Modelling and Simulation of Tractor-Implement Automation using OPC Unified Architecture for Next Generation ISOBUS

Matti Siponen
Ilkka Seilonen
Timo Oksanen
Modelling and Simulation of Tractor-Implement Automation using OPC Unified Architecture for Next Generation ISOBUS

Matti Siponen
Ilkka Seilonen
Timo Oksanen

Aalto University
School of Electrical Engineering
Department of Electrical Engineering and Automation
Modelling and Simulation of Tractor-Implement Automation using OPC Unified Architecture for Next Generation ISOBUS

Preface

This working paper presents proof of concept OPC UA information models for ISOBUS Tractor ECU and ISOBUS GNSS (GPS) devices of ISO 11783 networks; in addition to the previously developed model for ISOBUS TC.

The purpose of these information models is to demonstrate how OPC UA could be used in data exchange in a network consisting of a tractor represented by Tractor ECU, the Task Controller, an implement and two GNSS devices connected to the tractor and the implement respectively.

The information model for Tractor ECU extends the information model for the Task Controller and implements “Next Generation Task Controller for Agricultural Machinery using OPC Unified Architecture” designed by Matti Siponen. The information model for GNSS devices extends the device model defined in OPC Unified Architecture for Devices companion specification release 1.01.

The use of the new information models shall be demonstrated with a network consisting of a simulated implement, simplified TC, prototype OPC UA TECU and OPC UA GNSS Servers for the implement and the tractor.

This work was done as part of research project MaTyKo 2025 (Maatalousyökone 2025 - yhteensopivuus, palvelut ja tulevaisuudenstandardit).

Espoo, 10th January 2020

Matti Siponen, Ilkka Seilonen and Timo Oksanen
Preface ............................................................................................................. 1
List of Definitions and Abbreviations.......................................................... 3
1. Introduction .......................................................................................... 4
   1.1 ISO 11783 Standard Series ............................................................ 4
   1.2 OPC Unified Architecture............................................................. 6
   1.3 Past research on OPC UA and ISO 11783 Networks................. 10
2. Requirements .................................................................................... 19
   2.1 TECU............................................................................................ 19
   2.2 GNSS (GPS) Devices .................................................................. 19
   2.3 Simulation with OPC UA Applications........................................ 21
3. Information models for TECU and GNSS devices ....................... 24
   3.1 TECU........................................................................................... 24
   3.2 GNSS (GPS)................................................................................ 27
4. OPC UA Applications ........................................................................ 29
   4.1 OPC UA Simulation Server ........................................................ 29
   4.2 OPC UA Tractor GPS and OPC UA Implement GPS ............... 29
   4.3 OPC UA TECU............................................................................ 30
   4.4 OPC UA Implement Server v2 ................................................... 30
   4.5 OPC UA Task Controller v2 ....................................................... 30
5. Results ................................................................................................. 31
   5.1 Test Setup .................................................................................... 31
   5.2 Test Results ................................................................................ 32
6. Summary and Conclusions ............................................................. 39
References ................................................................................................. 40
### List of Definitions and Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AEF</td>
<td>Agricultural Industry Electronics Foundation</td>
</tr>
<tr>
<td>CAN</td>
<td>Controller Area Network</td>
</tr>
<tr>
<td>CF</td>
<td>Control function</td>
</tr>
<tr>
<td>COG</td>
<td>Course Over Ground</td>
</tr>
<tr>
<td>DDE</td>
<td>Data dictionary entity</td>
</tr>
<tr>
<td>DDI</td>
<td>Data dictionary identifier</td>
</tr>
<tr>
<td>DDOP</td>
<td>Device descriptor object pool</td>
</tr>
<tr>
<td>DET</td>
<td>Device element</td>
</tr>
<tr>
<td>DOR</td>
<td>Device object reference</td>
</tr>
<tr>
<td>DPD</td>
<td>Device process data</td>
</tr>
<tr>
<td>DPT</td>
<td>Device property</td>
</tr>
<tr>
<td>DTC</td>
<td>Data container</td>
</tr>
<tr>
<td>DVC</td>
<td>Device</td>
</tr>
<tr>
<td>DVP</td>
<td>Device value presentation</td>
</tr>
<tr>
<td>ECU</td>
<td>Electronic control unit</td>
</tr>
<tr>
<td>FMIS</td>
<td>Farm management information system</td>
</tr>
<tr>
<td>GNSS</td>
<td>Global Navigation Satellite System</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>HDOP</td>
<td>Horizontal dilution of precision</td>
</tr>
<tr>
<td>OPC UA</td>
<td>OPC Unified Architecture</td>
</tr>
<tr>
<td>OPC UA DI</td>
<td>OPC UA for Devices</td>
</tr>
<tr>
<td>PTO</td>
<td>Power take-off</td>
</tr>
<tr>
<td>RSC</td>
<td>Resource connector</td>
</tr>
<tr>
<td>SOG</td>
<td>Speed Over Ground</td>
</tr>
<tr>
<td>TC</td>
<td>Task controller</td>
</tr>
<tr>
<td>TECU</td>
<td>Tractor ECU</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
</tbody>
</table>
1. Introduction

1.1 ISO 11783 Standard Series

ISO 11783 is a communication standard for tractors and machinery for agriculture and forestry [1]. ISO 11783 standard series specifies a network for control and communication between tractors and implements. ISOBUS is an implementation of ISO 11783 promoted by the Agricultural Industry Electronics Foundation (AEF) to enhance compatibility between products from different manufacturers [2].

ISO 11783 networks consist of electronic control units (ECU) that provide control functions (CF) communicating with each other for data exchange and control [1]. CFs include Tractor ECU (TECU), Task Controller (TC) and various CFs for implements and their subsystems such as spray rate control and section on/off control.

TECU is a CF that represents the tractor in an ISO 11783 network [3]. TECU broadcasts messages containing information on the current state of the tractor to implements and receives control messages from implements to control the subsystems of the tractor. The information broadcasted by TECU includes the ground-based and the wheel-based speed of the tractor, hitch positions and states, PTO output shaft speeds and engagements and so on [4].

