OPC UA supporting the automated engineering of production monitoring and control systems

Miriam Schleipen
Fraunhofer Institute
for Information and Data Processing (IITB)
Fraunhoferstr.1
D-76131 Karlsruhe
Miriam.schleipen@iitb.fraunhofer.de

Abstract

Control systems monitor and control production plants and manufacturing systems. Before they can be taken into operation, they have to be configured. To automate this process, which is called engineering, a standardized exchange format is required that forms the basis of the work of all those who are involved in the project and that is understood by anyone involved. In addition, the contents to be exchanged have to be transmitted on the basis of a standardized communication mechanism with maximum efficiency. This article will present a standardized communication method capable of fulfilling these requirements. On the basis of an OPC UA framework acting as a communication, synchronization and processing mechanism, the opportunities of the Unified Architecture and its application in the automated engineering of Control systems will be evaluated.

1 Introduction

Efficient modifications to manufacturing systems confront today's business practice with considerable difficulties, resulting in increased demands on the underlying production plants in terms of adaptivity and interoperability. 'ManuFuture Germany', an initiative working out Germany's Strategic Research Agenda, identified adaptivity as the most relevant topic in the automobile industry in September 2007 [1] (see figure 1).

In order to be able to exchange information in an efficient and usable way, all systems involved have to interact as seamlessly as possible – even though they operate in a heterogeneous environment in most cases. This is called interoperability. Its increasing significance reflects constantly growing customer requirements and continually enhanced technologies. This results in ever new products and product variants. Since the potentials for increasing efficiency in the plant engineering process have largely been exhausted in other areas, the cost pressure in modifications to or re-engineering of existing plants and the development of new plants is increasing constantly. The perpetual changes result in plants having a significantly longer life cycle than products, compelling plant operators to re-configure plants dynamically and to integrate new equipment and components of equipment with existing production systems as quickly as possible. This also requires modifications to the production monitoring and control system.

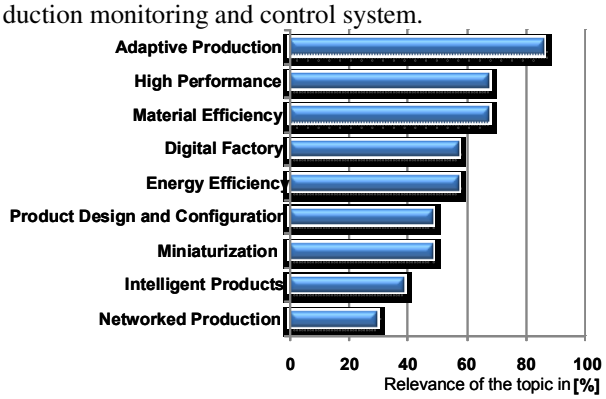

Figure 1. Adaptivity and interoperability as main topics in the automobile industry [1]

This problem is to be remedied by an OPC UA architecture supporting the automated engineering of production monitoring and control systems. For the ProVis production suite, there is a prototypical engineering framework (see figure 2), which imports the information describing the plant in the standardized CAEX format (Computer Aided Engineering Exchange, see [2]). As CAEX defines only the structural standardization, it has to be used as defined in [3]. On the basis of the CAEX data, the ProVis.Agent® production monitoring and control system is engineered dynamically using OPC UA (OPC Unified Architecture, see [4]), while the relevant process visualization is generated in ProVis.Visu®.

The exchange and processing of information in the modification framework of the production monitoring and control system is based on an OPC UA client-server architecture. Since this is a new communication standard which is not very wide-spread yet – unlike the established older OPC standards - a Web service was added to

avoid enquiries by the information provider regarding OPC UA.

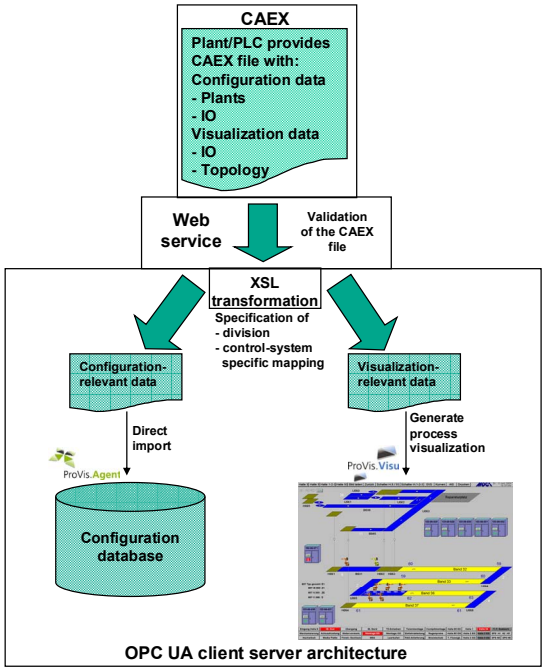

Figure 2. Engineering workflow

This Web service is also used in the context of ProduFlexil [5]. The CAEX-based plant description is first checked with respect to formal correctness. In this process, the CAEX file is validated against the standard CAEX-XML schema and rejected, if necessary. If the validation is successful, the semantics structured and contained in the CAEX file can be processed further.

