PLANTCockpit
Production Logistics and Sustainability Cockpit

SEVENTH FRAMEWORK PROGRAMME

PLANTCockpit White Paper

PLANTCockpit is supported by platte consult

The PLANTCockpit project (260018) is co-funded by the European Union under the Information and Communication Technologies (ICT) theme of the 7th Framework Programme for R&D (FP7).

Document History

Version	Date	Comments	Author
0.1	06/03/12	Document structure created	Dražen Nadoveza (EPFL)
0.2	16/07/12	Content added	Andreas Koukias, Dražen Nadoveza (EPFL)
0.3	19/07/12	Content reviewed	Vladimir Vasyutynskyy (SAP)
0.4	23/07/12	Content updated after review. Document formatted.	Andreas Koukias, Dražen Nadoveza (EPFL)
0.5	31/07/12	Content Review	Christian Hengstler (SAP)
0.6	07/08/12	Peer Review	Jessica McCarthy (Intel)
0.7	10/08/12	Project Board reviewing and corrections	Ellen Legziel (Platte Consult)
1.0	13/08/12	Document Finalized	Dražen Nadoveza, Andreas Koukias (EPFL)

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n°260018.

Table of Contents

Table of Figures

Executive Summary

This white paper presents the vision, implementation details and business benefits of the solution developed by the project Production Logistics and Sustainability Cockpit (PLANTCockpit), under the umbrella of the "Factories of the Future" Public-Private Partnership (FoF PPP), which was launched by the European Commission within the 7th Framework Programme.

The PLANTCockpit project develops a central environment for monitoring, visualizing and analyzing of all intra-logistical processes. PLANTCockpit proposes to the various users within a plant a platform that will gather and visualize all the necessary information from the various heterogeneous plant information systems, such as ERP, MES and SCADA and will offer the required overall visibility in order to make well informed decisions.

The PLANTCockpit consortium consists of a group of highly qualified industrial and academic research organizations that has been specifically affiliated to meet the challenges of modern industries. This is achieved by a mixture of 2 technology partners (SAP AG, ICONICS Europe) and 5 strong industry application partners (Acciona, BMW Group, Comau, Doehler Group, Intel) as well as 4 universities (Ècole Polytechnique Fédérale de Lausanne, Politecnico di Milano, Tampere University of Technology, Technische Universität Dresden) and 1 research centre (Tecnalia). The industrial application partners provide a real-world environment to evaluation opportunities and represent different industry segments from both the discrete industry such as automotive, semiconductors, industrial engineering, as well as the process industry, such as the beverage production business. Geographically, the consortium is distributed across 7 countries, namely the EU member states Czech Republic, Finland, Germany, Ireland, Italy, Netherlands, Spain as well as the associated country Switzerland.

The timeframe of the project is from **September 2010 to August 2013** and more information and documentation on PLANTCockpit can be found on the project website: http://www.plantcockpit.eu/. Should you wish to contact the PLANTCockpit consortium concerning the project activities, the project coordinator is **Dr. Volodymyr Vasyutynskyy** (Volodymyr.Vasyutynskyy@sap.com) from SAP AG.

Introduction

In this document we present the vision of PLANTCockpit and we demonstrate through the developed architecture why the proposed platform is an ideal solution for all modern industrial environments that require full visibility on all their plant activities and an integrated view of the various manufacturing processes.

Initially, the document shows the project motivation by presenting the current status of the area in the PLANTCockpit context as well as challenges that the five use case partners face. Also, the most important problematic areas that prevent the various users within a plant from having an overall plant visibility and thus make well-informed decisions for optimizing the plant efficiency are identified.

The most important related technologies and systems are shown in order to identify and demonstrate the major gap that currently exists in monitoring, analyzing and visualizing the integrated plant data and overall processes. The PLANTCockpit vision is then presented and its key functionalities as well as the most important design principles the development was based on in order to show why the PLANTCockpit solution would be ideal to cover this gap.

The overview of the PLANTCockpit architecture is presented, which is designed to be a flexible, extensible and easily configurable layer-based architecture with message-based uniform and standardized interfaces. Although it is designed to be technology independent, some typical technologies applicable to the solution are also presented.

Finally, the most important business benefits expected for the five PLANTCockpit use cases are presented, as representative of big industrial fields, such as the automotive, semi-conductor and beverage industries. The overall business solutions that PLANTCockpit offers to modern industries are then clearly described in order to summarize and highlight why PLANTCockpit would have a major impact on improving the overall activities of modern industries.

Motivation

Area overview

In modern plants, a variety of systems, tools and methods exists in order to monitor, analyse and manage the different enterprise areas, such as production, logistics, distribution etc. Common current systems such as Enterprise Resource Planning (ERP), Manufacturing Execution Systems (MES) and Supervisory Control and Data Acquisition (SCADA) systems, and various special-purpose solutions are rarely integrated with each other and typically provide no more than point-to-point interfaces between selected functionalities. These sporadic point-to-point integrations do not fulfil the requirements of today's dynamic markets where enterprises have to quickly judge complex situations, react to unexpected events, and make far-reaching decisions.

The growing complexity of plants nowadays forces supervisors to manage a great and constantly increasing amount of information from disparate data sources concerning plant activities. With the growing focus on sustainability, complexity grows even further as production supervisors have to manage energy and material consumption, carbon footprint, and waste output in addition to classical Key Performance Indicators (KPIs) like process efficiency, asset utilization, quality, scrap rate, and costs. Despite the vast technological advancements in production systems and the emerging of standards for developing interfaces between plant information systems, the overall knowledge about the current states of different production levels remains very limited since every system usually relies on its own data and visualisation. Visibility across the plant activities is restricted since accessing different systems is necessary in order to gather data manually. Furthermore, the performing of root cause analysis is time consuming and inefficient.