TC is a CF that controls implements based on tasks received from the Farm Management Information System (FMIS) [5]. These tasks range from logging data from implements to controlling the implements based on position data from a GNSS (GPS) device and prescription maps. Additionally, TCs may support automatic section control, which enables the TC to automatically turn sections of an implement on and off based on the implement’s section geometry and the travelled path to avoid treating field areas more than once. The data logged during a task is transferred to FMIS after the task has been completed.

The implements describe their structure to the TC with device descriptor object pool (DDOP) XML files [5]. A DDOP file consists of exactly one Device (DVC) object that provides general information on the modelled device, at least one Device Element (DET) object that represents the entire device and optionally other DET objects that represent the subsystems and components of the modelled device, any number of Device Process Data (DPD) and Device Property (DPT) objects that represent process data and properties of a DET object that references them with Device Object Reference (DOR) objects and Device Value Presentation (DVP) objects that specify equations for converting the values of DPD and DPT objects referencing the DVP object to chosen units. These objects are collectively called device descriptor objects. A DDOP diagram for a dual operation device capable of fertilizing and sowing is illustrated in Figure 1.
Each DPD and DPT object in a DDOP file has a data dictionary identifier (DDI) that corresponds to a data dictionary entity (DDE) in ISOBUS data dictionary [6]. These DDEs describe the data represented with DPD and DPT objects. DDEs include measurements and setpoints of application rates of applied products, boom section work states and positions and so on. The task files provided by FMIS identify which DPD objects the TC should monitor and control based on their DDI.

AEF has defined guidelines for implementing three separate TC functionalities:

- **TC-BAS** is required to read and write total values [7].
- **TC-GEO** is required to read and write total values and to control implements based on prescription maps and position data from a GNSS device [8].
- **TC-SC** is required to control implement’s sections based on a constantly updated coverage map and position data from a GNSS device [9].

These functionalities and instructions on how to make devices compatible with them were added to the second edition of ISO 11783 part 10 as an annex. This annex lists recommended and required DDEs for various device classes and provides sample DDOP diagrams such as the one seen in Figure 1.

In ISO 11783 networks, ECUs are connected with a single linear, 250 kbit/s, twisted, non-shielded, quad-cable and use Control Area Network (CAN) 2.0 B extended frame format with 29-bit identifiers and up to 8 bytes of data per frame [10]. Excluding the overhead of frame headers and considering the bus load of 50%, the theoretical payload capacity for all applications is limited to 64 kbit/s. The use of CAN bus is limiting the development of new functionalities for ISO 11783 networks such camera image and more precise section control with high rate GNSS receivers.
The next generation communication technology to provide higher bandwidth compared with CAN bus has been drafted by AEF project team [11]. The foreseen solution will be based on one twisted pair Ethernet, with the baudrate 1 Gb/s. The relevant available standard is 1000BASE-T1 which offers a solution for automotive Ethernet. However, this technology is limited only for physical layer and any protocol on top of IP based Ethernet may be used. Therefore, between the IP network and the functionalities of ISO 11783 (like virtual terminal), so middleware is required. OPC UA is one option for this middleware. The intention is to use the new backbone for communication not only for command and control of implements, but also camera systems, wireless in-field communication and other functionalities that have not been possible within the current ISO 11783 network. The project team also defines specific connectors for tractor-implement Ethernet. [11]

1.2 OPC Unified Architecture

OPC Unified Architecture (OPC UA) is a platform-independent communication standard for systems and devices published and maintained by the OPC Foundation [12]. The main purpose of OPC UA is modelling and communication of data with object-oriented techniques. OPC UA is the successor of a collection of standards known informally as OPC Classic.

OPC UA supports two communication models: OPC UA Client-Server communication model and OPC UA PubSub communication model. In the former communication model, an OPC UA Client and an OPC UA Server form a Session to exchange Service request and response Messages [11]. In the latter communication model an OPC UA Publisher sends NetworkMessages to a Message Oriented Middleware, which distributes the messages to subscribed OPC UA Subscribers [13]. The communication models are illustrated in Figure 2 and Figure 3.

Figure 2. OPC UA Client-Server communication model (Adapted from OPC 10000-1).
The available Services in OPC UA Client-Server communication model have been categorized into ten Service Sets including Session Service Set for forming a Session between a Client and a Server, View Service Set for browsing a Server’s contents, Attribute Service Set for reading values from and writing values to a Server and MonitoredItem and Subscription Service Sets for subscribing to receive notifications on changes of values and events on a Server [14]. A Server is not required to support all Service Sets and may conform to a Profile, which defines the Services provided by the Server to its Clients [15].

There are no inherent limitations on how OPC UA applications should form connections with each other. A single Client may form Sessions with multiple Servers and vice versa. Servers are allowed to limit the number of concurrent Sessions to be able to respond to requests from Clients in a timely manner. Similarly, a single Subscriber may receive NetworkMessages from multiple Publishers and vice versa.

An OPC UA application can be any combinations of Client, Server, Publisher and Subscriber. Applications that act as both Client and Server can be used to either form a chain of Servers or to enable peer-to-peer communication in a group of Servers. An application acting as Client and Subscriber and another application acting as Server and Publisher allows combining the communication models. It is also possible for a single application to act as Client, Server, Publisher and Subscriber at the same time.

The information exposed by an OPC UA Server is referred to as its AddressSpace and it consists of Nodes connected with References [16]. The Node model is illustrated in Figure 4. The type of Reference used to connect the Nodes defines their relationship. All Nodes have four mandatory Attributes: NodeId that identifies the Node, BrowseName and DisplayName that name the Node and NodeClass that defines the class of the Node.

OPC UA has eight NodeClasses: Object, Variable, Method, ObjectType, VariableType, ReferenceType, DataType and View. The first three NodeClasses are used for creating instances of Object, Variables and Methods. The following four NodeClasses are used for defining types of Objects, Variables, References
and data and are collectively known as TypeDefinitionNodes. The last Node
class is used for creating subsets of the AddressSpace.

