Scalability of OPC-UA Down to the Chip Level Enables "Internet of Things"

Jahanzaib Imtiaz[1] and Jürgen Jasperneite[2]

[1]inIT - Institute Industrial IT, D-32657 Lemgo, Germany
[1]jahanzaib.imtiaz@hs-owl.de
[2]Fraunhofer Application Center for Industrial Automation (IOSB-INA), D-32657 Lemgo, Germany
[2]juergen.jasperneite@iosb-ina.fraunhofer.de

***Abstract*—Many implementations of the OPC-UA are increasingly available. Until now, use of technologies like the OPC-UA was limited to embedded systems with a sufficient amount of memory resources. OPC-UA offers a scalable feature set that could be tailored according to application demands. We believe that a key to successfully enable Internet of Things for the industrial automation applications lies in bringing down the OPC-UA into low end resource-limited devices, such as sensors. Because the OPC-UA provides a powerful information representation that is exchangeable through interoperable services. This paper has investigated the OPC-UA as a middleware solution for such resource-limited devices. For a proof of concept we have implemented an OPC-UA server based on the "Nano Embedded Device Server profiles" of the OPC Foundation.**

***Index Terms*—OPC-UA and Internet of Things, Industry 4.0**

I. Introduction

Normally industrial automation systems are designed against fixed application requirements, which makes them less flexible for adopting changes. Due to shorter product lifecycles and changing market conditions, the current production systems must become more flexible [1], [2]. Industry 4.0, a term shaped by representatives of German industry leaders and researchers describes how the Internet of Things (IoT) will improve the industry's engineering, production, logistic and life cycle management processes. The number refers to the 4th industrial revolution. Starting in the 18th century, 3 major waves of technical changes modified the industrial landscape and increased the productivity. The IoT allows for a new way of organizing the industrial production: by connecting machines, warehousing systems, and goods, we can create smart production systems that basically control each other without requiring any manual intervention [3], [4].

In order to have such a plug and produce capability, adaptations to industrial devices at multiple levels are required and various technological challenges have to be addressed. Some of such challenges are described in Section II. While, this paper investigates the OPC Unified Architecture (OPC-UA) for integration of smallest devices in the IoT. This is to enable a vertical integration and a consistence view of a network's services to an application programmer [5], [6]. As we are looking for a seamless connectivity and an auto-configuration, the OPC-UA is a promising candidate for it. The problem with the current OPC-UA implementations is that they are overwhelmed with a lot of features that makes them bulky. Some implementations even take 10 MB of memory [7]. This is because the memory resources are directly proportional to the complexity and density of an application specific information model, and to a number of implemented OPC-UA features. However, a stack coming from the OPC Foundation can be reduced down to a merely 200 kilo bytes after optimization. Devices in IoT could be as small as having only a few kilobytes of memory available for an application. This makes an off-the-shelf implementation not suitable for a resource limited device with a basic functionality, which constitutes the research question for this paper: *"Is it possible to scale down OPC-UA to a chip level?"*

This work describes a possible scaling down of an OPC-UA server based on the "Nano Embedded Device Server Profile" [6] as defined by the OPC foundation. Followed by a prototype implementation for a simple sensor application that is based on a single chip with limited memory resources, furthermore the device is also integrated with a communication interface. It demonstrates the vertical integration of an industrial process application running at a resource limited device, to a commercial off the shelf OPC-UA client running at a higher end smart phone. This is while simultaneously communicating in real-time with a controller.

The paper is organized such as: Chapter II describes the background and the motivation for this paper. Chapter III provides an overview of the OPC-UA standard and describes its scalability aspects. This is followed by a chapter IV that presents a case study based on an industrial scenario as a proof of concept. At the end some conclusions are drawn and a possible future work is discussed in chapter V.

II. Background

The industry 4.0 provides a vision for an application of Cyber Physical Systems in the industrial automation [8]. The key requirements for the factory of the future as described in [9] includes:

- Smart objects that are technical devices with a decentralized intelligence.
- Universal networking that insures a communication ability of each of these smart objects in one network.

