

Executive summary:

The strategic objective is to strengthen EU leadership in production technologies in the global marketplace by developing innovative SELF-LEARNING solutions to enable tight integration of control & so called secondary processes (e.g. maintenance, energy efficiency) of production systems. The project developed highly reliable and secure service-based SELF-LEARNING solutions aiming at that integration. Approaches based on SOA principles, using distributed networked embedded services in device space, are the most appropriate for implementation of such SELF-LEARNING solutions. Context awareness, providing information about the processes and equipment and circumstances under which the services operate and allowing them to react accordingly, is a promising holistic approach to assure needed SELF-LEARNING adaptation to changes in processes and equipment states.

The key components of a SELF-LEARNING solution include:

- Context Extractor module: to allow for dynamic context extraction, processing and storage.
- Adapter module: to allow for a holistic process control of considered systems. The module is in charge of real time adaptation of control parameters, maintenance plans/execution and identification of parameters.
- A self-learning module: to allow for SELF-LEARNING relying on data mining and operator's feedback to up-date execution of adaptation and context extraction at run time.

The context modelling is following the ontology approach. The purpose of the SELF-LEARNING ontology is to define a fundamental data model for context extraction. The Context Extractor uses all monitored 'raw data' provided via the data access layer to derive the machine's current contextual situation. Using the ontology/context model the monitored data is evaluated and the context extracted. The Adapter is informed by the Context Extractor about a variation on the system (change of context) and adapts the system to handle the new reality. The Adapter is guided by a set of rules that describe how the system should behave in each particular context. These rules can be updated by learning based on lifecycle history data, context and user validation. The service infrastructure is based on the Security Framework concept which assures safe Information flow, data isolation, damage limitation. The secure multi domain monitor may have additional use beyond supporting the SELF-LEARNING solution.

The project is driven by three industrial application scenarios - business cases (BC), including both production equipment manufacturers and users, addressing different aspects of SELF-LEARNING holistic control of complex automation systems/manufacturing lines, in order to assure industrially relevant development of SELF-LEARNING production systems:

1. BC1 (Bosch Rexroth) - SELF-LEARNING idle-time recognition of machine tools
2. BC2 (DESMA) - SELF-LEARNING parameter identification for intelligent monitoring and adaptation of machines/automation systems for shoe industry
3. BC3 (FASTEMS) - SELF-LEARNING scheduling and dispatching in Flexible Manufacturing Systems (FMS) for automotive industry

The targeted SELF-LEARNING solutions are likely to have a high impact on manufacturing industries in terms of reduction of time and efforts needed for development/installation of control systems of production systems,

reduction of down times during product exchange and conflict situations,
reduction of maintenance time and costs, increase of Overall Equipment
Effectiveness (OEE) i.e. increase of plant availability and productivity.

Project context and objectives:

1.2.1 Project context - background

Modern highly flexible manufacturing processes require both effective control and secondary processes of production systems in order to assure their highest availability and efficiency. Currently, so-called secondary processes (e.g. maintenance, energy efficiency) and production control activities are often separated, leading to low efficiency and high costs for both users and manufacturers of production equipment, especially for those acting at the global market. As indicated in Manufuture (2006), merging of the world of secondary processes and the world of control in production systems may lead to enormous benefits regarding efficiency of manufacturing processes and maintenance activities, as well as regarding flexibility, equipment availability etc. Such an approach is also in full compliance with the lean production principles (LEADERSHIP, 2008). There is a clear need to establish a real-time management of secondary processes (see Manufuture, 2006) in order to open possibilities for new services and new business models. This is of a special importance for equipment manufacturers aiming at maintaining leadership of European manufacturing technologies.

In order to merge the two worlds in an effective way, the monitoring and control of production systems must exhibit high SELF-LEARNING adaptability to fluctuation in a flexible manufacturing process (e.g. product mix), changes of process and equipment parameters. Changes in process and equipment parameters have to be effectively identified and control and maintenance activities have to be adapted/synchronised accordingly. Such an integration of control and maintenance is related to considerable organisational changes, specifically within an Extended Enterprise (EE), addressing a needed tight collaboration between equipment manufacturers and users. On the other hand, in order to support this integration, implementation of complex models and algorithms for such SELF-LEARNING production systems require powerful ICT infrastructures. A service oriented approach opens entirely new perspectives in the embedded networking for SELF-LEARNING & self-adapting production systems. It is likely that approaches based on Service Oriented Architecture (SOA) principles, using distributed networked embedded services in device space (sensors, controllers etc.), are most appropriate for the implementation of SELF-LEARNING production systems in general, and especially those aiming to merge control & maintenance.

1.2.2 Strategic objectives

The strategic objective is to strengthen EU leadership in production technologies in the global marketplace by developing innovative SELF-LEARNING solutions to enable a tight integration of control and maintenance of production systems.

The objective of the SELF-LEARNING project is to develop highly reliable and secure service-based SELF-LEARNING production systems aiming at merging the world of secondary processes (e.g. maintenance, energy efficiency) with the world of control, i.e. aiming at holistic process control. Secondary (support) Processes are business processes that produce products or support primary processes that are invisible to the external customer but essential to the effective management of the business [Rummler 1995]. The key assumption of the project is that a

context awareness approach will allow for the adaptation and integration of control and secondary processes of production system.

To allow for such a breakthrough the specific project objectives are to be developed:

- SELF-LEARNING production systems able to self-adapt in response to changes in the context in which they operate, including changes in process and equipment parameters,
- Methodology addressing both organisational and technical aspects of such a radical change in production systems,
- SW service based infrastructure for the implementation of such SELF-LEARNING systems in manufacturing.

The project is driven by three industrial application scenarios (business cases - BCs) from 3 industrial beneficiaries, where SELF-LEARNING components and SOA infrastructure for integrated control and secondary processes are applied for machine tools, FMS in discrete manufacturing.

1.2.3 Project concept and objectives

The classical manufacturing concepts, often assume a 'division' of control and secondary process (e.g. maintenance) of production systems (complex assembly/manufacturing lines).

The objective of the project is to integrate these 'activities' by provision of a SELF-LEARNING adapter which 'adapts' and synchronizes both control and secondary processes. The challenge is to define an adapter able to handle wide scope of 'disturbances'/changes coming from either enterprise level, or process and equipment parameters changes, requiring harmonized adaptations of both control and 'secondary' activities. The proposed approach is to (on-line) identify current dynamically changing context in which the production system operates and to 'use' this identified context to adapt both control and secondary processes. Therefore, the proposed approach includes a SELF-LEARNING context extractor (as a generalized 'observer' providing current context) and an adapter (as 'active' part).

SOA principles are the most appropriate for implementation of such SELF-LEARNING production systems aiming to assure holistic control of processes.

Context awareness, providing information about the process & equipment and the circumstances under which the services operate and allowing them to react accordingly, is a promising approach to assure the needed dynamic SELF-LEARNING adaptation to changes in the context, including changes in processes and equipment parameters. The context can be seen as a bonding element between control and secondary processes of production systems and different services/networks.

This approach has not been explored up to now. The basic assumption is, that holistic, simultaneous and harmonized usage of context awareness, based on (on-line) extraction of a current context, for self-adaptation of production systems (aiming at holistic process control), is the effective way to assure considerable advantages regarding efficiency and availability of production systems, when applying SELF-LEARNING systems.