![Figure 4. OPC UA Node model [16].](image)

Each Object or Variable Node has to Reference a ObjectType or a Variable-
Type Node with HasTypeDefinition Reference. The value of the DataType At-
tribute of a Variable or VariableType Node must be a NodeId of a DataType
Node. References used to connect Nodes must also be defined with Reference-
Type Nodes.

OPC UA supports simple and complex TypeDefinitionNodes. Simple
TypeDefinitionNodes only define the semantics of the type while complex
TypeDefinitionNodes may also target other Nodes with HasComponent and
HasProperty References. These referenced Nodes are called InstanceDeclara-
tions. A complex ObjectType and its instance are illustrated in Figure 5.

![Figure 5. Complex ObjectType “AI_BLK_TYPE” and its instance “AI_BLK_1” [16].](image)

The relationship between a complex TypeDefinitionNode and its In-
stanceDeclarations is further described with ModellingRules, which are Man-
datory, Optional, ExposesItsArray, OptionalPlaceholder and MandatoryPlace-
holder. An InstanceDeclaration with Mandatory ModellingRule must exist in
all instances of the complex TypeDefinitionNode, while an InstanceDeclara-
tion with Optional does not. InstanceDeclarations of Complex VariableTypes
may use ExposesItsArray ModellingRule to declare that each value in an array
is represented by a separate Variable Node. Both Mandatory and Optional ModellingRules defined the TypeDefinitionNode and the BrowseName of the InstanceDeclaration, but OptionalPlaceholder and MandatoryPlaceholder only define the TypeDefinitionNode of the InstanceDeclaration, which means that instances of the complex TypeDefinitionNode may reference any Node of the chosen type instead of a specific Node.

OPC UA supports subtyping with inheritance and overriding. An InstanceDeclaration with OptionalPlaceholder or MandatoryPlaceholder ModellingRule may be replaced with any of its subtypes. Subtypes of complex TypeDefinitionNodes may add References to new Nodes, remove References to Nodes that their supertype Referenced and change ModellingRules of InstanceDeclarations of their supertype.

OPC UA also supports declaring TypeDefinitionNodes as abstract. No instances of abstract types may be created, but abstract types may have non-abstract subtypes and thus they can be used to organize TypeDefinitionNodes into groups and to define a basis for non-abstract subtypes.

A standardized configuration of an AddressSpace is called an information model. The base OPC UA information model is used as a basis for defining new information models [17]. Standardized information models have been defined by the OPC Foundation for Data Access [18], Alarms and Conditions [19], Programs [20], Historical Access [21], Aggregates [22] and Devices [23]. In addition to the base and standardized information models, OPC UA also allows users to define their own information models that best suit their needs.

OPC UA for Devices (OPC UA DI) defines device model for modelling devices [23]. The device model is illustrated in Figure 6. The main components of the device model are TopologyElementType and its subtype DeviceType that are used to model devices and their components and subsystems. TopologyElementType is an abstract and complex ObjectType with no mandatory components and its optional components are ParameterSet and MethodSet, which are used for listing the parameters and methods of a device or a device component. Additionally, the parameters and methods can be divided into groups with instances of FunctionalGroupType. DeviceType is also an abstract type and extends TopologyElementType with optional and mandatory properties that identify the device.
As both TopologyElementType and DeviceType are abstract ObjectType Nodes, the device model defined in OPC UA DI cannot be used on its own. Instead, non-abstract subtypes for TopologyElementType and DeviceType need to be defined for the target application. Companion specifications for various applications such as AutoID [24], FDT [25] and Commercial Kitchen Equipment [26] have been defined together with the OPC Foundation.

In April 2019, OPC UA DI release 1.02 was published by the OPC Foundation and was officially included in the OPC UA specification as OPC 10000-100. While some companion specifications have been updated to the latest release of OPC UA DI, some companion specifications are still using release 1.01 from 2013. Similarly, not all OPC UA software development tools have been updated to use release 1.02 so far.

1.3 Past research on OPC UA and ISO 11783 Networks

OPC UA over Ethernet has been identified as a potential replacement for CAN bus in ISO 11783 networks. The viability of using OPC UA in ISO 11783 networks has been studied previously by Piirainen et al. [27] by designing and evaluating an information model extending the device model of OPC UA DI for representing device descriptor objects of DDOP XML files as Object and Variable Nodes in an OPC UA Server’s AddressSpace. The ObjectType and VariableType Nodes designed by Piirainen et al. are illustrated in Figure 7 and Figure 8.

**Figure 6.** The device model [23].

**Figure 7.** ObjectTypes for representing DVC and DET objects [27].
DVC and DET objects were represented with instances of ISOBUSDeviceType and ISOBUSDeviceElementType respectively. DPD objects were represented with instances of either ISOBUSAalogDeviceProcessDataType or ISOBUSDiscreteDeviceProcessDataType and DPT objects were represented with instances of either ISOBUSAalogDevicePropertyType or ISOBUSDiscreteDevicePropertyType depending on whether the value of a DPD or DPT object was analog or discrete. The relationships between the objects were represented with suitable References and the information model omitted DVP objects.

An OPC UA Server application that reads an implement’s DDOP and generates Object and Variable Nodes based on it was designed and developed by Piirainen. The application was successfully able to generate the AddressSpace and allow OPC UA Client applications to write values to settable DPD objects. Thus OPC UA was found suitable for accessing the data of an implement.

An information model for data exchange between the TC and implements was designed and evaluated by Siponen [28]. The information model was inspired by the information model designed by Piirainen et al. and used similar naming scheme for TypeDefinitionNodes. The information model for TC and implements extended the device model of OPC UA DI release 1.01 and defined ObjectType and VariableType Nodes for representing device descriptor objects of the DDOP model.

In the information model designed by Siponen, DVC and DET objects were represented with instances of ISOBUSDeviceType and ISOBUSDeviceElementType respectively. The ObjectType Nodes for DVC and DET objects are illustrated in Figure 9 and Figure 10. The attributes of DVC and DET objects were modelled as Variables of ParameterSet Objects. Instead of connecting a parent DET object to its child DET objects with identifier numbers, the information model used suitable OPC UA References to connect a parent to its children. The information model did not use DOR objects either. In addition to implementing the modelling features of the DDOP model, the information model enhanced the DDOP model with new modelling features such as grouping DPD and DPT objects, modelling many-to-many relationships between DET objects and new device element type for booms.