For instance, a process manufacturer has the issue that he/she cannot immediately identify the reason for the disruption of a running production flow. He has to access several data sources to evaluate whether the disruption is caused by a delayed delivery of raw materials, by a delay in the internal material production and commissioning, by any failure in a machine or by any other reason. Ideally he would have the information on potential short-comings of material beforehand, to interfere and trigger new orders or to restructure the production schedule.

Due to these problems, it is clear that in the enterprise domain only an integration of all systems could provide the visibility and process integration needed to truly optimize the various plant processes, such as production, logistics, energy management etc. It is necessary to switch the focus from the isolated optimization of single areas to the consideration of an integrated interaction of all entities involved in a plant. Modern production environments require integrated views on different processes (manufacturing, logistics) at different levels to enable analysis and optimization of these processes.

PLANTCockpit use cases description

The industry partners involved in the PLANTCockpit project provide a real-world environment and evaluation opportunities and represent different industry segments from both the discrete and the process industry, with diverse and manifold use cases corresponding to various areas within a plant. An overview of the five use cases for Acciona, BMW Group, Comau, Doehler Group and Intel is described respectively below:

Acciona is a major business corporation and its Infrastructure division covers all aspects of construction, from engineering to project execution and maintenance, especially in the areas of transport and building construction.

Acciona's use case focuses on the continuous automated and highly complex process of pultrusion for manufacturing composites with constant cross-sections. Increasing production productivity is an important objective for the production manager and requires easy access to different data from different sources and a global vision of the pultrusion production process in order to gather all variables and make optimal decisions.

The most important issue during the pultrusion process is the production scheduling, where information such as raw material inventory, equipment, personnel availability and production stock is scattered among various systems, like SCADA and different databases. The production manager has to consider all these factors before making any decisions, thus turning the planning process into a very time-consuming and inefficient task.

BMW Group is one of the most successful worldwide automobiles and motorcycles manufacturers, with the highest standards in terms of aesthetics, dynamics, technology and quality.

Now more than ever the BMW Group is dependent on reliable interdependent Key Performance Indicators (KPIs), as aggregations of all related data, and current target values to cope with the challenges of an increasingly dynamic market environment and staying capable of making the right decisions and intervening to make fast corrective actions.

In order to face the complex set of challenges characterised by diversity both within and outside of the organisation, it is necessary to build an abstraction layer, established as a set of KPIs that will facilitate steering the very complex production network in the context of "order-to-delivery" process. More specifically, it is vital that the KPIs and their interdependencies are efficiently managed and visualized in order to control the various complex production and logistic processes and support a better root cause analysis.

Comau is a global supplier of industrial automation systems and services mainly for the automotive manufacturing sector, specializing in developing solutions for industrial production programs.

Comau has concentrated its activities on the optimization of plant asset management to improve the efficiency of the production process, and monitor, analyze and manage energy in the plant to evaluate and guarantee the manufacturing efficiency and implement process execution strategies while taking into consideration energy constraints.

However, the large number of tools involved in the different production phases leads to a lack of integration and automatisms for asset management, as well as the lack or shortage of exhaustive real-time data and information related to the several sources of energies. This generates the need for manual operations with high risk of mistakes for asset management, which has to be reduced.

Doehler Group is a global producer, marketer and provider of technology-based natural ingredients, ingredient systems and integrated solutions for the food and beverage industry.

As a producer of natural products, Doehler Group strives to produce finished products with a consistent quality to the customers on time according to the requested delivery dates. Doehler Group uses a group-wide ERP system and several subsystems to accomplish this task and ensure a smooth supply chain in a highly complex logistical system.

The large amount of daily scheduled production and filling orders and the urgent need for raw material availability stumble however on the very loose mapping between the supply chain, logistics and production. This causes major disruptions in feeding the information concerning possible delays in the procurement and transport of these raw materials and semi-final products to the production planner, causing valuable time lost in finding alternative raw materials and/or rescheduling production and filling process orders.

Intel is the world leader in silicon innovation and develops technologies, products, and initiatives to continually advance how people work and live, such as integrated circuits for the computing and communications industries and platforms for optimized user computing solutions.

Intel aims to achieve an overall operational control of energy in the manufacturing plant through management support to the energy program, continuous improvement towards the energy efficiency targets and the guarantee of the plant operational efficiency.

In order to deliver a sustainable repeatable improvement in energy efficiency, it is necessary to track and monitor the utility consumption (electricity, water and gas) at various levels of operations within the Intel factory. This can only be achieved through the integration of currently standalone energy management systems and providing systems and business processes integration.

Problem statement

Ideally the various users within a plant, such as production managers, should have all the necessary information available, on-time and properly visualized, in order to have an overall plant visibility and thus make well-informed decisions for optimizing the plant efficiency. However, modern production environments, as shown in the five use cases, face important problems that prevent users from having an integrated view of the various manufacturing processes and subsequently managing them efficiently and optimizing them. The key problematic areas within the PLANTCockpit context are identified below:

- **System interoperability:** Enterprises use a variety of information systems, tools and disparate data sources which often function isolated with point-to-point or no interfaces and no complete solution to monitor and manage all activities. Independent applications that do not allow automatic connections force the users to perform manually the data exchanges thus leading to wasting valuable time, efforts and resources to respond to business needs, as well as achieving operational and strategic targets with low efficiency, flexibility, reliability and with high risk of mistakes.
- **Data integration:** There's a major lack of data integration between different sources resulting into low visibility and information availability as well as inefficient flow of information, leading to high efforts for manual data integration and configuration. Further, integrated views on different processes e.g. manufacturing, logistics at different levels to enable analysis and optimization of these processes and the historical views are often limited.
- **Visualisation:** The user does not have a unified visualisation of data originating from different systems and has to access various sources in order to get an overall visual of the plant and evaluate the plant activities. Different systems use completely different visualisations and require an overall unification of visualisations to provide a complete understanding.
- **Eventing:** Events and alarms are generated and monitored within individual systems, where cross-boundary propagation and aggregation is beyond the state-of-the-art for most current operations, thus making it a difficult task to efficiently raise awareness in a complex situation. There are also some cases such as ERP that do not support eventing.
- **Root cause analysis capability:** The rising complexity of managing and understanding the interdependencies between related operating functions, KPIs and the causes of missed targets in combination with the huge amount of disparate data make the root cause analysis an extremely difficult and time-consuming task, performed only by experts.
- **User profiles management:** Different systems, within a plant, make different types of user profiles and accessibility rights for the same users coming from departments such as management, finance, engineering or technicians, instead of having one user profile, according to his/her position.

