FDT, OPC UA, mobility, and CANopen – a compelling combination

Manfred Brill, Schneider Electric Automation GmbH

FDT is a standardized device management technology, evolving to IIoT including OPC UA and mobile technologies. This will allow (existing) CANopen devices to be part of this evolution. This contribution shortly describes the concept of the FDT technology as it is specified today. Then it is explained how the use of OPC UA in conjunction with FDT allows to provide data of existing field devices (e.g. CANopen). Further on the concept of managing field equipment with mobile devices. The respective architecture in discussion shows how a standardized mechanism using FDT will help the user and avoid proprietary solutions. Finally an outlook is given how these pieces are building the basis to evolve into the IIoT world.

What is FDT?

FDT standardises device management for field devices in one tool. Device management is standardised regardless of the device manufacturer and the fieldbus/network used. This means that every device in an FDT tool can be configured, operated and maintained by standardised interfaces – regardless of manufacturer, type or communication protocol. The technology is promoted by the FDT Group and incorporated in the IEC 62453, ISA 103 and China GB/T 29618 standards.

The three key elements of FDT are:

- FDT interface
  The FDT interface is the specification which describes standardised data exchange between devices and the control system, or the engineering and asset management tools.
- DTM (device driver)
  The DTM (Device Type Manager) provides a standardised structure for accessing the device parameters as well as configuring and operating devices and performing fault diagnostics. DTMs range from a simple graphical user interface for parameterisation to a sophisticated application which can handle complex real-time calculations for diagnostics and maintenance. DTMs are divided into three categories.
  The Device DTM is supplied by the device manufacturer and represents the entire logic and parameterization of a device. It creates a standard interface with the FDT frame application and can be used in any of these frame applications. The GUI is based on the DTM style guide.
  The Communication DTM represents the communication component which provides the frame application the access to the network.
  The Gateway DTM represents a component that connects two different networks/fieldbuses.
- FDT frame application
  The frame application is a software program which integrates device, communication and gateway DTMs from various manufacturers and for different networks/fieldbuses. It offers a shared, standardized environment where the management of DTMs, users and data take place. The features available in a frame application can vary in a broad range. The functionality of a frame is not defined it could be any tool like a stand-alone tool, a PLC programming tool or an asset management system. The only requirement is that the tool provides the standardized FDT interface.

An important capability of the FDT technology is the so-called “nested communication”. This means the access to a device crossing different network/ fieldbus hierarchies. An example is shown in Figure 1 where the plant structure is reflected in the frame by the “connected” DTMs (logical topology). It shows the Device DTM at the bottom, two gateway DTMs in the middle and the Communication DTM at the top. This would allow to manage CANopen devices remotely.

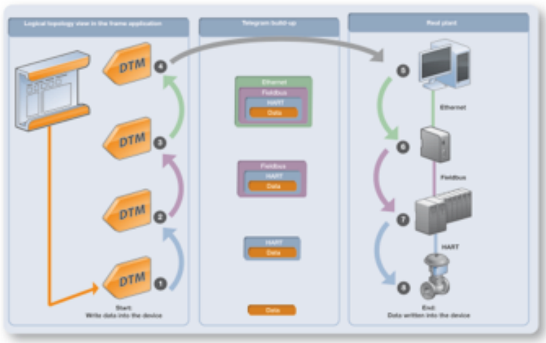

Figure 1: Nested communication

Unified device model with OPC UA

In the world of Industrie4.0/IIoT OPC UA is seen as an important part for the information exchange. Together with FDT it would be possible to provide a common information model for devices not supporting or are not able to support OPC UA (e.g. devices on existing fieldbuses).

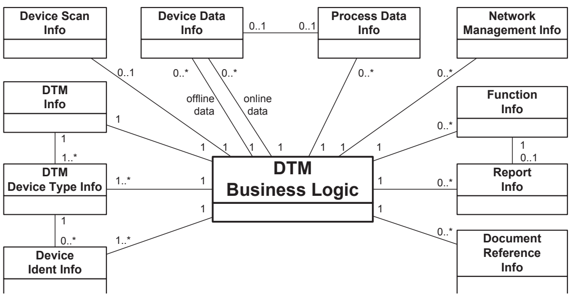

Figure 2: Information provided by a DTM

As shown in Figure 2, DTMs may provide various information, which is necessary for commissioning, operation and diagnosis of the respective device, for example:

- information for identification of a device or device type
- device parameters for configuration of a device, including semantic information for device parameters (e.g. value range, access)
- information about the I/O values of a device
- information about available configuration, diagnosis and maintenance functions for a device
- documentation of current data
- references to device documentation (e.g. manuals, technical documents, certificates)

To provide this information via OPC UA it is necessary to integrate an OPC UA server into the FDT frame application. Figure 3 shows the architecture of such a system. The frame application provides the topology and each device is represented by its corresponding DTM instance. The data and functions of each DTM further enrich the OPC UA information model. This information model is based on the companion standard “OPC UA for devices” from the OPC Foundation.
In this specification each device is modeled as Device Node. The child nodes of the Device Node provide access to information about the device, offline and online data, methods and network management information. Since this companion standard was chosen as a base for the integration of FDT data, it defines a compatible interface supporting all devices.
The FDT Group and the OPC Foundation worked jointly on a specification defining an information model for the mapping of DTM information. The development of the mapping of FDT2.0 to the OPC UA interface was based on an analysis of use cases for device integration. The following uses cases are supported:

- access to topology information
- access to device type information
- support of the different device types
- online identification of devices
- monitoring of device status
- access to device diagnosis
- access to offline and online data of devices
- access management for multi-user systems
- upload and download of device data
- monitoring of the network

The selected use cases have been the base for determining which information needs to be mapped from the DTM to the OPC UA information model. This mapping not only requires a transformation of data types, but

also the different concepts for device handling have to be considered. These differences are the reason, why certain information in OPC UA for Devices is provided in a different context than it is provided in FDT. For instance in FDT2.0 the device identity (e.g. represented by a serial number) is provided as online data, while in OPC UA the serial number is provided as part of the type information (see Figure 4).
After the definition of the use cases and the specification of the mapping the working group did some prototyping.
The prototyping has been organized into two phases. In the first phase a prototyping of the data types has been provided, in the second phase a full prototyping has been executed.
For the prototyping of the data types a complete information model for the OPC UA Server was generated. The FDT-specific data types have been implemented completely and were instantiated with simulated values. This means some device types (for communication adapter, gateway and field device) together with the respective device instance nodes were represented in the OPC UA information model. A simple device topology was implemented. – With that simulation it was possible to test the selected use cases completely with the OPC UA interface.
In the second step a Frame Application component was added to the OPC UA Server including an FDT project. The data representing the device types, the device topology, the device instance nodes and the respective data was based on information from the DTMs and the FDT project. Devices were represented with offline as well as with online data. Method execution could be tested and the retrieval of documents was implemented.
The combination of standards can enable new possibilities. In OPC UA a security model is part of the architecture. This allows establishing a secure connection between the OPC UA Client and the server (FDT frame application). The information model of the OPC UA server can be built automatically from the FDT project. Through the mapping of the FDT model to the OPC UA information model the existing fieldbus systems can be integrated into the future OPC UA eco-system. The information provided to OPC UA clients is standardized and common for all devices. It is independent from the fieldbus and the device vendor. With the nested communication of FDT fieldbus hierarchies can be covered. CANopen as a well known player in the automation field could be part of this journey. For example an asset management system using OPC UA would be able to get the needed information of CANopen devices (e.g. device type, vendor, etc.).

