Communication Systems for Building Automation and Control

WOLFGANG KASTNER, GEORG NEUGSCHWANDTNER, STEFAN SOUCEK, AND H. MICHAEL NEWMAN

Invited Paper

Building automation systems (BAS) provide automatic control of the conditions of indoor environments. The historical root and still core domain of BAS is the automation of heating, ventilation and air-conditioning systems in large functional buildings. Their primary goal is to realize significant savings in energy and reduce cost. Yet the reach of BAS has extended to include information from all kinds of building systems, working toward the goal of "intelligent buildings." Since these systems are diverse by tradition, integration issues are of particular importance. When compared with the field of industrial automation, building automation exhibits specific, differing characteristics. The present paper introduces the task of building automation and the systems and communications infrastructure necessary to address it. Basic requirements are covered as well as standard application models and typical services. An overview of relevant standards is given, including BACnet, LonWorks and EIB/KNX as open systems of key significance in the building automation domain.

Keywords*—Automation, building management systems, distributed control, field buses, networks, standards.*

I. INTRODUCTION

Home and building automation systems are—in the broadest sense—concerned with improving interaction with and between devices typically found in an indoor habitat. As such, they provide a topic with many facets and range from small networks with only a handful of devices to very large installations with thousands of devices. This paper, however, narrows its focus on the automation of large functional buildings, which in the following will be referred to as "buildings" for simplicity. Examples include office buildings, hospitals, warehouses, or department stores as well as large distributed complexes of smaller installations such as retail chains or gas stations. These types of buildings are especially interesting since their size, scale, and complexity hold considerable potential for optimization, but also challenges.

The key driver of the building automation market is the promise of increased user comfort at reduced operation cost. To this end, *building automation systems (BAS)* make use of optimized control schemes for heating, ventilation, and air-conditioning (HVAC) systems, lighting, and shading. Improvements in energy efficiency will also contribute to environmental protection. For this reason, related regulations sometimes mandate the use of BAS.

Costs can further be reduced by providing access to all building service systems in a centralized monitoring and control center. This allows abnormal or faulty conditions to be detected, localized and corrected at an early stage and with minimum personnel effort. This is especially true when access to the site is offered through a remote connection. A unified visualization scheme for all systems further eases the task of the operator. Direct access to BAS data from the corporate management level eases data acquisition for facility management tasks such as cost allocation and accounting.

Besides the immediate savings, indirect benefits may be expected due to higher expected workforce productivity or by the increased perceived value of the automated building (the "prestige factor," for both building owner and tenant).

Although investment in building automation systems will result in higher construction cost, their use is mostly economically feasible as soon as the entire building life cycle is considered. Typically, the operational cost of a building over its lifetime is about seven times the initial investment for construction. Therefore, it is important to choose a building concept that ensures optimal life-cycle cost, not minimum investment cost. The considerable number of available performance contracting offers strongly emphasizes that advanced BAS are indeed economical. In these models, the contractor

Manuscript received January 19, 2005; revised March 17, 2005.

W. Kastner and G. Neugschwandtner are with the Automation Systems Group, Institute of Automation, Vienna University of Technology, Vienna 1040, Austria (e-mail: k@auto.tuwien.ac.at; gn@auto.tuwien.ac.at).

S. Soucek is with LOYTEC Electronics GmbH, Vienna 1080, Austria (e-mail: soucek@ieee.org).

H. M. Newman is with the Utilities Computer Section, Cornell University, Ithaca, NY 14850 USA (e-mail: hmn2@cornell.edu).

Digital Object Identifier 10.1109/JPROC.2005.849726

takes the financial risk that prospective savings will offset the investment within a given time.

Benefits both in terms of (life-cycle) cost and functionality will be maximized as more systems are combined. This requires that expertise from different fields is brought together. Integrating fire alarm and security functions is particularly challenging due to the high demands made on their dependability. Engineers and consultants who used to work separately are forced to collaborate with each other and the design engineer as a team.

Integration is obviously far easier when systems that shall be joined talk the same language. For example, unified presentation is achieved at no additional engineering effort this way, potentially reducing investment cost. Especially large corporations with hundreds or thousands of establishments spread out over large distances certainly would want to harmonize their building network infrastructure by using a certain standard technology throughout. Yet this goal is effectively out of reach as long as different manufacturers' systems use proprietary communication interfaces, with no manufacturer covering the entire spectrum of applications. Here, open standards try to close the gap, which step into the breach, which moreover help avoid vendor lock-in situations.

In the past years LAN technologies have been pushing down the network hierarchy from the management level while fieldbus technologies are pushing upwards. This battle is still not over but what has already emerged from this rivalry is a new trend of combining fieldbus protocols with LAN technologies to better utilize an existing LAN infrastructure. Most approaches follow the principle of running the upper protocol layers of the fieldbus protocol over the lower layers of a typical LAN protocol such as IP over Ethernet. The synergies arising out of this very attractive combination are manifold.

For example, most corporations have established their own Intranet and are now able to leverage this infrastructure for managing their buildings. Still, all the device profiles developed with great effort over many years can be reused. Also, technicians trained on particular tools for many years do not find their existing knowledge rendered worthless despite the switch to IP-based building automation networks. IP-based communication also opens up new dimensions in remote management and remote maintenance.

II. Building Services, Automation and Integration

Building automation (BA) is concerned with the control of building services. Its historical roots are in the automatic control of HVAC systems, which have been subject to automation since the early 20th century. The domain of indoor climate control still is the main focus of this discipline due to its key role in making buildings a comfortable environment.

Initially, controllers were based on pneumatics. These were replaced by electric and analog electronic circuits. Finally, microprocessors were included in the control loop. This concept was called *direct digital control (DDC)*, a term which is still widely used for programmable logic controllers (PLCs) intended for building automation purposes. The DDC concept and its associated design methodology is, e.g., covered in [1].

The oil price shock of the early 1970s[1] triggered interest in the energy savings potential of automated systems, whereas only comfort criteria had been considered before. As a consequence, the term "energy management system" (EMS) appeared, which highlights automation functionality related to power-saving operation, like optimum start and stop control.[2]

Further, supervisory control and data acquisition (SCADA) systems for buildings, referred to as central control and monitoring systems (CCMS), were introduced. They extended the operator's reach from having to handle each piece of equipment locally over a whole building or complex, allowing the detection of abnormal conditions without being on-site. Besides environmental parameters, such conditions include technical alarms indicating the need for repair and maintenance.

Also, the service of accumulating historical operational data was added. This aids in assessing the cost of operation and in scheduling maintenance. Trend logs provide valuable information for improving control strategies as well. Often, BA systems with these capabilities were referred to as building management systems (BMS).

Other building service systems benefit from automation as well. For example, demand control of lighting systems can significantly contribute to energy saving. Recognizing the head start of the BA systems of the HVAC domain with regard to control and presentation, they provided the natural base for the successive integration of other systems (sometimes then termed "integrated BMS" (IBMS) [2]).

Today's comprehensive automation systems generally go by the all-encompassing name of BAS, although EMS, building EMS (BEMS), and BMS/IBMS are still in use, sometimes intentionally to refer to specific functional aspects, but often by habit. Fig. 1 illustrates these different dimensions. The relevant international standard [3] chooses *building automation and control systems (BACS)* as an umbrella term.

Comprehensive automation is instrumental to the demands of an *intelligent building*. This buzzword has been associated with various concepts over the past 25 years ([4] provides a comprehensive review). Although there is still no canonical definition, the current notion of intelligent buildings targets the demands of users and investors alike. Buildings should provide a productive and attractive environment to users while maintaining cost efficiency to maximize the investors' revenue over the whole life cycle. This specifically includes management issues. As facility management has to become more efficient, BAS services have to be tightly integrated into office and workflow automation. As an example, consider conference rooms to be air conditioned only (and

[1]During the 1973 oil crisis, an embargo policy by the Organization of Petroleum Exporting Countries made world oil prices quadruple for a five-month period, then settle at a 10% increased level.

[2]Automatic shutdown of air-conditioning equipment during nonoffice hours, but with start and stop times adjusted for system inertia (start earlier, stop sooner).

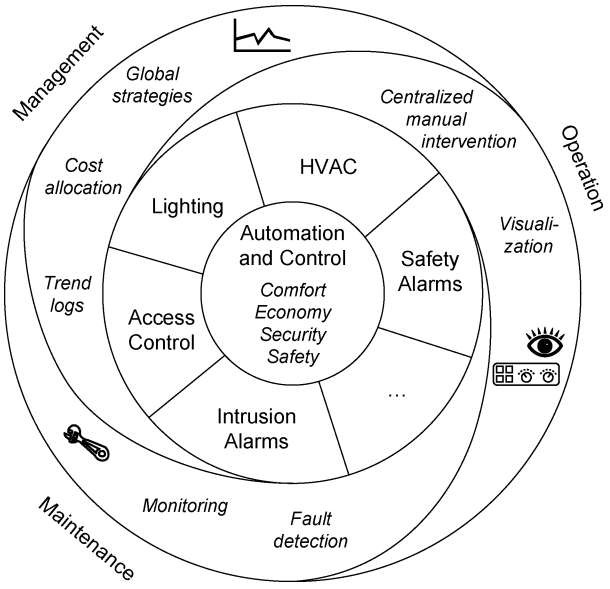

Fig. 1. Functional aspects of BAS.

automatically) when booked. Also, hotel management systems can automatically adjust HVAC operation depending on whether a room is currently rented or vacant. Cost allocation for climate control and lighting with live metering data from the BAS and optimum scheduling of preventive (or even predictive) maintenance based on automatic equipment monitoring and service hour metering are possible as well.