In a second step, XSL transformations and other mechanisms are applied to the received data. The resulting new CAEX files form the basis of the engineering of the production monitoring and control system, i. e. the data is adapted for ProVis.Agent® and imported in the engineering database. The visualization-relevant part of the data is transformed into the ProVis.Visu® process images directly.

2 Motivation

"Taking a process into operation is one of the most critical moments in its life cycle." [6] In this phase, efficiency can be enhanced by adaptive components and flexible systems that allow for a rapid re-configuration. In this context, the keyword is 'to configure rather than to program'. Owing to plug-and-play mechanisms, the majority of what used to be manual and cost-intensive customizing and engineering activities is rendered unnecessary or can be shifted to an earlier phase of the life-cycle. This allows for a higher degree of maturity. Thus, the main benefits of the mechanisms developed are the quicker start-up or restart of plants and the preceding debugging.

In order to implement the mechanism described in section 1, various single tasks need to be performed. They are depicted in figure 3.

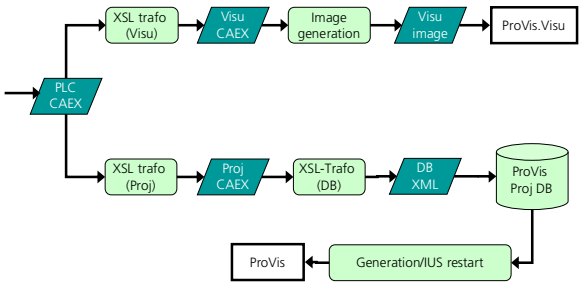

Figure 3. Overall procedure of control system engineering process

The communication concept to be developed is expected to meet a number of requirements. As it entails decisive changes in the production monitoring and control system, it is supposed to be future-oriented. It has to have a modular design so as to allow for the gradual integration of more and more contents. In order to enhance the functionality, extensibility plays a major part, too. Once a communication standard has been applied for the exchange of engineering contents, it is desirable to use the very same standard for the online communication of process signals. Hence, it should be possible to integrate process level communication as well. Owing to the multitude of data sources and existing IT infrastructures plant operators are faced with, the entire application must be able to run in a distributed environment. This means that the components operate on different computers or even different networks. And even though production monitoring and control system engineering usually deals with contents that are exchanged offline, the automated engineering calls for the online communication of data. For this reason, the volume of data should be reduced to a minimum for all mechanisms involved, and communication should be as efficient as possible.

Having examined current communication technologies, Fraunhofer IITB has found OPC's successor, the OPC Unified Architecture, to be the appropriate standard because it fulfills all the criteria required.

3 The OPC Unified Architecture (OPC UA)

As was mentioned above, the data transmitted to the change management of the production monitoring and control system is processed on the basis of a service-oriented client-server architecture applying OPC UA (OPC Unified Architecture). Currently, OLE for Process Control or OPC is *the* communication standard in automation technology. IITB's Production monitoring and control systems business unit has been a full-fledged ('corporate') member of the OPC Foundation since 2007,

which allows IITB to have permanent access to the latest developments and information.

In 2006, the OPC Foundation presented its Unified Architecture (see [7] to [18]), which standardizes the previous specification. For one thing, the specifications that had evolved in the course of time were to be unified, making it superfluous that a special server was needed for each partial functionality. For another, the binding to COM/DCOM brought about several drawbacks such as the dependence on the Windows operating system and frequent problems in the configuration of DCOM. In addition, COM/DCOM did not allow developers to be in 'genuine' control of the processes, and they were not able to fix errors because the source code was not available, as this technology is a black box. Moreover, there was no 'true' security, which is of major importance in this area.

Consequently, a new generation of OPC servers had to be developed, the OPC Unified Architecture. It allows for the access to various kinds of data and the vertical and horizontal exchange of data in a multi-functional ('unified') server. In addition, it offers extended reliability and interoperability. The robust transfer of data does not depend on communication protocols. Diagnosis is integrated in the OPC components. There are simple concepts for client and server redundancy. Keep-alive messages for the first time allow the clients to realize that the server or the communication channel have been disconnected. This is called heartbeat in OPC UA because this link is monitored in both directions. Timeouts can be configured for any of these services and messages that have not been received can easily be requested again using sequence numbers. Thus, interruptions in the connection do no longer result in the loss of data.