PLANTCockpit Vision

Available solutions

A variety of technologies exists in the field of data integration and manufacturing systems that PLANTCockpit deals with. Since it is not really possible to cover all the state of the art related to this work, the focus here is on the key technologies in the area.

Currently, the interconnected enterprise consists of a combination of specific components, systems and software tools such as SCADA, MES and ERP. Depending on the requirements from production areas, this is realized by implementing a selected combination of various stand-alone applications, each covering a limited group of functions.

Firstly, a typical system that is originated in the process control level is a Supervisory Control and Data Acquisition (SCADA) system. The SCADA systems are controlling and displaying technical processes in a plant and are often also a part of manufacturing execution level. The information content that is handled by a SCADA system is limited however to the processes that are controlled by the system (ZVEI, 2010).

MES undertakes responsibility in the plant layer, where it controls the workflow on a higher level than the SCADA systems and their functions, including functionalities such as production scheduling, shipping, product inventory control, quality assurance, material and energy control, and production control (ZVEI, 2010). Although MES in theory could be a complete solution, in reality each MES is a single vendor stand-alone system that offers a subset of the envisioned functionalities.

On the enterprise level, ERP systems provide extensive functionality for production logistics, planning and executing manufacturing operations, such as customer relationship management, financial management, human resource management, marketing or documents management. However, for real-time decision support today's ERP systems do not contain sufficient information. Further information about responsibilities and functions in this layer can be found for example in (Wagner & Monk, 2008).

On a different level, although there is a number of commercial cockpits with extensive visualizations like SAP Business Objects (SAP AG) or Siemens SPPA-M3000 (Siemens AG), they are as a rule targeting only one of the layers and require special middleware to access external data sources. In order to achieve a flexible and easily configurable cockpit solution, it is necessary to put the system functionality in re-usable and configurable components, such as the Function Blocks described in IEC 61499 (International Electrotechnical Commission, 2005). The IEC 61499 standard is developed for the programming of industrial controllers in a graphical way and the execution is event based. The IEC 61499 also introduces the so called service interface blocks, which encapsulate communication means with other systems or sources outside the Function Block approach (Vyatkin, 2011).

Lastly, state-of-the-art architectural patterns like Service-Oriented Architecture (SOA) and Enterprise Service Bus (ESB) can be very important in a heterogeneous, multi-vendor environment whilst aiming to provide interoperable services. SOA assists in integrating widely disparate applications by supporting a core service that abstracts from the heterogeneity of the systems the data comes from (Lorenz, 2006). Furthermore, the usage of an Enterprise Service

Bus (ESB) leads to a loosely coupled and highly distributable system (Groba, Braun, & Wollschlaeger, 2008).Overall, it is important to note that there is no generic one sizes fits all solution in the area of data integration and manufacturing systems. A true integration of applications and therefore a holistic view of production and business processes is still missing. This is where the EU funded research project PLANTCockpit intends to seize the opportunity to close the gap by providing a flexible and high-performance solution for modern industrial environments.

General vision of PLANTCockpit and functionalities overview

The vision of the "Production Logistics and Sustainability Cockpit"(PLANTCockpit) is to offer to manufacturing communities a central environment for monitoring and visualizing all plant processes, such as manufacturing and logistics. Virtually all modern manufacturing operations are multi-vendor environments, which instantly makes the harmonization of all systems a highly challenging goal. PLANTCockpit aims to provide to a variety of users within a plant, such as production managers and foremen, the necessary information and the required visibility in order to make well-informed decisions for optimizing plant processes. This includes the holistic visibility of the plan, the current status, deviations and exceptions, and bottlenecks. PLANTCockpit targets all three layers of the automation pyramid and tries to bridge the gap between them, by providing a platform for integrating the heterogeneous shop floor management systems including ERP, MES, SCADA and other special-purpose systems.

Figure 1 shows PLANTCockpit's position in the automation pyramid based on the IEC 62264 (International Electrotechnical Commission, 2007) which is the international standard for enterprise-control system integration) and the relevant layers serving as data sources for PLANTCockpit. These are the enterprise level (including e.g. ERP tools), the manufacturing execution level (typically including MES) and the process control level.

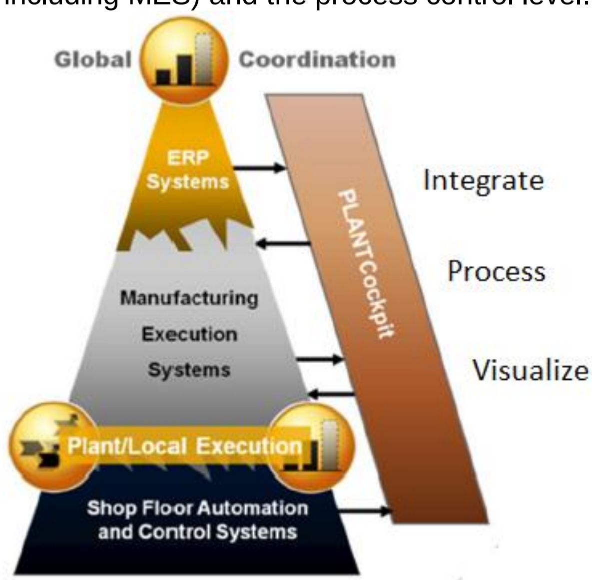

Figure 1. Place of PLANTCockpit in the automation pyramid.