Other dimensions of intelligent buildings are advanced infrastructures for data communication and information sharing to promote productivity, but also advanced structural design and innovative materials. For example, [5] mentions systems to improve the response of a building to earthquakes. Intelligent buildings are also expected to easily adapt to changing user requirements. Recent approaches even include the demand for them to automatically learn the behavior of the tenants and adjust the system performance accordingly.

A. Building Services

Buildings should provide supportive conditions for people to work and relax. This means they will usually be tuned toward human comfort parameters (comfort HVAC). Sometimes, zones or entire buildings are optimized for the particular demands of machines, processes, or goods, which may differ from human comfort (industrial HVAC). In any case, the environment needs to be safe, secure and provide the necessary infrastructure, including supply/disposal, communication/data exchange and transportation. These requirements vary significantly depending on the purpose of the building.

Buildings fulfill these demands through appropriate design of building structure and technical infrastructure, the latter being known as *building services*. For example, ventilation can be achieved through opening windows (a structural design measure) or forced ventilation (a mechanical building service).

Building services include elements usually perceived as passive technical infrastructure (such as fresh and waste water management and power distribution) as well as controllable, "active" systems such as HVAC. The boundary is not clear-cut, however. For example, water supply may include pressurization pumps, and power distribution may be extended with power factor monitoring or on-site cogeneration.

Different building types will have different requirements regarding presence and performance of these services. Table 1 highlights examples, grouped by building disciplines. For a comprehensive reference on building service systems, see, e.g., [6]. The remainder of this section will highlight selected properties of key domains where control is involved to a significant amount.

While the permissible environmental conditions for goods, machinery and processes are usually clearly specified, ensuring human comfort is a more complex affair. For example, thermal comfort does not only depend on air temperature, but also air humidity, air flow, and radiant temperature. Moreover, the level of physical activity and the clothing worn have to be taken into account. One and the same amount of air flow can be perceived as a pleasant breeze as well as a draft depending on thermal sensation. Also, the amount of control available to individuals influences whether they will consider otherwise identical conditions as comfortable or not. This for instance applies to the ability to open windows and having control over air delivery devices [7], [8]. Still, the thermal regulation system of the human body ensures comfort over a certain range of these parameters.

Space heating and cooling can be achieved in different ways. One possibility is to install convectors fed with hot or chilled water. Cooling ceilings are a special form of such convectors. Flow meters and valves are necessary to measure and control the amount of energy distributed. Convection may be fan-assisted, in which case the convector is referred to as a *fan-coil unit* (FCU). The feed water is centrally prepared in boilers and chiller plants. Electric heating elements are often substituted for hot water coils, especially where oil or gas is not available.

When forced ventilation is used, heating and cooling is usually provided with the supply air. In this case, central *air handling units* (AHUs) contain the convector coils (or cooling coil and heating element) together with air filters to remove dust and smoke particles, a humidifier and the necessary dampers and pressure sensors to control the amount of air exchange with the outside. With *variable air volume (VAV) boxes* instead of fixed outlets it is possible to finely control the amount of air released into the conditioned space in addition to its temperature, which allows saving energy.

The amount of air which needs to be exchanged to maintain proper air quality varies with the number of people present. Most frequently, a static value is assumed for smaller rooms and manual intervention is required for larger ones like lecture halls. Nevertheless, air quality sensors (cf. e.g., [9]) are available for automation.

Not all sections of a building can (or need to) be treated equally with respect to environmental conditioning. As an example, for access spaces like stairways, thermal comfort parameters are relaxed in comparison with habitable spaces.

Table 1
Building Service Domains

Climate control	HVAC systems (heating, ventilation, air conditioning) including cooling/refrigeration, humidification, air quality control
Visual comfort	Artificial lighting, daylighting (motorized blinds/shutters)
Safety	Fire alarm, gas alarm, water leak detection, emergency sound system, emergency lighting, CCTV (closed circuit television)
Security	Intrusion alarm, access control, CCTV, audio surveillance
Transportation	Elevators, escalators, conveyor belts
One-way audio	Public address/audio distribution and sound reinforcement systems
Supply and disposal	Power distribution, waste management, fresh water/domestic hot water, waste water
Communication and information exchange	Data networking, PBX (private branch exchange)/intercom, shared WAN (wide area network) access
Sundry special domains	Clock systems, flextime systems, presentation equipment (e. g. video walls), medical gas, pneumatic structure support systems (for airhouses)

Also, the sunlit south side of a building may require different treatment than the one facing north. Therefore, and for reasons of manageability in large complexes, buildings are split into control *zones*. With *room control*, every room forms a zone of its own. Conditions can then be optimized for taste or presence, using presence buttons or detectors.

Good HVAC control strategies can optimize the consumption of primary energy by capitalizing on information about thermal comfort conditions as well as properties of the building structure (e.g., high or low thermal inertia) and systems. Comprehensive sensor data and provisions for fine-grained control also work toward this goal.

Lighting systems fall into two subdomains: artificial lighting, where luminaires are switched and dimmed (by means of load switches, incandescent dimmers, and controllable ballasts) and daylighting. The latter is concerned with limiting the amount of daylight which enters the interior to avoid excessive light intensity and glare. Motorized blinds allow automation of this task. Lighting is traditionally dominated by simple open-loop control relationships in response to manual switches. Only recently, complexity has increased. Artificial light can be centrally switched off during nonoffice hours, also automatically on a given schedule. In this period, a time-limited mode of operation can be entered. Presence detector devices can be used to automatically turn off the lights in unused rooms. Both luminaires and blinds can be adjusted for the sun position according to the time of day. Advanced daylighting systems follow the sun to adjust mirrors which reflect daylight into interior zones. Also, luminaires and blinds can adapt to sky conditions to yield constant lighting conditions with optimum energy efficiency. Lumen maintenance can be achieved both in an open-loop (using a rooftop daylight detector) or a closed-loop manner (with lighting sensors placed in the interior). Anemometers and weather vanes allow determining when outside blinds have to be retracted to avoid damage. Recently, electrochromic windows have become available commercially. The translucence of electrochromic glass is continuously adjustable by applying a low voltage.

In *safety and security alarm systems*, no closed control loops exist. Alarm conditions have to be detected and passed on to appropriate receiving instances. This includes local alarms as well as automatically alerting an appropriate intervention force. Precisely distinguishing nonalarm from alarm situations is essential. Example sensors are motion and glass break sensors from the security domain; water sensors for false floors from the property safety domain; and smoke detectors, heat detectors and gas sensors from the life safety domain. Emergency communication can include klaxons or playback of prerecorded evacuation messages. Emergency lighting is also related to this field. Generally, high reliability is required in this domain, the exact requirements depending on the precise application. The requirements are highest for handling life-threatening conditions in the safety domain, most notably fire alarms. Also, no system components can be allowed to fail without being noticed. The inspections necessary to ensure this can be aided by automatic monitoring.

Like BAS, alarm systems gradually have implemented communication capabilities that reduce the cost of installation and operation. Traditionally, sensors had their alarm limits preset in hardware and were daisy-chained into loops. An alarm was triggered whenever a sensor broke the current loop, with the precise location and reason unknown to the system. This technique is still used in smaller systems. More recent systems allow communication with individual sensors, which may provide even more detailed information about the alarm condition this way, for example the gas concentration measured. [10] provides an overview on safety and security system technologies. As a final example, conveying systems are of significant complexity in their own right regarding control. Yet there is no need for modification of most of their parameters (like car speed or light level) in response to daily changes in building use, like it is the case in HVAC. Therefore, control interaction occurs on a high level only. Examples include putting the system into a reduced operation mode during night hours or controlled shutdown in case of a fire alarm. Additionally, signaling equipment on the landings (e.g., hall and direction lanterns) could be accessed through an open interface.

B. System Integration

Building engineering disciplines have evolved separately and are traditionally handled by independent contractors. Consequently, their respective automation systems are still entirely separate in most buildings today. Another good reason for this separation is that few companies currently cover all domains. Yet there are benefits when information

exchange between building systems is possible. For example, window blinds have considerable impact on HVAC control strategy, as incident solar radiation causes an increase in air temperature as well as in immediate human thermal sensation. Automatically shutting the blinds on the sunlit side of a building can significantly decrease the energy consumption for cooling.

A second area of overlap comprises doors and windows. Their state is of importance to both the HVAC system (to avoid heating or cooling leakage to the outdoor environment) and the security system (to ensure proper intrusion protection at night). The same holds true for motion or presence detectors. Also, motion detectors can provide intrusion detection at night and automatic control of lights during business hours. Such common use of sensors in multiple control domains can reduce investment and operational cost. On the other hand, it increases the complexity as different contractors needs to handle the functional overlap in their engineering systems.

As an important step, building control systems also need to accept control information from systems which are more closely related to the information technology (IT) world. This especially concerns access control systems. Data exchange is not limited to "pass/do not pass" signals sent to doors by RFID readers, card readers, biometric authentication devices, or simple key controls. Increasingly, scenarios such as lighting a pathway through the building and controlling elevators based on card access control at the gate are requested.

