INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT

Deliverable 2.2: State-of-the-Art Overview

CREATE

Creating Evolution Capable Co-operating Applications in Industrial Automation

Project number: ITEA 2 ip10020

Edited by: Silvia de la Maza (INNOVALIA)

Contributors: Fernando Perales (TRIMEK), Vadim Chepegim (TIE), Peter Funk (Malardalen University)

Date: 29.08.2012

Document version no.: *0.6*

INDEX

LIST OF FIGURES

1. Currently available system architectures in the area of industrial automation

1.1. Centralized

In the centralised architecture, a server keeps track of all the communication links and routes the data and messages to the correct receivers. This approach is neither scalable nor efficient, because the server will easily become a bottleneck and there is no direct communication between the senders and receivers. Fault-tolerance is also a problem, because a failure of the server will bring down the whole system.

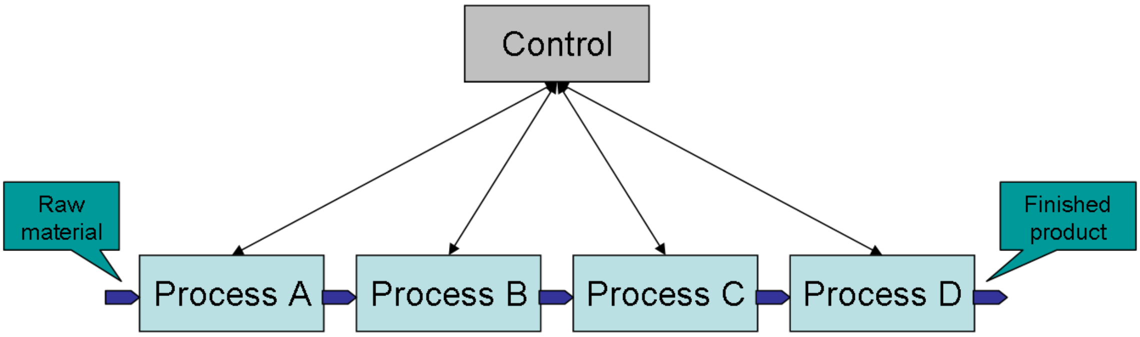


Figure 1 - Centralized automation control

1.2. Multi-layer Centralized

In this kind of system a computer is in charge of supervision chores, alarm management and data treatment and process control. Communication is performed by especial BUSes or LAN nets. These are real time executions and are designed for providing monitoring and supervision capabilities. Control system process elements can use a real time standard implementation of CORBA (Common Object Request Broker Architecture) for the communication among objects through nets. Indeed, the specification of the interfaces will be very important for the maintenance and conservation of the investment given the rapid technological changes. This is the reason why open standards such as RT POSIX or ATM and CORBA are used.

The necessary programmes and hardware is generally called SCADA.

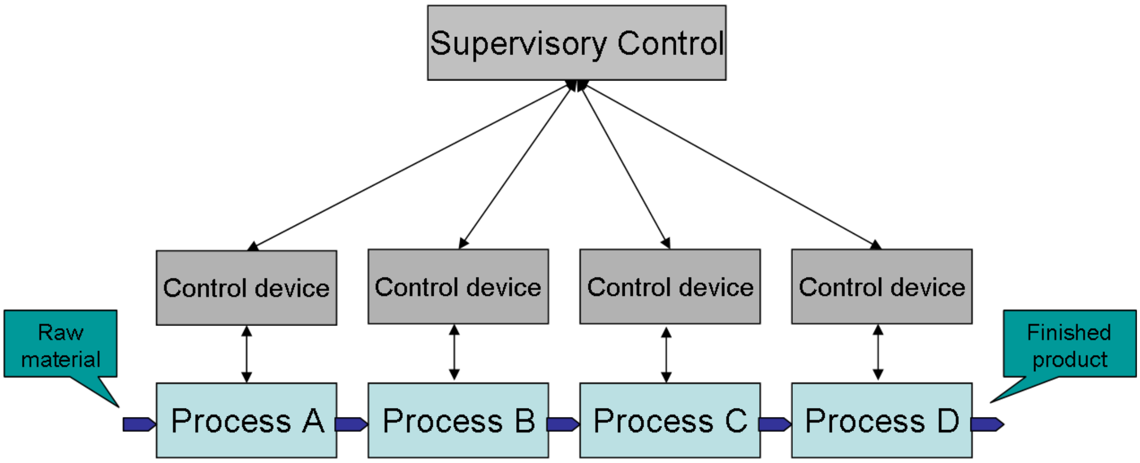


Figure 2 Multi layer centralized automation control

1.3. Distributed

Next figure shows a distributed control system scheme. As it can be seen it is very similar to the multilayer centralized one. The difference here is that there are several control units each of them controlling all the processes and a communication channel exists between the process controllers. In case of malfunction or overload, it is possible to transfer part of the whole work load to other units.

The possibility of bypass the work load avoids system jammings. In order to set this architecture a higher capacity of access, communication and data treatment is needed so that the different controllers can have a dynamic assignment.

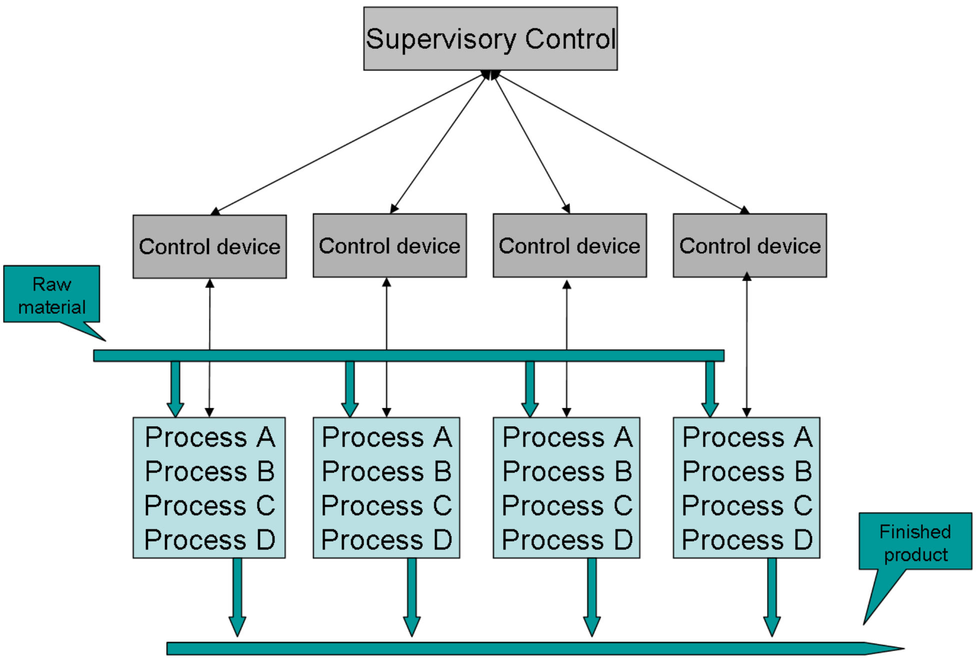


Figure 3 - Distributed automation system

1.4. Service oriented architecture

In Service Oriented Architecture (SOA) systems are built by composing independent well-defined units of functionality called services. One of the main characteristics of such architecture is that services are loosely bounded. Services do not posses any knowledge about the systems they will be used in. Similar, before their deployment, applications (consumers of services) do not know which are the concrete services that will deliver parts of concrete functionality.