978-1-4799-0752-6/13/$31.00 ©2013 IEEE

- Convertible and agile production systems which are an aggregation of smart objects to almost a self-configuring production system.
- The use of Internet Standards by an adoption of existing and proven standards for a cable-based and/or a wireless communication (e.g. TCP/IP, Ethernet, Bluetooth, WiFi etc.).
- Vertical integration in the Network by a replacement of a strictly horizontal and a hierarchical control architecture, by a stronger vertical integration and a consistency of network structures.
- Changed role of man through providing a stronger support to the user by an improved and a mobile access to production data and plants, e.g., via Tablet-PCs and Smartphones linked with a user-centric and a context-adaptive interaction design.

An important aspect of "Industry 4.0" paradigm is a decentralization of decision making, meaning that employees, machines and products communicate with each other in the so called "Internet of Things". One of the arguments for an Internet of Things is: allowing distributed yet interlinked devices, machines, and objects to interact with each other without relying on human intervention to set-up and commission the embedded intelligence [10]. To reach these ambitious objective not only the miniaturization of microelectronics is important, appropriate communication interfaces are also needed. While the network technologies such as Ethernet, Wireless or Bluetooth are well investigated and are wide spread, the software integration is still expensive and complex due to a variety of interfaces. Beside other technologies [11], [12], [13], the OPC-UA is an approach, which abstracts from the network technology and software application, and offers a generic communication interface. It has been seen as one of the key technology for a transparent data representation/transmission between heterogeneous system components. This paper presents a prospective of the OPC-UA as an enabler for the Internet of Things.

III. OPC Unified Architecture

The OPC-UA is a platform-independent industrial standard through which various kinds of systems and devices can communicate by sending messages between clients and servers over various types of networks. It is fundamentally about data modelling and transport. It uses object-oriented techniques, including type hierarchies and inheritance, to model information [14], [15]. An OPC-UA address space allows information to be connected in various ways. The base OPC-UA specifications provide only the infrastructure to model an information, and encourage additional, industry specific information model specifications to be defined by vendors and standards organizations. The OPC-UA can be mapped onto a variety of communication protocols and data can be encoded in various ways to trade off portability and efficiency. Furthermore, the OPC-UA systems architecture models the OPC-UA clients and servers as interacting partners. Each

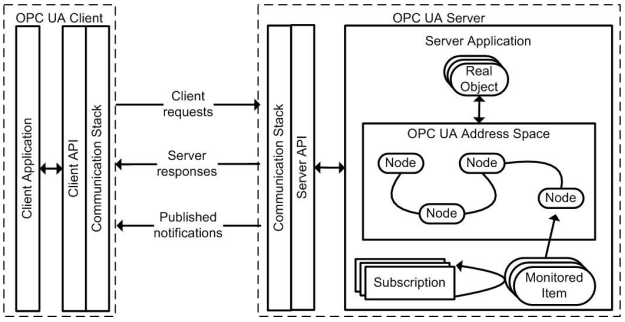

Fig. 1. OPC Unified Architecture [6].

system may contain multiple clients and servers. Each client may interact concurrently with one or more servers, and each server may interact concurrently with one or more clients [12]. An application may combine server and client components to allow interaction with other servers and clients at various levels in automation hierarchy for seamless vertical integration.

Figure 1 illustrates major elements of a typical OPC-UA client/server architecture and describes how they relate to each other. An OPC-UA client/server application uses an "API" to exchange the OPC-UA service requests and responses. The "API" provides an internal interface that isolates the application code from the OPC-UA communication stack. This stack handles transport mechanisms for the OPC-UA specific services/messages. Real objects represent the elements of a process application and are accessible via the OPC-UA means. Nodes in an address space virtually represent those real objects, their types and references to each other. Furthermore, the address space is a hierarchy of nodes accessible by the OPC-UA clients through a multitude of context specific UA services. A client creates server side references that are called "Monitored Items". These items reflect the changes in attributes and behaviour of address space Nodes against their real-world counterparts, when they detect a data change or an event/alarm occurrence, they generate a notification that is published to a client who has subscribed to that item. Furthermore, clients could control the rate at which a publishing should occur.