As a future prospect, data of multiple sensors may be fused for additional benefit. As an example, consider using the data provided by indoor air quality sensors for presence detection [9]. Control information can also be derived from CCTV imagery through image processing techniques. For example, presence detection and people counting for better HVAC or elevator control can be achieved this way. As another benefit, the state of doors and windows can be detected.

Yet, in all cases, the benefits reached by tighter integration come with a drawback. In an integrated system, examining groups of functionality in an isolated manner becomes more difficult. This introduces additional challenges in fault analysis and debugging as well as functionality assessment. Additionally, if multiple contractors are working on a single integrated system, problems in determining liability may arise.

The assessment problem is of special concern where life safety is involved. For this reason, fire alarm systems traditionally have been kept completely separate from other building control systems. Although a considerable degree of integration has been achieved in some projects, building codes still often explicitly disallow BAS to assume the function of life safety systems. This of course does not extend to less critical property safety alarms (e.g., water leakage detection). Similar considerations apply to building security systems.

These issues need to be addressed by carefully selecting the points of interaction between the different subsystems, with the general goal of making the flow of control traceable. First, this requires limiting the number of such points to the amount absolutely necessary to achieve a given task. Second, interfaces have to be defined clearly to ensure that no repercussive influence is possible. This may necessitate special measures to limit the direction of the control flow (dry contacts and the 4–20 mA interface remain classic examples). Third, points of contact have to be selected in a way that reasonably self-contained subsystems emerge when the links between them are cut. Such divisions may be vertical (e.g., separation into functional domains) as well as horizontal (e.g., a building wing).

Considerable benefits can already be achieved by establishing a highly limited number of interaction points at the highest system level. One prime example is that elevators only need the information that an evacuation condition is present—a single bit transfer—to be able to automatically stop loaded elevator cabins at the next floor level and shut down in case of a fire alarm. Integration at the device level, however, such as in the examples presented above, introduces a level of complexity that still remains a challenge to be handled.

It was stated above that the number of interaction points should be limited to the necessary minimum. While this is correct, it is also necessary to keep the system design flexible enough for future integration requirements. Since building installations are long-lived, system evolution is an important issue. A rigid system that solely satisfies the demands identified at design time often makes future extensions or tighter integration impossible.

C. Automation and Control

Building automation can be regarded as a special case of process automation, with the process being the building indoor environment (and its closer surroundings).[3] The process consists of numerous subprocesses, both discrete and continuous. The most complex processes by far[4] are present in the HVAC domain. Since HVAC processes involve large (thermal) capacities, changes in system parameters occur only gradually. Quick transients typically only have to be detected when optimizing system behavior. Since the process behavior is slow, requirements on controller response times are relaxed compared to industrial control applications. Despite the general absence of high-speed control loops, HVAC control is not without challenges. It has to deal with disturbances, which change over time as a function of load, weather conditions, and building occupancy. These influences are of stochastic nature and therefore not exactly predictable, although certain assumptions can be made. A comprehensive introduction to HVAC control is, e.g., provided in [11].

Closed-loop control is barely present in other building systems. Interestingly enough, timing constraints are tightest in certain open-loop control relations (most notably simple light control functions), where the response time is put in relation

[3]Although some applications, such as shading, will actually involve outdoor sensors and actuators, environmental conditions will typically only be controlled in the interior.

[4]At least concerning those controlled by present-day systems.

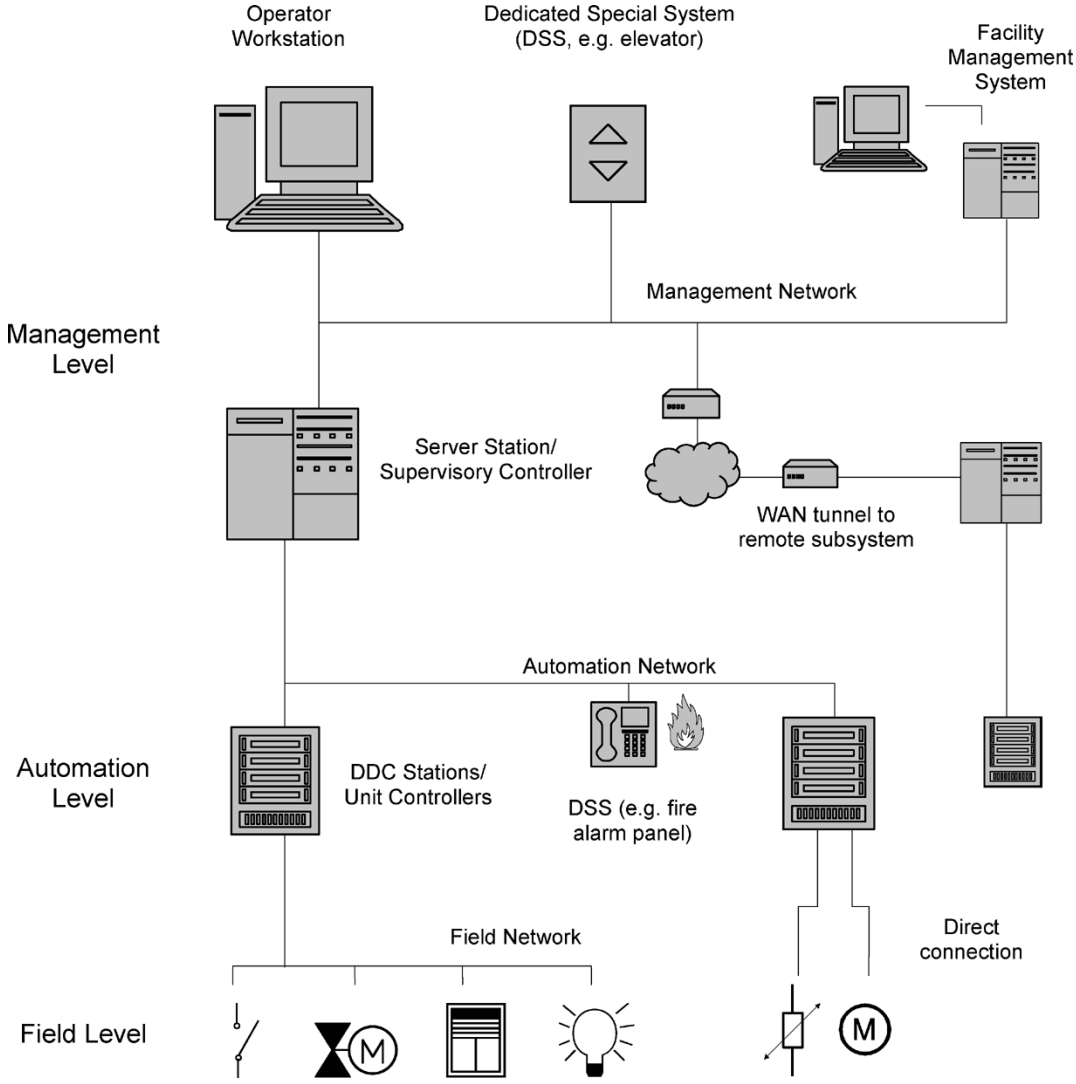

Fig. 2. Building automation, three-level functional hierarchy.

with the human perception time in the range of a few hundreds of milliseconds.

Regarding the required reliability (defined as the probability of a system to perform as designed) and availability (defined as the degree to which a system is operable at any given point in time), basic functions of automated systems have to measure up against conventional installations. As with timing constraints, demands are moderate, since the consequences for failing to meet them are merely annoying in the vast majority of cases. Exceptions do exist however, most notably in industrial HVAC (e.g., refrigerated warehouses) and medical applications. Dependable operation is also required when the integration of safety and security functions is desired.

Definitely, one key challenge in BAS is that large areas need to be covered especially in high-rise buildings or larger building complexes. Another challenge is that the domain is highly cost sensitive when compared with industrial automation. Also, systems have to be long-lived (at least in comparison with the IT world). They are required to be "future proof," which favors proven, technologically conservative approaches. Hence, the domain is very slow to accept and adopt new technological developments. Bid invitations often require systems to adhere to international standards, which lengthens the innovation cycle due to the delays inherent to such standardization procedures.

Finally, operators will seldom receive intensive training, which is why ease of use and robust operation are of significant importance. This is especially the case for all system components which are meant to be operated by tenants.

D. Automation Hierarchy

A general system model designed to accommodate all kinds of BAS[5] is described in [3]. Key elements are shown in Fig. 2. In this model, aspects of system functionality are broken up into three levels, presenting the incarnation of the automation pyramid for BAS.

At the field level, interaction with the physical world takes place. Environmental data are collected (measurement, counting, metering) and transformed into a representation suitable for transmission and processing. Likewise, parameters of the environment are physically controlled (switching,

[5]One should note, however, that the scope of BAS in this model encompasses HVAC and lighting only.

setting, positioning) in response to commands received from the system.

Automatic control, including all kinds of autonomously executed sequences, is assigned to the automation level. It operates on data prepared by the field level, establishing logical connections and control loops. Processing entities may also communicate values of more global interest to each other, for example the outside temperature or whether night purge is to be activated. This type of process data exchange is referred to as *horizontal* communication. In addition, the automation level prepares (possibly aggregate) values for *vertical* access by the management level. This includes the accumulation of historical data at specified rates (*trending*).

At the management level, information from throughout the entire system is accessible. A unified interface is presented to the operator for manual intervention. Vertical access to automation-level values is provided, including the modification of parameters such as schedules. Alerts are generated for exceptional situations like technical faults or critical conditions. Long-term historical data storage with the possibility to generate reports and statistics is also considered part of this level.