Therefore, the S/T objective of the project is to use context awareness to allow for self-adaptation of production systems, integrating control

and maintenance of production plants, the objectives of the project were to:

- elaborate SELF-LEARNING solutions for adaptation of integrated holistic process control, including a set of prototypes of high level SW services for real-time SELF-LEARNING adapter aiming to dynamically adapt
- control parameters (both feed-forward and feedback loop parts), and/or
- management of secondary processes in production systems (both planning and execution), as well as
- parameter identifier (identifying/monitoring of manufacturing/assembly process-es and/or tools/parts parameters/characteristics),
- develop a dynamic context model and a prototype of the real-time SELF-LEARNING context ex-tractor needed for self-adaptation of production systems (adaptation of SW ser-vices for parameter identifier, control and secondary processes management based on current context),
- develop a methodology for the introduction of the holistic process control approach, and
- provide a SOA infrastructure for the implementation of services for the proposed SELF-LEARNING production systems.

The project addresses the key S/T problems:

- a) self-adaptation ability of production systems, based on SELF-LEARNING algorithms, in response to changes of context for both control and maintenance activities,
- b) organisational problems related to holistic process control activities,
- c) context awareness as a prerequisite for such dynamic self-adaptation
- d) SOA infrastructure including a security & trust framework for embedded services in manufacturing industry.

Project results:

1.3 Scientific and technical results

1.3.1 The key results

The project is seeking to maximise the benefits to the European manufacturing industry and is making the core project technologies available in open source format to enable broad industry access and exploitation. The following core technologies of the SELF-LEARNING system will be made available as open source.

- Prototype of SELF-LEARNING Adapter
- Prototype of Context Extractor
- Prototype of Service Infrastructure

Parts of the open source products resulting from SELF-LEARNING are extensions to existing open source products (e.g. Infrastructure) and as such will also be submitted as updates or specific versions to existing open source repositories.

Submission to standards body: The following SELF-LEARNING project results will be published and submitted to the appropriate standards bodies as revisions or extensions to industry standards.

- Methodology for context modelling and extraction. This specification describes the methodology to select and customise context modelling, and the SELF-LEARNING context extracting approach by providing a description of the services and algorithms for SELF-LEARNING context extraction. The overall concept is presented and each of the services are described.
- Specification of Service Infrastructure. The specification addresses the underlying framework to support the deployment of SELF-LEARNING adapters and context extraction processes. The document describes the design of a decentralised and verifiable security framework that is scalable and able to integrate context data sources and support adaptations of control systems while maintaining the required security protocols and integrity of the various operational domains.

Published specifications: The following SELF-LEARNING project results will be disseminated as published specifications in order that other technology vendors and industrial technology providers are able to create components and platforms that exploit or interoperate with SELF-LEARNING technologies.

- Methodology of Integrated Control & Maintenance
- Methodology for Context Modelling and Extraction
- Specification of Service Infrastructure

Commercial products: The following project results will be made available as commercial products from three industrial partners within the project that provide production systems to manufactures in Europe and around the globe.

- SELF-LEARNING system for machine tools
- SELF-LEARNING system for machines/automation systems
- SELF-LEARNING system for FMS Cells

The project has developed and makes available training materials that support industrial use of the SELF-LEARNING core technologies and supports industrial exploitation of the project results.

1.3.2 Approach applied.

Shortly, the steps of that process were:

1. Detailed analyses of the application cases by the three industrial partners.
2. Creation of the textual descriptions of the application cases and extraction of needs and requirements.
3. Collection of the information/insight into the market available solutions and into the state-of-the-art of corresponding application.
4. On top of that the RTD performers have created an in-depth analysis of the state-of-the-art R&D activities in the relevant areas, what was used, enriched by the expertise (of RTD performers), for creation of a generic set of requirements and generic application scenarios.
5. All participants in the above described activities have also provided technical visions and innovation ideas to complete the generic requirements. The attempt was done to introduce the long-term visions for the future improvements of the solutions.
6. The generic scenarios are to be observed also as a kind of contribution to the generic requirements upon the SELF-LEARNING components functionalities.
7. Based on the defined requirements the key SELF-LEARNING components are specified in de-tail.

Based on the defined concept, the Implementation framework and services have been specified and developed, where following steps were taken:

1. Based on the defined concept, the Early Prototype (EP) of the core services and architecture was implemented. In parallel to EP specification BCs were specified.
2. The early specification of the BCs included elaboration of (limited number of) instances/scenarios, Application Specific Services and Core Services needed in each BC and their customisation, as well as elaboration of the BCs infrastructures.
3. Based on feedback on services within BCs, full prototype was specified and implemented, which also includes full specification of BCs.
4. The implemented FP of core serves and architecture were customised and combined with the application specific software within each BC and tested by the users. The results of testing were used to finally enhance the services.

1.3.3 SELF-LEARNING ICT concept

As the figure indicates, the SELF-LEARNING environment consists of conceptually several different layers, mainly focused around the Core Services. The SELF-LEARNING project addresses a generic solution for context based self-adaptation of production systems. Project results derive required features and functionality for the overall SELF-LEARNING architecture to meet industry specific needs, driving the applications of common project results to the wider scope of discrete manufacturing industries.

Context awareness allows for dynamic self-adaptations and learning in production systems. The system adapts to run time critical contextual changes and learns from it. Learning is also en-hanced by operator's feedback and experiences.

The architecture is designed following SOA principles as an add-on to the standard control following the conceptual approach.

The components of the proposed system include:

- Context extractor, adapter and SELF-LEARNING services - see the text to follow

- Expert collaboration platform (user validation) module: the identified solutions are required to be validated by user, where the user can manually/automatically accepts/rejects any new solution. The user validation UI sends the feedback to the adapter and learning module.

- Evaluator: Performance of adaptation and context extraction are measured by the evaluator either manually via operator's feedback or automatically via mapping against objective functions at run time. Evaluation results are sent to the Learning module.

- Data access layer: Generic component, responsible for accessing the device layer (data from shop floor infrastructure). Information from ERP level, devices or plant data servers are brought to data access layer directly or via middleware depending on plant specific equipment and communication protocols.

- Data Processing: These services are responsible for the bidirectional processing of information and perform e.g. pre-processing of monitored raw data acquired via the data access layer, before the context is identified. Main functionality is to transform the raw data in a format which serves as basis for context identification. The Model Repository contains ontology based plant specific models for equipment, production processes and products. The models are shared by different software components at run time. The Context Repository allows update and storage of extracted/processed contextual information for later retrieval. Information flow among the modules is event driven in some cases and time based in other cases.

- Service Infrastructure: underpinning framework ensuring information is securely gathered from trusted context data sources and that the control updates are securely communicated to control systems with appropriate levels of authentication. The communication authentication components ensure seamless and secure connectivity with existing manufacturing and information system communication protocols and security mechanisms.

- Middleware: Information from ERP level, devices or plant data servers are brought to data access layer directly or via middleware depending on plant specific equipment and communication protocols.

The SELF-LEARNING system has been implemented as a generic system thanks to an SOA approach and to the context model, making it easy to adapt it for different organisations and contexts. For this, different knowledge base could be produced to adapt the system usability in specific context. Specific tools can be connected to the SELF-LEARNING services to respond to the specific or-rganisations' needs. The SELF-LEARNING solutions can be deployed in different contexts. In the scope of the project, it was deployed to three different solutions.

1.3.4 SELF-LEARNING components

Component/result description

Context Extractor Use all monitored 'raw data' provided via the data access layer to derive the machine's current contextual situation. Using the ontology/context model the monitored data is evaluated and the context extracted. Based on the identified context, situations can be compared to previous ones and stored.

Context model/ontology The context modelling is following the ontology approach. The purpose of the SELF-LEARNING ontology is to define a fundamental data model for context extraction. Basically, the SELF-LEARNING ontology defines two ontologies: Generic Device Context Model and sector-specific context ontology.

Adapter informed by the context extractor about a variation on the system (change of context) and adapt the system to handle the new reality. The Adapter is guided by a set of rules that describe how the system should behave in each particular context. These rules can be updated through learning based on lifecycle history data, context and user validation.