A. Scalability of OPC-UA

The figure 2 shows an overview of the OPC-UA's complex feature sets, which also formally referred as facets and grouped together to address needs of a various application requirements. As the OPC-UA is designed in a way that individual implementations do not need to support all features, but can be downscaled to a limited scope if desired [16]. A service-based OPC-UA implementation can be tailored to be just as complex as needed for the underlying application. The OPC foundation described various ConformanceUnits/Profiles to describe (and test) which features are supported by an OPC-UA compliant product. An application (client or server) shall implement all mandatory ConformanceUnits in a profile to be compliant with it. Some profiles may contain optional ConformanceUnits which in turn may exist in more than one

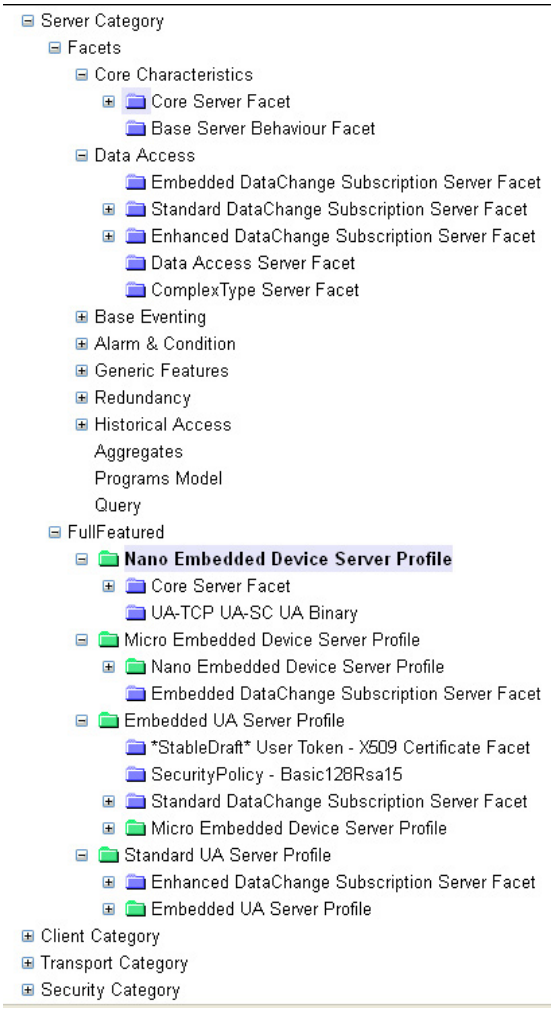

Fig. 2. Snapshot of OPC-UA Features and Server Profiles [6]