It is evident that the amount of (current and historical) data present for access within a given device increases when ascending through the levels. The task of the field level is a distributed one by nature. Automation is typically handled in a distributed manner as well, with multiple processing units responsible for locally contained (or functionally separate) subprocesses. The benefits of distribution are manifold, such as reducing latencies in control loops, avoiding single points of failure, reducing the risk of performance bottlenecks and allowing for subsystems to be out of service due to failures or scheduled maintenance without affecting other parts. Certainly, distributed systems are harder to design and handle than centralized ones. Yet the increase in complexity for the overall system will be mitigated when "divide and conquer" is applied properly, with the added benefit of the resulting subsystems being more transparent.

A BACS design could choose to actually distribute the functions described above over separate devices. As illustrated in Fig. 2, sensors and actuators are either directly connected to controllers via standard interfaces (like dry contacts, 0–10 V, or 4–20 mA) or by means of a field network. Process control is performed by DDC stations (unit controllers). A server station performs supervisory control, logging and trending for a group of unit controllers (e.g., in the central plant room or a building wing). Supervisory and unit controllers are connected via their own automation network. In addition, dedicated special systems (DSS) can connect at this level. For instance, a fire/security panel could put HVAC unit controllers into smoke extraction mode when a fire alarm is raised on its line. An operator workstation uses the data prepared by the server stations to present the user interface. DSS which are not to be integrated into a tight automatic control scheme can be tied in at this level as well. This can, on the one hand, be done with the goal of achieving single-workstation visualization for all systems. On the other hand, metering and other usage data can be transferred into enterprise-level databases such as computer-aided facility management (CAFM) systems for predictive maintenance and cost allocation. Remote stations are integrated into the management network on demand via a dial-up connection (or other WAN tunnel) when data exchange is required. Alert messages may be forwarded to the operator via cellular short message gateways or electronic mail.

The system architecture of today's BACS, however, seldom coincides so closely with the functional architecture described by the three-level model. For example, visualization software packages usually include soft PLC functionality. This allows leveraging the integration effort spent on integrating diverse systems to offer uniform visualization from a single workstation, which is a standard requirement on many projects. Intelligent field devices—as those connected to a field network—can easily perform simple control functions as well.

A trend toward a flatter hierarchy can be observed. Automation-level functions are being assumed by devices typically associated with the adjacent levels: supervisory control and data aggregation are integrated with management-level functions while continuous control is incorporated in field devices. Still, dedicated controllers will help to address the complexity inherent in larger installations or where special performance requirements exist. Depending on the particular demands and structure of a project, multiple approaches to distributing the necessary functionality are viable.

III. Building Automation and Control Networks

In distributed control applications, there is an inherent need to communicate. Actual and actuating values need to be transferred between sensors, controllers and actuators. As building automation has changed over the years, the exchange of control information did as well.

Pneumatic control systems transmitted information in the form of air pressure levels, typically in the industry-standard 0.2 to 1 bar (3–15 lbf/in^2) pressure range. In electrical and electronic systems, voltage or current levels, e.g., the well-known 4–20 mA interface, served (and still serve) this purpose. However, monitoring and control from a central location can only be achieved for a limited number of values this way. To reduce the amount of cabling necessary, CCMS used matrix multiplexing. Soon, wires were even more efficiently used by data networking.

As a consequence of this otherwise desirable evolution, achieving interoperability between controllers, sensors and actuators by different manufacturers has become a significantly more complex issue than simply setting up value range mappings in an identical way.

This section covers how the characteristics of building automation applications translate into requirements on the underlying networks used for this purpose. This encompasses quality-of-service aspects as well as appropriate services and the standard "point" data model. It also touches aspects of network architecture, integration through gateways and routers, and the topic of open systems.

A. Basic Characteristics

General demands on a building automation system (whether in the traditional sense or as an integrated system) were already discussed in Section II-C. These are immediately related to the requirements on data networks within such a system, which are either instrumental in achieving these objectives or will improve the price/performance ratio in doing so.

Key criteria regarding the required quality-of-service are throughput, timeliness, dependability and security. As for necessary *throughput*, BA applications usually do not generate high traffic load at the field level due to the absence of high-speed control loops. Also, event load from stochastic sources (e.g., light switches) is low. Moreover, the spatial locality of control relationships is high. Still, considerable amounts of traffic can accumulate when data have to be collected in a central location from all over a large system. Data, however, seldom need to be available with full spatial and temporal resolution in real-time at a management workstation.

For example, it can be perfectly acceptable for the state of a luminaire to be updated with the central monitoring application every two minutes. Proper response time to tenants' requests is ensured by the local unit controller. Supervisory controllers can summarize the heating or cooling loads determined by subordinate HVAC zone controllers for the purpose of calculating the necessary amount of primary energy conversion, but still log the information in detail for future operator review. Nevertheless, as a general rule, management and automation level functions are more demanding in terms of network throughput to provide acceptable speeds for larger block data transfers like trend logs or DDC program files.

The previous example already hints at the fact that *timeliness* is of different concern for the three layers. Actually, real-time data is only exchanged on the field and automation level. Here, moderate requirements apply to all time constraints (periodicity, jitter, response time, freshness/promptness, time coherence; cf. [12]). No special mechanisms (e.g., dynamic scheduling) for handling these constraints are necessary. It is sufficient even for more demanding applications to be able to state certain upper bounds on transmission delays.

Dependability (robustness, reliability, availability, safety) translates into the ability of the network to detect transmission errors, recover from any such error or other equipment failure and meet time constraints. Guaranteed performance (still with relaxed timing requirements) is only mandatory for life-safety applications. Loss of control has no catastrophic consequences otherwise. Still, a certain amount of fault tolerance is desirable on the field and automation level in the sense that a single failing unit should not bring down the whole system. As long as these layers remain operational, having management functions unavailable for some time is usually acceptable.

The network should also provide appropriate noise immunity (and not generate unacceptable levels of noise itself). Robustness in this respect is desirable especially at the field level, where cables are laid in the immediate vicinity of the mains wiring. Apart from this, the environment of BACS networks is not particularly noisy, especially in office buildings.

Table 2
Selected Service Requirements and Related Mechanisms in Industrial and Building Automation

Industrial Automation	*Building Automation*
Tight timing requirements	Relaxed timing requirements
Periodic traffic, higher volume	Less regular traffic, lower volume
Graceful degradation seldom useful	Graceful degradation desirable
Polling (time-driven)	Change-of-value (event-driven)
Master/slave	Peer-to-peer

Reviewing these requirements, peer-to-peer, event-driven communication schemes appear well suited to BACS. Medium access control using deterministic Carrier Sense, Multiple Access (CSMA) variants, possibly supporting frame prioritization, will allow efficient use of the "raw" throughput capacity available as well as fulfill timeliness requirements for the lower levels.

This is different when compared to industrial automation, where high-speed control loops favor time-driven master–slave approaches. Also, regarding fault tolerance, the focus typically is on redundant design (if necessary) rather than graceful degradation of functionality as systems need every sensor, actuator or controller to be operational to fulfill their purpose.

Table 2 summarizes the main differences with respect to functions involving real-time data. Management-level operations may use any "office-type" network.

For managing the large scale of BA systems, network protocols need to support hierarchical subdivisions and appropriate address spaces. Larger installations will run into thousands of meters of network span as well as thousands of nodes. Networks should also be able to transparently include wide-area connections, possibly dial-on-demand.

Historically, the level of communications *security* provided by the variety of proprietary, undocumented protocols mostly proved to be appropriate for isolated building automation systems. Nowadays security concerns are increasing rapidly, however. In part, this is due to the fact that more sensitive systems like access control and intrusion alarm systems are being integrated. Moreover, office networks are used to transport automation system data and remote access is standard on present-day systems (as will be discussed in more detail below). Protection against denial-of-service attacks becomes more of an issue as buildings get more dependent on automation systems. In any case, the security focus is on authentication. For example, it is usually not a secret that a door was unlocked; however, only a trusted entity should be able to do so.

Securing connection points for remote access is of particular importance. Since they often allow access to management level functions, attacks on them will have a higher chance of global effect. BACS field networks are exposed to (inside) attack as well, especially when run through publicly accessible spaces.[6] Open media such as wireless and power-line signaling further increase vulnerability, since access to

[6]The case of equipment being located on the premises of the adversary is particularly relevant for the related field of remote metering.

the medium can be gained in an unobtrusive manner. Further, the shift to open systems reduces the knowledge barrier for intruders.[7]

Considering cost, many sensors and actuators (e.g., light switches or controllable breakers) are cheap. Providing them with fieldbus connectivity must not be inappropriately expensive. Costs are also an issue in manpower involved. Therefore, installation and configuration have to be as simple as possible.

Wiring can be significantly simplified when a network supports free topology. One can think of free topology as increasing the stub length in a bus topology until the bus character disappears. The two bus terminators in a bus topology will be replaced by a single bus terminator for the free topology network installed anywhere on the network. Cables should also be easy to run through ducts. Supplying power to the nodes over the network cable (also known as *link power*) both saves additional power wires and allows compact, inexpensive power supplies. For inaccessible or hazardous areas or special aesthetical requirements, wireless technologies are deployed. Wireless access is also interesting for management functions like log file access for service technicians or presenting user interfaces to tenants on their personal mobile devices. Retrofit applications will also profit from the ability to use power-line communication.