Service Infrastructure The service infrastructure is based on the Security Framework concept which assures safe Information Flow (contextual data originates only from authorised sources, contextual data is delivered only to the intended SELF-LEARNING control solution, source of contextual data is authenticated to SELF-LEARNING control solution), data isolation, damage limitation. The secure multi domain monitor may have additional use beyond supporting the SELF-LEARNING solution. The partitioning approach to secure data monitoring may also be an important concept for achieving greater integration of control systems and enterprise systems for manufacturing organisations.

Learning module SELF-LEARNING (both for Extractor and Adapter) relying on data mining and operator's feedback to up-date execution of adaptation and context extraction at run time

1.3.4.1 Context extractor and context model

This section describes the services/modules of the full prototype of the ontology and the context extractor. It describes all services, features and requirements that are part of the full prototype. The services are separated into ontology, monitoring, context extraction and supporting services, by their functionality as well as their requirements and dependencies with regard to other services and work packages.

Prepare the monitoring service for the integration of legacy systems (i.e. file systems, external devices)

Standardized monitoring Data Develop a schematic set for 'standardized data'. This serves as foundation for the context extraction. It is the interoperable exchange basis of Monitoring Services interfacing with SELF-LEARNING, which needs to map onto the structure. Must be correlated with context extraction services.

Enhanced monitoring services Integration of legacy systems fulfilled and several external systems are monitored and provide data as a foundation

for context extraction. External systems (.NET WebServices, OPC-UA services, file systems)

Parsers and analysers domain- and monitoring source specific parsers and analysers which create the standardized monitoring data enhanced monitoring services, standardized monitoring data

Monitoring processors pre- and post-processors for monitoring allow updates and changes of identified monitoring data before it is persisted and/or send to the context identification module analyzers, monitoring repository

Monitoring repository provides capabilities for persisting and retrieving monitoring data and enables traceability of Context Extraction outcomes as well as adaptation results. Monitoring Services

Provide interfaces for user to input context manually.

Context reasoning Based on the SELF-LEARNING ontology and user defined domain specific rules, deduce indirect high-level, implicit context from direct low-level, explicit context. Based on the output of context identification and SELF-LEARNING ontology definition.

Use deductive reasoning to check context consistency and reliability which might be brought in by imperfect monitoring.

Context similarity Measure context similarity in terms of statistical reasoning, rule based reasoning and ontological reasoning. Requires context information from context reasoning.

Weighting mechanism to regulate / balance results of comparison results of different contexts and monitoring data sets. Requires the information gathered by context statistical reasoning.

Dynamic query generation (SPARQL) used for retrieval of additional information belonging to contexts Requires context information from context reasoning.

Context Repository Provides capabilities for persisting and retrieving contexts (raw as well as refined) and enables traceability of Adaptation results. Context Extraction

Context processors pre- and post-processors for context extraction allow updates and changes of identified context data before it is persisted and/or send to the SELF-LEARNING module Context Repository and Context identification.

Solution Starter Services General runtime framework allowing easy setup, configuration and execution of SELF-LEARNING solution.

SELF-LEARNING context model. The purpose of the SELF-LEARNING ontology is to define a fundamental data model for context extraction. As it is known, the discipline of knowledge management is not like knowledge engineering or artificial intelligence, it doesn't intend to answer questions such as how to build a knowledge base or how to realize automatic knowledge creation by reasoning, but tries to enhance computer as well as people during the knowledge process and handling of knowledge. The main purpose of KM is to provide the right knowledge to the right

device at the right time. In SELF-LEARNING the ontology is mainly used to model the device context. So the SELF-LEARNING ontology does try to bring full description of context, but to index context to help to identify context.

Basically, the SELF-LEARNING ontology defines two ontologies: Generic Device Context Model and sector-specific context ontology. Both Context Ontologies model the knowledge context (including information of goal, activity, resource, etc.).

The SELF-LEARNING ontologies defined in such a way that it can be expanded: it is a layered ontology, with two parts, the generic part forms the core ontology and the business case specific part forms the domain specific ontology.

Monitoring services. These services are used in a wide range of application areas, monitoring conditions such as temperature or humidity using sensors or probes, or accessibility and performance of e.g. a computer network.

Context Monitoring services are used to monitor machine states and human-machine-interaction in order to monitor the (inter)actions in and extract the actual context to further use it for adaptation. For that, a machine is being closely monitored, 'raw data' collected and enriched with available 'knowledge'. The implementation of monitoring services involves the development of two types of services generic and application specific monitoring services.

Generic monitoring services provide basic monitoring functionalities to monitor production systems. Focused on machines, generic monitoring services provide access to e.g. sensor data, ambient conditions. Monitored data is enriched with pre-tagged knowledge such as type of monitored machine/control enabling context extraction services to better correlate the monitored data and its purpose.

As a precondition for comprehensive context extraction, monitoring services are customizable to support particular applications, taking into account all available individual contextual knowledge that can hardly be extracted with generic monitoring services and need to individually set-up. Therefore, application specific services are developed to support particular applications, sensors or external systems; implementing specific functionality already defined in generic monitoring services.

For both types of monitoring services, the monitoring process itself is the same. The monitoring process separates into 4 main services. The Parser converts the raw data coming from Monitoring Services (e.g. data from a temperature sensor), depending on the source system and formats it to the systems readable data for the Analyser to interpret.

Furthermore the information is constructed into the standardized Monitoring Data Repository containing instanced information, based on the schemata for later processes to rely on. Resulting is a recurring cycle of Context Monitoring that monitors and maps different sources of information on one context-sensitive monitoring data repository.

Context extractor. The Context Extractor is used to extract context during daily machine operation, and to provide the extracted context to the Adapter services. There are three basic functionalities: context

identification, context reasoning and context provisioning. Context identification receives knowledge and knowledge context from the monitored information. Context reasoning deduces high-level, implicit context from low-level, explicit context, and checks the context consistency and reliability as well. Context provisioning provides context for the Adapter services to realize intelligent and context sensitive adaption. The extracted context (modelled as ontology instances) is stored in the context repository and the ontology definition in the context model repository.

There are two potential input sources for context identification: one is the monitored 'raw data' provided by the machines; the other is by the user manually inputting knowledge context information.

One of the reasons that existing knowledge management solutions do not work well in practice is because people are bothered too much to provide input manually. It can be a big burden for people to collect context manually. Therefore, the monitoring interfaces provide as much context information as possible. They monitor e.g. the machine states and extract a set of 'raw data', structured for further context processing. In this way, the context identification process just needs to map the delivered data onto the ontology.

With context identification, the solution can only acquire context information directly provided by monitoring interfaces or by the user. In many cases, this is not enough to support Adapter services. For example, monitoring services might provide such context: a sensor indicates that the pressure inside machines goes up, which normally would mean a critical event for the production. But what a context sensitive service might need to know is whether the increasing pressure comes from temperature changes of the material to be pumped. There is a gap between what context identification provides and what Adapter services need.

This gap could be closed by context reasoning. Based on the ontology definition and user defined domain specific rules, it can deduce indirect high-level, implicit context from direct low-level, explicit context. The core technologies used here are deductive and probabilistic reasoning as well as context similarity measurement. Deductive reasoning is a basic technique in logics, in which conclusions (deductions) are reached by previously known facts (premises). Deduction is a well-understood general technique in general logics, and particularly relevant in logic programming. Since we use Resource Description Framework (RDF) and Web Ontology Language (OWL) to model the ontology, deductive reasoning can be supported very well.

Context Extraction. Context Extraction is based on set of embedded services responsible for identifying changes in the context of the environment. The current identified context is used to extract available context knowledge. The results of the Context Extraction are used in the Adapter which is responsible for updating the system behavior.