Figure 4 shows the OPC UA in interaction with the different levels. The striking feature is that it can even be used to transfer data to the enterprise level.

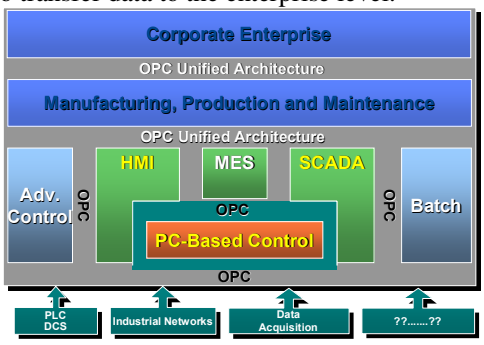

Figure 4. OPC UA in interaction with different levels [19]

OPC UA is based on standards such as TCP/IP, HTTP, SOAP and XML and marks the transition from DCOM to a service-oriented architecture (SOA). This is achieved by using WSDL (Web Service Description Language), which can be converted to COM and various Web service protocols. This makes OPC UA independent from platforms and operating systems and, compliable with the internet, which extends OPC's 'reach' enormously. A three-layer model consisting of 'user level' (user authentification), 'application level' (exchange of digitally signed certificates) and 'transport level' (optional encryption of messages) helps to enhance security tremendously. In addition, there is an information model defining the structure in which the data is held on the server or how data and information is managed in the address space of the server. The information model does not only consist of a hierarchy of 'folders', 'items' and 'properties', as was the case with OPC. Instead, it is a full-mesh network made up of nodes allowing all kinds of meta and diagnosis information to be transmitted. OPC UA has been defined in the basic specifications listed in the references below ([7] to [18]).

In the automation industry, OPC has established itself as an industry standard in the last few years and provides a sophisticated, innovative infrastructure of the underlying information and data model by presenting the specification of the UA and the associated software development kit (hereinafter abbreviated as SDK). The SDK contains the objects and services described in the specification. It is meant to support developers of OPC UA applications.

The infrastructure of OPC UA unifies all previous OPC-based technologies under a 'platform-independent umbrella'. OPC UA provides mechanisms for the standardized, asynchronous, distributed communication.

Owing to the multitude of options OPC UA offers, it was chosen as the standard on which the communication within IITB's framework should be based.

3.1 Address space of the OPC UA server

In the following section, the address space model of the central server will be presented.

To model the server address space, the standard UA notation described in figure 6 (presented in [19]) was chosen because it had proven to be very suitable for mapping the information very easily. The boiler example, both presented in [19] and implemented in the UA SDK for .NET, was used for demonstration purposes.

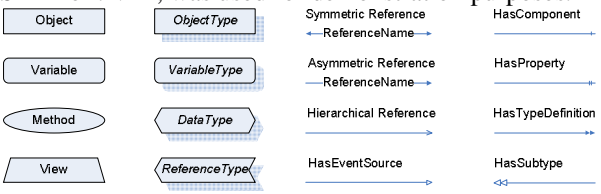

Figure 5. OPC UA notation for modeling the address space [19]

The idea behind the architecture is based on a central OPC UA server and its address space, on which the individual components of the production monitoring and control system, the associated engineering and the process visualization operate as OPC UA clients. Owing

to the new UA concepts, the clients (i. e. the individual components) are able to integrate further information in the address space, i.e. to create new nodes. In this process, the components only monitor those sections of the address space for changes that are relevant to them (see figure 7). They are only notified by the server if information is received that is relevant to them.

Hence, the OPC UA server supports the synchronization of processes in the automated engineering of production monitoring and control systems.

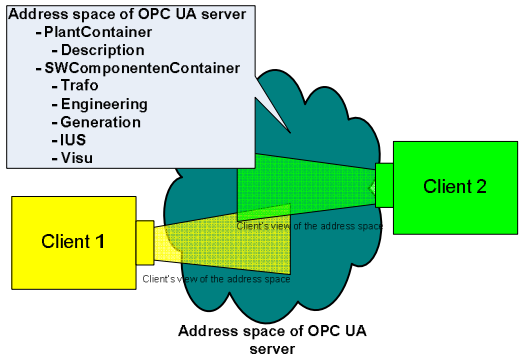

Figure 6. Extract from the architectural concept

Figure 8 shows the initial state of the modeled address space, which is created when the OPC UA server is started.