The most important PLANTCockpit functionalities are now briefly described in order to provide a deeper overview of the solution and show the drivers for its development. The PLANTCockpit project addresses 5 use cases from different industries covering a variety of common and heterogeneous use case specific requirements. The key functionalities for all use cases focus on accessing a diverse set of external source systems, mapping and transforming the incoming information into PLANTCockpit specific data and then visualizing it to the end-user, with a wide significant range of valuable visualization tools like charts, data grids or traffic lights. Different types of information, such as KPIs (Key Performance Indicators), tree hierarchies, processes, reports, historical values and events are efficiently imported and visualized to the user.

The PLANTCockpit solution is a central environment for monitoring and analysis that spans disparate information systems. It will vastly facilitate decision support by defining and displaying targets, identifying thresholds, performing calculations, providing comparisons and most importantly supporting the invaluable ability to drill-down for root cause analysis in order to assist the decision makers in getting detailed information faster than it is the case now. Furthermore, a very powerful aspect of the proposed platform suggests that the system will provide an efficient and configurable notification and alerting mechanism to the end-user, according to defined notification rules. In this way he/she can access all necessary information in a timely manner e.g. in case of a status change.

A key PLANTCockpit feature is the provision of a single sign-on mechanism for all systems whilst at the same time guaranteeing security access to the end-user. In addition, the user will have the ability to define, assemble and change predefined and user specific views, using the available visualization components and according to their area of responsibility which improves efficiency. The system will also provide the possibility to create and persist reports, as well as storing historical information which is important for keeping record on previous situations for further analysis as well as from legal point of view.

Lastly, a major characteristic of PLANTCockpit is that it will remain flexible, extensible and configurable so that it can easily add new source systems and new functionalities, making it an ideal solution for all types of modern industries.

Design approach and principles

The aim of the solution is to provide a platform that is capable to cover diverse use cases and act on different automation levels, as demonstrated in Figure 1, as well as to support a large range of operational scenarios in terms of scope and breadth of deployment within different industrial environments. Based on the requirements analysis method for heterogeneous requirements, the proposed solution uses SOA principles in order to enable the architecture to evolve and provide all the functionality required in a flexible and structured manner, which can be adapted to the use case requirements. This allows the development of an extensible and lightweight framework, capable of supporting multiple heterogeneous environments. An important benefit of using SOA as the basis of the architecture is that it supports great flexibility in deployment and controlled iteration in development. The key aspects of establishing a successful architecture hinge on the robust development of contractual interfaces for self-describing services and a reliable mechanism for service discovery.

The principles of architecture summarize consensus, universal rules and the policy for the development of systems architecture of PLANTCockpit. In order to provide the ability to satisfy heterogeneous requirements from a variety of diverse use cases, the chosen architectural approach is based on four principles:

Loose coupling between independent layers and between their components: This approach demands generic interfaces and meta-models to support the integration between the layers and the single components within them. The communication is realized by a message-based/event-driven architecture that provides three main advantages to the final solution. Firstly, the architecture allows customization in order to adjust to different use cases. Secondly, it provides the possibility to use different technologies for the realization of the components. Lastly, loose coupling allows easy extensibility for new functionalities and models.

- Standardized message exchange mechanism: Based on the previous principle, the technical architecture has to provide standardised mechanisms for incorporating the user's additional or subsequently built functionalities, based on a standardized structured message exchange mechanism.
- Easy configuration of the components on different levels as well as of the entire solution.
- Independent and extensible solution: Based on the previous principles, it is necessary to provide at the same time an independent solution and also the ability to incorporate existing solutions into the final application.

Thus the principles for the proposed architecture can be summarized as providing a layered, independent, standardized, and extensible architecture.

Solution Details

PLANTCockpit Architecture

The PLANTCockpit architecture was developed based on the concept of the layer-based architecture, focusing on the major functional aspects concerning production and logistic cockpits and driven by the user requirements and the four architecture principles. The architecture is designed to be layer-based, in order to separate different functional aspects and facilitate customization and configuration, allow the use of different technologies for the implementation and provide easy extensibility for new functionalities. The connection between these layers and different components of the solution is provided by message-based uniform and standardized interfaces. Each layer provides various functional components with specific functionalities and connections between them. All the layers and their inner components are described in detail in the following subsections.

The proposed PLANTCockpit architecture for production and logistics cockpits is demonstrated in Figure 2 and consists of the following layers:

- Function Engine layer
- Persistence Engine layer
- External Systems layer
- Visualization layer
- Presentation Layer

Figure 2 displays how the autonomous layers can be placed and connected, while selecting different technologies and configurations for their implementation. The architecture presents two clear borders which are described in the following subsections in details. The first border separates the solution from the External Systems Layer, which is not a part of the architecture and is accessed by interfaces from the Adapter and Transformation component. The second border is within the architecture between the Visualization Engine Layer on the server side and the Presentation Engine Layer on the client side.