Context Extraction uses all 'raw data' provided via the data access layer to derive the machine's current contextual situation. Using the ontology/context model the monitored data is evaluated and the context extracted. Based on the identified context, situations can be compared to previous ones and stored. A continuous process, which is coordinating with the monitoring and followed by the adaption process to give current

contextual meaning to the provided knowledge, is built around the main extraction of current contextual.

The core modules of the proposed Context Extractor Architecture are the following:

Adapter interface: Represents the interface to the adapter. Via this inter-face the Context Ex-tractor and the Adapter exchange data and call specific functionality on both sides.

Data access layer: Generic component, which is responsible for accessing the device layer.

Data processing: This module is responsible for the pre-processing of monitored raw data acquired via the data access layer, before the context is identified. Main functionality is the normalization of monitored data to transform the raw data in a format which serves as basis for context identification

Context Identifier: Main component of the Context Extractor. It is responsible for the identification of the current context, based on monitored raw data, the ontology and historic context information stored in the context repository.

Rule Engine: Responsible for providing appropriate rules for the identification of context.

User Interfaces: User interfaces for maintaining and administering the rules and the context repository.

Context Repository: Inside this repository the identified context are stored for further processing and reuse.

Model Repository: Repository for the ontology. This repository should eventually be shared with other parts of the system (e.g. Adapter).

Business Case Specific Modules: This module is a placeholder and represents all components and user interfaces which needs to be developed for each business case individually

1.3.4.2 Adapter

This section describes the services/modules of SELF-LEARNING Adapter. It describes all services, features and requirements that are part of the full prototype.

The following services are separated into Adapter, Learning and Adaptation Repository Services, by their functionality as well as their requirements and dependencies with regard to other services and work packages.

Allow the retrieval of Adaptations data made to the sys-tem during system lifecycle.

Adapter Services. These services are implemented and exposed to the SELF-LEARNING infrastructure to allow others components to trigger the Adaptation process, but also to setup initial con-figurations. In order to understand all the involved components during the Adaptation process and to clarify the complexity of the process itself the Adapter architecture is shown and a brief description of all its components is given; however more detailed information about the Adapter specification and generic architecture is available in deliverable D2.2.2.

The task-oriented components of the proposed architecture are the following:

- Context Change Handler: responsible for asynchronously handle notification events sent by the Context Extractor, whenever a change in

context is detected. These events are the trigger of the Adaptation process.

- Repository extractor: responsible for retrieving the necessary information from the Data Access Layer repositories related to the current context change. The retrieved data set includes all the information necessary to support the Adaptation process that, in turn, determines the appropriate Adaptation proposal, i.e. machines/processes parameters and/or configurations adaptation to the new context.
- Repository parser: the data set retrieved from Data Access Layer repositories contains raw information that needs to be arranged in particular way in order to be properly processed by the Learning service. In summary, the Repository Parser creates a generic data structure that serves as input for the Learning service.
- Learning parser: Similarly to the repository parser, this component acquires the result of the Learning service reasoning task and parse it to create a generic data object (Adaptation), which includes all the information needed by the system expert for validation issues. Furthermore, it is also responsible for receiving a complete Adaptation (including the proposal and result of the system expert validation). This information is crucial to support the accuracy of future adaptation proposals.
- UI Comm: handles the interaction between the adapter and the expert collaboration user interface (UI) providing a communication channel between the system expert and the SLPS deployment. This component is also responsible for informing both the UI whenever a new adaptation proposal is ready and for detecting/retrieving an adaptation that was entered to the system through the Expert Collaboration UI.
- Adaptation distribution: responsible for distributing an Adaptation object instance along the SELF-LEARNING environment after it was transferred into the real system. It stores the current Adaptation instance in the adaptation repository and it informs Context Extractor that an adaptation was done in the system.
- Proactive learning: embodies the proactive behaviour of the adapter component by performing two main tasks: the first one is an event-triggered task and the second one is a cyclic task. The major goal is to improve future adaptation proposals and exploit system idle times by running learning tasks.

The first service operation is used to configure the Adapter Module, i.e. to instantiate all the modules of the Adapter architecture that in turn is used during the Adaptation process, along with the thresholds to be used by the Proactive behaviour.

Looking to the service interface, the ApplicationScenario parameter allows to instantiate the particular implementations of the modules while the Mode parameter and the Algorithm parameter are used to specify what type of processing execute (generate a new model or use the latest one) and which learning algorithms to use according to the particular BC; e.g. Rule Induction, Neural Network, Naive Bayes, Support Vector Machines, Least Mean Square, or ID3.

All the information necessary for the Adapter configuration is defined in a XML file.

The configuration file enables instantiation of all the necessary classes for running the Adapter, and subsequent adaptation processes. As said before, the informAboutMonitoredData operation triggers an Adaptation process. The parameters className and Identifier are both used to

retrieve the according monitoring data from the Monitoring Data Repository that contains all the necessary information about a particular context representing a picture of the system in a particular time instant. Each Adaptation process runs on its own Java thread and a new one is launched for each context change notification sent by Context Extractor. Then, the adapter reasons on the retrieved monitoring data, creating an adaptation proposal that reflects the most suitable set of system parameters values.

The Adapter most visible result concerns the transmission of the current Adaptation to the current production system. Although having to deal with three different BCs, the process can be considered generic following the proposed architecture. The Adaptation object is always transmitted through a Comm UI interface implementation to the Expert Collaboration UI, which is then responsible for deploying it to the concrete system, after user validation.

This way, the output for each one of the BCs is:

- BC1 (Bosch Rexroth): iCal format message including the calendar event details regarding the identified idle time pattern, which is sent to the system via a OPC-UA client.
- BC2 (DESMA): a CSV (Comma-Separated Values) file specifying the appropriate machine parameters values to be set in for the current context.
- BC3 (Fastems): specification of the priority rule to apply in current context through the invocation of a web service.

These last three BCs are completely dissimilar from each other, which validates the objective of delivering a generic architecture able to adapt to the majority of application domains. Besides the generic architecture, the majority of the developed code is the same for the three BCs, with the exception for the communication with the real systems and data parsers.

Besides being triggered by the Context Extractor whenever a change in context is detected, the Adapter is also capable of monitoring its own state during system operation in order to identify suitable instants in time to proactively launching new learning tasks - proactive learning).

Moreover, when a learning task is launched, i.e. whenever a learn command is given and a new learning model referring to present context is inferred from the context data, the Adapter is able to validate the new model against the latest context data available. This will allow the system to check if the actual parameterisation is still valid with the updated model, or if a more suitable one is now available.

During system operation, the adapter verifies both the number of performed adaptation tasks and the elapsed time since last adaptation for detecting when a model can be considered out-dated and, then, perform a new learning task taking into account all the context data available since the last learning task. The level of proactivity is customizable and can be defined through a configuration file, which can be configured by the user in order to specify the thresholds for triggering new learning tasks.

Learning Services: The Learning Services represent the reasoning entity employed by the adapter. These service operations are used during the Adaptation process and also during the Proactive behaviour. When an

Adaptation process is triggered, the monitoring data are retrieved and en-capsulated into a structure (ReasoningInput) that in turn is sent to the Learning module to be pro-cessed. The result is encapsulated into a ReasoningOutput object.

The selectAlgorithm operation instantiates the algorithm(s) to be used for current Adaptation process, i.e., based on the ReasoningInput object structure the Learning module is able to instantiate the necessary number of algorithms to face the current application scenario. The ReasoningInput is generated using the MonitoringData coming from the ContextExctrator and then contains a list of contexts representing the actual/current system status allowing the Adapter to know exactly how many algorithms instantiate.