Figure 8. VariableTypes for representing DPD and DPT objects [27].
In the current DDOP model, each DPD and DPT object must reference exactly one DDI of a DDE that specifies the definition, the role and the unit of the
data represented by the object. Thus, adding new specifications requires defining new DDEs and adding them to the ISOBUS data dictionary. Siponen proposed that instead of using DDIs, a three-part identifier consisting of Definition, Role and Unit could be used to describe DPD and DPT objects. This concept was extended by grouping related DPD and DPT objects with Data Container (DTC) objects that specify their contents with a three-part identifier consisting of Definition, Structure Configuration and Unit. With the use of DTC objects, new Definitions, Roles and Units could be added separately and their combinations could be approved as seen fit. Siponen defined mapping from DDEs to Definitions, Roles and Units for all DDEs in ISOBUS data dictionary and provided rules for grouping related DPD and DPT objects. The chosen units belonged to the SI system and necessary unit conversions could be handled by the TC. The VariableType Nodes for DTC and DPD objects are illustrated in Figure 11 and Figure 12. The information model used PropertyType defined in OPC 10000-5 for DPT objects.

**Figure 11.** ISOBUSdataContainerType for DTC objects [28].
Only parent-child relationships are used to model relationships between DET objects in the current DDOP model, which are unsuitable for modelling many-to-many relationships between DET objects. As modelling such relationships could be beneficial in future implements, Siponen proposed the use of Resource Connector (RSC) objects to connect a DET object acting as a resource user to another DET object acting as a resource object enabling modelling many-to-many relationships between DET objects. Additionally, Siponen proposed that RSC objects should be allowed to contain DTC objects related to application rate controls to control the application rate separately for each pair of boom and bin. The ObjectType Node for DTC objects is illustrated in Figure 13.
Figure 13. ISOBUSResourceConnectorType for RSC objects [28].

The concept of booms was added to the second edition of ISO 11783 part 10. To enable backwards-compatibility between the first and the second edition, the standard suggests that booms should be modelled as devices when the implement has only one boom or as functions when the implement has more than one boom [5]. Siponen proposed a new device element type for booms and modelling rules for boom and other types of DET objects to make implements compatible with TC-BAS, TC-GEO and TC-SC functionalities.

Two OPC UA applications utilizing the information model were designed and developed: OPC UA Implement Server acting as a simulated implement and OPC UA Task Controller acting as a simplified TC. These applications were used to evaluate the information model.

The OPC UA Implement Server was designed to provide access to three different simulated product application devices: a sprayer with four booms and one bin per boom with application rate controlled at a boom level, a seed drill with one boom and six bins with application rate controlled at a section level and a spreader with one boom and three bins with application rate controlled at a boom level and configurable bin cultural practices and application rate units. The OPC UA Implement Server generated an AddressSpace based on the selected implement and updated values of Variable Nodes 100 times per second.

The OPC UA Task Controller was designed to control connected implements with OPC UA Write and Call Services based on the provided tasks consisting of products to be applied. Each product had a cultural practice identifier, an ap-
plication rate unit identifier and a prescription map. A prescription map for the simulated seed drill is illustrated in Figure 14. The compatibility of the chosen task and the connected implement was verified by comparing the cultural practices and application rate units of products in the task and bins in the implement. The OPC UA Task Controller provided simulated tractor positions based on predetermined paths to perform site-specific application and automatic section control. The predetermined path for the simulated seed drill is illustrated in Figure 15.

![Figure 14](image1)

**Figure 14.** A prescription map for the simulated seed drill [28].

![Figure 15](image2)

**Figure 15.** The predefined path for the simulated seed drill [28].

The OPC UA applications were evaluated by running them on two different computers connected with an Ethernet cable and monitoring their data exchange with Wireshark to measure delays between Service request and response Messages. The OPC UA Implement Server was configured to represent different simulated implements and the OPC UA Task Controller was given tasks with varying numbers of products to measure the performance of the applications when increasing the number of Service requests per second.
The OPC UA applications could successfully transfer a simulated implement's DDOP from the OPC UA Implement Server to the OPC UA Task Controller with View and Attribute Services. The OPC UA Task Controller could also send proper commands to the OPC UA Implement Server in both automatic section control and site-specific application, as illustrated in Figure 16, Figure 17 and Figure 18.

**Figure 16.** The working width of the boom of the simulated seed drill shows that automatic section control is working as intended [28].

**Figure 17.** The coverage map produced by the OPC UA Task Controller for a product applied by the simulated seed drill [28].
Figure 18. A comparison of the setpoint application logged by the OPC UA Task Controller and the setpoint in a prescription map [28].

The performance of the OPC UA applications was evaluated by measuring the duration of OPC UA data exchange per iteration of the command algorithm and found to be sufficient to satisfy the goal of controlling an implement at 100 Hz, as illustrated in Figure 19. However, there were outlier cases where the data exchange could take ten times the average duration. Siponen speculated that the inconsistent performance could be caused by the used software development tools and running the OPC UA applications in Windows environment. Logging data from implements at frequencies higher than 10 Hz could not be tested due to the limitations of the used software development tools.

![Figure 19. A histogram durations measured by The OPC UA Task Controller with OPC UA Implement Server configured to act as a seed drill with 400 sections applying 6 products [28].](image-url)
2. Requirements

2.1 TECU

The proof-of-concept Information model for TECU shall extend the information model designed for TC and implements by Matti Siponen [28]. The tractor represented by the TECU shall be modelled similarly to implements as a device consisting of device elements that represent the tractor’s components and subsystems and their process data and properties. The ingoing and outgoing signals shall be represented with DTC Variables and Methods as seen fit. New identifiers for DTC Definitions and Structure Configurations shall be defined as necessary. The geometry of the tractor shall be modelled with connector and navigation reference DET Objects.

Instead of modelling all signals defined in ISO 11783-7 as DTC Variables and Methods, a subset of these signals has been selected for the proof of concept information model for TECU. The selected signals are listed in Table 1. Signals related to valves and lights have been omitted from the information model for TECU.