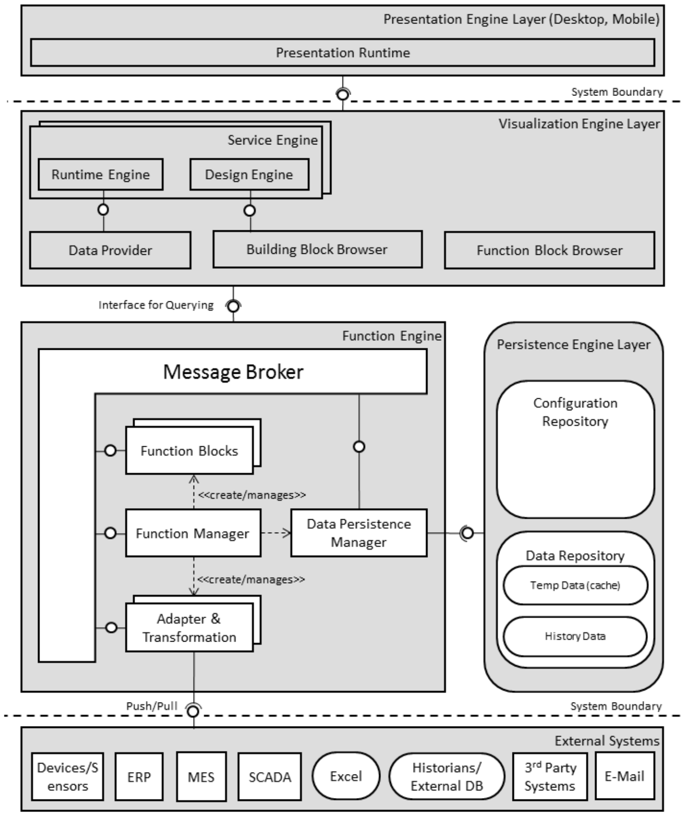

Figure 2. Layered architecture for production and logistics cockpits

Function engine layer

The Function Engine Layer is the core part of the proposed architecture. It provides a flexible, adaptable and highly extensible execution environment to manage and accomplish analytic process logic. It can be considered as a framework that supports the initiation and execution of functions, which are realized as components called Function Blocks, managed by a Function Manager. The complexity of the Function Blocks may vary from simple calculation to sophisticated data processing. For example, a Function Block can calculate data based on input data and transfer the results to other Function Blocks or layers. The Function Engine Layer predefines some common functions used by all Function Blocks and provides the ability to add new ones or extend existing ones with new functionality via a plug-in capability. At its core, the Function Engine layer utilizes the event-driven architecture pattern promoting the generation, detection, consumption of and reaction to events, thus building a responsive and scalable framework.

The Function Engine Layer specifies an interface to the External Systems Layer via Adapters, to support query for data and event-based message notification of external systems, and to the Persistence Engine Layer via the Persistence Manager in order to manage all the internal data of the framework. Further, the interface to the Visualization Engine Layer is defined via the

Message Broker to execute data requests, triggered by user interaction. The basic components included in the Function Engine Layer are described in detail below:

- Function Blocks (FB): Adapted from the object-oriented programming and IEC 61499, Function Blocks can be considered as a component type that is composed from structural and behavioural constituents. In general, each Function Block type provides specific executable program code as well as input and output parameters declaring its data structure as Data Points. A Function Block type is a reusable and modular construct that can be composed by the Function Manager like a blueprint, to create Function Block instances by providing them with specific configuration. Figure 3 illustrates a Function Block type structure. Those instances are lightweight components which can be deployed on the Function Engine and their business logic can be executed during runtime. Communication between Function Blocks is supported by the Message Broker. The proposed approach pursues a small baseline framework with only common functions, leaving other exotic functionality to be implemented by plug-in extensions and uploading of new Function Blocks, thus enriching the functionality range. This structure provides many advantages for the framework, such as extensibility, reusability, interoperability, reduced application size and process orchestration of Function Blocks.
- Function Manager: The component manages the whole lifecycle of Function Blocks, beginning with the creation (instantiation), provision of configuration (initialization), activation, deactivation and subsequent destruction.
- Message Broker: This publish-subscribe component is responsible for facilitating the communication between Function Blocks and loosely coupled components via messaging mechanisms negotiating service contracts. The duration of a contract between components is long-term rather than a one-time invocation and is maintained by the Message Broker. Each Function Block has the responsibility to update subscriptions of data from a publisher, based on their information needs. The publisher has service agreements to propagate the information to the subscriber by engaging services provided by the Message Broker, such as event propagation, message routing and queuing. The aim is to achieve security, trustful communication, reliable, consistent and guaranteed message delivery, unicast (one-to-one) and multicast (one-to-many) message delivery. The idea is to use one of the off-the-shelf message broker solutions, like ActiveMQ, in order to achieve higher interoperability level, compliance with existing standards, avoid message congestion and to fulfil high requirements for time criticality.
- Adapters and Transformation components: These components are responsible for maintaining the connection and integration of the system with a variety of external systems, such as ERP systems, MES and SCADA systems, OPC servers, files, devices or sensors. The aim is to unify the data in these components from heterogeneous interfaces into internal data formats, which can then be processed further in the Function Blocks. It is designed to be flexible, extensible and reusable, by using plug-in mechanisms to guarantee that the possible extension to other external source systems is covered. These concepts are also applied in IEC 61499.

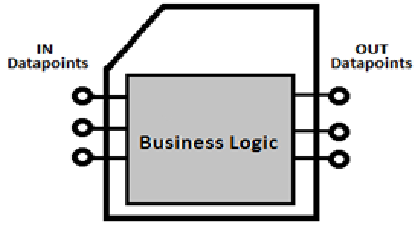

Figure 3. Function Block Type.