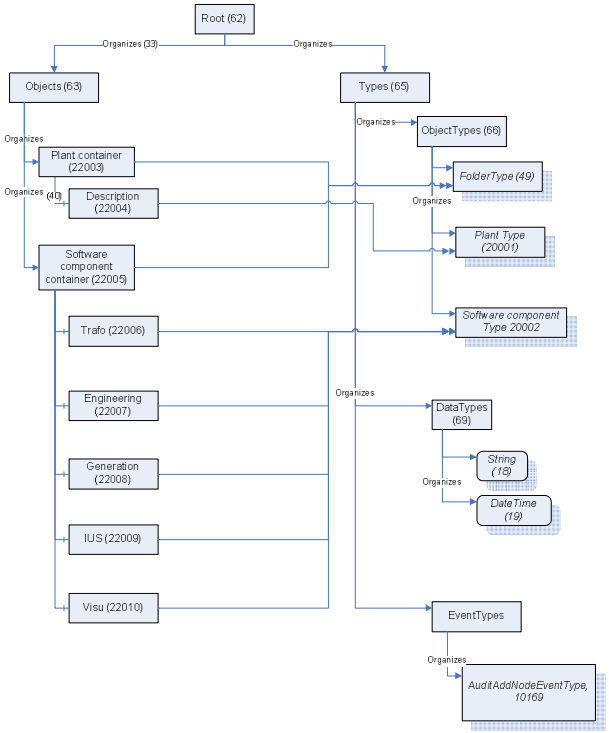

Figure 7. Initial state of the OPC UA address space model

The round brackets behind the name of each element indicate the associated defined node IDs within the address space. The supraordinate structure, consisting of the 'root' node and the objects 'objects' and 'types' (also of the type 'folder type') are already predefined.

Appendix A of [9] provides helpful information about address space modeling. It says: "Objects are used to represent systems, system components, real-world objects, and software objects."[9] "Objects are used to group Variables and other Objects in the AddressSpace." [9] The type-instance concept of the address space was specified on the basis of those two statements.

In order to group the address space, there is a 'plant container' and a 'software container' object. They own a 'HasTypeDefinition reference', which links it to the standard type namely 'folder type'.

Beneath the 'plant container', there is the 'description' object. It represents the real-world components in a CAEX file describing the plant. For this reason, it owns a 'HasTypeDefinition reference' to the self-defined 'PlantType'. If a new CAEX-based description of a plant is added, the appropriate OPC UA client will instruct the server to add a new node to this object. The client responsible for transforming the CAEX files subscribes, for instance, to the events of the description object and will be notified by an 'AuditAddNodesEvent' containing the necessary information. This mechanism is identical for all other clients.

The 'software component container' contains the software components required for the process. They represent software objects, which is why they have been modeled as objects in the address space. All of them own a 'HasTypeDefinition reference' to the SoftwareComponentType'.

To the 'trafo' object, a node is added when the PLC-CAEX file has been transformed into the Proj-CAEX and the Visu-CAEX files. A new node is added to 'engineering' once the contents of the Proj-CAEX file have been imported in the engineering database, and 'generation' receives a new child node once the contents of the engineering database have been generated and transmitted to IUS (ProVis.Agent® runtime components). A new node is inserted into the hierarchy of 'IUS' after IUS has been restarted, and 'Visu' gets a new child node when the process visualization has been generated.

3.2 OPC UA communication within the framework

To make the procedure somewhat clearer, the communication on which this approach is based will be described in more detail. First of all, the individual clients register for the relevant sections of the address space. This is marked by '1' in figure 9.

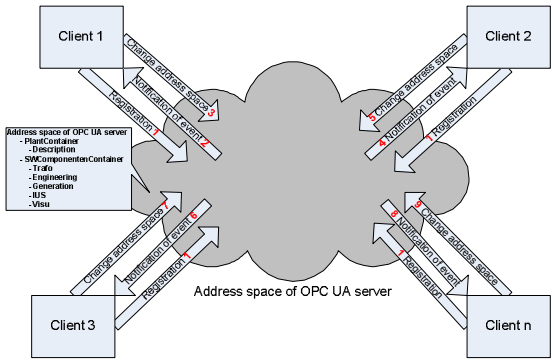

Figure 8. Interaction between clients and server

Once a client has completed its task, it adds a new node to the appropriate object in the address space (figure 9, step '3'). This node contains the necessary information. The client in charge of creating the CAEX file has fulfilled its task as soon as the CAEX description has been completed. Therefore, it adds a new node to the description object containing the storage location of the file and the date of creation. So the transformation client monitoring this object for changes is notified (figure 9: '2') and can get the appropriate information from the server. If this client, in turn, has fulfilled its job, i. e. transforming the file into an engineering- and a visualization-relevant part, it adds a node to the trafo object. This is repeated until all components have completed their processing tasks and a new plant component has been customized and the visualization been generated

The procedure and the services applied in one cycle will be illustrated by UML sequence diagrams. The method names used in these diagrams refer to the terminology of the OPC UA services (to be verified in [10]). Figure 10 provides an overview of the overall procedure.