B. Application Model and Services

In the distributed system constituted by a building automation and control application, a number of nodes (sensors, actuators and controllers) are connected over a network and communicate through a certain protocol. The data transported are values from the sensors, which are processed and sent to the actuators (horizontal communication). In addition some nodes also send data directly to actuators (e.g., a manual override or set point change from an operator workstation), or only consume data from sensors (e.g., for trend logging; vertical communication).

To the application developer the network represents itself as a set of elementary data elements, called *data points* (or simply *points*). These data points are the logical representation of the underlying physical process, which control network nodes drive or measure. Each node can be associated with one or more data points. In the logical view each data point represents a single datum of the application. It can correspond to an aspect of the real world (such as a certain room temperature or the state of a switch) or be of more abstract nature (e.g., a temperature set point).

The data points are connected through a directed graph, distinguishing output points and input points. The application is defined by this graph and a set of processing rules describing the interactions caused by the change of a point value. The logical links which this graph defines can be entirely different from the physical connections between the nodes.

The main characteristic of a data point is its present value. How the digital value is represented is determined by the basic point type, such as integer, floating point, Boolean or enumeration types. To further qualify their value, data points are associated with additional meta data (attributes), which are important in the context of the control application.

A unit attribute adds a semantic meaning to the present value by describing the engineering unit of the value. This attribute is often implied by a certain complex point type defined for a specific application, such as "Temperature." These type attributes are often used to ensure compatible connections between data points.

A precision attribute specifies the smallest increment that can be represented. Attributes such as minimum value, maximum value, and resolution may describe the observable value range of the data point more precisely. The resolution can be the actual resolution of the physical sensor and may be less than the precision.

A key attribute is the location of a point, which is often correlated to a name. Building planners may design the point name space according to geographical aspects, such as building, floor or room and/or according to functional domain aspects, such as air conditioning or heating. The name space hierarchy need not correspond with the network topology (although it often does, especially with a geographic hierarchy). An example pattern is Facility/System/Point, e.g., "Depot/Chiller1/FlowTemperature."

Often, alarm indicator attributes are used. By presetting certain bounds on a data point value the data point can switch from normal mode to alarm mode, e.g., when a temperature limit has been exceeded. This attribute can be persistent so that it can be used to detect alarms also after the value has returned to be in bounds again.

An additional, important concept for data points are point priorities. In building automation applications it is common that multiple output points are associated with a single input point. If the output point values are in conflict with each other the more prioritized one succeeds, e.g., a window contact overrides the air-conditioning thermostat.

Typically, points in the data point graph can be logically grouped to describe specific functions of the system. Such groups forming a coherent subset of the entire application (both data points and the processing rules that belong to them) are referred to as *functional blocks*. While these profiles do not influence the graph as such they allow a functional breakdown of the system and aid in the planning and design process by giving the planner a set of building blocks for the distributed application. Functional blocks can also be grouped to form larger functional blocks.

This concept is illustrated in Fig. 3. The vertices in the graph represent the data points, the thin-line edges network connections and the bold edges processing data flow connections in a field unit. By grouping certain points by their functional relation, the functional blocks FB1 and FB2 are formed. These may or may not coincide with actual physical nodes. At higher levels of abstraction the application engineer may work with aggregates of functional blocks. The aggregate behaves like its own functional entity with the bold

[7]See also the related discussion on standard protocols in Section III-C.

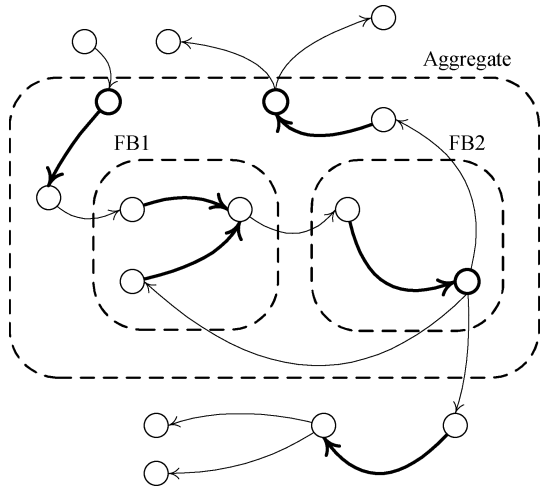

Fig. 3. An example for a data point graph using different functional blocks and aggregation.

vertices being the interfacing data points. Using this technique planners can construct templates of complex functionality and instantiate them multiple times without repeated engineering effort.

This high-level view, which is an accepted standard for BA applications, should be reflected by the data model and services of the network protocol. Data points then serve as the external interface for accessing device functionality. Establishing the communication links for horizontal communication to build the application graph at installation time is known as *binding*.

Since the time characteristics of horizontal traffic are known at the design stage, the application can be subjected to *a priori* performance analysis. [13] discusses how to quantify the amount of such identified data in building automation applications.

As one output data point will often be bound to multiple input data points, horizontal communication benefits from protocol support for multicast relationships. Such support also may facilitate obtaining set coherence (which, for example, can be of interest when switching groups of luminaires). Since the multicast destination groups will be static as well, labeling them with logical identifiers can simplify addressing. This also enables a publisher–subscriber style of communication where producers and consumers of information need not be aware of each other.

Generally, producer–consumer relationships seem most suitable for horizontal communication. For the node application programmer, a shared-variable model is particularly convenient. On every change of a specially designated variable (possibly holding the present value of an output data point), the nodes' system software will automatically initiate the necessary message exchange to propagate the updated value to the appropriate receivers.

When bindings can be defined without changing the node application, the latter can be created independent of its particular use in the overall system. This is especially useful for smart field devices. Unlike DDC stations, their programming is more or less fixed due to resource limitations. Standardization of the functional blocks they represent ("device profiles") is instrumental for enabling interworking between such nodes.

Table 3
Horizontal Versus Vertical Communication

Horizontal communication	*Vertical communication*
Process communication	Management/engineering communication
Publisher-subscriber	Client-server
Multicast (set coherence)	Unicast (engineering: reliable connections)
Static/identified	Ad-hoc, sporadic
Shared variables	Data points (including meta data)

Vertical communication can be divided into services related to accessing and modifying data from within the application, for example adjusting a set point or retrieving trend logs (frequently referred to as *management communication*), and others concerned with modifying the application itself, for example changing binding information or program transfer (*engineering communication*).

While horizontal communication only involves the exchange of present values (or alarm indicators) since a consistent interpretation of their semantics by all communication partners was ensured at setup time, for both management and engineering tasks access to the meta data (descriptive names, units, limits, ...) pertaining to a data point is relevant as well.

Vertical communication typically relates to information stored within a single node, which suggests unicast as the prevalent mode of communication. Engineering communication is supported by the availability of reliable point-to-point connections. Still, broadcasts are needed to support functions like device or service discovery and clock synchronization.

Vertical communication is initiated on-demand, i.e., the communication targets are chosen *ad hoc*. Related services therefore most often follow a client-server model. Table 3 compares the different properties of horizontal and vertical communication.

Data points which need continuous monitoring can be polled cyclically. Additionally, more elaborate protocols provide an event-based mode of communication. In such a model, services exist for clients to subscribe to (and unsubscribe from) change-of-value (COV) notifications, which are generated when selected point values change by a specified amount. Alternatively, notifications may only be generated when the value exceeds or falls below certain limits (coming/going alarms).

For engineering tasks, it is desirable that services are provided which allow devices present on the network to be discovered automatically. They should also be able to provide descriptive information pertaining to the data points (and possibly functional blocks) they provide. Configuration information (e.g., binding information or the device location) should be retrievable as well to minimize dependence on external, possibly inaccurate databases.

In addition to the manual configuration of bindings, system concepts may include support for devices to provide self-binding capability. Usually, the system integrator is responsible that the processing rules associated with data points bound to each other yield a sensible combination. Automatic binding schemes may use standardized identifiers

for particular functional blocks to replace this knowledge. This necessarily reduces flexibility as it requires a stringent high-level application model. As an aside, such self-binding capabilities are a prime example of "vertical" communication between devices of the same stratum, illustrating that the three-level model can be considered a functional classification only.

C. Network Architecture

Although the three-level model from Fig. 2 suggests a matching three-level hierarchical network architecture, strictly implementing this concept is not appropriate in many cases. It was already discussed that devices implement a mix of appropriate functionality from all three levels. Network architectures have to cater to this mix of services and appropriate requirements.

In particular, intelligent field devices incorporating controller functionality render the notion of a separate automation network absurd. A strictly three-tier network would also unnecessarily complicate sharing devices (like sensors in particular) between functional domains.

Still, cost-efficient device networking technologies cannot accommodate the throughput requirements created by log file transfers or central real-time monitoring of numerous event sources. Therefore, a two-tier architecture has become popular where local *control networks* are interconnected by a high-performance network *backbone*.

A typical building network infrastructure consists of independent control networks on every floor, which connect sensors and actuators at the room level. Control networking technologies are geared toward cost-efficient implementation of field-level and automation level tasks where throughput is less an issue than timeliness.

These networks are connected through a backbone channel for central monitoring and control, remote maintenance and diagnostics, which may also span building complexes. Plant networks may use a separate controller network, although DDC stations will often connect to the backbone directly.

While BACS traditionally use dedicated transmission media, most modern buildings are also equipped with structured cabling for office data networking throughout the building. The IT infrastructure has become an integral part in modern buildings. The attempt to leverage this infrastructure for automation purposes is a natural consequence.

