No.6

EIB's Object Interworking Standards

Your key to Distributed Application development.

1 *Ask any system integrator: Interworking is the magic which makes an installation tick in a multi-vendor environment typical for open systems. At EIBA, we have understood this challenge since the very beginning and made it into a cornerstone of our technology. Find out how to put this experience to your advantage.*

Interworking - EIB's Open System Commitment

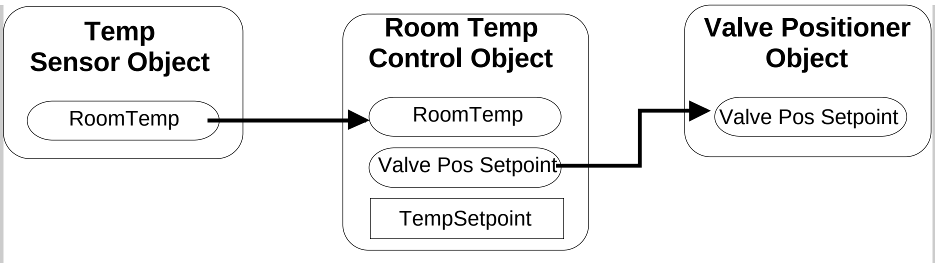
Answering all automation challenges for residential and commercial buildings, EIB is the first solution for Home & Building Electronic (HBE) networks. EIB's decentralised, open network technology is the choice of more than 100 leading companies from industry and engineering. Under various brand names, they market certified, EIB compatible equipment for:

- electrical installation,
- measurement and control,
- heating, ventilation and air conditioning (HVAC),
- security and alarms,
- household appliances, etc.

Originally launched on Twisted Pair communication, EIB now supports all relevant fieldbus media including Powerline and Radio Frequency. EIB.net introduces automation-level capability based on Ethernet. Since 1991, EIB has pioneered the concept of operational run-time Interworking between components of Distributed Applications through its EIB Interworking Standards (EIS) standardisation effort, backed up by a meaningful (i.e. not merely "token") certification of implementations. The EIB Object Interworking Standards (ObIS) extend the EIS data format specifications with object-based models for Distributed Applications and their components.

The Age of Distributed Applications

Developing for a peer-to-peer distributed automation network such as EIB, calls for an entirely different approach to application design. Rather than using the automation network to emulate master-slave systems, design must optimise efficiency and flexibility by creating truly distributed applications with locally autonomous components.



A simple Object Application Model showing objects and their properties, which can be configured for run-time linking via group addresses, or set as local parameters.

Another keyword is data orientation: focusing on information (data), modern automation avoids rigid dictionaries and command languages. EIB's multicast "group addressing" scheme is particularly suited to the data-centred view.

The resulting "weak typing" allows for multi-purpose solutions. A presence detector which can both control the lights during day and trigger an intruder alarm after office hours certainly helps to reduce the combinatorial overhead for development and logistics. Combined with powerful project engineering tools such as the EIB

Association's ETS, this brings unprecedented freedom and control to project or installation designers.

In an open system, the balance between common standards and a manufacturer's need for a distinct individual profile can sometimes be delicate. EIB's Object Interworking Standards provide the necessary binding force to bring and hold all these parts together.

The EIB Association provides the ideal framework to find this balance: all EIB Interworking Standards are forged in close cooperation with domain experts from industry and engineering.

Beyond Interfaces: EIB's Application Object Models

Heating. Room temperature control. Valve positioner. Set value. This example nicely illustrates the structure of Distributed Applications as reflected in the hierarchy of EIB's Object Interworking Standards.

Application domains cover some particular domain such as *Heating (Control)*; the system is not aware of the entire domain, so this is a somewhat informal ("meta"-)concept which typically consists of several Distributed Applications like *Room Temperature Control*.

The EIB Object Interworking Standards capture a Distributed Application in an Application Object Model (AOM). The AOM shows how the required functionality is distributed, and how the resulting objects are linked.

Each of these objects is specified individually as an Object Type Definition (OTD). First of all, the OTD lists the individual properties which together build the object. These actually constitute the communication and control interface for the corresponding function of the local application.

The communication interface as such is quite meaningless without at least a basic "commitment" concerning the actual functionality behind it, on the part of the component which implements it. This explains why things such as state machines appear in the description. They specify the behaviour of one component, reacting on an event in the Application Object Model.

Various Application Object Models have been defined, for example for Security Alarm systems and Demand-driven Forward Flow Temperature Control.

Precise algorithms and their dedicated specific parameters are left open: this is where the manufacturer's domain and implementation know-how come into play.

Data Types: at the Basis.

Finally, the Property Type Definitions (PTD) define the data formats for the individual properties. (Remember that a property may also be "published" as a shared variable, so that it becomes accessible through group addressing.)

Among the properties defined for EIB, we find:

- Boolean
- (un)signed short integer
- (un)signed long integer
- short float
- IEEE float
- date
- time
- 4-bit control
- multi-state control
- etc.

Specific identifiers are defined for all relevant physical values.

Summarising, this leads to the following hierarchy of EIB Interworking concepts:

- Object Application Models
- Object Types
- Property Types

The Advantage

With the newest generation of open HBE systems, multi-vendor installations have become the norm, rather than the exception. EIB's ObIS ensure seamless integration across and importantly, also within application domains. This opens interesting opportunities for niche vendors who can concentrate on their core know-how.

More than 5 years of experience with industrial-quality implementations have even shown that the combinations of interfaces and their functional descriptions are a welcome guideline for development.

The resulting distributed applications lead to a degree of modularity which is highly appreciated by the installation professional in the field.

Finally, it is widely known that via the ETS (EIB Tool Software), the EIB system is "tool enabled" for custom installation configuration. The EIB Object Interworking Standards add a new dimension: tool support for novel, high-level project design - based on linking Objects or even entire Application Models in one go. Supervision and control tools similarly gain smooth access to ObIS-based installations.