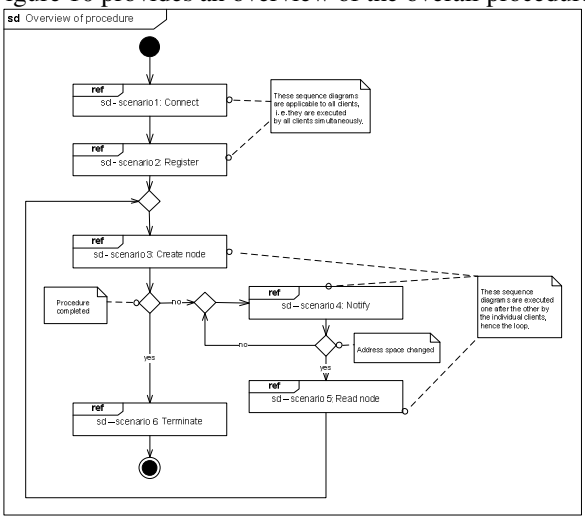

Figure 9. Sequence diagram for the procedure

It includes references to the individual scenarios. Once the clients have been connected to the server, the objects to be monitored are registered. The first client creates a new node when the PLC CAEX description is available. The creation of the new node triggers a notification to all monitoring clients, which export the newly created node, perform their tasks and create a new node in turn. This is repeated until the last client, in this case the one responsible for generating the visualization, has finished its job and created a new node in the address space. If the automated engineering is to be terminated, the clients disconnect from the server and the server can be terminated.

Scenario 1 (figure 11), which establishes the connection between the client and the server, will now serve as an example that is described in more detail. To do this, the 'DiscoveryServiceSet' finds 'endpoints', which provide services. When the system was developed, the services of a Discovery Server had not been available yet. However, the features of that ServiceSet were included to complete the picture. The implementation, however, is based on the assumption that the client knows which endpoint to address. If the endpoint is known, a secure channel to the client is established by means of the SecureChannelServiceSet, and the SessionServiceSet allows creating a session between client and server.

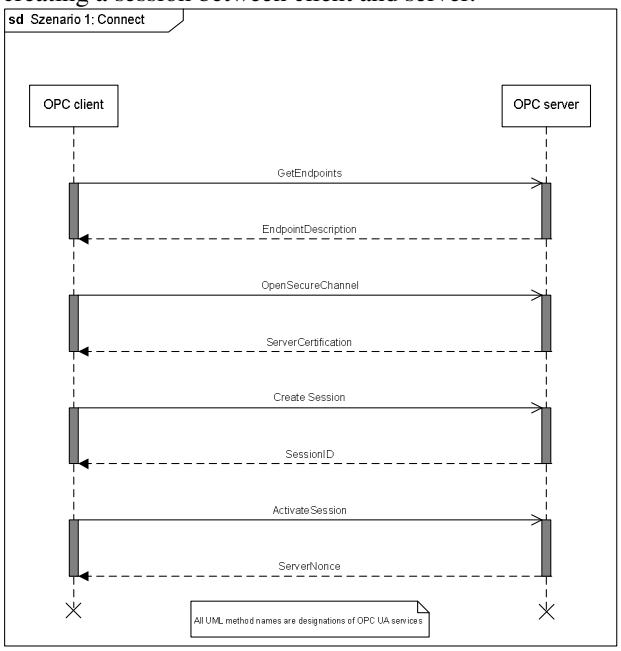

Figure 10. Sequence diagram for scenario 1: Connect

Consistently, the appropriate services are used in each scenario. In addition, the formal specification of the applied services can be looked up in the OPC UA specifications (see [7] to [18]).

4 Implementation

4.1 Components

The part of the solution concept that refers to communication was developed on the basis of the 'OPC UA SDK for .NET' (version 1.00.115). It includes the services of the Unified Architecture. These services are encapsulated into methods by the SDK, which makes them easy to use. In addition, the SDK includes a combined SampleServer and SampleClient, which uses some functions of the Unified Architecture by way of example, such as read and write access to attribute values.