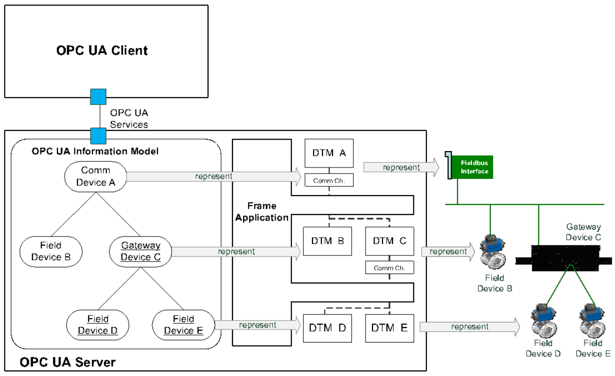

Figure 3: Architecture with OPC UA

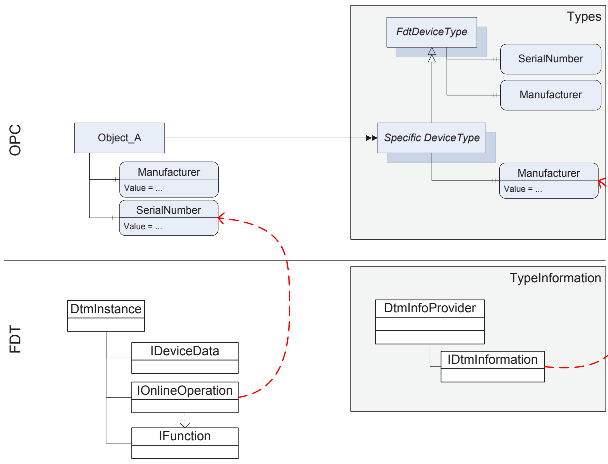

Fig. 4: Mapping of device type information

Mobile devices for device management

Beside OPC UA the use of mobile devices in the environment of Industie 4.0 and Ilot is another growing requirement. Today the FDT technology is based on .NET. For the graphical user interfaces WPF and/or Winforms controls are used. The FDT frame applications are running on desktop or notebook systems depending on the type of frame application (e.g. stand-alone tools are often used on a notebook).
To cover the needs of using mobile devices on the plant floor the FDT Group has started a new working group to specify the necessary enhancement of the current FDT specification.

The solution to be specified should fulfill the following requirements:

- target users are doing commissioning and maintenance activities (e.g. no project building but ability to navigate a project)
- nested communication should work as it does today
- all DTM functions and graphics should be possible
- support of different mobile operating systems (Windows, Android, iOS)
- existing DTM business logic should remain unchanged

A main objective of this solution is of course to manage different devices in a common and standardized way. Currently many proprietary solutions are entering the market creating an app “zoo” for the end user. Each vendor has a specific solution which could be avoided with the FDT solution.

Figure 5 shows a small scenario how a CANopen device could be managed with a mobile device. As the device itself has no wireless connection the FDT client is connected to the FDT application server, where the DTM business logic is running. From there via nested communication the data exchange to the device takes place.

The current FDT architecture is already client/server based. This architecture allows the separation of the DTM user interface and the DTM business logic as shown in figure 6. But this implies a rich client and the communication between client and server is not specified and potentially a proprietary implementation of the frame vendor.