Persistence Engine layer

The Persistence Engine Layer includes the components necessary to store all the required internal data for the framework. This data can mainly be classified into either configuration data, which is necessary to configure the internal components of the system, or analytical data, such as runtime and business data. This layer is also responsible for persisting generated and retrieved runtime data like raw data, aggregated data or historical data. The Persistence Engine layer provides an interface and exposes the data via the Persistence Manager in the Function Engine Layer. The main components involved in data persisting are:

- Persistence Manager: The component is responsible for storing, long-time archiving, retrieving and deleting of any kind of data. It provides the access to all persisted configuration and analytical data in the Persistence Engine Layer. It manages the configuration settings of the different systems, both internal and external. Further, it updates the configuration whenever the user modifies it, by transferring all changes made in design time phase to runtime and reinitializing the system with new settings and parameters. The Persistence Manager and Repositories can be realized either as a single instance or as multiple instances in order to ensure reliable data persisting, load balancing and to avoid performance bottlenecks.
- Configuration Repository: The component contains the configuration data of the system's components, which can be stored at design time or at run-time. In run-time phase of the solution lifecycle the system is running and fulfilling its functionality. In design-time phase, the system is configured and prepared for the running. Design-time configuration data can be changed only through the process of actualization and can be used at design and at run time. Run-time configuration data can be changed during run-time, without changing the design data, and are in most cases held by components, persisted only in selected cases. The repository includes a variety of configuration data, such as adapter configuration data, meta-models, visualization settings, Function Blocks settings, communication logic or general framework settings.
- Data Repository: It contains all the data that is necessary during the analytical process in the solution. Most of the data contained in the external systems should not be duplicated in the system but queried at runtime to ensure the data consistency. However, in some cases the data should be temporarily cached in order to improve the performance. The Data Repository may contain run-time data, stored only temporarily and available only during one working session or historical data, archived for a long time. Various types of data are possible, such as run-time raw data, data created in the current solution, business objects data, high-level or low-level KPIs, notification data, visualization data or logging data.

External systems layer

This layer is not a part of the solution and is only needed for comprehensibility and completeness of the architecture and to emphasise that PLANTCockpit solution is not a standalone solution and exists only on top of existing external systems. This layer bundles existing and required external open or proprietary data sources that may be connected to a production and logistics cockpit e.g. ERP, web services, databases, Excel files, sensors, actuators, OPC (UA) servers or other 3rd party systems. Each external system provides access to its data via proprietary or standardized interfaces. The individual components in the layer are the source systems which are accessible via the adapting and transformation components in the Function Engine.

Visualization layer

The Visualization Engine Layer processes the data coming from the Function Engine and prepares the visualization information for the Presentation Engine. It’s a server-side layer, responsible for preparing the UI for presentation and providing easy access for consumption of the solution’s data. This covers the mapping of the solution’s internal data format to the required format and values for the visualisation elements as well as the processing of the data changes and their subsequent transforming into visualization updates. The Visualization Engine Layer provides interfaces to the Presentation Engine Layer consisting of graphical updates and user interactions information and to the Function Engine Layer for data requests towards the Message Broker. The layer consists of the following components:

- Building Block (BB): Is basic visualization element which is displayed in the Presentation Engine Layer. The building block defines a simple or complex data structure required to represent data and subscribes for data from data points through the Data Provider component. It generates the visualization updates which are delivered by the Service Engine to the Presentation Engine. The concept of Building Blocks is extremely flexible and extensible, supporting predefined and supplemental visual elements.
- Service Engine: A back-end component creating the visualization information necessary for the front-end clients. It is a manager component that provides data for configuration of Runtime Engine and Design Engine components. Each front-end client has a single Service Engine dedicated to it.
- Runtime Engine: The component contains the UI’s composition of Building Blocks that have been configured for the user. It creates and destroys the Building Blocks and serves them with the data updates from the Data Provider.
- Design Engine: It creates the configuration consisting of building blocks for the Runtime Engine, and defines the assignments of data points to building blocks. The Design Engine loads UI components (Building Blocks) via the Building Block Browser to user interface.
- Data Provider: The component aggregates all the service engine data requests for the Message Broker component to get the data and delivers them to the Runtime Engines.
- Building Block Browser: It provides all available building blocks to the Design Engine in order to build a UI’s Runtime Engine.
- Function Block Browser: It shows available data points, creates new Data Points in Function Blocks and allows their assignment to building blocks.

Development technologies

As mentioned previously, the PLANTCockpit solution is based on independent layers whose functionalities and configuration are technology independent. However, based on the described generic functionalities, this section presents the key technologies used for the PLANTCockpit implementation in order to provide an overview of some key applicable technologies in the solution.
Adapters are a key component for connecting PLANTCockpit to external data sources, by tying into the Function Engine, exposing a function block interface towards the framework. These adapters are related to the specific requirements of the external systems and the most important ones are briefly described as follows:

- **SAP ERP / BAPI / Business Objects:** Data stored in the SAP ERP system is modelled as Business Objects (BOs), as real world objects like a customer order or an employee. In order to allow the integration between PLANTCockpit and SAP ERP system, Business Application Programming Interfaces (BAPIs) are used to enable an object based communication and the execution of business functionalities in SAP ERP, without any need for the user to know the underlying implementation details(SAP, http://help.sap.com ,2012).
- **SAP BW / BEx Queries:** The data connection between PLANTCockpit and SAP Business Warehouse is based on SAP BAPI's and its corresponding JAVA connector. With the developed adapter it is possible to query arbitrary mdx queries in the business warehouse and perform analysis on the query results. Those queries can be created through the SAP Business Explorer (BEx) (SAP, sdn.sap.com , 2012).
- **Web Service:** Web Services technology can be used to implement a service-oriented architecture, where SOAP messages are the basic unit of communication. The interface is described in an XML-based machine-readable format, called Web Services Description Language and machines interact with the Web Service by exchanging SOAP messages, typically using HTTP (Haas & Brown, 2004).
- **DPWS:** The Devices Profile for Web Services (DPWS) defines a minimal set of implementation requirements for dynamic discovery, service description, secure messaging, and events and subscriptions, aiming to provide interoperability analogous to Universal Plug and Play (UPnP) for networked devices, fully aligned with Web Services technology (Microsoft, http://msdn.microsoft.com/en-us/library/bb736556(v=VS.85).aspx , 2012).
- **MS Excel:** Excel files are accessible through Apache POI, Automation, as well as using comma-separated values format (CSV).
- **MS Project.** It is a project management tool, designed to help project managers plan project stages and tasks. Although not primarily designed with the goal of data exchange between applications, it is accessible based on the MPXJ library and OLE Automation.
- **OPC Unified Architecture:** A relatively new specification from the OPC Foundation for data exchange between systems in industrial applications, based on web-service concepts. OPC UA is designed for accessing large amounts of real-time device data using standard network infrastructure, while maintaining sufficiently high performance.
- **SQL:** It's a programming language designed especially for access to relational database management systems that belongs to the family of data manipulation languages (DML) and is used to retrieve and manipulate data, through various approaches (ISO/IEC9075, http://www.contrib.andrew.cmu.edu/~shadow/sql/sql1992.txt , 1992).