This is determined in run-time, when applications discover available services and bind to them. Service-oriented systems can also utilize run-time reconfiguration. During run-time new service providers can be introduced (so called “Plug and play”) or existing ones can become unavailable. Applications can dynamically change their bindings to services based on service availability and the quality they can provide.

An essential part of SOA is service discovery. Service consumers do not know in advance which service providers will be available while the system is running. Because of that there must exist a mechanism for consumers to learn what service providers can deliver the functionality they need.

Implementation of service discovery can either be centralized or distributed. In the centralized approach there exists one service registry to which services announce their presence and consumers can query for a service. Distributed implementations depend on service providers listening for queries from the consumers on some kind of computer network. Providers then send responses for service queries they can satisfy.

Once a consumer discovers services that implement desired functionality it can collect more information about them by requesting service descriptions. Such descriptions can contain both functional and non-functional attributes of services. The consumer then selects a service provider that fits the best and initiates binding to the service.

CREATE is a step towards a SOA approaches in manufacturing where SNMs have different levels of intelligence and can communicate with other SNMs. Increasing demands on flexibility and reconfigurable manufacturing systems with increasingly advanced technology and intelligent parts makes SOA architectures desirable in manufacturing. SOA is inspired by agent based system and ultimately enables manufacturing systems that are robust and adapt to dynamic changes in their environment and resolve internal and external disturbances in an intelligent way. The SNM concept enables such an evolution of manufacturing systems and is a move towards so called holonic manufacturing systems (autonomous co-operating agents).

1.5. Enterprise Service Bus

The Enterprise Service Bus (ESB) is the bare bone of the CREATE architecture. It provides the functionality for communication among different components or SNMs connected within the CREATE middleware.

CREATE follows principles of Service Oriented Architectures (SOA) to assist the development team in focusing just on the different services that CREATE platform should provide, independently of the way that the different components are implemented. ESB's primary use is in Enterprise Application Integration of heterogeneous and complex landscapes.

In [1] ESB is defined as following: "*an Enterprise Service Bus is a software architecture for integrating enterprise applications at service level. It is implemented as middleware that provides the means for standardized communication among applications and supports service, message, and event-based interactions among applications*."

ESBs provide functionality bundled into different functional areas [2] as follows (Figure):

- **Architecture**. Support for fault tolerance, scalability and throughput, the ability to federate with other ESBs, the supported topologies, and features supporting extensibility
- **Connection**. Includes support for a wide range of messaging standards, communications protocols, and connectivity alternatives
- **Mediation**. Deals with key requirements related to dynamic provisioning of resources, transformation and mapping support, transaction management, policy meta model features, registry support, and Service Level Agreement (SLA) coordination
- **Orchestration**. This area provides lightweight orchestration of services and more robust Business Process Execution Language (BPEL) and/or Business Process Modelling Notation (BPMN) support
- **Change and control**. The main components are design tooling, life-cycle management, technical monitoring, and security
- **Commodity services** like event handling and event choreography, data transformation and mapping, message and event queuing and sequencing, security or exception handling, protocol conversion and enforcing proper quality of communication service.

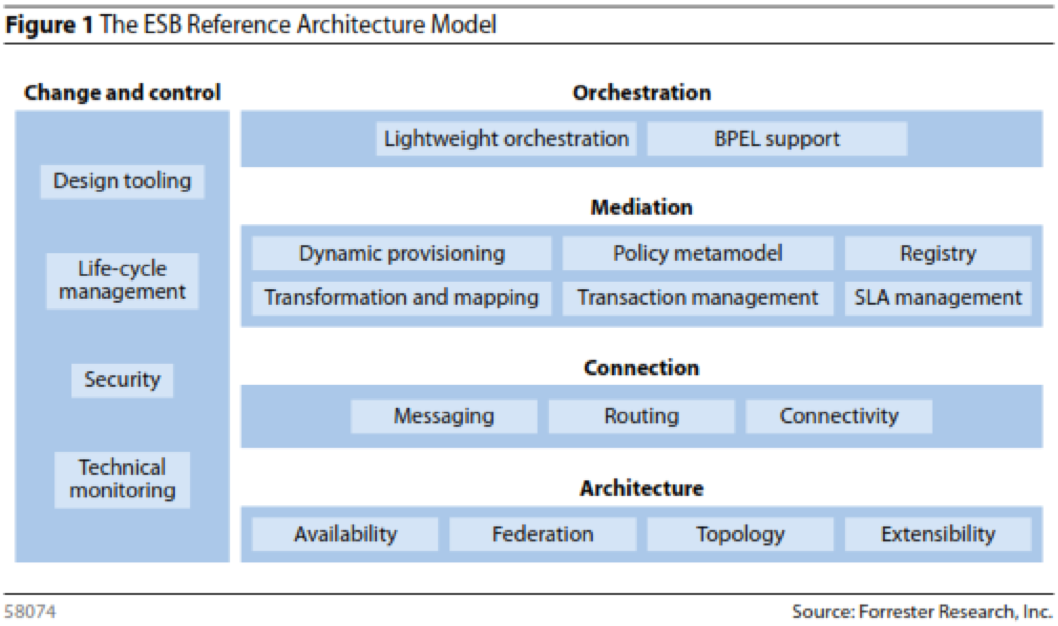


Figure 4 - ESB Reference model (from Forrester Research)

The following can be considered general ESB core functionalities, based on [3]:

ESB core functionality	Description
Location transparency	The ESB helps with decoupling the service consumer from the service provider location. The ESB provides a central platform to communicate with any application necessary without coupling the message sender to the message receiver.
Transport protocol conversion	An ESB should be able to seamlessly integrate applications with different transport protocols such as HTTP(S) to JMS, FTP to a File batch and SMTP to TCP.
Message transformation	The ESB provides functionality to transform messages from one format to the other.
Message routing	Determining the ultimate destination of an incoming message is an important functionality of an ESB that is categorized as message routing.
Message enhancement	An ESB should provide functionality to add missing information based on the data in the incoming message by using message enhancement.
Security	Authentication, authorization and encryption functionality should be provided by an ESB for securing incoming messages to prevent malicious use of the ESB as well as securing outgoing messages to satisfy the security requirements of the service provider.
Monitoring and management	A monitoring and management environment is necessary to configure the ESB to be high performing and reliable and also to monitor the runtime execution of the message flows in the ESB.

Table 1 - ESB core functionalities

Existing industry level Enterprise Service Busses

There are several ESBs in the market, covering many aspects and (i) given that the selected ones are more complete than the disregarded ones, and (ii) to avoid extending this section, the CREATE project has analysed many of them but covers in detailes only for 2 the most promising of them:

- TIE Kinetix SmartBridge
- Apache ServiceMix.