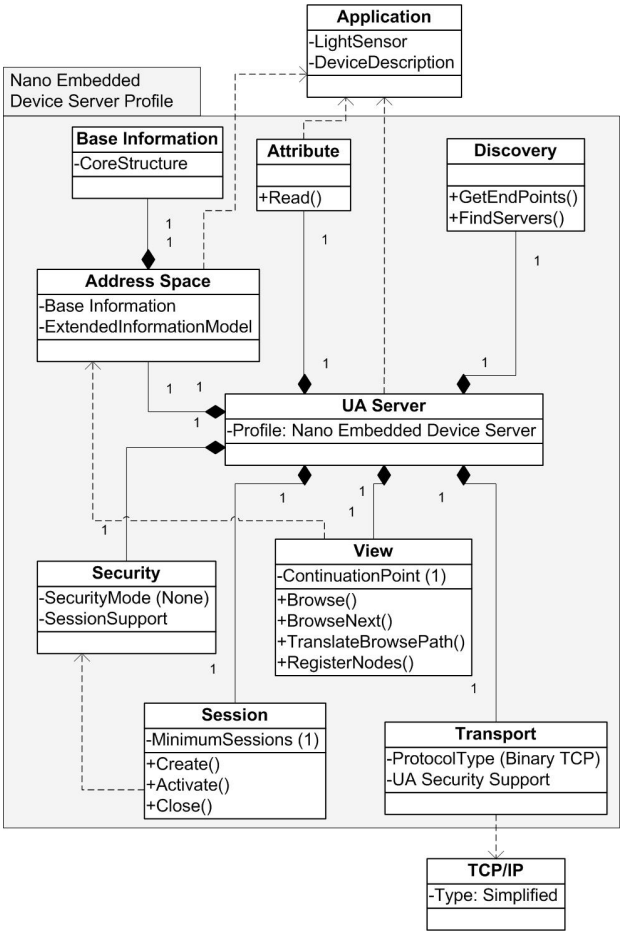

Fig. 3. Core components of Nano Embedded Device Server Profile

profile. Up to now, the OPC Foundation has released more then 60 OPC-UA profiles. However, it is expected that the list will be extended over time even by other organizations than the OPC Foundation [17], [18]. This presents a complex picture of the OPC-UA from a product implementers point of view. Because on one hand there is a wide range of ever increasing (for now) features and on the other hand there are specific application needs of different technical domains, making design decisions difficult for OPC-UA products (e.g. servers). We believe that the OPC-UA applications should be tailored around an application domain. Therefore, in this paper we investigated the bare minimum features of the OPC-UA. We considered a specific industrial application domain that deals with self configurable production systems [19]. Such systems involve specific automation applications, technologies and requirements on data communication within and between these technologies. Knowing which OPC-UA profile is designed to fulfil the given communication requirements, allows to adopt a design strategy for the OPC-UA stack for a particular application. For example, the "Nano Embedded Device Server Profile" can be used for a flexible industrial automation application that intends to enable a plug and work mechanism at a low end field device. Since the existing OPC-UA SDKs are very generic and provide extensive libraries to accommodate different kind of applications, therefore they are bulky and not suitable for devices with limited ROM/RAM. After analysis of the OPC Foundation's software stack, we have identified mandatory design components of this profile and implemented a corresponding software stack completely from scratch. This new implementation provides a reduced complexity and a downscaled footprint. The following section explains core components of the profile.

B. *Nano Embedded Device Server Profile*

It is a full featured Profile that is intended for chip level devices with limited resources. This Profile is functionally equivalent to a core server facet and it defines an OPC-UA TCP binary protocol as a required transport profile facet [16]. Furthermore, it describes standard compliant bare minimal OPC-UA server functionality. Figure 3 describes core components of this profile, including:

- An *Address Space* component that handles an application specific description of information model , which is exposed to the external world.

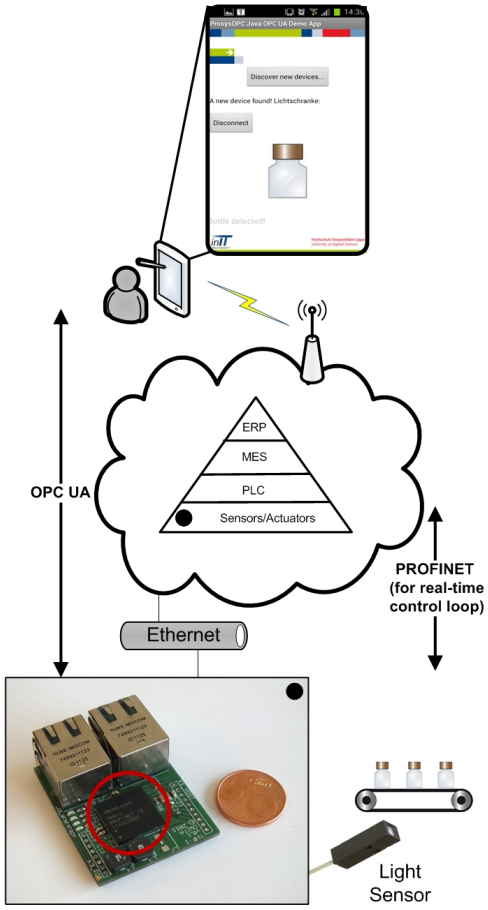

Fig. 4. Case Study: Scenario for Internet of Things in Industry 4.0

- The *Security* component that handles security related topics, this profile does not provide stringent security features but it has to comply with a security profile "None" as defined by the OPC foundation.
- A *Session* component that handles an establishment of a secure session instance between a client and the server, this profile supports one session at any given time.
- A *Transport* component that describes/handles a binary transport mechanism on top of TCP/IP.
- A *Discovery* component that handles a discovery service as described by the OPC foundation, this profile does not support a registration to an external discovery server, however it answers self-discovery queries from external clients.
- A *View* component that handles browsing of an application specific address space and traversing through the information model.
- An *Attribute* component that allows reading and/or writing of specific attributes to an object in the address space.