Since management level services do not impose any timeliness constraints worth mentioning, office networks will always be able to assume functions of this level.[8] Given the fact that BA applications are not exceedingly demanding in terms of timeliness and reliability, IT technology is also in a position to handle automation level services. Extending the unification process to the field level is still a theoretical possibility though, as cost efficiency, robustness and ease of installation can not yet match dedicated solutions.

It should be noted that adopting "office-standard" technologies need not necessarily mean having office and automation traffic use the same wires. Adopting IT networks for control purposes is actually a three-fold decision. First, one can employ IT technology at the physical and data link layer only, running custom protocols above. In this case, mainly questions of design performance have to be considered. Second, one can choose to adopt standard office networking protocols. This facilitates integration, but already has security implications, as standard protocols and especially their off-the-shelf implementations provide a broader area for attack; the ability to make use of approved and tested security measures is generally considered to offset this disadvantage.[9] Third, control and IT communications can be actually run over the same network. This makes an integrated assessment of network quality of service necessary. They may or may not use the same upper-layer protocols in this case, although adopting standard IT practice will certainly make administration easier.

Today, "IT network" has effectively become a synonym for "IP network." Making use of the associated standard application-related protocols as well holds considerable potential for building integrated systems, including greatly facilitating remote connections via the Internet [14].

Although IP networks cannot fulfill the quality of service requirements of more demanding control applications yet, since delay cannot be fully controlled (cf. [15] for a comprehensive discussion), they are definitely suitable (and also applied in practice) for use as a backbone network in building automation systems. Still, individual control networks should depend on the backbone just as little as unit controllers should on a central station. To provide additional reliability (for example, for safety-related functions), an additional control backbone (possibly using a fiber optic ring network) may be installed in parallel to the common office network backbone.

D. Network Interconnection

Building automation systems may span a variety of different networks, which again may or may not share a common notion of their distributed application (i.e., resource models, services, and namespaces). Discontinuities especially occur when integrating special-purpose systems, no matter whether centralized or distributed.

In the general case, gateways are needed to handle the interconnection. Gateways effectively need to maintain a database of mappings between network entities on either side. This translation does not only introduce considerable engineering effort, but also has to be provided with a multitude of application-related parameters to fill the gaps which will necessarily occur in mapping protocol constructs between both sides. Also, it uses considerable processing power.

Therefore, gateway functionality is usually integrated in nodes which are designed to perform customizable processing anyway. Traditionally, this applies to controllers and server stations (which therefore also handle network transitions in the classic three-tier model).

[8]This is actually part of the design of the automation pyramid.

[9]Still, some of these measures (like continuous software updates) do not translate well into the automation domain.

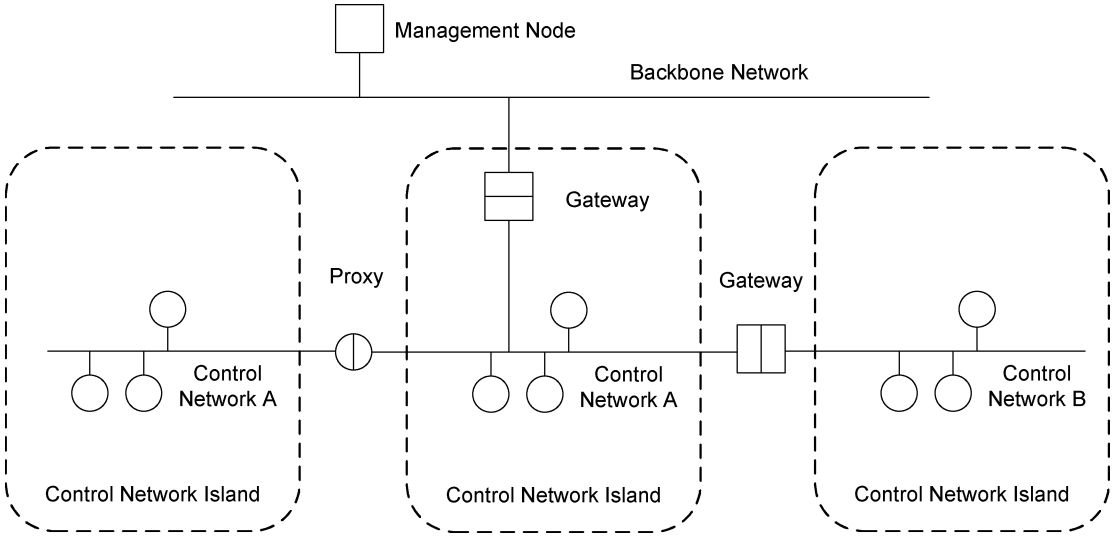

Fig. 4. System with control network islands and horizontal proxy nodes/gateways.

In the two-tier model, gateways are in an ideal position to assume additional tasks as well. At the intersection of control network and backbone, they can for example perform trend logging, thus freeing the backbone network from real-time concerns and taking load off the control network. They could also perform logic control. Dynamic application frameworks such as OSGi [16] allow providing gateways with the necessary flexibility.

With the gateway approach, control applications on every network use their native protocols to communicate with each other, with the gateway establishing the semantic connection. No half needs to deal with any protocol specifics of the other half. All the intricacies of the specific protocol can be abstracted and hidden behind the gateway.

This approach is desirable when applications need to be working across the boundaries of different control network systems and can operate with the common denominator of the present services. In building automation, gateways typically operate on the abstraction level of data points, which represent the common denominator regarding the application model thanks to the real-world orientation of their concept. This is especially the case for data point connectivity during regular operation. The gateway functions needed for this type of integration are limited to a small set of services, such as read value, write value and change-of-value subscription.

Gateways can directly translate between two control networks, providing horizontal connections from data points in one system to data points in another system as depicted in Fig. 4. This is especially appropriate for decentralized control tasks. As an example, consider a lighting system using control network A using information from presence detectors connected to the HVAC system using control network B. This is, however, a less commonly used technique.

More frequently, both control networks use gateways for vertical connection to a third, common standard. This may be the backbone network or also a software platform on a management server. This is more frequently done, since accepted common standards for integration exist. Thus, different control networks only need to provide one mapping to the common standard each instead of multiples to each other to achieve integration. Fig. 5 illustrates this concept.

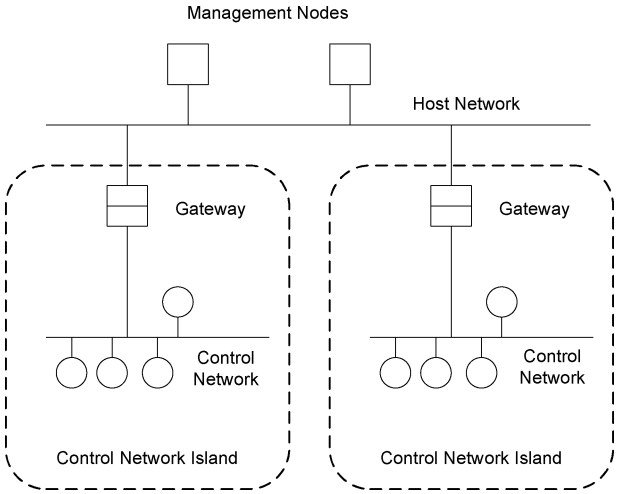

Fig. 5. System with control network islands and a common backbone.

A key limitation of the gateway approach is that mapping all intricacies of a protocol is extremely hard (and thus often a theoretical possibility only). While the data point abstraction will serve as the common denominator for the exchange of process-related data, most engineering services are impossible to translate because these services are usually highly technology-specific. Actually, they typically already require the communication partners on both ends to know the protocol in full detail. The only problem which remains is that the intermediate network does not support those services natively. A beneficial approach in this case is to transfer all protocol layers of the control network over the intermediate (backbone) network. The intermediate network basically functions as a transmission medium for the control network protocol. This method is known as the *tunneling* of a control network protocol over an intermediate network (e.g.,

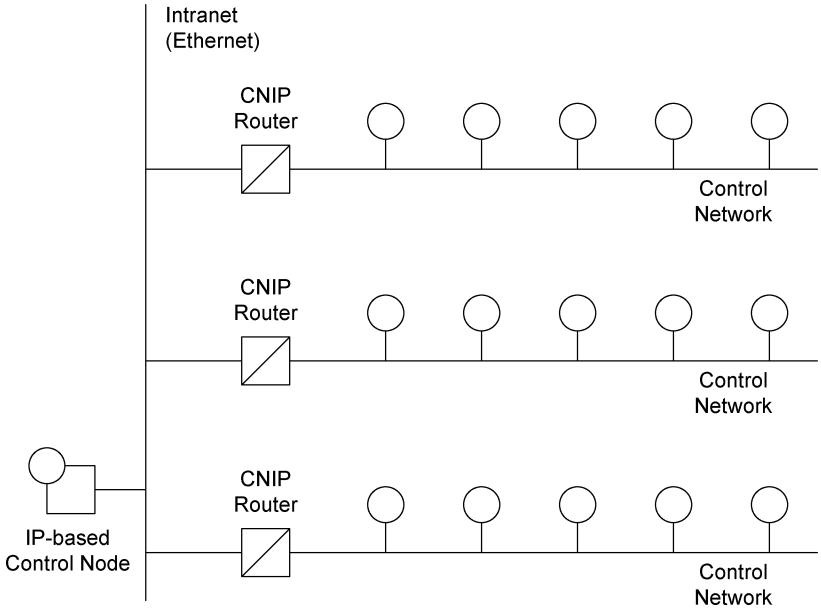

Fig. 6. System with tunneling routers.

an IP backbone). The devices at the boundaries of the intermediate network, which build the ends of the tunnel between two or more control network segments, are called *tunneling routers*.