TIE Kinetix SmartBridge (TSB)

TIE Kinetix SmartBridge[1], is provided by CREATE partner namely TIE Kinetix. It is a Business Integrated Platform and a complete and efficient integration solution born from the B2B world. It provides tools for seamless integration of back-office solutions, by implementing different interoperability strategies. Based on a core ESB, it is compounded of a hub for transferring single document/messages among the applications connected to it. TSB is especially designed for optimizing and automating the different

[1] http://businessintegration.tiekinetix.com/node/802

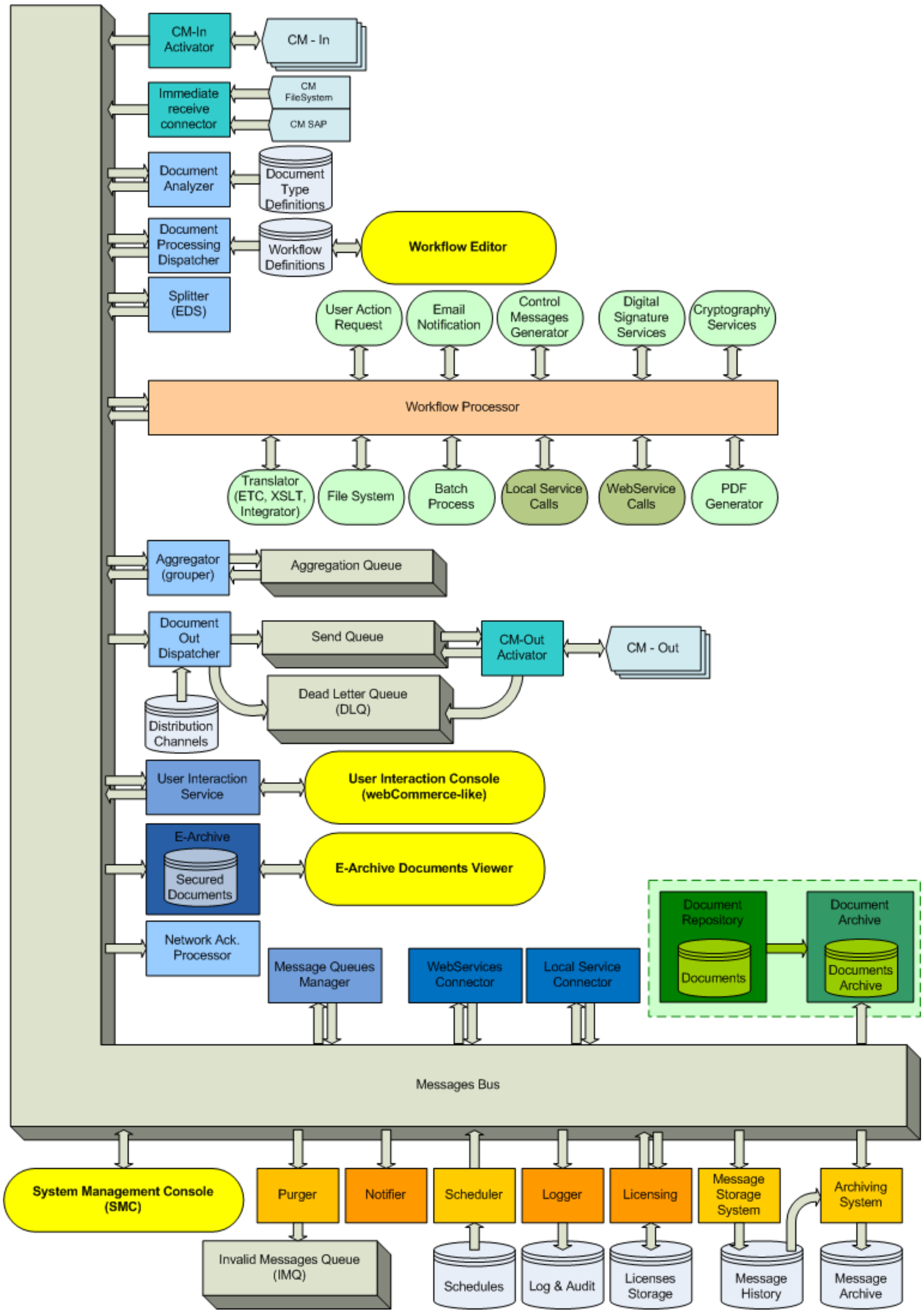


Figure 5 - TSB Message Bus

processes present in Supply Chain Management. One of the main features of the TSB is the B2B message exchanging broker able to transport B2B messages (XML, EDI, Flatfiles, etc.) from a source point to its destination, performing the adequate message transformations and routing. TSB also provides a mechanism for disaster recovery.

The bus implemented in TSB (Figure) allow different services to communicate with each other through messages. New services can be added by using the plug-in mechanism resulting in an increase of the flexibility and the scalability of overall system.

TSB provides for communication purposes producers, subscribers and polling consumers which respectively push and pull data (Figure). The more complex item is the subscriber which is compound of several local queues (one per subscriber) to attend the different demands, a mediator service to assign the tasks coming from the queues to the different workers, which are the last element. When a new message is inserted in the queue, the queue launches a "new message" event, the mediator gets the message, puts it in the "executing bag" and tries to find an available worker where the message is executed. If this search produces no result, it spawns a new worker. There are multiple workers per subscriber and when it finishes the task it is executing, it requests a new task to the mediator.

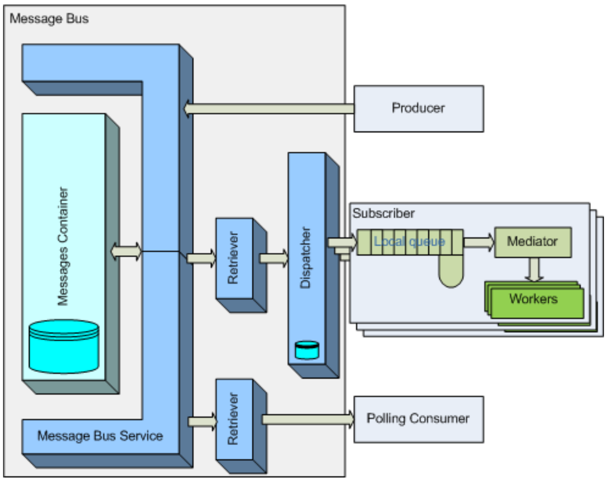


Figure 6 - Message bus in detail

In summary, TSB is a core ESB but constrained and tailored to benefit throughput and fault tolerance.

Apache ServiceMix

Apache ServiceMix[2] is a flexible, open-source integration container unifying the features and functionality of Apache ActiveMQ[3] for messaging, Camel[4] (for routing), CXF, ODE, and Karaf[5]. The result is a powerful runtime platform used to build new

---

2 http://servicemix.apache.org
3 http://activemq.apache.org
4 http://camel.apache.org
5 http://karaf.apache.org

integration solutions. It provides a complete, enterprise-ready ESB exclusively powered by OSGi[6]. See Figure for a complete picture of the ServiceMix architecture.

Apache ServiceMix is designed according to the Java Business Integration (JBI) specification. As an ESB it allows disparate applications, platforms and business processes to exchange data in a protocol-neutral way. the JBI specification (JSR 208) defines the manner in which this communication will take place.

The JBI specification requires two types of components. They are defined by JSR 208 as:

- **Service Engine (SE)**: provide business logic and transformation services to other components like e.g. transforming XML data to an HTML format
- **Binding Component (BCs)**: provide connectivity to services external to the JBI installation in the protocol used by the external application. BCs convert protocol and transport specific messages, such as HTTP, SOAP, and JMS messages to a normalized format that is used within the JBI infrastructure.

Besides the usage of JBI and its components (both SEs and BCs can be service providers or service consumers or both), ServiceMix makes use of JMI for binding components and Camel and XSLT as Service engines. The communication between the different ServiceMix components is performed via XML files using a Normalized Message Router to communicate both SEs and BCs.