In the following section we have identified an use case of the IoT in an industrial automation domain. In order to enable the IoT concept in a resource limited device, we needed a middleware service interface that not only can be fitted into such resources, but also provide interoperability by means

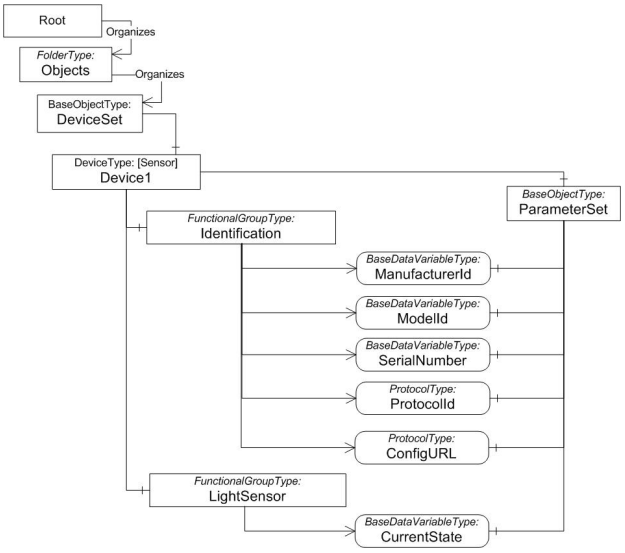

Fig. 5. OPC-UA Specific Information Model for Light Sensor Process Application.

of information modelling. For a proof of concept, we have implemented a Nano OPC-UA server by tailoring the OPC-UA according to a given application.

IV. Case Study

An application scenario used for this case study consists of a simple process model that is driven from the Lemgo Smart Factory (LMF) [20]. It includes a conveyor belt that carries some bottles and moves around a pick and place robot. The robot picks a bottle from a predefined position (e.g: x,y) on the belt and passes it to another module for further processing. In order to identify that if a bottle has reached the target position, a field device that is equipped with a light sensor is attached to the position "x,y". A controller communicates with the device over a real-time channel and uses the sensor's sate to determine the presence of a bottle. Later on, the controller instructs the robot to move its gripper to that position to pick the bottle for further processing.

The field device uses two different communication channels to exchange data with other devices. While, the standard TCP/IP channel is used for parameterization and/or a configuration data, the real-time channel is used for a standard cyclic data transfer. Real-time communications bypass the standard TCP/IP interface to expedite the time-critical data exchange.

Over time the field device needs replacement for maintenance purposes. In state of the art automation applications, such a replacement requires a manual engineering effort at various levels. The operation of a whole process has to be suspended, the new device has to be configured before making the process operational again. Such a configuration included an IP address assignment, a device description and a parameterization for an industrial communication protocol. In principle it could be any industrial protocol, for this case study we have adopted the PROFINET to realize such a communication. For the given scenario a field device is extended with many