IITB's application was developed on the basis of the combined server and client. To this end, client and server were separated. Currently, the application consists of one server (see figure 12) and four clients (figure 13). Owing to the performance of the sample computer, the number of clients was limited. However, it can be increased at will. It is also possible to run the entire procedure with just one client. The interfaces were adjusted to allow for easy visualization. A few features were added.

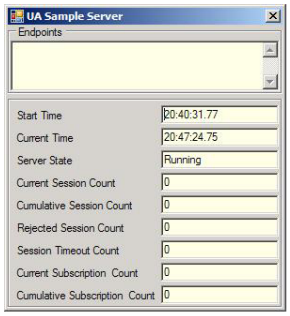

Figure 11. User interface of OPC UA sample server

In the sample client, parts of the client are linked to the user interface and implemented there. For this reason, the tasks to be performed by the appropriate client, e. g. an IUS restart, are processed in separated threats so the functionality of the interface is not affected.

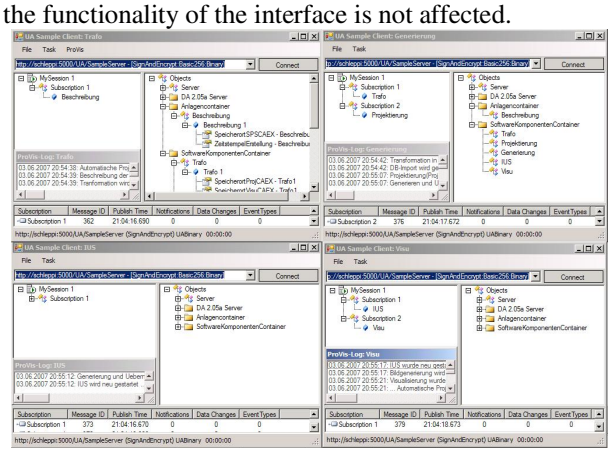

Figure 12. OPC UA sample clients

At the beginning of the automated engineering, the configurator developed by IITB (see figure 14) allows to choose whether the change management should be started in a central and local or a distributed way. A distributed start can be used, for instance, if the OPC UA server runs on a computer different from the clients.

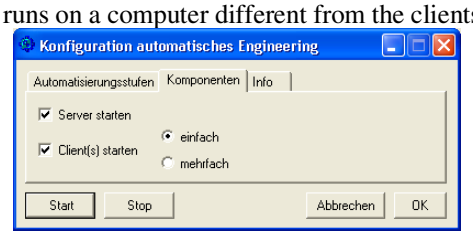

Figure 13. Configurator for automated engineering

Figure 15 shows one of the clients. In the address line on top, users can select an endpoint to which they want to connect. This is confirmed by clicking the Connect button. After the connection with the server has been established successfully, the window on the right side shows the objects and their structure in the address space. In the left window, the subscriptions assigned to the session are depicted as well as the associated MonitoredItems. Even if there is no event for this subscription, keep-alive-messages are exchanged between client and server at regular intervals. They can be seen in the bottom window. The red circle marks the column which shows whether there is a notification for a subscription.

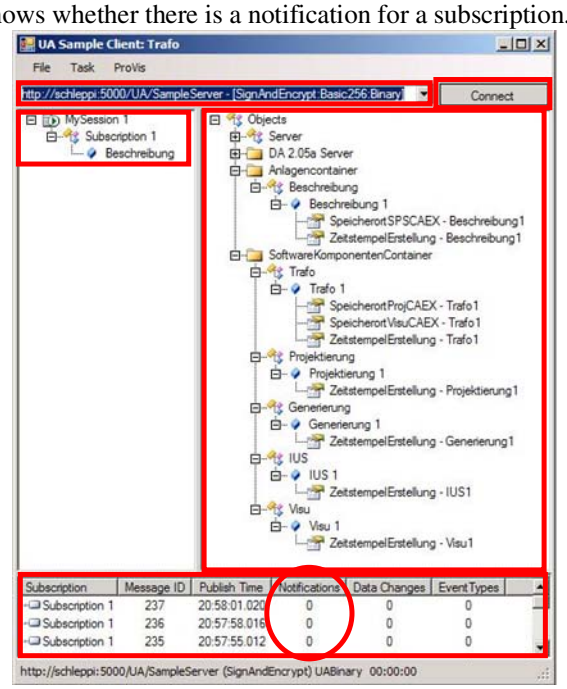

Figure 14. OPC UA client and address space

The automated engineering is started by selecting the appropriate menu item ('Automatische Projektierung', see figure 16) or by using the Web service interface. The figure also shows the address space at the beginning of the procedure. Subsequently, a new node is added to 'Beschreibung' (description) to signal that the description

of a new plant has been received. The node contains information about the storage location of the file. This process is supported by the AddNodes service call.