Furthermore, some key technologies for the visualisation part of PLANTCockpit are presented below. It is important to note that due to the diversity of the use cases, a combination of technologies was employed.

- **Windows Presentation Foundation:** A graphic framework distributed by Microsoft that supports many features including graphic hardware acceleration, functionality for 2D and 3D graphics, media support and predefined user interaction interface components (e.g. panels, buttons, menus, list views). WPF also allows, using the Microsoft's .NET software framework, the creation of sets of user interface components and the ability to save them as one unit (Microsoft, http://msdn.microsoft.com/en-us/library/ms754130.aspx , 2012).
- **HTML5:** A hypertext mark-up language elaborated to create and present content used in the Web, with features designed to make it easy to include and handle multimedia and graphical content, without needing to resort to proprietary plug-ins and APIs (W3C HTML Working Group, http://www.w3.org/TR/html5/ , 2012).
- **Scalable Vector Graphics:** SVG is a family of XML- based specifications for two-dimensional vector graphics, both static and dynamic, which combined with HTML5 and JavaScript offers an efficient platform for covering a variety of visualisation requirements (W3C SVG Working Group, http://www.w3.org/TR/SVG/ , 2012).

Most importantly, in order to achieve the desired integration between different systems, a mix of various technologies may be used:

- **Apache ServiceMix**: It is a flexible, open-source integration container that unifies the features and functionality of various tools into a powerful runtime platform which can be used to build integrations solutions. It provides a complete, enterprise ready Enterprise Service Bus exclusively powered by OSGi (ASF ServiceMix, http://servicemix.apache.org/, 2012).
- **Java Message Service:** It provides standard Java APIs that Java developers can use to access the common features of enterprise message systems. The communication with JMS between different components is loosely coupled, asynchronous and reliable (Sun Microsystems, http://docs.oracle.com/cd/E19957-01/816-5904-10/816-5904-10.pdf, 2012).
- **Apache ActiveMQ:** An open source messaging and Enterprise Integration Patterns server that serves as a message broker and fully implements Java Message Service specification (ASF ActiveMQ, http://activemq.apache.org/ , 2012).
- **Apache Camel:** An open source integration framework based on known Enterprise Integration Patterns by supporting Bean Integration to facilitate message routing and mediation (Apache Camel, http://camel.apache.org/eip.html, 2012).
- **Spring:** It provides a powerful and flexible set of tools for development of enterprise Java applications and is used to create high performing, easily testable, reusable code without any lock-in (SpringSource, http://www.springsource.org/, 2012).

Business Benefits

PLANTCockpit presents a platform for different industries and application scenarios where various business benefits are expected from its implementation. The proposed architecture has already been validated through the development of several prototypes for all use cases, where its applicability to all use cases was proven and the business benefits were demonstrated. Firstly, the benefits for each PLANTCockpit use case are separately presented in order to demonstrate the various impacts of PLANTCockpit among different kinds of industries. In the second part, the overall benefits of the PLANTCockpit platform are shown to present why PLANTCockpit is an effective solution to improve the overall activities of any modern plant.

Business benefits per use case

PLANTCockpit will provide a platform to increase the productivity in the highly complex production process of pultrusion, through extended visibility and vast improvement of the management and control of all the main parameters involved.
Through the optimization of the pultrusion production process and the early identification of deviations, Acciona expects from the PLANTCockpit solution to have the following impacts:

- Reduction of process stops time by 15%
- Reduction of waste by 20%
- Increase of productivity by 30%
- Reduction in power and materials consumption of 10%
- Cost reduction by 15%

PLANTCockpit will allow the user to have a customised view of the processes in his/her area of interest and immediate access to all aggregated information, in order to make optimal and transparent decisions. Value nets will replace the traditional value chains and cross-dependencies between different sectors of industry - like for automotive to high tech suppliers – will have a rapid and pervasive impact to area of supply. Quality issues will have immediate global effects and on-time delivery will be ensured for the customers through rapid and thorough resolving of problems in the areas of supply or quality. Overall, the BMW Group expects to achieve a global optimum of its processes.
Through PLANTCockpit, the BMW Group specifically expects to be able to monitor and analyze the KPIs and their complex interdependencies in order to achieve a holistic visibility of processes and perform an improved root cause analysis. This will lead towards the following impacts:

- Proper car bodies always available in assembly
- Reduction of production time

- Stable processes by ensuring on-time-delivery

PLANTCockpit will provide a solution for an efficient asset management in the extended enterprise environment and all asset information will be put to effective use to improve the plant planning and scheduling activities in order to respond to market demand, whilst also considering the external supply chain. Furthermore, PLANTCockpit will help to develop an advanced energy monitoring, analysis, and management system to enable Comau plant management to overcome present limitations in the area of sustainable production.
The overall expected impacts of PLANTCockpit project are:

- Lead time reduction of up to 15%
- Cost reductions up to 20% through optimized processes and better energy/resources management
- Reduction waste production of up to 20%
- Overall efficiency of processes + 15%
- Confirmation of the certification European Standard EN 16001 – Energy Management
- Cost reductions up to 20% through optimized processes and better energy management
- Increased overall sustainability of manufacturing processes

