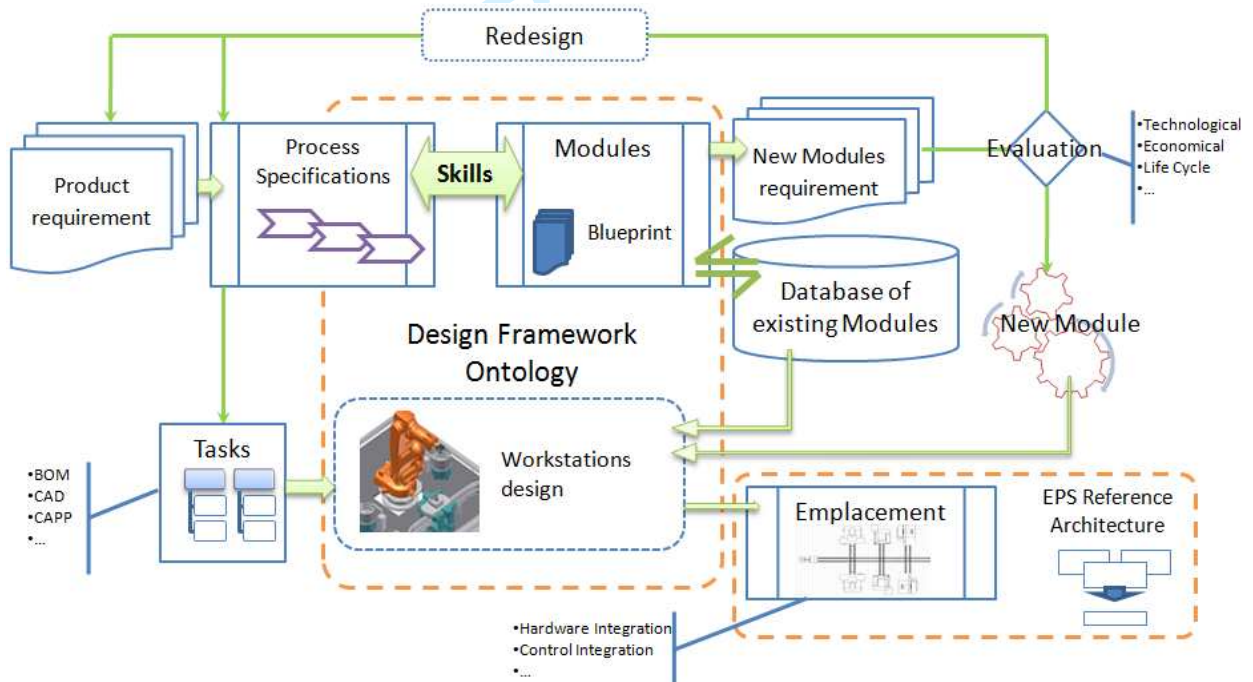


Fig.8 - The preliminary EPS approach



Fig. 9.0a- The Evaluation Test Case Setup



Fig. 9.0b- Industrial Test Case Setup

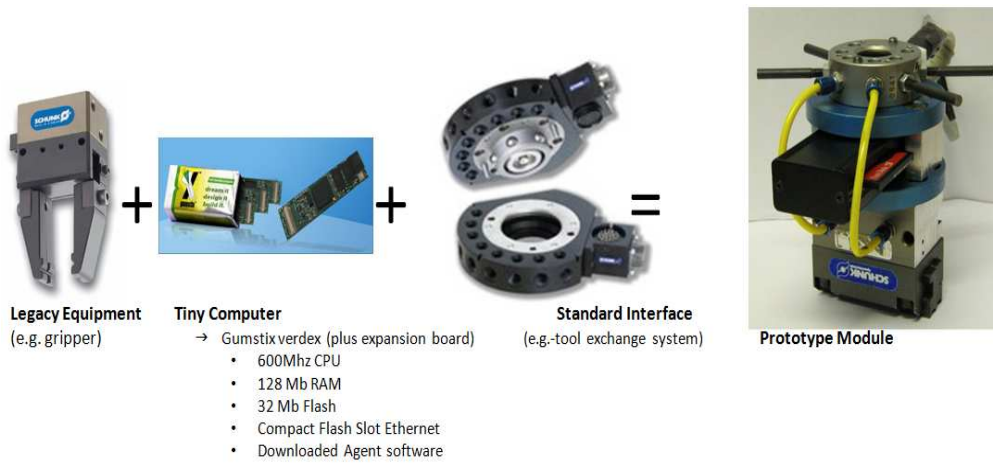


Fig. 10- The Intermodular Receptacle



Fig. 11- The EUPASS System at FHNW, Switzerland