[6] http://www.osgi.org

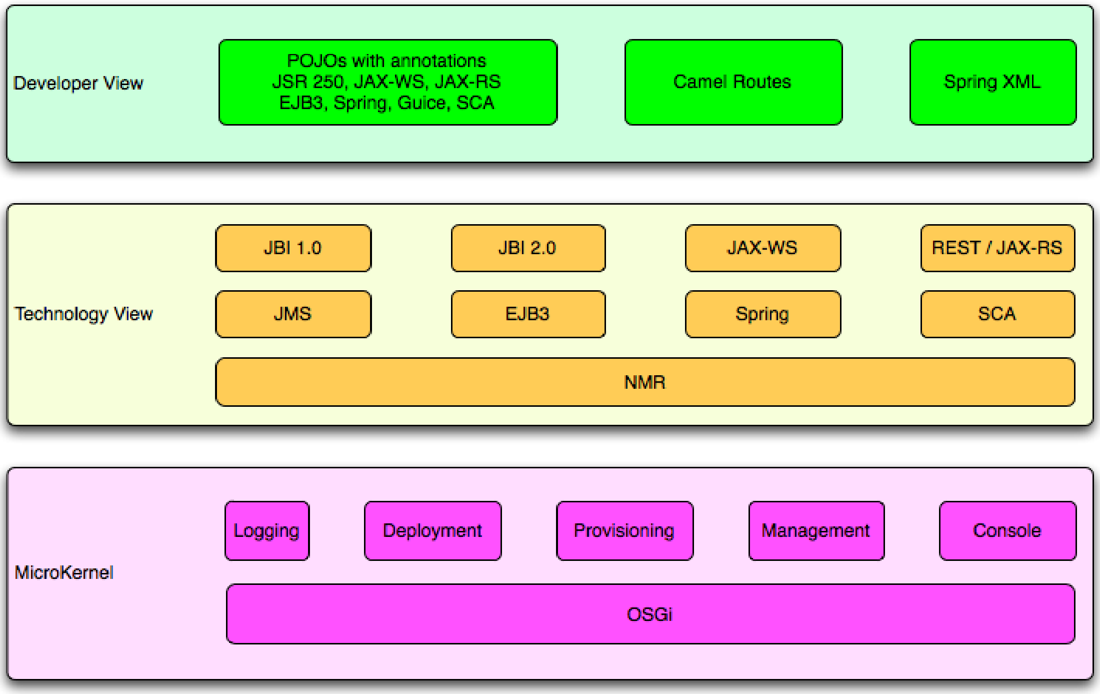


Figure 7 - ServiceMix Architecture

Apache Karaf is the ServiceMix runtime and is based on OSGi. It provides a lightweight container where hot deployment of components and applications, and dynamic configuration of services can take place. It also comes with logging and provisioning facilities as well as security services and direct native OS integration.

Apache ActiveMQ is the ServiceMix messaging server. It provides support for Enterprise Integration Patterns (EIP) in several developing languages. Regarding the messages themselves, ActiveMQ supports JMS and has REST APIs for communicating the different components plugged to ServiceMix.

Finally, Apache Camel is the integration framework used by ServiceMix which supports EIPs and performs the basic activities such as routing and mediation. It also has powerful Bean Integration.

References

1. D. Chappell, Enterprise Service Bus, O'Reilly, 2004
2. Forrester ESB 2011. (n.d.). Retrieved from http://www.oracle.com/us/corporate/analystreports/infrastructure/forrester-wave-esb-q2-2011-395900.pdf
3. Tijs Rademakers, J. D. (2008). *Open Source ESBs in Action.* Manning.

2. Technologies utilised in modern automation systems

2.1. Robotics

Robots used in manufacturing system have an increasing degree of flexibility and ability to adapt and learn compared with only a few years ago where industrial robots where highly inflexible and ever move and operation programmed in detail. In modern automation system the flexibility increases, a production cell with a robot may be used for different tasks depending on the need of the manufacturing. E.g. there are on-going evaluations of industrial robots on wheels able to move to the locations where they are needed. Also the use of industrial robots for smaller productions (short product series or many variations) is increasingly explored. This are all features of importance for the CREATE project and one of the reasons to introduce the SNM concept, but robotics in itself will not be explored in CREATE.

2.2. Artificial Intelligence (AI)

In modern manufacturing system individual components become more and more advanced and intelligent. To achieve this AI is increasingly used. Artificial intelligence is a set of algorithms and methods that enable systems to learn and improve their performance. Amongst these methods we can mention: Artificial Neural Nets (ANN), fuzzy logic, Rule Based Reasoning (RBR), heuristic planning and search techniques and Case-Based Reasoning (CBR). ANN, RBR and fuzzy logic, are used in industry for solving specific tasks and are successful within their scope of competes. CBR is a methodology where a case library with past experience, both successful and less successful is used for solving new tasks and problems. Every new solved task or problem is added to the case library with its outcome, hence the experience and knowledge increases. CBR systems often use different techniques, e.g. ANN, RBR, fuzzy logic, Bayesian nets and statistical approaches to perform its task. CBR has been used successfully for a number of applications in medical applications, decision support systems but R&D projects exploring industrial us are rare.

2.3. RFID

RFID improves process automation systems by allowing items to be tagged and traced throughout the manufacturing process. This tag and trace information can be used to reduce inventory, improve quality, and automate manual process.

Bar codes can be used to track and trace, but RFID is a better technology as RFID tags can be read in bulk at distances up to 30 feet and do not require line of sight, can store large amounts of information, and are re-writable.

Automation of manual processes: As RFID tags are applied in production, a time and date code is entered into a database for every product's serial number. With this information pallet tags can be automatically developed instead of manually keying in information.

Increased safety and quality control: Prior to RFID, plant personnel had to hold entire pallets if problems were suspected. To err on the side of safety, longer time frames were often used and more product than necessary was placed on hold. With RFID, it can easily and precisely be known what groups of products are affected during the time of a hold order. This minimizes hold orders and rework.

Increased productivity: The old process of inducting pallets into the automated storage/retrieval system (ASRS) relied on personnel to precisely place pallet tags so that scanners could read the tag when the pallet was inducted into the ASRS. With RFID, the pallet tag can be read as long as it is on the correct side, regardless of its exact location, greatly increasing the efficiency of the ASRS.

Improved shipping accuracy: Easy identification of the products and avoids errors in shipping with the correspondent save in time and money

Improved insight into inventory: Currently, counts often show differences due to numerous variables. RFID technology can help with such discrepancies, because each unit will have an individual serial number on it. This technology allows the counting of units to be accurate as they pass.

2.4. DCS

Have their background in large-scale continuous applications such as manufacturing systems, process or any kind of dynamic system and regulatory control in the process industry. DCS platforms are integrated product families usually provided by one manufacturer and include facilities for user interfaces and information management.

2.5. PLC, PAL

Were originally developed to replace hard-wired relay logic in discrete manufacturing are mainly limited to low-level control functions. Essentially a PLC is an assembly of solid state digital electronic elements designed to make logical decisions and provide outputs. They provide programmable and reconfigurable control.

2.6. HMI

The **user interface**, in the industrial design field of human–machine interaction, is the space where interaction between humans and machines occurs. The goal of interaction between a human and a machine at the user interface is effective operation and control of the machine, and feedback from the machine which aids the operator in making operational decisions. Examples of this broad concept of user interfaces include the interactive aspects of computer operating systems, hand tools, heavy machinery operator controls, and process controls. The design considerations applicable when creating user interfaces are related to or involve such disciplines as ergonomics and psychology.

A user interface is the system by which people (users) interact with a machine. The user interface includes hardware (physical) and software (logical) components. User interfaces exist for various systems, and provide a means of:

- Input, allowing the users to manipulate a system
- Output, allowing the system to indicate the effects of the users' manipulation

Generally, the goal of human-machine interaction engineering is to produce a user interface which makes it easy, efficient, and enjoyable to operate a machine in the way which produces the desired result. This generally means that the operator needs to provide minimal input to achieve the desired output, and also that the machine minimizes undesired outputs to the human.