Modelling rules for different types of tractors shall be defined. It is assumed that all modelled tractors have at least one connector either in the rear or in the front. These connectors may include either hitch and PTO or just one of the two. Additionally, a GNSS (GPS) device may be installed on the modelled tractor.

2.2 GNSS (GPS) Devices

The proof-of-concept Information model for GNSS devices shall extend the device model defined in OPC UA DI release 1.01 by defining a subtype of DeviceType for representing GNSS devices. This subtype shall define mandatory FunctionalGroups for organizing the Variable Nodes representing the signals sent by a GNSS device.

A GNSS device in the ISO 11783 network uses the NMEA2000 protocol. The fields of NMEA2000 messages included in the proof of concept information model as Variable Nodes are listed in Table 2. In addition, each GNSS device needs to be able to identify itself and its user in networks with multiple GNSS devices.
Table 1. Selected TECU Signals.

<table>
<thead>
<tr>
<th>Parameter Group</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ground-based speed and distance (B.2)</td>
<td>Ground-based machine speed (A.5)</td>
</tr>
<tr>
<td>Ground-based speed and distance (B.2)</td>
<td>Ground-based machine distance (A.6)</td>
</tr>
<tr>
<td>Ground-based speed and distance (B.2)</td>
<td>Ground-based machine direction (A.7)</td>
</tr>
<tr>
<td>Wheel-based speed and distance (B.3)</td>
<td>Wheel-based machine speed (A.8)</td>
</tr>
<tr>
<td>Wheel-based speed and distance (B.3)</td>
<td>Wheel-based machine distance (A.9)</td>
</tr>
<tr>
<td>Wheel-based speed and distance (B.3)</td>
<td>Wheel-based machine direction (A.10)</td>
</tr>
<tr>
<td>Maintain power (B.4)</td>
<td>Maintain ECU power (A.13)</td>
</tr>
<tr>
<td>Maintain power (B.4)</td>
<td>Maintain actuator power (A.14)</td>
</tr>
<tr>
<td>Secondary or front hitch status (B.6)</td>
<td>Front hitch position (A.19.1)</td>
</tr>
<tr>
<td>Secondary or front hitch status (B.6)</td>
<td>Front hitch in-work indication (A.19.5)</td>
</tr>
<tr>
<td>Secondary or front hitch status (B.6)</td>
<td>Front draft (A.19.7)</td>
</tr>
<tr>
<td>Primary or rear hitch status (B.7)</td>
<td>Rear hitch position (A.19.2)</td>
</tr>
<tr>
<td>Primary or rear hitch status (B.7)</td>
<td>Rear hitch in-work indication (A.19.6)</td>
</tr>
<tr>
<td>Primary or rear hitch status (B.7)</td>
<td>Rear draft (A.19.8)</td>
</tr>
<tr>
<td>Secondary or front PTO output shaft (B.8)</td>
<td>Front PTO output shaft speed (A.20.1)</td>
</tr>
<tr>
<td>Secondary or front PTO output shaft (B.8)</td>
<td>Front PTO output shaft speed setpoint (A.20.3)</td>
</tr>
<tr>
<td>Secondary or front PTO output shaft (B.8)</td>
<td>Front PTO engagement (A.20.7)</td>
</tr>
<tr>
<td>Primary or rear PTO output shaft (B.9)</td>
<td>Rear PTO output shaft speed (A.20.2)</td>
</tr>
<tr>
<td>Primary or rear PTO output shaft (B.9)</td>
<td>Rear PTO output shaft speed setpoint (A.20.4)</td>
</tr>
<tr>
<td>Primary or rear PTO output shaft (B.9)</td>
<td>Rear PTO engagement (A.20.8)</td>
</tr>
<tr>
<td>Hitch and PTO commands (B.10)</td>
<td>Front hitch position command (A.19.3)</td>
</tr>
<tr>
<td>Hitch and PTO commands (B.10)</td>
<td>Rear hitch position command (A.19.4)</td>
</tr>
<tr>
<td>Hitch and PTO commands (B.10)</td>
<td>Front PTO output shaft speed set point — Command (A.20.5)</td>
</tr>
<tr>
<td>Hitch and PTO commands (B.10)</td>
<td>Rear PTO output shaft speed set point — Command (A.20.6)</td>
</tr>
<tr>
<td>Hitch and PTO commands (B.10)</td>
<td>Front PTO engagement (A.20.13)</td>
</tr>
<tr>
<td>Hitch and PTO commands (B.10)</td>
<td>Rear PTO engagement (A.20.14)</td>
</tr>
<tr>
<td>Machine selected speed (B.28.1)</td>
<td>Machine selected speed (A.30.1)</td>
</tr>
<tr>
<td>Machine selected speed (B.28.1)</td>
<td>Machine selected direction (A.30.3)</td>
</tr>
<tr>
<td>Machine selected speed (B.28.1)</td>
<td>Machine selected speed source (A.30.4)</td>
</tr>
<tr>
<td>Machine selected speed command (B.28.2)</td>
<td>Machine selected speed set point command (A.30.5)</td>
</tr>
<tr>
<td>Machine selected speed command (B.28.2)</td>
<td>Machine selected direction set point command (A.30.6)</td>
</tr>
</tbody>
</table>
Table 2. Selected NMEA2000 Fields.