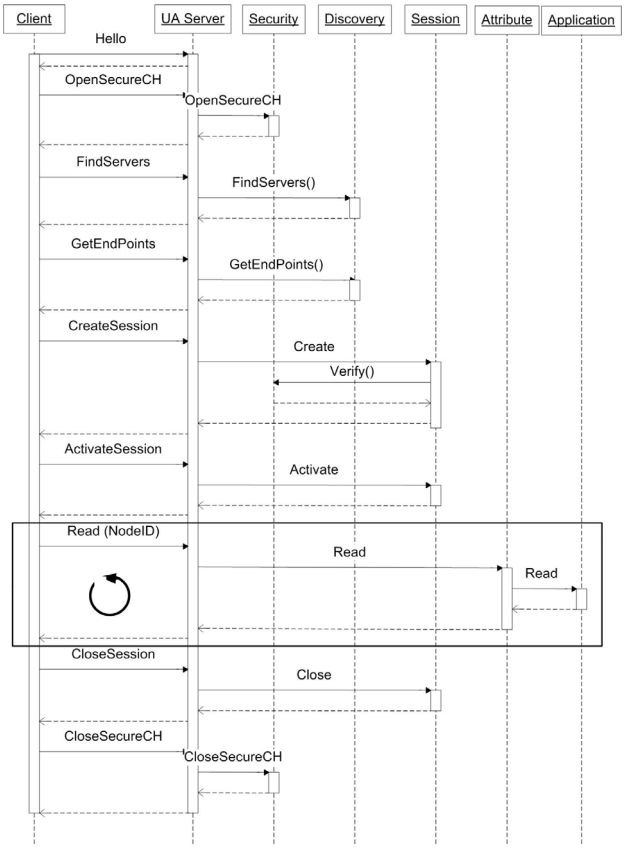

Fig. 6. OPC-UA Message Exchanges between the Client and the Server.

technologies, to support dynamic configuration and operations, and to become part of the operational system with least manual efforts and a production disruption. Beside a real-time communication channel for a control loop, an OPC-UA based non real-time channel has been used to integrate the device with a high level infrastructure for a vertical integration (e.g. MES, ERP, Internet), this is for a monitoring and/or a diagnostic application.

Furthermore, the field device has very small memory and processing resources. It is based on the single chip solution TPS-1 (ARM9@100MHz) with less then 64 KB of the available memory [21]. It provides two Ethernet interfaces and all PROFINET related protocol handling is implemented in the hardware. The corresponding software is based on the real-time OS "embOS". Figure 5 describes a simple information model implemented for the light sensor application in this device. Beside a general device description, it includes a configuration URL object that has a predefined reference to a GSDML file, a UA client at the controller uses the URL to retrieve a complex device description, and later on use it to configure the PROFINET functionality for the device. In [14] we have described a way to use such a information model for the device auto-configuration. The address space also includes a $LightSensor$ object together with its current state, which is used by an external OPC-UA client (a smart phone in this case as described in figure 4).

The figure 6 describes interactions of an android based

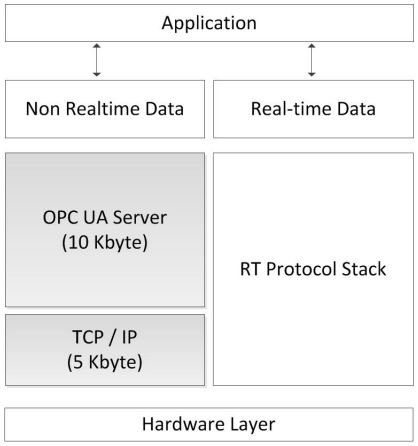

Fig. 7. Implementation architecture of OPC-UA at TPS-1 evaluation platform.

OPC-UA client hosted on a smart phone with different components of our implemented Nano OPC-UA server at the given device. All messages are request/response pairs. Once the OPC-UA client at the controller is aware of the presence of a new device (via OPC-UA discovery or as in this case study via PROFINET's DCP), it directly connects to the relevant OPC-UA server at this device using the discovered server's credentials. After opening up a secure channel, the client has to first verify the latest service definitions through *GetEndpoints*. A session has to be created via *CreateSession*, that exchanges some security credentials. Once the session is activated using *ActivateSession*, the *Read* service can be invoked to actually learn a multitude of objects from the address space that include device description and/or the sensor's states etc. There are two possibilities to communicate to a particular object in the address space. One is by browsing the information model an OPC-UA client will identify a NodeID of a required object and using it the client reads the Node's attributes. Secondly, the NodeID is well defined and previously known to the client, this can reduce the browsing overhead. We have opted for the second option for this case study. Once connected and the session has established, a client can call a read service indefinite number of times. After read activity has finished, the session must be closed, as well as the secure channel.