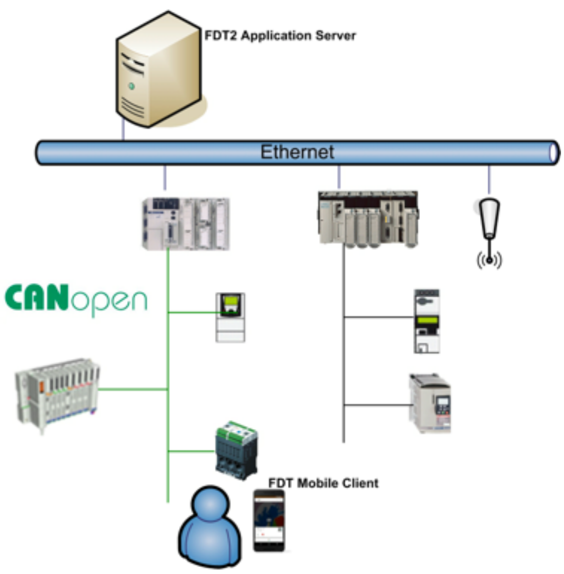

Figure 5: FDT mobility scenario

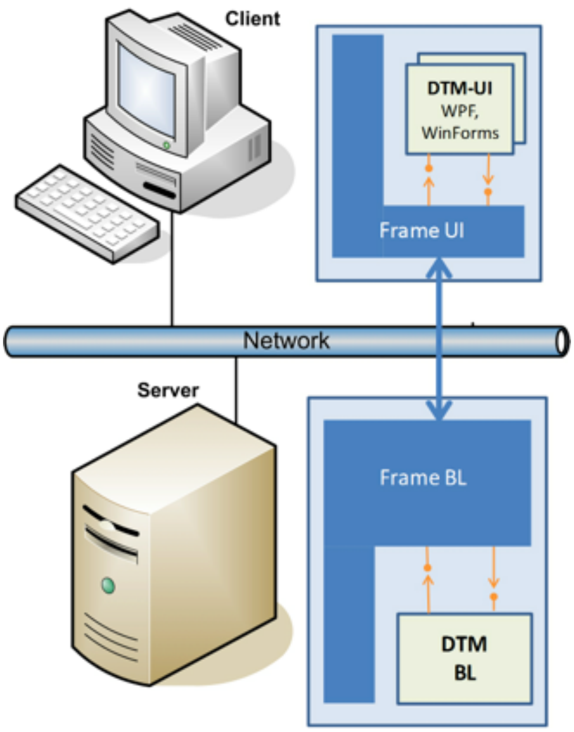

Figure 6: FDT client/server architecture

This architecture needs some extension to cover the mobility scenario. A final solution is not yet defined. Currently different types of clients are under discussion.

Figure 7 shows the potential architecture with the different client types and the extensions on the FDT server. The frame business logic is amended with a web server. It provides the external interface to the mobile clients.

The thin client is using a browser to connect to the FDT server. An end user would be able to start with a mobile device which do not need any specific software. The browser would get a frame UI which is hosted on the FDT server in the component Frame Mobile UI Provider. With the selection of a device the respective DTM mobile UI would be loaded on the client. This is hosted on the FDT server by the component DTM Mobile UI Provider. This component has an interface to the DTM business logic to manage the data exchange between the mobile UI and the business logic. This interface is specific to the device vendor and needs not to be standardized. A similar interface exists between the frame business and the Frame Mobile UI Provider.

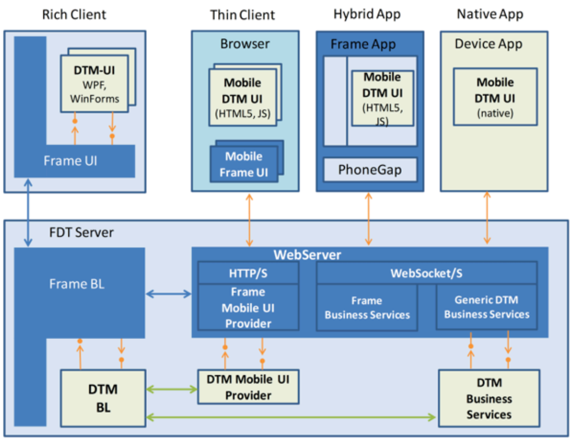

Figure 7: FDT mobile architecture

Another type of client could be a hybrid app. Hybrid apps are cross-platform apps which are developed with frameworks like PhoneGap. The objective of such frameworks is to develop one app which can be used on the different mobile operating systems. The client would have a frame app which will render the mobile UI from the DTM. For the communication between client and server the WebSocket protocol makes a bi-directional channel available. This allows more interaction between client and server. The server can send data to client without the need of being polled by the client. Depending on the outcome of the final architecture it might be useful to provide additional services from the frame and/or the DTMs. For this reason the components Frame Business Services and Generic DTM Business Services are shown in the architecture. The latter one is seen as common for all DTMs. For DTM specific services the component DTM Business Services is the placeholder. It has a proprietary interface to the DTM business logic. The same holds for the interface between the Frame Business Services and the frame business logic.

The third version of a client could be a native app using the communication mechanism and the components mentioned before to exchange data with the DTM business logic. The targeted user of this app will be most probably not an end user as this app depends on the mobile device operating system and the handling cannot be standardized as the other client types. It might be developed for special use for the vendors personnel.

Before the FDT specification amendment for mobility is written an intensive prototyping will be done. Here different aspects will be taken into account like user authentication, web service concepts, performance, usability, etc.