The two core operations learn and reason, allow the training of the algorithm using a particular model and elaborate a result. The number of necessary models used to train the instantiated learning algorithms depends on the particular business case and in the same business case several models can be used depending on the current ApplicationScenario, also in this case the number of instantiated algorithms determines the number of necessary models to use. The result of the pro-cess depends on not only the actual context comprised within the ReasoningInput instance but also on the learned model, since the quality of the learned model is essential for assuring a good algorithm performance. Finally, the updateModelEntry operation is used to execute an update of the existing learning models with the last Adaptation process result. These new models are then available for future learning tasks.

The Learning Service has been implemented using the RapidMiner API library, which allows the employment of the necessary learning algorithms; however, their input data has to be ar-ranged/organized in a well-defined manner in order to enable the algorithm handling. Thus the RepositoryParser is responsible for organising the MonitoringData in order to be properly processed by the Learning algorithms.

For the Full Prototype the following learning algorithms are possible to be executed through the Learning Service: ID3 Learner, Naïve-Bayes Learner, Support Vector Machine, Neural Net-works, Rule Induction, Least Mean Square.

As said before, the Learning Service supports a straightforward integration of further algorithmic components to be available under the same service interface.

Adaptation Repository Service. In order to store the Adaptations made to the system during its lifecycle, the Adaptation Repository has been implemented. The Adaptation Repository Service offers to the overall platform a collection of data to take into account during the lifecycle evolution of the system in order to improve its performance.

The first operation allows the retrieval of an Adaptation instance corresponding to the defined adaptationID. The operation getLastAdaptations allows the retrieval of the last 'count' adapta-tions performed to the system or all the Adaptations in the specified TimeFrame.

Finally, the storeAdaptation operation is used each time a created Adaptation instance has to be stored into the Adaptation repository. The repository has been implemented in a layered style.

When an Adaptation instance has to be stored or retrieved, the according implemented operation can be used to execute it, which in turn use the functions provided by the Jena API to create and manage OWL models.

1.3.4.3 Service infrastructure

This section describes the modules of the full prototype of the Service Infrastructure. It summarises the capabilities, features and requirements that have been included within the full prototype developments. The full prototype of the Service Infrastructure is separated into multiple modules for Communications Authentication, multiple modules for the Control Processor, the QoS Monitor, and the Separation Kernel, all of which represent the core functionality of the platform that enables the SELF-LEARNING system.

1.3.4.4 Implementation

For the implementation of the SELF-LEARNING Prototype several different development tools and IDE have been used. For the overall development and orchestration of all system modules and components the Eclipse IDE has been used. The tested and widely accepted Open Source development environment for Java offers through a modular system a large plug-in community. Through all these techniques selected for the implementation of the systems architecture and services can be summed up in one environment.

The repositories are based on a model- and object-oriented POJO approach. These are the foundation of the data flow and storage and can be easily extended or altered according to the changing requirements and embedded into the system again. Relations and models have also been used in correlation with the serialization for sending and hibernation. The storage layer is, as described, based on a Java JPA solution. The JPA specification is referenced by the Hibernate project which gives the ability to create and use storage systems based on models, JPA-Specifications and PO-JOs.

The implemented solution, based on Java as described, runs in a Service Execution Environment given through the Service component architecture by Apache CXF and running on the application server Tomcat. Together this application environment is the foundation for the SOA modules and orchestration / composition.

These have been used to develop and run the SELF-LEARNING system against the systems concept and hereby specified functionality. It resembles state-of-the-art tools and software to provide a modular, extendable and expandable service-oriented approach.

1.3.4.5 SELF-LEARNING methodology

The methodology provides an approach on how to use SELF-LEARNING solutions to integrate control and secondary processes in production systems, pushing applications of the generic project results in industry. It addresses both organisational aspect and new business models allowed by the SELF-LEARNING solutions. From the organizational point of view two

main problems are addressed: Extended enterprise issue, integration of primary and secondary processes - following lean approach.

1.3.5 Evaluation of results and lessons learned

The proposed concept has been developed and applied in three different scenarios. The three application scenarios belong to different industrial sectors (although all three address discrete manufacturing and machine vendors' views).

Company main business application scenario objectives/technical issues addressed

BC1 - Bosch Rexroth control, automation and drive systems Control systems of machine tools Improve current service platform by integration of control and secondary processes (e.g., power saving, maintenance), specifically by improved transparency of complex machines, improvement of tools and methods for the analytical optimisation of secondary processes (e.g. maintenance plan / scheduler, NC-program with 'Power save commands'). SELF-LEARNING idle-time recognition of machine tools

BC2 - DES-MA machines and automation systems for shoe industry Control systems of machines/automation systems under development Enhance machines with SELF-LEARNING features by allowing machines to inspect statistically the condition of products and equipment, report and analyse proactively the gathered statistic values, enabling the machines to decide and adapt the parameters and keep them always inside the 'optimized' working range. 3 challenging scenarios defined aiming to identify the unique SELF-LEARNING solution

BC3 - FASTEMS highly customized FMS systems FMS experimental cell Optimisation of usage of FMS by maximum utilization rate of production machines, minimization of the lead time of production orders. Optimisation of reactive scheduling model for SELF-LEARNING scheduling and dispatching in FMS.

Throughout these three distinct business cases, several users were involved in the assessment of the SELF-LEARNING Full Prototype. Each Business Case covered specific test cases correlated with the companies use cases.

The results of testing from all Business Cases and assessments have been collected. These issues and possible refinements served as an additional foundation how to extend, fulfil and conclude the SELF-LEARNING solution for the full proto-type.

During the whole implementation phase of the Early Prototype as well as the Full Prototype JUnit tests were created, updated and executed to ensure previously validated functionality works throughout the entire project life cycle.

A common code repository for all the software parts were used during development. In the optimal case, developers should be integrating and committing code into the code repository every few hours, whenever possible. Instead of integrating all software parts after every update manually, for the SELF-LEARNING solution the approach of continuous integration was used. Continuous integration (CI) is a software engineering practice in which isolated changes of code are immediately

tested and reported on when they are added to a larger code base. The goal of CI is to provide rapid feedback so that if a defect is introduced into the code base, it can be identified and corrected as soon as possible.

The usual CI rule for each team member is to submit work on a daily (or more frequent) basis and for a build to be conducted with each significant change. When used properly, continuous integration provides various benefits, such as constant feedback on the status of the software. Because CI detects deficiencies early on in development, defects are typically smaller, less complex and easier to resolve.

For the SELF-LEARNING solution development the continuous integration platform Jenkins was used. The Java based open-source continuous integration platform allows compilation of code coming from a central code repository after every commit. Moreover all previously defined unit tests are executed if the previous build succeeds. This allows easy integration of work of different RTD partners and ensures an always up-to-date code base, where every critical error or mal-function of code or failing unit tests are automatically detected and reported to the developers.

The users which were using, testing and assessing the SELF-LEARNING solution within the three business cases, were overall satisfied with the functionality provided by the solution. Overall, several conclusions can be made based on the test and assessment concerning the acceptance and usability of the SELF-LEARNING solution within the three business cases.