For implementation of the concept, we have extended the TPS-1 firmware by introducing a tailored OPC-UA stack to transport the messages in binary encoding on top of a micro TCP/IP stack. Figure 7 shows the abstract architecture which includes two distinct function blocks: one is the PROFINET specific stack controlling a real-time communication on top of the TPS-1 hardware Layer. Other is an OPC-UA function block on top of the same hardware for best effort traffic. For the OPC-UA server function together with a micro TCP/IP stack, only 15 KB of memory footprint (ROM size) is required to implement a basic application. The protocol stack is implemented in ANSI C and consists of about 2000 lines of code. For the verification of an interoperability the implemented OPC-UA server is tested with the standard OPC-UA client coming from the OPC Foundation, the UaExpert OPC-UA

client coming from Unified Automation and the Java based client for Android coming from Prosys.

The given application in this case-study is very simple and does not involve a lot of dynamic application data. Since the device description is statically defined, the only dynamic attribute is a boolean, i.e light sensor, value that could be represented by one bit (on/off), which is realized as a global variable. This leads to the very low RAM requirements and instead of application data the RAM is mainly consumed by the OPC-UA protocol overhead (an average frame size for OPC-UA read response is about 200 Bytes). Also the RAM size is directly proportional to the complexity of the application, data type and a number of dynamic objects.

Being a research project the aim of this study was not to develop an optimum quality software (in comparison with existing SDKs and passing the certification etc.), but to demonstrate that scalability of OPC-UA enables and contributes towards the internet of things, which is a foundation for the industry 4.0. Furthermore, an OPC-UA client would need to adopt its interactions with the implemented server, as it can only access a limited functionality that is offered by this server. OPC-Foundation may define a certification process for compatibility with the "Nano Embedded Device Server profile" that could optimize such interactions.

V. Conclusion

The work presented in this paper has demonstrated that the OPC-UA can scale down to a chip level while it still retains its prominent features to remain useful. The implementation uses the binary transport profile. For the server functionality with a basic set of services (scale down to allow browsing the address space and reading the application data) only 10 KB of the memory space is needed (excluding the TCP/IP). To our knowledge, this is one of the smallest OPC-UA servers [22] that demonstrates the high scalability of the OPC-UA. Also it makes software integration easy for simplest devices in the Internet of Things. This also shows that a hybrid approach is possible such as, besides a real-time channel for the control loop, an OPC UA based best effort channel could be used for a seamless communication from a sensor to the Internet. This is without any additional engineering efforts at different levels of the automation pyramid. An interesting future work could be to investigate porting such a Nano OPC-UA server, to devices with different physical transport technologies (e.g. Bluetooth, CAN bus).

Acknowledgment

This work was partly funded by the EU FP7 STREP project IoT@Work under the grant number ICT-257367 as well as the German Federal Ministry of Education and Research (BMBF) within the CAPRI project (FKZ: 17N0609) and the Leading-Edge Cluster "Intelligent Technical Systems OstWestfalen-Lippe" (it's OWL).

References

[1] Jahanzaib Imtiaz and Jürgen Jasperneite. Common Automation Protocol Architecture and Real-time Interface (CAPRI). In W.A. Halang, editor, *Echtzeit 2012: „Kommunikation unter Echtzeitbedingungen", Boppard*. Springer-Verlag,, Nov 2012.

[2] Jürgen Jasperneite, Jahanzaib Imtiaz, Markus Schumacher, and Karl Weber. A Proposal for a Generic Real-time Ethernet System. *IEEE Transactions on Industrial Informatics*, 5(2), May 2009.

[3] Stefan Ferber. *Industry 4.0 – Germany takes first steps toward the next industrial revolution*. http://blog.bosch-si.com/.

[4] S. Hodek and J. Schlick. Ad hoc field device integration using device profiles, concepts for automated configuration and web service technologies: Plug amp;play field device integration concepts for industrial production processes. In *Systems, Signals and Devices (SSD), 2012 9th International Multi-Conference on*, pages 1 –6, march 2012.