<table>
<thead>
<tr>
<th>Parameter Group</th>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position, Rapid Update (129025)</td>
<td>Latitude (DD064)</td>
</tr>
<tr>
<td>Position, Rapid Update (129025)</td>
<td>Longitude (DD065)</td>
</tr>
<tr>
<td>GNSS Pseudorange Noise Statistics (129542)</td>
<td>STD of Major axis (DD075)</td>
</tr>
<tr>
<td>GNSS Position Data (129029)</td>
<td>Altitude (DD204)</td>
</tr>
<tr>
<td>GNSS Position Data (129029)</td>
<td>Method, GNSS (DD067)</td>
</tr>
<tr>
<td>GNSS Position Data (129029)</td>
<td>Number of SVs (DD006)</td>
</tr>
<tr>
<td>GNSS Position Data (129029)</td>
<td>HDOP (DD055)</td>
</tr>
<tr>
<td>COG &amp; SOG, Rapid Update (129026)</td>
<td>Course Over Ground (DD165)</td>
</tr>
<tr>
<td>COG &amp; SOG, Rapid Update (129026)</td>
<td>Speed Over Ground (DD044)</td>
</tr>
</tbody>
</table>

2.3 Simulation with OPC UA Applications

The use of the proof of concept information models for TECU and GNSS devices shall be demonstrated with a network of OPC UA application utilizing these information models. The network consists of the following applications:

1. **OPC UA Simulation Server** that provides simulated data for TECU and GNSS devices
2. **OPC UA TECU** that represents a tractor as defined in 2.1
3. **OPC UA Tractor GPS** that represent a GNSS device installed on the tractor as defined in 2.2
4. **OPC UA Implement Server v2** that represents a simulated implement
5. **OPC UA Implement GPS** that represent a GNSS device installed on the implement as defined in 2.2
6. **OPC UA Task Controller v2** that represents a simplified TC

This network of OPC UA applications is illustrated in Figure 20 with arrows pointing from a Client to a Server.
The OPC UA Simulation Server uses a predetermined path to generate tractor positions for OPC UA TECU, OPC UA Tractor GPS and OPC UA Implement GPS. The predetermined path consists of the first driving along the border of a 250 m x 250 m field and then covering the field in zig zag pattern as illustrated in Figure 21. Additionally, tractor speed and key switch state are generated for OPC UA TECU. There is no need to define an information model for the OPC UA Simulation Server and thus any suitable Nodes of the base OPC UA information model defined in OPC 10000-5 may be used. The OPC UA Simulation Server should be controllable with Methods and setpoint Variables.
Figure 21. The predetermined path for OPC UA Simulation Server. The path begins from (0,0) and ends in (221.1,250).

The OPC UA Implement Server and OPC UA Task Controller applications designed and developed by Siponen need to be modified to read information on the tractor from OPC UA TECU, OPC UA Tractor GPS and OPC UA Implement GPS instead of generating this information on their own or using predefined constant values. These modified versions shall be called OPC UA Implement Server v2 and OPC UA Task Controller v2 respectively.
3. Information models for TECU and GNSS devices

3.1 TECU

To model tractors and TECU with the information model for TC and implements, the following DTC Definition identifiers have been added:

174. Key Switch State
175. Working Position
176. PTO Engagement
177. Draft
178. Ground-based Speed
179. Wheel-based Speed
180. Ground-based Distance
181. Wheel-based Distance

Key Switch State is used for representing the state of a tractor’s key switch as an enumerated Readable State. Working Position is used for representing the positions of hitches and uses a Structure Configuration that includes Actual Value and Setpoint Value and a Readable State for hitch in-work indication. PTO Engagement is used for representing the engagements of PTOs as enumerated Readable State and Setpoint State. Draft is used for representing horizontal forces applied to hitches by implements. Ground-based Speed and Wheel-based Speed are used for representing ground-based and wheel-based speed of a tractor respectively. Ground-based Distance and Wheel-based Distance are used for representing ground-based and wheel-based distances travelled by a tractor respectively. With these and previously defined Definitions, the signals selected in 2.1 can be represented as DTC Variables except for “Maintain ECU power” and “Maintain actuator power”, which are represented with Methods.

Unlike implements where totals can be set to zero by the TC, the totals of TECU should not be settable. A new Role called TECU Total is introduced to represent such totals. This Role is used in DTCs representing ground-based and wheel-based distances and is a part of Structure Configuration 26, Analog TECU, Task Effective and Ineffective Totals. No new Units were defined for modelling TECU and tractors.

Like implements, a tractor represented by a TECU is modelled with a single DVC Object and multiple DET Objects and their DTC Variables. In addition to the root DET Object, each tractor is required to have at least one connector either in the front or in the rear represented with a connector DET Object. If a tractor has a rear connector, it may also have a rear hitch and a rear PTO or just one of the two. Similarly, if a tractor has a front connector, it may also have a front hitch and a front PTO or just one of the two. Hitches and PTOs are
represented with function DET Objects. Additionally, a tractor may have GNSS device installed on it represented with a navigation reference DET Object.

Two different tractors are modelled in Figure 22 and Figure 23. The tractor modelled in Figure 22 represents the minimal tractor configuration consisting of only a rear connector without a hitch, a PTO or a GNSS device. The tractor modelled in Figure 23 represents the maximal tractor configuration with hitches and PTOs for both front and rear connectors and a GNSS device. The abbreviations used for data items of DTC Variables are explained in Table 3.

Figure 22. A tractor with only a rear connector.
Table 3. The abbreviations for data items.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AV</td>
<td>Actual value</td>
<td>Represents a measured value of a physical process or quality</td>
</tr>
<tr>
<td>SV</td>
<td>Setpoint value</td>
<td>Represents a value that is used for controlling a physical process or quality</td>
</tr>
<tr>
<td>CV</td>
<td>Constant value</td>
<td>Represents a constant value of a physical process or quality</td>
</tr>
<tr>
<td>ET</td>
<td>Effective total</td>
<td>Represents a resettable total value of a physical process accumulated when the device has been working</td>
</tr>
<tr>
<td>IT</td>
<td>Ineffective total</td>
<td>Represents a resettable total value of a physical process accumulated when the device has not been working</td>
</tr>
<tr>
<td>TT</td>
<td>TECU total</td>
<td>Represents a non-resettable total value of a physical process accumulated since the last booting of TECU</td>
</tr>
<tr>
<td>RS</td>
<td>Readable state</td>
<td>Represents a state of a device or a process that can be read but not written</td>
</tr>
<tr>
<td>SS</td>
<td>Setpoint state</td>
<td>Represents a setpoint state that affects another state of a device or a process</td>
</tr>
<tr>
<td>X</td>
<td>X-offset</td>
<td>Represents an offset along the x-axis</td>
</tr>
<tr>
<td>Y</td>
<td>Y-offset</td>
<td>Represents an offset along the y-axis</td>
</tr>
<tr>
<td>Z</td>
<td>Z-offset</td>
<td>Represents an offset along the z-axis</td>
</tr>
</tbody>
</table>
### 3.2 GNSS (GPS)

GNSS (GPS) devices are represented with instances of GPSdeviceType ObjectType. GPSdeviceType is a subtype of DeviceType defined in OPC UA DI release 1.01 and it adds References to the following mandatory FunctionalGroups:

- Identification
- Position Rapid Update
- GNSS Pseudorange Noise Statistics
- GNSS Position Data
- COG & SOG Rapid Update

Identification contains parameters that identify the GNSS device and its user and the other functional groups contain fields selected in 2.2. GPSdeviceType is illustrated in Figure 24 and the parameters of ParameterSet are listed in Table 4.