Ever since the increased use of personal computers and the relative decline in societal awareness of heavy machinery, the term user interface has taken on overtones of the graphical user interface, while industrial control panel and machinery control design discussions more commonly refer to human-machine interfaces.

Other terms for user interface include **human–computer interface** (**HCI**) and **man–machine interface** (**MMI**).

2.7. SCADA

SCADA is a software application specially designed for working in computers in the manufacturing control providing communication with the field devices (autonomous controllers, PLCs, etc.) and automatically controlling the process from the computer. It also provides with all the information generated during the production process to different users, either the ones in the same level or the supervisors (quality control, supervision, maintenance, etc.).

A SCADA package offers the following benefits:

- Possibility of creation of alarm panels that require the presence of the operator in order to stop or raise the alarm. Event log.
- Generation of reports of the historical plant alarms that can be downloaded to a spreads sheet.
- Execution of programs that can modify the control law of a robot or even cancel or modify the tasks associated to a robot under certain conditions.

- Numeric programming possibility that allows to perform high resolution calculations in the CPU of the computer.

Data capture, signal analysis, screen prints or data storage applications can be created out of the SCADA´s main characteristics.

All the actions are actions are carried out by functions that include the possibility of using general use programming languages. This gives it great versatility. Some SCADA offer function libraries for general purpose languages that allow to customize the SCADA applications.

Requirements for a SCADA systems:

- They must be open architecture systems, capable of growing and adapt according to changing necessities of the company.
- They must be able to easily and efficiently communicate with the team in plant and the rest of the company.
- They must be simple installing programs without excessive hardware requirements and easy to use

3. Consideration of the system engineering approaches and SW tools

3.1. J2EE

Sun Microsystems´ J2EE is a platform for developing, building and deploying multi-tier, web-based enterprise application. It is based on Java 2 Standard Edition.

The type of architecture of J2EE is the distributed architecture. Applications are divided into multiple tiers such as client tier (to provide a user interface), middle tier (to provide services and business logic) and backend (to provide data management). The next figure illustrates the typical J2EE environment.

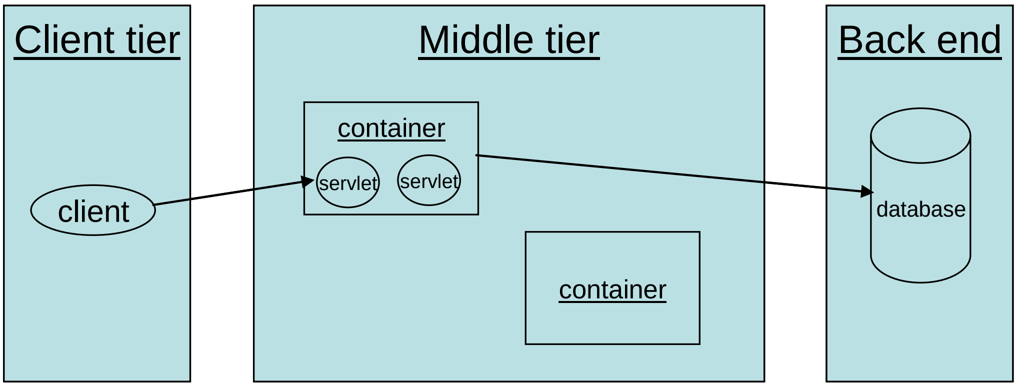


Figure 8: J2EE Architecture

J2EE platform reduces the cost and complexity of the multi-tier development. J2EE applications can be rapidly deployed and easily improved in order to facilitate the flexibility of the company.

Application components (J2EE Application clients, Applets, Servlets and services such as HTTP, HTTPS, Java Transaction API…) are managed in the middle tier by containers, which are the providers of services (this container-based model separates business logic from system infrastructure).

These series of components create the perfect settings for the development and deployment of scalable applications on the web that have the following properties:

- Portable: An application than is written in a Windows or Linux based machine can be used in any platform for which a Java Virtual Machine is available.
- Scalable: If the company increases the number of clients, new J2EE components can be added to a web application without having to re-write all the code again.

- Highly supported: Almost all the software companies have web containers or application servers with J2EE.
- Safe: While other business application models require specific security policies, the security environment of J2EE platform allows defining security restrictions in the moment of the deployment of the application. This enables to skip the step of the implementation of security policies in the applications. J2EE platform makes a great complexity of the security policies portable.

3.2. .NET

.NET is Microsoft developer model that allows the software to be independent of the platforms and the hardware. Moreover, it makes that all the data and information can be accessible by internet. The basic infrastructure that is below the .NET is the .NET Framework.

One of the main features of .NET is the open architecture. This platform generates and executes the next generation of Windows and Web apps.

This environment supports a myriad of languages, and can profit from the high predisposition to accommodate to next generation languages. This makes .NET a solid solution for present and future developments. In this sense, it also represents an ideal candidate for migration and reengineering projects where applications are required to relocate their infrastructures. Another benefit from .NET is the variety of options provided to developers, allowing them to select the most suitable language depending on the requirements of the task. This usually leads to an increase of their productivity.

NET main technologies are:

1. NET Framework: This is the structure where the applications and Web XML services are developed and executed. The basis of this framework allows all the applications to be developed with the same suite of tools and codes, providing the needed environment to be easily integrated.

The two main components of the .NET Framework are the Common Language Runtime (CLR) and the Base Class Library.

2. Windows Server System: A type of industrial server, integrating, executing, operating and managing applications and services Web XML.
3. Fundamental building blocks services: a set of user-centred Web XML services moving the user data control from the apps to the users.
4. Visual Studio .NET: this tool allows the generation of WebXML services, Windows and Web applications to improve the user experience.

In the Figure 9, it is displayed the relationship of the CLR and the class library to the applications and the other components of the systems.

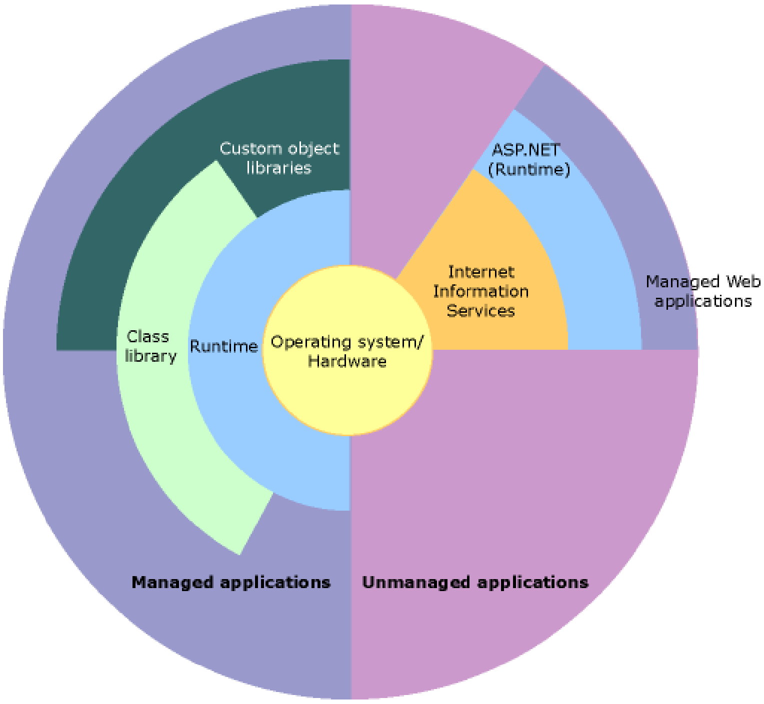


Figure 9: .NET Framework (from: www.msnd.microsft.com)

3.3. EAI, B2B

Development of interconnection middleware solutions synchronous / asynchronous systems of information between the manufacturing plant and ERP / MRP / SCM company.