The main advantage of the tunneling approach is the transparent connection of control nodes over IP networks. This is especially convenient for two purposes. First, separate control network segments may be connected over a higher performance backbone network. Second, for remote administration, a system's native tools can be run on a node on the host network to manage (e.g., commission, monitor, control, visualize) the control network. The software in this case is specifically written for a given control network protocol. In this case, the host network node usually implements the tunneling router functionality itself. Fig. 6 depicts a system with tunneling routers and IP-based control nodes.

Technically, the tunneling approach principally has to overcome the problem that packets on an arbitrary packet-switched network may be reordered, delayed, duplicated, or dropped. A number of techniques have been standardized to address these problems for all field network protocols of relevance in building automation.

With the tunneling approach, control network segments are not decoupled. They have to be considered as a whole both for troubleshooting and functionality assessment. When gateways are used, the coupled systems stay independent. They can be commissioned (and, if necessary, repaired) separately. As an important side note, this independence is sometimes a desirable property (cf. the discussion regarding "loose coupling" in Section II-B). Therefore, even connections between systems with identical network stacks may be established on the application layer in certain cases by *proxy nodes*. A proxy node is also included in Fig. 4.

Gateways and tunneling routers are no panacea, however. For high-level integration, the common semantics of data points suffice for integration. Intelligent buildings and reaping the benefits of sensor synergy, however, demand deeper integration on the device level. Obviously, it is not feasible to integrate complex gateway functionality into every device.

Also, customers want to mix and match components from different vendors to build best-of-breed systems and realize hitherto unattained levels of functionality. Escaping vendor lock-in is especially significant given the fact that BACS have high life expectancy and need to be capable of continued evolution. Not being bound to one original vendor can significantly lower the total cost of ownership.

To achieve this, all aspects of interfacing with a system have to be *open*. Very different notions exist concerning the meaning of the word "open." For the purposes of this discussion, a system technology is considered open if its full specifications are made available to the general public and can be implemented at nondiscriminatory conditions. Such systems can be repaired, modified, and extended by everyone with the necessary basic qualifications without having to rely on the original manufacturer. Unlike gateways, which need only expose data points defined at contract time, open systems are indeed future-proof. Besides the specification of the network stack with its protocols and services, data point attributes and functional profiles have a key role in the specification of open systems whose parts will interwork and interoperate, respectively.

The effort to engineer an open system is still considerable, since many parameters still have to be aligned to achieve interoperability. "Open" does not mean "plug and play"; it merely ensures that interoperability can be achieved without further involving equipment manufacturers. To the end user, a system must always appear homogeneous, no matter how complex the interplay of its components may be.

Therefore, the benefits of open systems are not free. The reduction in lifecycle cost thanks to the flexibility gained, however, is generally considered to offset the initial additional hardware and engineering cost.

IV. Standards Overview

The field of building automation has been dominated by a plethora of proprietary solutions for a long time. Its moderate performance requirements still encourage ad hoc approaches. Yet pushed by market demand for open systems, even market leaders are gradually abandoning proprietary designs.

Official standards bodies ensure that the standards they maintain and publish fulfill the conditions of open systems as outlined, i.e., nondiscriminatory access to specification and licensing. Hence, adherence of equipment to such formal standards is required in an increasing number of tenders. Standards directly related to building automation system technology are created in the United States[10] and in a number of European and international standards bodies.

ISO TC 205[11] (Building Environment Design) is publishing a series of international standards under the general title of Building Automation and Control Systems (BACS). The series includes a generic system model describing hardware [3], functions, applications and project specification/implementation of a BACS (the latter parts still to appear). It also contains the BACnet standard [17] discussed in the following section.

CEN[12] TC 247 (Building Automation, Controls, and Building Management) is responsible for paving the way in European BA protocol standardization through cumulative prestandards of industry-standard protocols for the automation and field level [18], [19] which also included a collection of standardized object types for the field level [20]. TC 247 also made significant contributions to [3].

CENELEC[13] TC 205 (Home and Building Electronic Systems, HBES) oversees the EN 50090 series, a standard for all aspects of HBES tightly coupled to KNX (which will also be presented in the following section). Its scope is the integration of a wide spectrum of control applications and the control and management aspects of other applications in and around homes and buildings, including the gateways to different transmission media and public networks.[14] Moreover it takes into account all matters of EMC and electrical and functional safety.

ISO/IEC JTC1 SC25 WG1[15] (Information Technology, Home Electronic System) focuses on the standardization of control communication within homes. Its work specifically includes residential gateways between the internal Home Electronic System network and external wide-area networks such as the Internet. Despite its focus on the home environment, the work of WG1 may be relevant since it also looks at similar management functions in commercial buildings. This especially concerns [21] for field-level functionality.

A number of standards—closed and open company standards as well as formal ones—further contribute to the overall picture by providing important directions for BACS subsystems. These will be covered in the remainder of this section.

A. Subsystem Solutions

On the management level, IT standards prevail for connectivity, as was already discussed. Application level issues will be covered in the next subsection. On the automation level, EIA-485 is very popular, with many (proprietary) protocol variants on top. The most notable example which also provides a certain degree of openness is Johnson Controls Metasys N2.

Fieldbuses which are well-established in factory and process automation (like Interbus, CAN-based protocols as Devicenet or CANOpen, and Profibus DP[16]) are largely irrelevant in BA, except for occasional use in "plant room network" controller-to-controller communication (specifically including variable frequency drives for fans and pumps).

Although never formally standardized, Modbus can definitely be regarded as an open protocol. This protocol was designed in the late 1970s and is currently supported by most programmable logic controllers (PLCs) in some form. Implementation of the Modbus protocol is license-free, which makes it especially interesting for integration and interfacing between BAS and other systems. It still is supported to some extent by numerous BA controllers, especially for the purpose of HVAC controller-to-controller-communication (e.g., with chillers). Moreover, Modbus is also present in devices belonging to other building disciplines, like electricity meters or fire alarm systems.

The Modbus application layer is basically confined to reading and writing of register values using a simple request/response protocol. This yields a very flexible/versatile application layer, but causes high engineering effort, since even the format of primitive data types has to be coordinated. Modbus supports serial communication using a simple master–slave protocol over EIA-485. A total of 247 different slaves can be addressed. The typical data rate is 19.200 b/s.[17] A mode of transmission over TCP/IP is also defined, in which every node can be both client and server.

At the field level, wireless technologies hold great promises for reducing the effort spent on sensor cabling and installation. Yet to realize this benefit, nodes have to run on batteries for months, or even better years. Control applications in BA do not require high bandwidth, but still demand reasonably low latency. Support for large device arrays is an

[10]Although the following paragraphs will not cover U.S. standards developing bodies in detail, they will be referenced as their respective standards are discussed.

[11]International Standards Organization, Technical Committee 205

[12]Comité Européen de Normalization, European Committee for Standardization.

[13]Comité Européen de Normalization Electrotechnique, European Committee for Electrotechnical Standardization.

[14]Not every aspect of this comprehensive scope is covered by published standards yet.

[15]ISO/International Electrotechnical Commission Joint Technical Committee 1, Subcommittee 25, Working Group 1.

[16]A Profibus FMS profile for building automation existed (albeit never as a formal standard), but shared the fate of FMS in that it is no longer relevant in practice today for new installations.

[17]It is acknowledged that coding and protocol design have considerable impact on the effective data rate of a communication system. Due to lack of space, these details cannot be covered sufficiently here. The bits per second figures are quoted to allow a rough estimate. For LonWorks and EIB/KNX, [22] provides a comparative discussion.

added benefit. Popular office wireless standards like IEEE 802.11 are obviously not optimized for these requirements. Even Bluetooth is designed for being embedded in devices which consume more power. Therefore, these technologies are better suited to management-level functions.

IEEE 802.15.4 defines physical and medium access layers for low-rate wireless personal area networks. It contains methods to provide (cumulatively) long periods of deep sleep, which are necessary to save power (making use of the quick transitions between sleep mode and active state possible with current silicon). A coordinator periodically can transmit beacon frames, which among other things are used to synchronize attached devices. Devices which expect data (periodically, at an application-defined rate, e.g., sensors) can wake up only for the beacon frame, which indicates whether data is actually available for them. Devices which only intermittently have data to transmit (at an application/external stimulus defined rate, e.g., light switches) can wake up, synchronize with the beacon, transmit and go to sleep again. Small packets and CSMA ensure that nodes only transmit when necessary.

The Zigbee alliance [23] adds additional layers (whose specification is not openly available) to IEEE 802.15.4. They provide network layer functionality with additional security including AES (Advanced Encryption Standard) and routing functionality for extending the typical 50 m range of a "segment" by supporting mesh topologies for dynamic creation, consolidation and splits. Zigbee also adds an application support layer with discovery and binding plus "application objects" (functional blocks), which currently cover building automation, plant control and home control applications. Latencies of 15 ms from sleep to actual transmit are achieved and a significantly smaller and less resource-consuming stack than with Bluetooth are advertised.