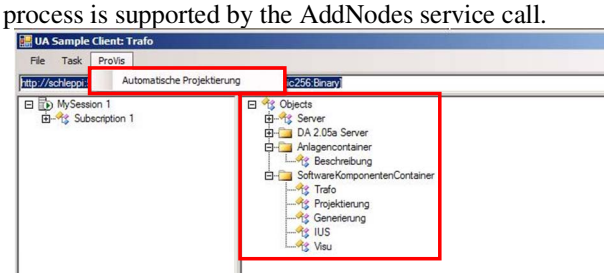

Figure 15. Invocation of the automated engineering in the OPC UA client and address space at the start of the application

4.2 Different change scenarios to be supported by the automated engineering

During engineering or reengineering, different sce nar ios have to be taken in account:

- New engineering
- Adding (several) plant(s)
- Removing (several) plant(s)
- Changing existing plant(s)

These options can be set respectively communicated via the Web Service interface integrated in one of the OPC-UA clients.

The first case is a totally new engineering. In this case, all previous engineering information is discarded.

One or several new plants can be added as well. In this case, existing plants will not be affected and go on working. Therefore, it is essential that the reengineering takes place at an appropriate time.

To turn off working plants or remove them from the production process, there should be the possibility to delete them.

The last case consists of modifications within an operating plant, for example if several parameters or even parts of the plant changed. This case should be realized with an online data management in OPC UA, where its real advantages take effect. It is already scheduled, but not yet implemented. The need to use simulation or engineering data parallel to the production process has also been recognized by [20].

4.3 Modification of address space

When a CAEX-based engineering operation has been completed, various new nodes have been created in the address space of the server. The clients communicate about the contents of these nodes. The OPC UA server receives information about the location of the corresponding CAEX file, for example, and the server, in turn, can convey this information to other clients. Figure 17 shows what the address space looks like.

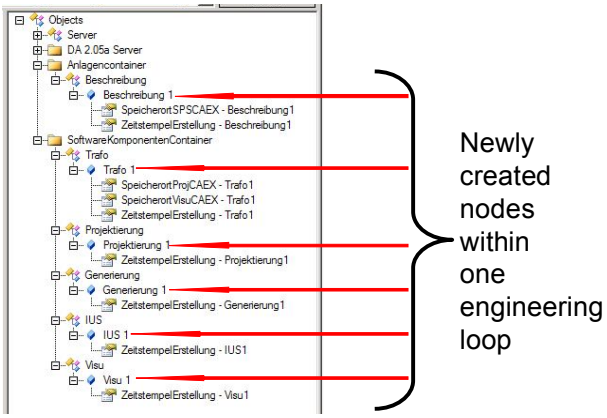

Figure 16. Address space of the server following the engineering of a plant

The nodes marked by red arrows have been newly created by the individual clients during the operation. If another plant is added, a new node is added into the hierarchy of the 'Beschreibung' (description) object which is called 'Beschreibung 2'. For a third plant, it would be called 'Beschreibung 3'. Consistently, new nodes are added to the other objects while a new plant is engineered.

4.4 Result

The visible result of the engineering of production monitoring and control systems is the visualization that has been generated and linked to the process. Figure 18 shows a process control image generated on the basis of a CAEX description using the aforementioned mechanisms.

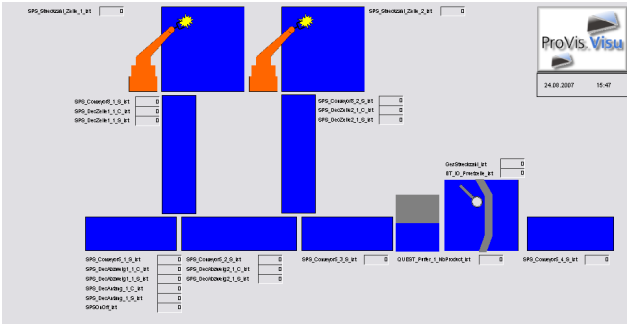

Figure 17. Generated process control image to visualize an application example (defined in cooperation with Daimler's Research Center Ulm)

The associated engineered production monitoring and control system information can be viewed using the current graphical engineering tool of ProVis.Agent®. This data is depicted in figure 19.