Taking into account the novel monitoring and eventing tools that PLANTCockpit will provide, Doehler Group expects to see a big improvement in visualizing the gaps across the supply chain and the deviations in the actual production process.
As the main parts of the supply chain (planning, logistics, production) will be made visible in the new PLANTCockpit platform, the aim is to achieve:

- Improvement in planning and processing times within logistics, production and quality control
- Improvement of the delivery reliability and overall delivery service
- Increased customer satisfaction

The PLANTCockpit solution will integrate the standalone energy management systems and allow the utility consumption, i.e. electricity, water and gas, to be tracked and monitored at various levels of operations within the Intel factory.
Specifically, the PLANTCockpit solution will benefit Intel in the following ways:

- Provide mechanisms to be the more energy and cost efficient
- Faster response to business needs through real-time visibility extending from the shop-floor to business levels, e.g. rapid response to changes in product priority
- Reduced cognitive workload through better integration of diverse information systems
- Seamless communication between all stakeholders
- Optimised interaction strategies for data access/manipulation
- Capability to support improvements in specific targeted use cases

Overview of the solution impact

The implementation of the PLANTCockpit solution can be the key for any modern industrial environment to integrate all information from various plant areas to provide complete visibility of plant activities and greatly improve the efficiency of its operations. More specifically, PLANTCockpit provides:

- **High usability:** PLANTCockpit offers a user-friendly environment, with various visualisation tools, such as traffic lights, Gantt charts and data grids, in order to provide a complete picture of the plant activities, according to the user's area of interest.

- **High adaptability**: The solution fits to various use-cases and covers very heterogeneous requirements concerning production and logistics cockpits, with minimal customization efforts needed only for specific use-case requirements.

- **High technology flexibility:** The architecture is designed in a way which allows implementing different components and layers with different technologies and tools.

- **High development flexibility:** The concepts of loosely coupled components and a standardized message exchange mechanism allow parallel development and subsequent integration of different functionalities.

- **Easy deployment:** The final solution can be deployed in a distributed environment or on a centralized machine. Also it is not necessary to have dedicated servers to run the system and it is possible to deploy it on already existing IT infrastructure.

- **Increased productivity:** The extended visibility from the shop-floor to the enterprise level guarantees efficient and swift decision making and thus increased productivity.

- **Adherence to well-known standards** such as OPC and EN16001.

- **Ensured return on investment**: The use of open-source software in PLANTCockpit and the possibility to deploy on existing infrastructure make for minimal costs whereas the increased productivity guarantees the financial gain.

- **High speed of implementation:** The technologies and functionalities as well as the standardised interfaces make for an easy implementation of the solution in an industrial environment with additional effort required only for use-case specific functionalities.

- **Efficient asset management:** All asset information within a plant can be transparent so the user can optimize their performance.

- **Improved energy management strategy:** Increased visibility over energy management systems allows an improved monitoring of all related activities, thus improving the energy efficiency.

- **Efficiency and optimization of processes:** Holistic visibility of plant activities and early identification of possible deviations leads towards the optimization of plant processes.

Summary

With this document we have presented the innovative PLANTCockpit solution that enables an integrated view on the various industry processes e.g. manufacturing and logistics. PLANTCockpit proposes the development of a cockpit solution enabling generic access on the various plant information systems, flexible analysis and visualization of the information as well as easy user and application configuration. The outstanding features of the architecture make the solution applicable in many diverse use cases and industries.

The solution has been so far validated within the PLANTCockpit project through the development of several prototypes for the project use cases, where the business benefits were clearly demonstrated. The platform is currently under development where we address a more detailed revision of the infrastructure components to enable an ease of use and adoption of the framework. We are also further working on building new prototypes for the creation of a cockpit which will be easily adjustable to the use cases from different industries, allowing a new level of the optimization of manufacturing processes under involvement of heterogeneous information systems.

References

Groba, C., Braun, I., & Wollschlaeger, M. (2008). A Service-Oriented Approach for Increasing Flexibility in Manufacturing. *IEEE International Workshop on Factory Communication Systems WFCS.* Dresden.

Haas, H., & Brown, A. (2004, February). *Web Services Glossary, World Wide Web Consortium (W3C) Working Group Note*. Retrieved 2012 July, from World Wide Web Consortium: http://www.w3.org/TR/ws-gloss/

International Electrotechnical Commission. (2005). IEC 61499-2: Function Blocks - Part 2: Software tool requirements, International Standard, First Edition.

International Electrotechnical Commission. (2007). IEC 62264 1-3: Enterprise-control system integration.

Lorenz, M. (2006, October). *IBM Certified SOA Solution Designer certification prep, Part 1: SOA best practices.* . Retrieved July 2012, from IBM developerWorks: http://www.ibm.com/developerworks/webservices/tutorials/ws-soacert1/section2.html

SAP AG. (n.d.). *See Your Business Clearly*. Retrieved July 2012, from SAP BusinessObjects Dashboards: http://www.sap.com/uk/solutions/sapbusinessobjects/large/business-intelligence/dashboards/sapbusinessobjects-dashboards/index.epx

Siemens AG. (n.d.). *SPPA-M3000 Plant Management*. Retrieved July 2012, from Siemens: http://www.energy.siemens.com/co/en/automation/power-generation/sppa-m3000/sppa-m3000-plant-management.htm#content=Description

Vyatkin, V. (2011). IEC 61499 as Enabler of Distributed and Intelligent Automation: State-of-the-Art Review. *IEEE Transactions on Industrial Informatics, Bd.7, Nr. 4* , 768-871.

Wagner, B., & Monk, E. (2008). *Enterprise Resource Planning.* Course Technology.

ZVEI. (2010). *Manufacturing Execution Systems (MES).* Frankfurt.