3.4. Decision support systems (DSS)

A DSS is a computer-based information system that supports business or organizational decision-making activities. DSSs serve the management, operations and planning levels of an organization and help to make decisions, which may be rapidly changing and not easily specified in advance.

DSSs include knowledge-based systems. A properly designed DSS is an interactive software-based system intended to help decision makers compile useful information from a combination of raw data, documents, and personal knowledge, or business models to identify and solve problems and make decisions.

Typical information that a decision support application might gather and present are:

- Inventories of information assets (including legacy and relational data sources, cubes, data warehouses, and data marts),
- Comparative sales figures between one period and the next,
- Projected revenue figures based on product sales assumptions

Decision support systems helping operators, technicians and engineers to perform their tasks better in manufacturing is rare but there are a number of examples. SNMs may include DSS to increase overall performance if humans are part of an SNM.

3.5. ERP, CRM

Enterprise resource planning (ERP) systems integrate internal and external management information across an entire organization, embracing finance/accounting, manufacturing, sales and service, customer relationship management, etc. ERP systems automate this activity with an integrated software application. Their purpose is to facilitate the flow of information between all business functions inside the boundaries of the organization and manage the connections to outside stakeholders.

ERP systems can run on a variety of computer hardware and network configurations, typically employing a database as a repository for information.

Customer relationship management (CRM) is a widely implemented model for managing a company's interactions with customers, clients, and sales prospects. It involves using technology to organize, automate, and synchronize business processes—principally sales activities, but also those for marketing, customer service, and technical support. The overall goals are to find, attract, and win new clients; nurture and retain those the company already has; entice former clients back into the fold; and reduce the costs of marketing and client service. Customer relationship management describes a company-wide business strategy including customer-interface departments as well as other departments. Measuring and valuing customer relationships is critical to implementing this strategy.

3.6. Requirements for future automation architectures

To summarise the discussion of the previous sections, there is a need for an integrated, flexible and low-cost system platform that supports design reuse and allows intelligent features to be easily implemented. To make this possible, the following general requirements should be considered:

- In order to guarantee sufficient semantic coherence and interoperability, applications should be based on the common concepts and reference architectures of the intended application domain.
- The platform should support reuse of design and software by individuals, companies and application domains. In addition to custom-made components and vendor specific libraries, it should be possible to use independently developed Commercial Off-the-Shelf (COTS) components.
- The use of common hardware and software standards is critical for reducing development costs and time to market, as well as for improving system reliability. Standards are also the way to easier integration with external systems and, thereby, larger component markets.
- New design tools should provide application-oriented programming languages with powerful interaction mechanisms and hide unnecessary implementation issues. For instance, this includes name-based addressing, location transparency and the freedom to define detailed system functions before deciding the hardware structure.
- An industrial system must be scalable from small stand-alone control and data acquisition units to integrated applications with hundreds of nodes. The platform should apply horizontal peer-to-peer (P2P) communication instead of centralized system architecture. Due to functional integration, it must be possible to combine different types of functions, such as real-time control, parameter and software modifications, historical data storage and reporting.
- Industrial applications must be maintained for periods longer than the life-time of their individual components. So, in addition to easy component integration, the platform must provide services for remote diagnostics and maintenance. Further, flexible plug&play mechanisms are needed for adding, updating and removing components without interruptions in system operation.
- Security and reliability are critical requirements for an industrial application. The system must be protected from unauthorised access and denial of service attacks. In addition, malicious and incompatible components must be detected before their installation. On the other hand, a mechanism is needed to protect the intellectual properties of component developers. In order to cope with the often harsh operating environments and communication uncertainties, control systems should have built-in mechanisms for detecting problems and management of various abnormal situations.

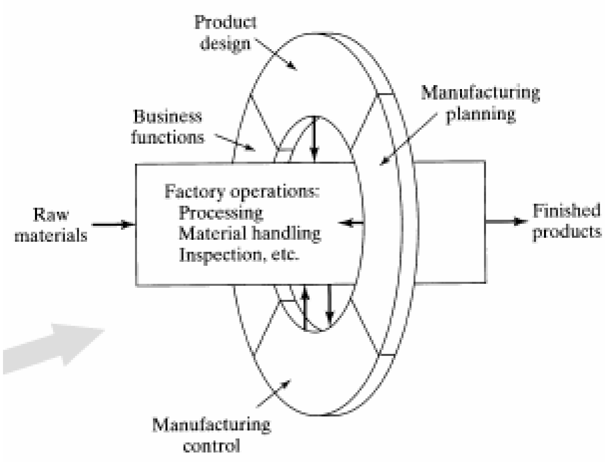


Figure 10 Integration between infrastructure and support systems

4. Most relevant technologies and identification of technology gaps in:

This section aims to identify and concrete the important technologies currently used as well as the gaps and improvements to be solved in the CREATE project for each demonstrator.

4.1. Flexible material Flow systems (CCM)

Technologies that can be used in the CREATE project for a flexible material flow system, completely based on services from SNMs, can be grouped in the following categories:

- Material flow with flexible routing
- Material flow with extended functions and decision support
- Material flow with adaptive processes

Generally, there are no technology gaps identified, and the surplus value of the CREATE project is the availability of the technologies as services offered by SNMs, and the facility to build systems with these services that incorporate flexible material flow.

Material flow with flexible routing

Definition: The material flow can be routed along different flow paths.

Pick-and-place robots

Any device that is capable to pick a material from any location, and to place it to any other location, is considered a pick-and-place robot. The material flow is flexible when the number of locations is 3 or more. A typical application is in warehouse repositories, the actual number of existing applications is quite large.

Conveyor belt (de)multiplexers.

In a conveyor environment, devices exist that multiplex material flow from multiple belts into a single belt, or demultiplex material flow from a single belt into multiple belts. Alternative devices perform similar tasks, by pushing materials onto or from a single belt.

AGV's

An Automated Guided Vehicle (AGV) is a mobile robot used in industrial applications to move materials around. With AGV's, it is possible to move material between devices that have no physical interconnection.

Manual product exchange

In a mechanical material flow environment, devices are present that allows human workers to take away products from, or to place products into the material flow.

Material flow with extended functions and decision support

Definition: Functions that can be added to an existing material flow system, with the purpose to influence the material routing.

Product identification

Product identification is a technology that identifies material, and can be used to decide on the progress of the material flow. Product identification can be based on a deliberately attached tag (like RFID, barcode, datamatrix code), or by some product recognition technology (like vision).

Product qualification

Product qualification is a technology that qualifies material, and also can be used to decide on the progress of the material flow. Product qualification is generally based on one or more physical measurements (like: weight, size). More and more often, product qualification is implemented on the basis of vision technology.

Material flow with adaptive process

Definition: A device can perform multiple operations on multiple types of materials.

Smart stations

A smart station is a device with at least one of the following capabilities:

- Perform multiple operations (process steps) on a single type of material or product.
  Example: SMT placement equipment, where many surface mount components are placed on one carrier product.
- Perform a single operation on multiple types of materials or products. Example: Automotive production machine, where different vehicle types are produced in a single material flow.
- Perform a data-dependent operation on a material or product. Example: An invoice printer, where each printing operation makes a product unique.
  Notes:
    - The data-dependency can be either to another device (e.g. supporting product qualification) or to the system framework (e.g. production on demand).