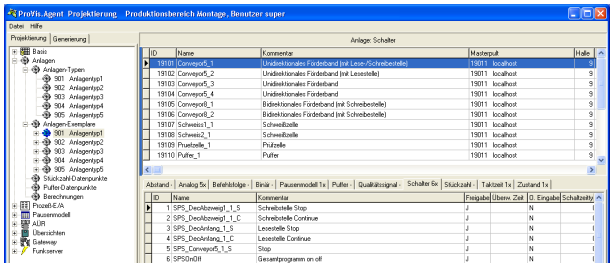

Figure 18. Depiction of engineered plant samples of the application example in today's engineering interface

5 Summary and outlook

By combining OPC UA and CAEX, a framework has been created fulfilling tasks that range from obtaining the file describing the plant to visualization and engineering. To this end, first implementations of the OPC Unified Architecture were performed, and the methods and opportunities it offers were tested. At present, CAEX is used as data format for the exchange of engineering information. It is checked for formal correctness and processed within the individual components. UA is currently used for communication, processing, synchronization of the procedure, and the framework architecture.

As soon as the OPC UA SDK for .NET is released in a final version, more information will be integrated in the address space of the UA server to tie both technologies closer together. This includes the engineering data itself, for instance, which will then be available online, thus allowing for even quicker re-engineering. Therefore, the production monitoring and control system has to operate on these data and to support an online engineering. Even though, security, redundancy, consistency and availability should be ensured. The engineering data should then be integrated in the CAEX format into the OPC UA server address space which makes a full OPC UA-CAEX type model necessary

References

[1] W. Schreiber, "Die Top-Themen der deutschen Automobil-Industrie", in: Tagungsband zur MANUFUTURE Germany Konferenz: die strategische Forschungsagenda Deutschland, pp. 90-94, 12 September 2007.

[2] "IEC/PAS 62424 - Ed. 1.0: Specification for Representation of process control engineering requests in P&I Diagrams and for data exchange between P&ID tools and PCE-CAE tools", June 2006.

[3] M. Schleipen, R. Drath, O. Sauer, "The system-independent data exchange format CAEX for supporting an automatic configuration of a production monitoring and control system", IEEE International Symposium on Industrial Electronics – ISIE 2008, pp. 1786-1791, Cambridge, June 30 - July 2, 2008.

[4] OPC Foundation, "UA Specification", http://www.opcfoundation.org, March 2008.

[5] "ProduFlexil", http://www.produflexil.de/, April 2008.

[6] F. Alsmeyer, "Durchgängige Nutzung von Prozessdaten im Lebenszyklus verfahrenstechnischer Anlagen", VDI-Berichte no. 1980, 2007.

[7] OPC Foundation, "OPC UA Part 1 - Concepts 1.00 Specification", http://www.opcfoundation.org, July 2006.

[8] OPC Foundation, "OPC UA Part 2 - Security Model 1.00 Specification", http://www.opcfoundation.org, July 2006.

[9] OPC Foundation, "OPC UA Part 3 - Address Space Model 1.00 Specification", http://www.opcfoundation.org, July 2006.

[10] OPC Foundation, "OPC UA Part 4 - Services DRAFT 1.01.05 Specification", http://www.opcfoundation.org, February 2007.

[11] OPC Foundation, "OPC UA Part 5 - Information Model 1.00 Specification", http://www.opcfoundation.org, July 2006.

[12] OPC Foundation, "OPC UA Part 6 - Mappings RC0.93 Specification", http://www.opcfoundation.org, June 2006.

[13] OPC Foundation, "OPC UA Part 7 - Profiles Draft0.23 Specification", http://www.opcfoundation.org, July 2006.

[14] OPC Foundation, "OPC UA Part 8 - Data Access 1.00 Specification" http://www.opcfoundation.org, September 2006.

[15] OPC Foundation, "OPC UA Part 9 - Alarms Draft 0.62 Specification", http://www.opcfoundation.org, April 2006.

[16] OPC Foundation, "OPC UA Part 10 - Programs 1.00 Specification", http://www.opcfoundation.org, January 2007.

[17] OPC Foundation, "OPC UA Part 11 - Historical Access 1.00 Specification", http://www.opcfoundation.org, January 2007.

[18] OPC Foundation, "State Machine Appendix to OPC UA Part 5 RC1.70 Specification", http://www.opcfoundation.org, December 2006.

[19] T. J. Burke, "OPC Foundation – OPC DevCon", OPC DevCon, 10-12 October 2006, Munich, 2006.

[20] S. Kain; F. Schiller; S. Dominka, "Reuse of Models in the Lifecycle of Production Plants, Using HiL Simulation Models for Diagnosis", IEEE International Symposium on Industrial Electronics – ISIE 2008, Paper CF-015261, Cambridge, June 30 - July 2, 2008.