[5] U. Enste and W. Mahnke. OPC Unified Architecture. *Automatisierungstechnik*, 59(7):397–404, 2011.

[6] OPC Foundation. *OPC Unified Architecture*. http://www.opcfoundation.org/.

[7] Unified Automation GmbH. *OPC UA Demo Server for Windows*. http://www.unified-automation.com/opc-ua-servers/.

[8] J. Schlick. Cyber-physical systems in factory automation - Towards the 4th industrial revolution. In *Factory Communication Systems (WFCS), 2012 9th IEEE International Workshop on*, page 55, may 2012.

[9] Detlef Zühlke, Wolfgang Wahlster and Klaus Mittelbach. *Smart Factory KL*. http://smartfactory.dfki.uni-kl.de/en/content/pressekonferenz2.

[10] Louis Coetzee and Johan Eksteen. The Internet of Things – Promise for the Future? An Introduction. In *IST-Africa 2011 Conference Proceedings*, 2011.

[11] B. Bony, M. Harnischfeger, and F. Jammes. Convergence of OPC UA and DPWS with a cross-domain data model. In *9th IEEE International Conference on Industrial Informatics (INDIN)*, pages 187–192, 2011.

[12] G. Cândido, F. Jammes, J. de Oliveira, and A. Colombo. SOA at Device level in the Industrial domain: Assessment of OPC UA and DPWS specifications. In *8th IEEE International Conference on Industrial Informatics (INDIN)*, pages 598–603, 2010.

[13] G. Candido, A. Colombo, J. Barata, and F. Jammes. Service-Oriented Infrastructure to Support the Deployment of Evolvable Production Systems. *IEEE Transactions on Industrial Informatics*, 7(4):759–767, 2011.

[14] Lars Dürkop, Jahanzaib Imtiaz, Henning Trsek, and Jürgen Jasperneite. Service Oriented Architecture for the Auto-configuration of Real-Time Ethernet Systems. In *3. Annual Colloquium "Communication in Automation (KommA 2012)*, Lemgo, Nov 2012.

[15] Lars Dürkop, Henning Trsek, Jürgen Jasperneite, and Lukasz Wisniewski. Towards Autoconfiguration of Industrial Automation Systems: A Case Study Using PROFINET IO. In *17th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2012), (best paper award)*, Krakau, Poland, Sep 2012.

[16] OPC Foundation. *OPC UA Part 7 - Profiles 1.01 Specification*. http://www.opcfoundation.org/.

[17] D. Grossmann, K. Bender, and B. Danzer. OPC UA based Field Device Integration. In *SICE Annual Conference, 2008*, pages 933 –938, aug. 2008.

[18] Dirk van der Linden, Maarten Reekmans, Wolfgang Kastner, and Herbert Peremans. Towards Agile Role-based Decision Support for OPC UA Profiles. In *ICIW 2012 : The Seventh International Conference on Internet and Web Applications and Services*, 2012.

[19] Hans-Peter Huth Amine Houyou. Internet of things at work European Project: Enabling Plug and Work in Automation Networks. http://www.embedded-world.eu/review-2011.html, February 2011. Embedded World Conference 2011.

[20] inIT - Institute Industrial IT of Hochschule Ostwestfalen-Lippe - University of Applied Sciences. *inIT: Lemgoer Modellfabrik (Lemgo Smart Factory)*. http://www.hs-owl.de/init/research/lemgoer-modellfabrik.html.

[21] Roland Hess, Andreas Steinmetz, Sebastian Schriegel, and Markus Schumacher. Profinet und Power-over-Ethernet: Einfache Vernetzung dezentraler Sensorik. In *Industrial Ethernet Journal III/2012*, pages 902–904, August 2012.

[22] Nathan Pocock Michael J. Bryant, Thomas J. Bruke. OPC-UA at Chip Level for Device-to-Device Connectivity. In *OPC Foundation General Assembly Meeting Webinar*, pages 33–34, http://www.opcfoundation.org/DownloadFile.aspx/Presentations/GAM /GeneralAssemblyMeeting2012Slides.pdf?RI=946, December 2012.