![Figure 24. GPSdeviceType.](image-url)
<table>
<thead>
<tr>
<th>Functional Group</th>
<th>Parameter</th>
<th>NodeId</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification</td>
<td>GPS ID</td>
<td>ns=2;si=GPS.Id</td>
</tr>
<tr>
<td>Identification</td>
<td>GPS NAME</td>
<td>ns=2;si=GPS.NAME</td>
</tr>
<tr>
<td>Identification</td>
<td>User NAME</td>
<td>ns=2;si=GPS.User</td>
</tr>
<tr>
<td>Position Rapid Update</td>
<td>Latitude</td>
<td>ns=2;si=GPS.Latitude</td>
</tr>
<tr>
<td>Position Rapid Update</td>
<td>Longitude</td>
<td>ns=2;si=GPS.Longitude</td>
</tr>
<tr>
<td>GNSS Pseudorange Noise Statistics</td>
<td>STD of Major Axis</td>
<td>ns=2;si=GPS.STDofMajorAxis</td>
</tr>
<tr>
<td>GNSS Position Data</td>
<td>Altitude</td>
<td>ns=2;si=GPS.Altitude</td>
</tr>
<tr>
<td>GNSS Position Data</td>
<td>GNSS Method</td>
<td>ns=2;si=GPS.Method</td>
</tr>
<tr>
<td>GNSS Position Data</td>
<td>Number of SVs</td>
<td>ns=2;si=GPS.SVs</td>
</tr>
<tr>
<td>GNSS Position Data</td>
<td>HDOP</td>
<td>ns=2;si=GPS.HDOP</td>
</tr>
<tr>
<td>COG &amp; SOG Rapid Update</td>
<td>COG</td>
<td>ns=2;si=GPS.COG</td>
</tr>
<tr>
<td>COG &amp; SOG Rapid Update</td>
<td>SOG</td>
<td>ns=2;si=GPS.SOG</td>
</tr>
</tbody>
</table>
4. OPC UA Applications

4.1 OPC UA Simulation Server

The OPC UA Simulation Server is an OPC UA Server application that provides access to simulated data related to tractor position, speed and key switch state. This data is used by OPC UA TECU, OPC UA TECU GPS and OPC UA Implement GPS to generate values for variables. The OPC UA Simulation Server calculates new tractor position every 10 ms based on the predetermined path described in 2.3.

The OPC UA Simulation Server provides a settable Variable for controlling the speed of the simulated tractor. The maximum acceleration for the simulated tractor is 0.5 m/s\(^2\) and the maximum deacceleration is 7.5 m/s\(^2\). The simulated tractor can only move forward and its speed can never be less than 0 m/s.

In addition to Variables, the OPC UA Simulation Server provides Methods for starting the tractor up, shutting it down and powering it down. Starting the tractor up changes key switch state from “Key switch OFF” to “Key switch not OFF”. Shutting the tractor down changes key switch state from “Key switch not OFF” to “Key switch OFF”. Powering the tractor down turns off ECU power and actuator power although this has no impact on the simulations.

4.2 OPC UA Tractor GPS and OPC UA Implement GPS

OPC UA Tractor GPS and OPC UA Implement GPS are Client-Server hybrid applications that calculate GNSS data every 10 ms based on simulated tractor positions read from the OPC UA Simulation Server and pre-configured offsets given to the applications as XML configuration files. The applications calculate COG and SOG based on current and previous GNSS coordinates. The values for Method, Number of SVs and HDOP are updated periodically with pseudorandom values. Noise is added to longitude and latitude. The maximum absolute errors due to noise are 0.058 m for longitude and 0.11 m for latitude. The values of the Variables of the OPC UA Tractor GPS and OPC UA Implement GPS are updated every 10 ms.

The identification parameters provided by OPC UA Tractor GPS and OPC UA Implement GPS were intended to be used OPC UA TECU and OPC UA Implement Server v2 to identify the correct GNSS device after discovering them with the help of a Local Discovery Server. However, the Local Discovery Server provided by the OPC Foundation had trouble registering the OPC UA Tractor GPS and OPC UA Implement GPS and OPC UA TECU and OPC UA Implement Server v2 were configured to connect to OPC UA Tractor GPS and OPC UA Implement GPS directly.
4.3 OPC UA TECU

OPC UA TECU is a Client-Server hybrid application that provides information on the internal state of the tractor based on both data from OPC UA Simulation Server and OPC UA Tractor GPS and local simulations of hitch positions and PTO speeds and engagements. The ground-based speed of the tractor is a fusion of speeds provided by OPC UA Simulation Server and OPC UA Tractor GPS and the wheel-based speed is the ground-based speed multiplied by 1.1. The values of the Variables of the OPC UA TECU are updated every 10 ms.

4.4 OPC UA Implement Server v2

OPC UA Implement Server v2 is a modified version of the OPC UA Implement Server developed by Siponen. It is a Client-Server hybrid application that provides access to a simulated implement’s DDOP. Unlike the original OPC UA Implement Server, the OPC UA Implement v2 is connected to both OPC UA TECU and OPC UA Implement GPS. Instead of assuming that the implement is moving at a constant speed, the value of SOG of the OPC UA Implement GPS is used as the implement’s speed. The OPC UA Implement Server v2 also monitors the key switch state of the OPC UA TECU and has been configured to call Maintain ECU power Method five times with one second intervals when it detects a change from “Key switch not OFF” to “Key switch OFF”.