- The SELF-LEARNING full prototype was well accepted by the users / operators
- The users / operators found the system having some intelligence / awareness to understand what the monitored systems and machines were doing
- Configuration of the system and the application scenarios allowed an easy installation and maintenance. It was not necessary to implement bigger changes in the existing IT infrastructure within the business cases.
- The users / operators highlighted the simplicity of using and integrating the SELF-LEARNING solution into existing system. From their point of view there is no need to drastically change their usual workflow after integrating the SELF-LEARNING solution
- Administrators and personnel dealing with security and privacy issues were impressed by the security infrastructure and the secure handling and transfer of data between existing solutions and the SELF-LEARNING full prototype components
- Nevertheless there were some new requests for features and enhancements to the full proto-type which could be included in another iteration of the software when it comes to commercialisation:
- The graphical user interfaces should be even more simplified (i.e. content configurable) for some application scenarios and business cases to ensure usability for ALL users
- The security infrastructure is a very complex and powerful system and could be more light-weight for future applications in terms of configuration and setup
- The overall performance of extracting and identifying context and in a next step making adaptation proposals could be increased (faster response times, less computational resource utilisation)

1.3.6 Innovations and future work

The project developed a novel approach in production systems, being context aware, adaptive to contextual changes at run time and learns from adaptation and operator's action. The solution proposed addresses adaptation of various process/control parameters to achieve integration of control and secondary processes. Several aspects will be specifically elaborated in future research.

High complexity of data acquisition and real-time data analysis algorithms and tools for SELF-LEARNING production systems imposes high demands on quality and security of SW services which are implementing such SELF-LEARNING algorithms. Therefore, the main obstacles to fully utilize the opportunities offered by service-based SELF-LEARNING systems are problems related to reliability, availability, interoperability and security & trust of diverse SW services, especially real-time services in device space. This will be the subject of further investigations.

As indicated above, an effective integration of control and secondary processes requires solving both organizational and technical problems. The tight integration of control and secondary processes may lead to critical organizational problems taking into account current 'separation' between these activities by many equipment users. Effective integration requires consistent application of EE principles (e.g. trust between equipment manufacturers and users), since e.g. equipment manufacturers often need to be strongly involved in the maintenance activities which are, according to the proposed approach, tightly integrated with control of production. Lean production principles (such as minimization of 'waste' activities, where 'wastes' are all activities not bringing direct added value to product) in combination with the technical concept will be used as a leading approach.

Further research will focus on advanced algorithms for SELF-LEARNING based on extracted context to (semi-)automatically update the context model. In addition the context model itself will be addressed by further research to allow better utilisation of the presented model for other companies as well as for other application domains. High complexity of data acquisition and real-time data analysis algorithms will be addressed in further research to 'fully' utilise the opportunities offered by service-based SELF-LEARNING systems.

The applications of the presented approach in real world industrial scenarios are pointing at promising results.

Potential impact:

1.4 The potential impact

1.4.1 Overall impact

The project impact can be expected at various levels:

- S&T impact
- Impact upon end-users involved in the consortium and wider audience
- Socio-economic impact
- Impact through dissemination and exploitation activities.

In the text to follow these potential impacts are briefly analysed.

Science and technology (S&T) impact: Exploring one of the most critical problems of the SELF-LEARNING systems of how to assure effective self-adaptations of the production systems in order to achieve holistic process control and assure reliable and secure work of SELF-LEARNING production systems, via context awareness, the project provides solutions on how to apply SELF-LEARNING approaches for holistic process control adaptation and integration and on how to extract context from networks/services and processes and reuse it for highly reliable SELF-LEARNING services to comply with the above mentioned requirements regarding reliability and security of control & secondary processes integration. One could say that these solutions are relevant for the 'entire' networked embedded control systems in manufacturing industry. SELF-LEARNING shows how a context can be used as a bonding element between control and secondary processes of production systems and different services/networks. Due to the wide impact and applicability in diverse application scenarios it is envisaged that the SELF-LEARNING research may have an impact on many other S/T areas, while fitting with the challenges as identified within both MANUFACTURE/LEADERSHIP and the embedded systems research roadmap, and serving as a bridge between ICT and NMP domain.

Benefiting communities: The communities most likely to benefit from SELF-LEARNING results are:

- (a) Manufacturers of industrial installations: equipment, automation and electronic devices,
- (b) small and medium-sized enterprises (SME) and LE users of industrial automation in different sectors (first target is automotive, aerospace and machine industry as direct customers/partners of the three industrial partners: BR, LSW, FASTEMS)
- (c) ICT community (embedded SW and service providers - both large and small, where BR and TOG belong).

All these communities are explicitly represented in the project.

The benefits and impact upon the end-user groups in three BCs are presented in Section 1.4.2.

Societal-economic impact analysis is presented in section 1.4.3.

1.4.2 Impact within business cases

1.4.2.1 Business case 1

The main challenge in the bosch-rexroth (BR) case is related to the issue of optimization of non-productive ('secondary') processes in machine tools.

The objectives of this scenario are the use case maintenance and energy usage optimization as relevant examples for secondary processes. SELF-LEARNING system derives secondary state information (e.g. tool wear, power consumptions) from explicit control parameters and provide modified/extended action description list (e.g. energy efficient NC code) through adaptation. Learning is based on operator's feedback to adaptation.

SELF-LEARNING solution integration to the existing service platform in this use case will provide:

- Improved transparency of complex machines (e.g. energy consumptions in NC blocks, symptom to damage model).
- Enhanced plug-in framework for diagnosis and analysis methods.
- Availability as a plug in framework for OEM or user defined methods.
- Avoidance of resource conflicts by requirements descriptions for diagnosis and analysis methods.
- Tools and methods for analytical optimization of secondary processes (e.g. maintenance plan, NC programs with 'power save commands').

The most important fact to understand in the BR application scenario is that the machine tool manufacturer stands in between BR (as control system provider) and the operator of the control, which gives BR a more indirect role in the provision of production services, more as a platform provider than as service provider. The variety of machines and machine classes supported by BR controls even enlarge the number of different applications. The specific characteristics of these numerous applications entail a complexity in monitoring and optimization of control solutions that can hardly be mastered effectively without context aware solutions, learning about their characteristics and abilities and using available knowledge to adapt for various application conditions. In order to bring such technologies to practical use by enabling for customer defined service plug-ins in a limited resource domain, where resource conflicts are foreseeable and have to be sufficiently managed, BR puts emphasis to the Quality-of-Service aspects of modular embedded services. To provide such a generic service platform gives BR the opportunity to fulfil demands of OEM customers without too many efforts on customisation to specific domains and to provide and exploit internal solutions in several domains quickly and on demand. The most urgent and promising domains for BR in the field of optimising secondary processes as a service in production automation are efficiency and maintenance.

Innovation/Benefits expected: Issues like resource efficiency and maintenance costs will gain importance, compared with traditional assessment criteria like productivity, process quality and (investment) costs.

The SELF-LEARNING system will give substantial support in mastering the optimization of production equipment with respect to such 'secondary processes' and will change the capability of automation systems qualitatively because of three main reasons:

1. SELF-LEARNING capabilities will gain access to information about the organisation of production, which is not explicitly available on process level normally. This will reduce communication needs between manufacturing execution- (MES-) and production layer to an acceptable level, thus enabling for complex optimization of automation systems.
2. Secondary process models will be extractable from control and drive states, thus enabling for the application of optimization rules. In fact the control will learn about the machine it's controlling, resulting in some self-awareness of the machine.
3. The whole concept of integrating models of machine and environment in the control is a paradigm change in automation, fundamentally increasing transparency and flexibility by adding semantics to software-objects directly linked to physical objects.

The SELF-LEARNING concept will pave way for a holistic approach to the concepts in control of factory automation. It will improve the performance of automation solutions in total and the division of labour in the life cycle of production facilities.

Future application / usage: Beyond application in efficiency and maintenance the SELF-LEARNING concept and -architecture lay basis for a general intelligent 'job scheduling' in automation systems. With this, time is an explicitly known, meaningful category in the automation system on a process control level. The knowledge about the duration of operations, which is gathered by intelligent systems, not by pre-designed data connections, allows for the rule-based scheduling of actions following various purposes, while keeping the data network focused on process relevant data only. BR has intentionally de-coupled the SELF-LEARNING-system from its application to keep the future possible field of operation as wide as possible.