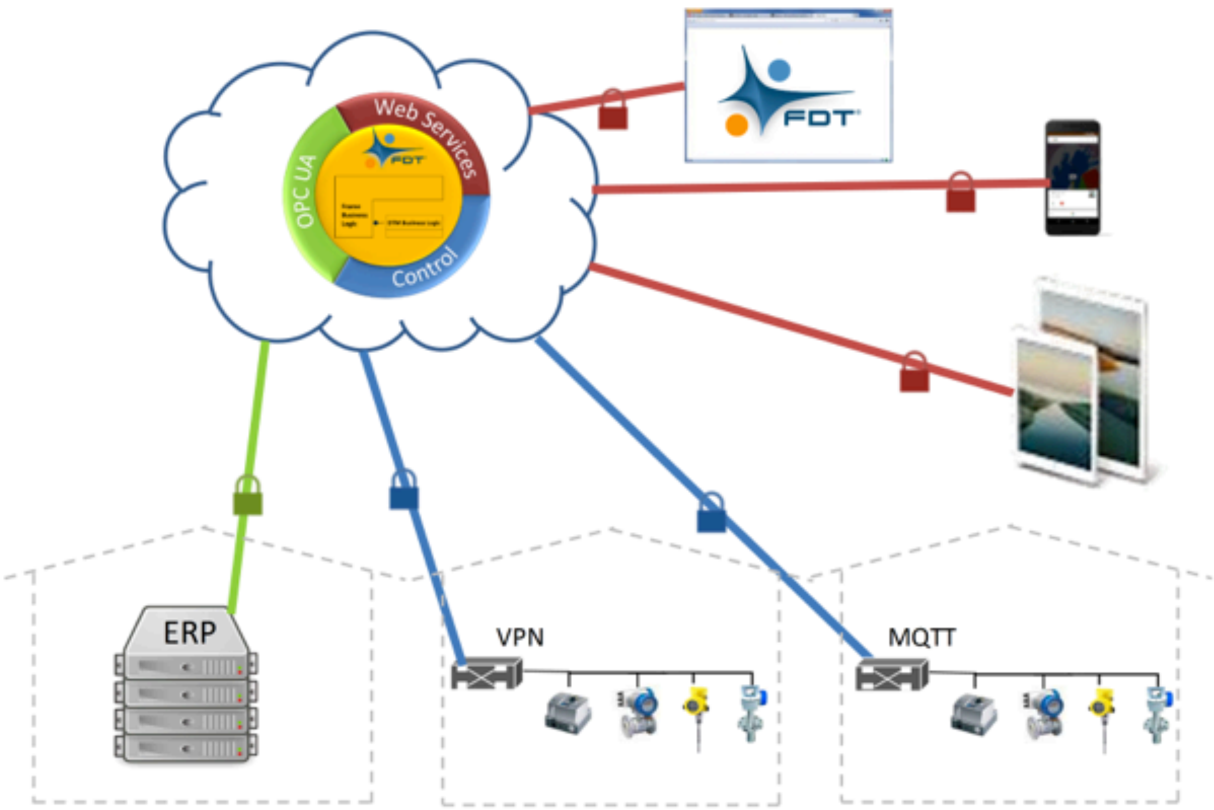

Figure 8: FDT mobile architecture

Other aspects of mobile devices are the touchability and different screen sizes. The presentation of functions and data will therefore be various. The existing DTM style guide needs to be extended to guide DTM UI developers and achieve as much as possible a standardized user experience. With this outlined concept FDT provides a standardized device management using mobile devices and it allows to integrate existing field devices (e.g. CANopen).

Outlook

Putting together all the pieces (access to control networks, OPC UA and mobility) the so-called FDT IIoT Server (FITSTM) can be created and is the key component for the further development of the FDT technology. Figure 8 summarizes the manifold connections which are supported by the server. The center part of the server contains the frame business logic and the business logic of the DTMs. The OPC UA interface provides asset information of the connected control networks and devices to MES or ERP systems. The web interface provides new possibilities like browser based access and web services. The connection to the control networks are traditional direct connections and connections using new protocols like MQTT to an Edge device or VPN to a firewall. The open architecture of FDT allows adding networks to meet changing requirements. A Gateway DTM e.g. for the Edge device would be needed to connect to the control network. All this is based on nested communication.
To leverage all these prospects to its members the FDT Group plans to provide so-called common components which implement the standardized parts of the architecture.

References

www.fdtgroup.org
OPC UA for devices, OPC Foundation 2013
OPC UA Information model for FDT, 2017
FDT2.0 technical specification 2012

Manfred Brill
Schneider Electric Automation GmbH
Schneiderplatz 1
DE-97828 Marktheidenfeld
Tel.: +49-9391-606-2273
Fax: +49-9391-606-4001
manfred.brill@schneider-electric.com
www.schneider-electric.com