In case of adjustment or calibration data-dependency, the product is not necessarily made unique: the data is necessary for the process, not for the product.

4.2. Industrial Metrology (TRIMEK)

Given the global market where Europe´s SMEs are competing, fast reactions, comprehensive quality, efficient use of resources and sustainability are the four pillars that support the modern industry. In order to achieve this, CREATE proposes an improvement in the production lines. This improvement is based on the integration of the high level business software with the lowest one at the tool machine level.

Efficiency in production lines is directly related to the capability of adaptation and reconfiguration. The market is tending to demand customized products instead of pre-designed standard ones. In order to produce smaller and more customized batches, tool changing downtimes and software reconfiguration periods have to be reduced significantly. Otherwise the impact in the final price will be too high to be competitive in a global market.

The industrial metrology use case to be developed in CREATE works in this direction. It will consider the Smart Neighbourhood Modules (SNMs) to develop intelligent, on-line decision making production lines that can easily be adapted to the manufacturing of different products. In this case, the proposed modules are going to be applied to scenario based on a production line of EPC, which is a supplier of camshafts for the naval and automobile industry. These pieces, due to its working conditions, are required to meet high quality dimensional requirements; for that reason TRIMEK will provide M3 inspection system with DATAPIXEL´s camera and STAUBLI SPAIN its know-how in robot controls and communications. From the software point of view, CBT will develop the metrology programs and trajectories and Innovalia Association will develop the communication channels and. On the other hand CEESA will adapt its ERP to meet the new demands of flexible production lines. Ultimately, SQS will test all the SW developed.

The well-known strict supply management that these short of industries demands and the high specialization and customization, there is a clear opportunity for EPC to develop an improved production line to meet the demands its clients. The capacity of real time control and adaptation that SNMs provide will be applied in the inspection machines to make automatic on line decisions with the aim of maximizing the potential of the tools available at EPC. The SNMs inside the inspection machines and the rest of the machine tools will be able to execute different actions for each manufactured piece depending on the results of the dimensional measurements and the QR codes used to identify each piece. Actions such as filtrate them given their different manufacturing errors, decide whether these errors are fixable or not and separate them in different production lines or send them back to re-manufacturing, communicate with the machine tools and modify its parameters to meet the client demands for each piece.

An important development, apart from the contribution for the **automation** of the line, is the flexibility that provides the use of QR codes to identify pieces and its manufacture requirements. Given the identification of the camshafts, specific test or manufacturing steps can be done for each unity. As the machines are directly connected, not only among themselves, but also with other departments of the company, a customization proposed directly by the client can be updated in real time so that the company can adapt their products to the changing client's needs.

Another contribution to the **flexibility** provided by the SNMs, is the ease of addition of new machines. If the demand requires new tools to manufacture and inspection the camshafts (e.g. the camshaft will go in a lower power engine so the shaft can be thinner and it requires a longer time grinding process) thanks to the SNMs, the new tools will easily be adapted and configured to the line and already existing machines will also change their performance to adapt to the new coming machine.

The next figure shows the scheme of the proposed production line.

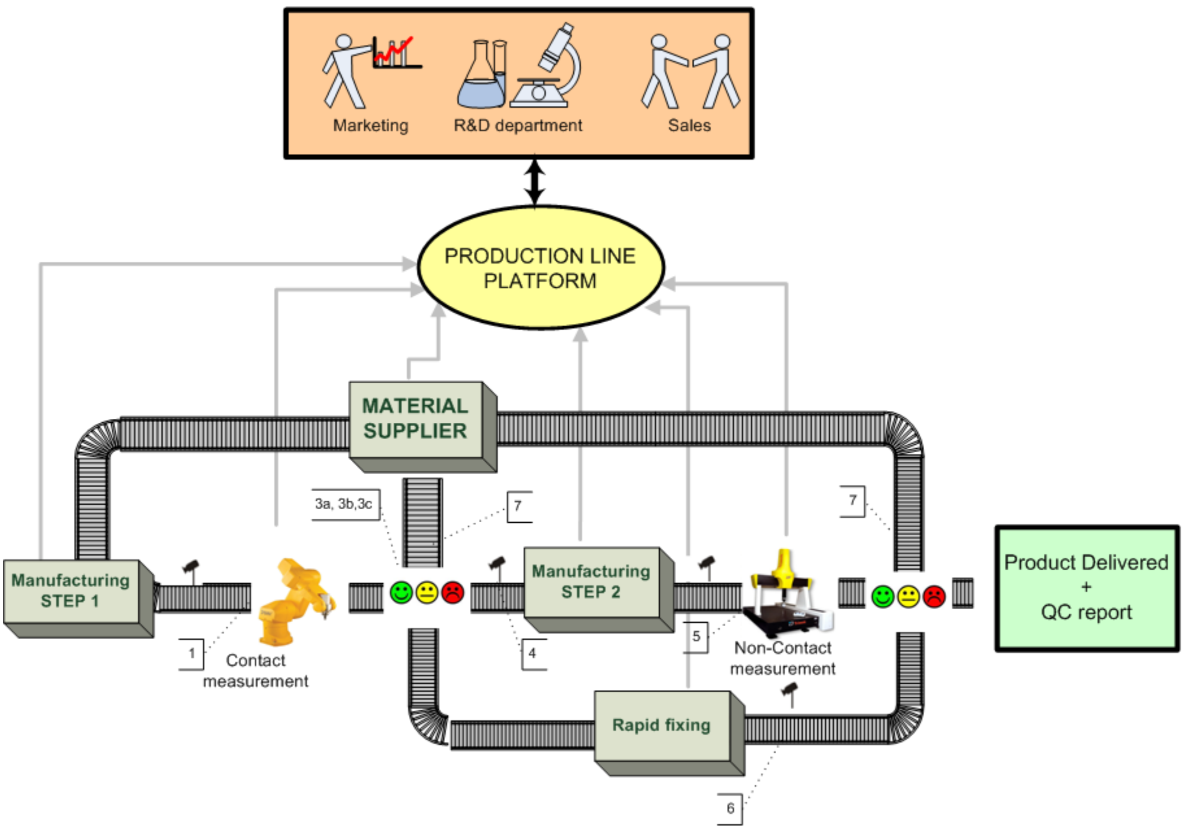


Figure 11 Industrial metrology use case

According to the figure, the proposed production line would work as follows:

1. After the first manufacturing step, a simple camera reads the product QR identification code. Once the code is read, all the information of the piece will be available to the machine. This information contains the theoretical dimensions of the piece, material, future working conditions and client. This data is real time updated so that CEESA´s ERP will be able to always locate the piece, steps it has gone through, time and steps left to be finished and energy and materials utilised.
2. Once identified, the next step is a fast inspection carried out by a robot equipped with a contact inspection machine. It will conduct a rapid measure of the most critical dimensions (few pre-defined points) given the information of the piece.
3. Once the robot has measured the piece, three different results can be possible:
4. The piece does not meet the expected dimensional requirements and it is not worth fixing them. Therefore the piece goes back either to the first step or to be recycled.
5. The piece does not meet the demands and tolerances but they are still fixable by a rapid fixing machine prepared for these cases.
6. The piece does meet the dimensional requirements, either because the errors are fixable in the next (grinding) process or because it is correctly manufactured.
7. Next manufacturing process of the piece. Each of the machines, thanks to the SNMs, will always be connected among each other and the server. As the pieces can be different, all the machines will adapt their performance to the manufacturing needs.
8. After the final machining, the optical inspection machine will perform a total scanning of the piece in order to generate the Virtual Part (VP) and then, detect any kind of failures that could lead to not meeting the expected results. Moreover, the virtualization of each piece so that it can easily be compared to the theoretical one, so that discrepancies can be easily located.
9. After the optical measure, other three different results can be possible, similarly to the third step.
10. The piece does not meet the expected dimensional requirements and it is not worth fixing them. Therefore the piece is recycled.