Nevertheless BR intends to apply the results to the use-case 'efficiency-optimisation of machine-tools' first. The next step will be the transfer to others than machine tool controls. To this aim the architecture and the realisation have to be adapted to a real-time operation system environment. Appropriate first activities have already been started.

The SELF-LEARNING principles of the BR-demonstrator will be the core of any future 'intelligent services' product in BR-controls.

1.4.2.2 Business case 2

The second case is involving DESMA and is focusing on parameter identification for intelligent monitoring and adaptation of machines/automation systems for shoe industry. The objective of this use case scenario is the adjustment of single parameters from different parameter sets. SELF-LEARNING system integration is focused to react to the changing scenarios associated with variations in different parameter sets. The parameter variations are mostly in terms of pressure and temperature, speed frequencies of drives and pumps, proper material mix ratio and filling of materials into shoe forms.

Identified key objectives of SELF-LEARNING solution integration to this use case define:

- Intelligent monitoring of the state of the machines/parts using contextual information.
- Intelligent monitoring of (ambient) working conditions of the machines/parts using contextual information.

Provide adjusted parameter sets to human operator for final decision.

DESMA is fully aware that the operation challenges, as well as the diagnostics and maintenance challenges have to be collaboratively solved in an Extended Enterprise (EE) context, by designers and maintenance staff from DESMA, as well as the operators in the shop-floor workshops and the maintenance service departments at the customer sites. To support such approach (by provision of data needed for collaborative work with geographically distributed teams) and in order to avoid potential problems with their automated production systems, DESMA intends to apply intelligent monitoring systems: this means to significantly extend the existing monitoring of the state of the machines/parts and replace/repair them before problems occur which may ask for quick actions and may cause stopping of the line. Such information can be used for both preventive maintenance and on-line problem solving.

DESMA intends to equip their system installations with a number of additional sensors and different ICT solutions allowing for monitoring of the status and performance of the automated systems and promptly react to any disturbances. The problems are how to process this information and correlate with existing knowledge on the machines/automation systems to create higher level information needed to optimise the maintenance of these automated systems and provide better services. To achieve this target, on mid-term the company intends to apply the SELF-LEARNING services in their system solutions. The SELF-LEARNING services will be connected with control of the machines/automation systems, sensors, man-machine interfaces and other sensors (temperature, variation, noise, humidity etc.), The system will be at DESMA connected to their PDM system, intending to transfer (re-use) the knowledge (e.g. on problems) from one line to another.

Innovation/Benefits expected: DESMA sees a huge winning point in the usability of complex production equipment by the integration of SELF-LEARNING and self-adapting control modules. The worker or operator is not forced to bring highest skills to run the equipment and generate a profit for his society but more concentrating on the core processes while the production equipment is self-controlling all relevant parameters to keep the process running and manufacturing the products on the highest quality level. The SELF-LEARNING modules are integrating more intelligence into the production and generating a higher benefit for the producer while less investment in the human resources are required.

The customers of DESMA can calculate finally a higher profit so the integration of the technology generates a competitive advantage for DESMA and the customer.

Future application / usage: All developments within the project SELF-LEARNING are just a first step into the direction of self-adapting machines and units, which will be one of the key-products for future equipment. The SELF-LEARNING methodology can be used in a broad field of processes and operations within the DESMA customer industry.

The transfer of the technology will be supported by the clear architecture of the software tools. As the base for the developments always the service infrastructure, the adaptor and the extractor will be used in the existing or maybe a slightly modified version.

DESMA will try to generate a toolbox of modules for an easy configuration of SELF-LEARNING control and software systems with the combination of the physical processes.

Another concentration will be the integration of sensors for a variety of applications because with an intensive use of sensory data the SELF-LEARNING systems can be used multi-flexible in different applications.

1.4.2.3 Business case 3

The third Case relates to FASTEMS in Finland and focusses on scheduling and dispatching in flexible manufacturing systems (FMS) for automotive industry. The objective of this FMS use case is to automatically select candidate clamping jobs into a priority order. The scheduling in FMS cells consider certain objectives as maximization of machine utilization rate, avoiding starvation at loading stations, keeping due delivery date of production orders etc. These optimizing criterion are usually not static. Run time changes occur in scheduling due to production profile, process state and shift model adjustments from enterprise level. Hence, the use case is mostly dealing with reactive scheduling.

SELF-LEARNING solution integration to this use cases' existing service platform is aimed to improve the reactive scheduling model by considering the following aspects:

- Taking into account the operator supervision concerning the optimisation criteria.
- Introducing resource planning features.
- Identifying process states and operator supervision and learning from them.

FASTEMS intends to apply intelligent monitoring systems in order to allow optimization of scheduling rules. This means to significantly extend the existing monitoring of FMS utilization rate avoiding starvation at loading stations, keeping due delivery date of production orders etc.

Innovation and benefits: In general, customer that is using FMS system has his own idea of optimal operation of the system.

These objectives can be for example one of the following items or combination of them:

- Maximum utilization rate of production machines
- Minimize lead time of production orders
- Keeping the due delivery dates of production orders
- Minimize the tool flow in production machines

These optimization objectives are normally not static. They are changing depending on the production profile but also depending on the process state and shift model. For example, when most of the production load is addressed for direct customer orders it is necessary to keep the due dates. When most of the production is for Kanban manufacturing (stock batches) then it is natural to try to keep the machine utilization high.

Operators on the other hand, are less educated and they may be using a specific system just short time, Therefore it is important to facilitate operators in their daily decision making and to make systems more transparent. The final goal is that operators are capable to run systems in more optimum way in the constantly changing environment. This BC is

focused to achieve these targets by using novel technology to guide operators make the right decisions.

Future application / usage: Fine scheduling systems are becoming more and more common in factory automation. MES systems in broad perspective are covering the gap between ERP/MRP systems and factory floor. Essential and also a core property of a MES system is a comprehensive fine scheduling system that is capable of simulation production over the whole factory. A modern fine scheduling system does not take into account only workstations and job capacities but also resources as operators, material and tools.

A special case of a fine scheduling system is a FMS fine scheduling system in which production lots are split into machining pallet jobs. These entities are then used in the simulation process that creates the schedule.

When creating a fine schedule, there need to be a set of policy attributes that define the criteria in which way the schedule is generated. By changing the values of these attributes user can express what kind of manufacturing performance he appreciates in the manufacturing context in concern. When the manufacturing situation changes, user need to activate rescheduling with changed at-tributes.

However, it is not desirable that rescheduling necessitates manual intervention because it occupies extra workforce. Instead it would be desirable that the fine scheduling system could adapt automatically to changing process context.

In MES systems, fine schedule is implemented and work queues are generated on workstations. MES system normally is aware of the process context by receiving job progress reports and de-vice state reports. User at workstation may also differ from the schedule by selecting not the top job but some other later or lower priority job.

SELF-LEARNING technology could be used when:

- Improving schedule accuracy: Schedule accuracy is dependent on the accuracy of phase times of manufacturing operations; these are normally defined based on empirical data or algorithmic model. Real operation phase time data can be logged in the MES systems. When sufficient quantity of this data has been gathered, a learning algorithm could be used to feedback operation time data back to the fine scheduler.
- Improving the usability and flexibility of the fine schedule: Created fine schedule is based on rule that defines how candidate jobs of a workstation are prioritized.