4.5 OPC UA Task Controller v2

OPC UA Task Controller v2 is a modified version of the OPC UA Task Controller developed by Siponen. It is a Client application that controls implements based on given tasks. While the original OPC UA Task Controller provided internally generated simulated tractor positions, the OPC UA Task Controller v2 is connected to the OPC UA Tractor GPS and the OPC UA TECU and uses their data as the position and the speed of the tractor. The OPC UA Task Controller v2 is also connected to the OPC UA Simulation Server to shut the simulated tractor down once a task has been completed, although this operation could be handled by an external OPC UA Client application such as UaExpert.
5. Results

5.1 Test Setup

All the OPC UA applications were run on a single PC. The computer used for the tests is described in table 5. The communication between the applications was not monitored nor was the performance of the OPC UA Task Controller v2 logged.

Table 5. Computer used for the tests.

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Processor</th>
<th>RAM</th>
<th>Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows 10 Enterprise</td>
<td>i5-6300U @ 2.40 GHz</td>
<td>8 GB</td>
<td>SSD</td>
</tr>
</tbody>
</table>

The OPC UA Simulation Server v2 was configured to simulate a sprayer with two booms and two bins. The maximum working width of the booms is 25 m and there are 200 sections per boom. The first bin of the simulated sprayer contains liquid fertilizer and the other bin contains liquid crop protection product.

The task given to the OPC UA Task Controller v2 involved application of a fertilizer and a crop protection product. The prescription maps used in the task are illustrated in Figure 25 and Figure 26. The dimensions of the maps were 5000 x 5000 grid cells with 0.05 m cell width. The maximum application rate was 100 l/ha for the fertilizer and 300 l/ha for the crop protection product. The maps were based on an NDVI satellite image of a field in Illinois downloaded from Sentinel Hub [29].

Figure 25. Prescription map for fertilizer.
Figure 26. Prescription map for crop protection product.

5.2 Test Results

During the execution of the given task, the OPC UA Task Controller v2 logged data from the simulated sprayer and tractor positions received from OPC UA Tractor GPS. Log entries were added to the data log 10 times per second.

The tractor positions logged by the OPC UA Task Controller v2 are shown on Figure 27. These logged positions match the predetermined positions shown on Figure 21.
The correctness of the section control performed by the OPC UA Task Controller v2 can be evaluated by comparing the logged working widths and positions of booms. Decreases in the working width of the boom imply that sections have been turned off. A comparison of the working widths and positions of the first boom is shown in Figure 28.

Figure 27. Logged tractor path.
There are several instances where sections have been turned off incorrectly. The OPC UA Task Controller v2 turns sections off only when the sections are either entering an area that has already been treated or exiting the field. With the former case in mind, the heading of the tractor, which is also the heading of the booms, was compared to the working width of the first boom as seen on Figure 29. There are instances where the heading of the tractor spikes and results in sections being turned off incorrectly. One such heading error is demonstrated in Figure 30. The correct heading would be approximately $\pi$ and the wrong heading received from the OPC UA Tractor GPS is causing sections to be turned off incorrectly. The effects of incorrect measurements on the control algorithm could be reduced by implementing a Kalman filter that filters the positions received from the OPC UA Tractor GPS to reduce effects of noise.

*Figure 28.* The working width of the first boom (Boom 0) as a function of its position.
Figure 29. The heading of the tractor and the working width of the first boom (Boom 0).

Figure 30. The position of the simulated tractor and implement after receiving an incorrect heading for the OPC UA Tractor GPS.
The coverage map generated by the OPC UA Task Controller v2 for the fertilizer applied by the first boom is shown in Figure 31. In the Figure, green areas have been treated and grey areas have not been treated. The predetermined path was chosen such that the corners of the field would be left untreated. However, there are also a few small stripes in the field that were not treated.

![Coverage map for the fertilizer applied by the first boom.](image)

**Figure 31.** Coverage map for the fertilizer applied by the first boom.

The setpoint application rate for the fertilizer applied by the first boom and the prescription map for the fertilizer are compared as functions of boom position in Figure 32. The prescription map presented values with a precision of seven decimals while the OPC UA Task Controller v2 rounded these values to six decimals when logging data. However, the maximum absolute error between the prescription map and the logged application rate setpoints was 0.0012 l/m². Figure 33 shows this absolute error as a function of boom position. Figure 34 and Figure 35 show the same information for a part of the travelled path to make the figures easier to read.
Figure 32. The setpoint application rate and the prescription map for the fertilizer.

Figure 33. The absolute error between the setpoint application rate and the prescription map for the fertilizer.
Figure 34. The setpoint application rate and the prescription map for the fertilizer for a partial path.

Figure 35. The absolute error between the setpoint application rate and the prescription map for the fertilizer for a partial path.
6. Summary and Conclusions

This working paper presented proof-of-concept OPC UA information models for TECU and GNSS (GPS) devices in ISO 11783 networks. The information model for TECU was based on past research on the use of OPC UA in data exchange between TC and implements. The information model for GNSS devices was based on the device model specified in OPC UA DI release 1.01.

The use of the proof of concept information models was demonstrated with a network of OPC UA applications acting as a tractor represented by its TECU, a GNSS device connected to the tractor, TC, a simulated sprayer implement and a GNSS device connected to the implement. The applications were run on a single computer and they were able to successfully exchange data required by TC and implements in product application tasks including logging of data, section control and site-specific application.

Based on this and earlier study reported in the thesis by Matti Siponen, OPC UA is suitable for the following ISOBUS Functionalities: TC-GEO, TC-SC (and evidently also TC-BAS), and furthermore TECU and GNSS. These ISOBUS functionalities require plug-and-play connectivity between client and server and signal types and data rate are suitable for OPC UA Client-Server. A complete tractor-implement system was used to test the control concept (presented in Figure 20) and the proposed roles of Servers and Clients was also tested functional.
References


Modelling and Simulation of Tractor-Implement Automation using OPC Unified Architecture for Next Generation ISOBUS

Matti Siponen
Ilkka Selonen
Timo Oksanen