11. The piece does not meet the demands but they are still fixable by a rapid milling machine prepared for these cases or by second grinding step (therefore it would pass through the same grinding cabin again).
12. The piece meets the expected demands and it is ready to be mounted in the engine.
13. In case that the pieces have errors that are difficult to fix, they will be recycled.

Trimek´s M3 inspection centres will adapt their software to the standards proposed by CREATE in order to facilitate the communication between SMNs. The software will be real time connected to the rest of the machines and departments and will automatically detect what parts are worth to be inspected more carefully.

In case that a new machine (both manufacturing and quality control) has to be placed due to production requirements, the special configurations of the SNMs will automatically configured the new work station. To facilitate the communications between machines standard devices and protocols will be utilised. The connexions among themselves will be based on PROFIBUS and ETHERNET protocols.

Special focus will be set in the aspect of **compatibility and scalability** of the software and communication standards due to the fact that CREATE aims to apply these production lines to the manufacture of different products. The system will be prepared to grow and to be accessed and configured from different users and from different platforms.

4.3. Monitoring and Quality Control (VOLVO)

Monitoring, Quality Control and Diagnostics (MQD) is an essential part of manufacturing today and it is often too costly to allow producing products not meeting all requirements. Monitoring and immediate adjustment or corrections is needed to secure an overall high output. Today both monitoring and adjustments are surprisingly **often manual even if modern tools and sensors are used such as laser measurements or** sound/ temperature/ current, the result is often based on experience of individual technicians in the production line. The result therefore varies largely based on individual experience and skill often acquired over long time and by past mistakes and successful adjustment and corrections.

SNMs with the task to perform MQD in manufacturing will be an important part in future automation systems. SNMs with MQD tasks will also make decisions or assist in decision support for corrective and preventive actions. By giving SNMs increased intelligence by enable them to collect, reuse and share experience, the manufacturing system will learn and improve performance. Also the manufacturing history may be preserved since certain action on a produced product later on may be linked to certain future problems on the after market, e.g. transmissions that where manufactured in a specific way may show up better performance in warm climate or car doors with parts from a certain subcontractor may be easier to repair after a car accident, valuable facts that can be used to improve product/production/after market.

By giving SNMs the ability to learn from past cases they will be able to prevent mistakes from occurring more than once as the experience. By sharing experience with other SNMs performing similar tasks experience will spread enabling the production system to learn and improve beyond individuals SNMs. In discussions with technicians and engineers, reoccurring mistakes is one important key for cost efficient and competitive manufacturing.

One of the key methods that are explored for adding intelligence is Case-Based Reasoning (CBR) based on a cognitive model of how human learn. The advantages are that similar to humans and before sufficient data has been collected for use of quantitative methods, the CBR approach is able to solve problems and find solutions. Recorded sensor signals and following decision connected to a specific manufactured part enables feedback to previous steps in the manufacturing and design process. For example if certain problems occur more or less frequently after a design alteration, the design department will immediately be notified about it and necessary actions can be taken to further improve the product or production process.

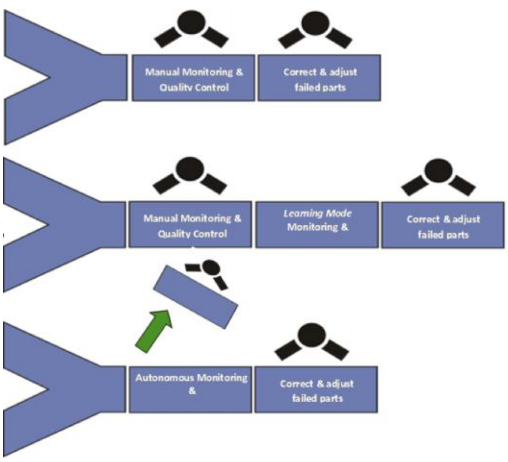
SNMs performing MQD and communicating with other SNMs in earlier steps in the manufacturing allow necessary adjustments to secure quality. Also communicate with SNMs later on in the manufacturing process enable necessary steps to ensure the final quality is within given parameters.

SNMs may be small or large, include basic or advanced sensors, include advanced computer programs collaborating with experienced technicians or SNMs may be fully automated with industrial robots. SNMs may be able to perform more than one service and do what is mostly in demand in the current manufacturing. SNMs will also communicate with other systems, e.g. design department (if certain quality problems increase it may be caused by factors outside the manufacturing, change of design), production planning (if quality problems exist leading to delays) or manufacturing management (frequency of certain problems may be caused by some changes in manufacturing on a meta level to the SNMs and require alterations/changes on a higher level. For this a SNM may need ability to communicate with both other SNMs, with humans helping that may perform a part of the SNMs task, with management e.g. via a web interface. Some of this communication may involve asking humans for advice if actions are outside the SNMs ability or authority. It may be argued that some of these features already exist in some manufacturing equipment but by using the concept of

SNMs we move towards a new concept and take a step towards a new paradigm in manufacturing where SNMs collaborate dynamically to meet production requirements.

In Volvo we will develop SNMs with complex tasks that are able to improve their performance based on experience and communicate both with technicians when help is needed to solve a difficult task and communicate with production management (system or human) so necessary actions can be taken to meet long term production plans. One SNM will be developed at Volvo Car Corporation and one by Volvo Construction Equipment. Each SNM will not include all the important features of an MQD SNM, instead a number of features have been selected to explore.

The following figure is a schematic example how a SNM learns by observing a human operators decision and once the SNM has collected sufficient experience to perform as well or better than a human operator it can take over the MQD task. Both Volvo SNMs are based on this principle.



Flexible Monitoring, Quality Control and Diagnosis MQD

If a production series is small manual MQD may be most cost efficient.

When production increases, the best solution may be a MQD SNM. The SNM learns from past cases.

When the SNM has learnt enough and proven to perform to an acceptable standard, equally good or better than a human operation, then the SNM can start performing the MQD task autonomously and only ask for help if needed.

Figure 12: Example for Flexible Monitoring, Quality Control and Diagnosis

A basic example of a MQD SNM would be of a scale after a manual quality control where the operator types in the diagnosis and let the production line take care of further corrections. The scale finds a correlation between certain diagnoses made by the worker, e.g. right side of product -3 gr. is classified as a missing rubber damper in the product. Once the intelligent learning algorithm and the sensors are able to diagnose as well or better than a human (monitoring work is often highly monotone and often leads to mistakes) it is able to make an autonomous diagnosis, and the manual MQD can be moved to a place in the production where it is of greater benefit. Manual MQD may be suitable for small series but in bigger production batches it quickly becomes a considerable cost. In CREATE we will apply learning algorithms, artificial intelligence and case-based reasoning and move the front line of what can be done forward and use the two Swedish demonstrators to prove the concept. We will develop generic novel methods able to learn MQD fast and reliable and able to perform MQD for a wide variety of tasks using sensors such a laser sensors, sound, vision etc.