Backward scheduling rule accomplishes different kind of schedule compared to the one that has been created based on the forward scheduling rule. If the fine schedule, for a reason or another, is not feasible, then users tend to break the schedule by forwarding jobs outside the schedule or jobs that originate from a later time slot. Self-learning technology could be used to reason whether the fine scheduling primary tenet is intact or not. SELF-LEARNING application could then pro-pose alternative schedules that are based on different schedule generation methods when it observes that the schedule is not advancing as it should.

1.4.3 Socio-economic impact

Employment: Using the project results to optimise SELF-LEARNING solutions in industry will allow for considerably higher productivity and more effective control and maintenance of manufacturing systems, thereby improving competitiveness and business development of manufacturing industry. On the other hand, ICT and industrial automation and electronic devices vendors will be able to offer new services to their customers, also enabling European industry to reinforce its major strengths in the supply of hardware and software components and their integration and deployment into intelligent systems in manufacturing plants. This will have a direct positive benefit for employment.

Working conditions and quality of life: The project makes significant contribution to improving working conditions and quality of life. Higher quality of SELF-LEARNING solutions for integrated control and maintenance will reduce stress on employees of both equipment manufacturers and users. For example, massive introduction of embedded services in manufacturing industry will allow for early provision of required information in complex networks concerning process execution deviations, breakdown of machinery, change of transport systems or schedule changes, re-duce stressing maintenance tasks under high time pressure etc.

1.4.4 Dissemination activities

The assessment of the dissemination achievement was both qualitative and quantitative. General-ly speaking, the consortium refers to the outreach and broadness of dissemination actions and the quality of publications. Based on the defined indicators the project performed a semi-annual assessment and suggested corrective actions. As important dissemination successes may indicate: 1 articles submitted to journals, 7 papers in conference proceedings, several local seminars organised by the partners as well as 4 PhD (most of them still in process) based on the SELF-LEARNING RTD activities. The SELF-LEARNING dissemination activities were intensive from the beginning of the project in terms of presenting basic ideas within RTD and industrial communities, followed with results presentation. Some of the most important dissemination activities are:

1.4.5 Exploitable results

1.4.5.1 Exploitation approach

The SELF-LEARNING project will provide technologies for SELF-LEARNING production systems, which will be comprised of a suite of software components and platform technologies intended to ad-dress a wide variety of production systems and industries.

The key principles on which the project development work is based include the following:

- Using existing open standards wherever feasible
- Submission to standards bodies of any extensions or refinements made to standards in order that these are adopted by industry
- Publication of interfaces used within the SELF-LEARNING results in order that additional components and functionalities from organisations outside the project can eventually be included in the SELF-LEARNING framework

- Use of a component approach where production systems suppliers are able to select the features, platform and other elements that best meet their needs

These underlying principles are manifest in the dissemination and exploitation planning for each of the technologies that have been developed within the project.

As a partially funded European commission (EC) project, it's important within the exploitation planning to provide opportunities for the partners to recover the substantial investments in the project that are not funded by the EC. While, at the same time, the partners recognise that the broad take-up by industry of SELF-LEARNING project technologies requires that certain project results be made public in order that other production systems technology vendors also invest in complementary components that operate with the SELF-LEARNING platform, and industrial users have confidence in the openness and continued evolution of the SELF-LEARNING framework.

Achieving a good balance of different exploitation paths will increase momentum in the market for SELF-LEARNING production systems, which in turn will provide opportunities and benefits for each of the SELF-LEARNING project partners.

1.4.5.2 Exploitation paths

As the project has completed its final phase, the project partners have identified the exploitable technology components from the project and the appropriate exploitation paths that are envisioned.

The project partners will utilise that several primary exploitation paths, which are summarised in the following:

Publication of specification and submission to standards body. The Open Group hosts the real-time and embedded systems (RTES) Forum, which is a global grouping defining standards for real-time systems software including operating systems, languages, security and platform architectures. The real-time and embedded systems forum includes over 40 members from leading technology vendors, manufacturers and services providers, and research organisations and also has formal liaisons with other embedded systems standards bodies. The project partners are also active contributors to a range of standards bodies including OMG, IEEE and many others. The partners through the RTES Forum and participation in other standards bodies are fully capable of submitting specifications that meet industry requirements for standardisation. In addition, the partners are familiar and have extensive experience with the processes of building consensus and reaching agreement on new standards.

Commercial products. The SELF-LEARNING consortium includes three industrial partners that are commercial technology vendors delivering production systems and services directly to manufacturing industries in Europe as well as global markets. As described in Section 2.2.1.3, each partner has specific focus areas for their commercial offerings, but all three partners are successful in introducing new products and services to major industrial manufacturing organisations. The approach for distribution varies amongst the partners from direct sales and service, to use of nation-al or regional distributor channels, while also partnering in some cases with value added resellers or integrators. The

combined annual turnover of the industrial partners exceeds 4 billion EUROS annually. The industrial partners in the project have fully sufficient operations in commercial production systems and services for widespread exploitation of SELF-LEARNING results.

Open source products. The partners will utilise open source products as a basis for many components within the SELF-LEARNING platform, and are familiar with the procedures and mechanisms for open source distribution. One partner operates an open source product platform addressing middleware technologies, while other partners have made contributions to well-established open source platforms and are active in shaping their evolution. The partners are fully capable of either utilising existing open source product mechanisms for those technologies extended by the SELF-LEARNING project, or if required have established facilities for creating and managing a new open source platform for disseminating and exploiting SELF-LEARNING technologies.

Associated services. The SELF-LEARNING consortium includes four research and technology partners that work closely with industry to support technology transfer and exploitation of innovative technologies. These organisations each provide software development services, typically on a cost recovery basis, to commercial companies within the manufacturing sector. Each partner has many years of experience managing projects to develop custom implementations of control technologies, context analysis or infrastructure technologies that meet industrial requirements and exploit state-of-the-art technologies that result from advanced research and develop from many sources. Each partner has well established procedures for proposing and implementing custom development projects and is capable of tailoring one or more of the SELF-LEARNING technologies for specific industrial applications.

It's worth noting that these exploitation paths identified above are not mutually exclusive. For example, it's possible that technology that proceeds down an exploitation path of an open source product is also exploited by one or more industry partners packaged as a commercial product with associated commercial services and support. Also, publication of a specification and submission of a specification to a standards body are often synchronised as part of a strategy to achieve broad industry consensus on a newly proposed standard.

1.4.5.3 Exploitable results

The core project technologies will be exploited by targeting the manufacturers of production equipment, similar to the industrial partners that participate in the project, so that these third parties provide SELF-LEARNING capabilities for their production systems. The specific implementations of the core technologies tailored to the production equipment from the industrial partners will be exploited by productisation and commercial sales within the industrial production equipment markets where the industrial partners operate.

1.4.6 Timeline

The SELF-LEARNING partners have established a set of action plans related to the project results that are intended for commercial exploitation. These actions will lead to the fulfilment of the exploitation and dissemination objectives described in the sections above and provide a timeline for when first commercial products that exploit the project results will be introduced in European and other global markets.

The main steps that are involved in creating commercial products from the SELF-LEARNING project results vary in length and effort between industrial partners due to a number of factors including the scope of the product line, the complexity of the SELF-LEARNING optimisations, the size and number of personnel or distributors involved in selling and supporting products, and many other factors. For example, a partner with a large international direct sales and service support organisation might require more time to organise sales and technical support training than a partner with a smaller organisation or several distributors.

The three industrial partners Bosch-Rexroth, DESMA and Fastems each plan to carry out the above steps as they progress towards commercial availability and exploitation of the SELF-LEARNING project results.

List of websites:

<http://www.selflearning.eu>
<http://www.atb-bremen.de>
<http://www.boschrexroth.com>
<http://www.desma.de>
<http://www.fastems.com>
<http://www.tut.fi>
<http://www.uninova.pt>
<http://www.opengroup.org>