As for standards covering specific building service domains only, Digital Addressable Lighting Interface (DALI) is an IEC standard and widely accepted for lighting applications. Its primary focus is on replacing the traditional 0–10 V interface for dimmable electronic ballasts. A DALI loop can contain up to 64 individually addressable devices. Additionally, each device can be a member of 16 possible groups. Devices can store lighting levels for power-on, system failure and 16 scene values, plus fading times. There are also immediate commands (without store functionality) and commands for information recall (like lamp status). Loops can be up to 300 m long, with free topology. The data rate is 2400 b/s using a master–slave based protocol.

DALI also accommodates operation buttons, light and presence detectors. Addresses and all other settings are assigned over the bus. The necessary functionality can be provided by hand-held programming devices, gateways, or wall-box controllers which add it to their operation button functionality.

Finally, for remote meter reading, M-Bus [24] has gained a certain degree of importance in Europe. Its application layer supports various metering applications and includes support for advanced functionality like multiple tariffs. It operates on low-cost twisted pair cabling, with the data link layer based on the IEC 870-5-1/-5-2 standard for telecontrol transmission protocols. A serial master–slave protocol with data rates between 300 and 9600 b/s is used. A segment can contain up to 250 devices and cover a maximum distance of 1000 m (multiple segments are possible). In the master-to-slave direction, data is transmitted using voltage modulation, while in the reverse direction, current modulation signaling is used.

B. Open Management Integration

At the management level, office network and automation standards prevail. Mapping BA functionality and system states to protocols and representation formats used in the IP-dominated IT networks is of particular interest.

A variety of Web servers for BACS visualization and control are available. For user interfaces, HTML/Java Applet user interfaces are especially convenient in office environments when light walls are used, reconfiguration is frequent and room control is desired as they eliminate the need for room controllers. [25] details how rights management on a per-workstation basis (for functions with local scope) as well as on a per-user basis for administrative-level functions can be implemented.

Protocols like HTML are designed for operator-machine communication, not for transmitting information from one machine to another. For integration of BACS with other enterprise computing applications such as, for example, facility scheduling, maintenance management, and energy accounting, a suitable data model and corresponding services are needed.

For manipulating single control variables over a gateway the use of the Simple Network Management Protocol (SNMP) proved to be a practical approach [26]. In this case, the control variables are mapped to management information base (MIB) variables that can be accessed over the Internet via SNMP. While this method illustrates the gateway concept, it has less practical relevance in building automation.

Today, it is common practice to model data structures as objects, including those in the control domain. Several standards for distributed object-oriented systems are commonly used in the Internet and thus are candidates for usage in application layer gateways. Object access protocols over IP networks are provided by the Common Object Request Broker Architecture (CORBA), the Java Remote Method Invocation (RMI) interface, the Microsoft Distributed Component Object Model (DCOM), or the Simple Object Access Protocol (SOAP) using XML notation [27], [28]. All these technologies are found in proprietary gateway solutions, for example [29]–[31].

One of the first open standards for accessing process data using an object-oriented approach which found broader acceptance by different vendors is open process control (OPC) [32]. OPC, which is based on DCOM, is also widely used in building automation. The OPC gateway acts as a server providing data from the control network to the client. The namespace is organized as a tree. Services implemented are not limited to data access and exchange, but also include alarms and events and historical data access. A number of

PC-based OPC servers and clients (e.g., visualization tools) for BAS are available.

The main disadvantage of plain OPC is its tight relation to Windows-based systems. Because of this platform-dependence a new trend of standardization focuses on SOAP/XML to access the data objects in the building. In the recent past a number of initiatives are producing platform-independent gateway standards based on XML/SOAP.

For the OPC data access services the OPC XML/DA standard enables the access of data on an OPC server through web services. Two upcoming standards designed specifically for the building automation domain are of particular interest and find support by important manufacturers in the area: oBIX [33] and BACnet/WS (covered in the following section).

V. Open System Solutions

BACnet, LonWorks and EIB/KNX are open systems claiming the ability to cover BA applications in their entirety. They all have achieved considerable significance in the worldwide market (in case of BACnet and LonWorks) or in the European market (in the case of EIB/KNX) and are often chosen by both customers and system integrators for complete system solutions.

This section introduces the following aspects of these systems: standardization and certification; physical characteristics including supported media and network topologies; communication paradigms; application data model; and services. In addition, standard hardware components and commissioning tools are discussed where appropriate.

A. BACnet

The Building Automation and Control Networking Protocol (BACnet) [1], [34] was developed specifically to address the needs of building automation and control systems of all sizes and types. Capabilities vital to BA applications were built into BACnet from the beginning in order to ensure the highest possible level of interoperability in an environment possibly involving multiple vendors and multiple types of building systems.

The development of BACnet began in 1987, when an American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE) project committee could not find an existing protocol that satisfactorily met all of the criteria the committee members had in mind for a suitable standard communication protocol for building automation applications.

The development effort was finally completed in 1995, when BACnet was first published as an ANSI/ASHRAE standard. In 2003 BACnet was adopted as both a CEN and ISO standard [17] and has been, or will be, adopted as a national standard by the 28 member countries of the EU pursuant to CEN regulations. It has also been adopted as a national standard by Korea and is presently under active consideration in many other countries including Russia, China and Japan. The current version is [35].

The BACnet specification is under continuous maintenance and further development. It is maintained by ASHRAE Standing Standard Project Committee (SSPC) 135 [36]. SSPC members represent all sectors of the industry. Additionally, participation in the development process is completely open. Any interested parties are actively encouraged to provide comments and suggested changes.

Based on surveys conducted in the United States, Europe, and Japan in 2003, there are now more than 28 000 installations in 82 countries and on all continents. The use of BACnet is free of any licenses or fees.

"BACnet Interest Groups" (BIGs) exist in Europe (BIG-EU), North America (BIG-NA), AustralAsia (BIG-AA) and the Middle East (BIG-ME). Additional BIGs are in various stages of development in China, Japan, Russia and Sweden. The Swedish group may expand to include the other Scandinavian countries. Each BIG has its own unique character: while the majority of BIG-EU members represent corporations, for example, almost all members of BIG-NA come from colleges and universities. In the United States, BACnet manufacturers have formed the BACnet Manufacturers Association (BMA) which, in turn, operates the BACnet Testing Laboratories (BTL). All these organizations are, to varying degrees, active in promotional activities, educational programs, the exchange of practical field experiences, interoperability issues, testing and certification and, last but not least, standards development.

While BACnet messages can, in principle, be conveyed over any network, a small number of network types were standardized for BACnet's use in order to maximize the probability that any two devices of comparable functionality would use the same type. The network types chosen cover a range of speed and throughput. They are Ethernet, ARCNET, Master–Slave/Token-Passing (MS/TP), LonTalk, and Point-to-Point (PTP). Each local area network type, except MS/TP and PTP, is a standard, off-the-shelf technology. MS/TP addresses connectivity over twisted pairs using EIA-485 signaling while PTP supports dial-up communications and other point-to-point applications using EIA-232 and, possibly, modems or other data communication equipment. Note that the use of the LonTalk protocol is limited to transporting BACnet-specific messages. In particular, BACnet does not make use of the LON standard network variable type (SNVT) concept. An analysis of MS/TP performance is provided in [37]. [38] discusses the determination of the optimum packet length and buffer sizes for BACnet on Ethernet.

The desire to be able to make use of the Internet Protocol (IP) was recognized early on and in 1999 "BACnet/IP" was finalized. The protocol stack was extended with a "BACnet Virtual Link Layer" (BVLL) which allows underlying protocols, such as the User Datagram Protocol over IP, to be used as if they were in themselves a datalink layer. Thus, IP networks are now natively supported by the existing BACnet network layer which allows BACnet devices to communicate using IP directly, rather than via tunneling routers, as had been specified in the original standard. The MS/TP EIA-485 medium provides a low-cost, well-established means for communication up to 78.4 kb/s and is useful with traffic loads such as would typically be experienced with unitary

or application specific controllers. BACnet/IP and BACnet over Ethernet are more suited to communications involving higher data volumes. ARCNET is also widely employed for controller-to-controller communication in the United States and Asia due to the advent of the low-cost and relatively high-speed (156 kb/s) twisted pair version. PTP is still occasionally used but has been largely superseded by the Internet, at least for workstation traffic. Only two companies are known to have offered BACnet over LonTalk simply because more cost-effective alternatives, such as twisted pair MS/TP or ARCNET, are readily available. As for wireless communications, a transparent bridging solution based on IEEE 802.11 has been recently presented [39] while BACnet over wireless Ethernet has been around for years, proven largely at trade show exhibitions.

The base element in the BACnet network topology is the *segment*. Segments are physical runs of cable, which can be coupled using repeaters and bridges to form a *network*. BACnet networks (of possibly different media types) are connected by routers to form a BACnet *internetwork*. Only one path may exist between any two devices on an internetwork. BACnet also provides support for intermittent connections (like PTP) managed by *half-routers.*

A BACnet network address consists of a 2-byte BACnet network number and a local address of up to 255 bytes. The local address is specific to the link layer medium, e.g., an IP address for BACnet/IP or a MAC address for LANs. The BACnet routers which connect the individual networks route packets based on the network numbers. These routers are required to be self-learning. Provided with the network numbers for each of their ports, they are able to learn the topology by using appropriate router network management services, such as Who-Is-Router-To-Network.

BACnet represents the functionality of a BACS as a set of *objects*. Each BACnet object is a collection of data elements, all of which relate to a particular function. These objects correspond to the data points of the control application. The individual data elements are called the *properties* of the object. For example, an analog input object that reports room temperature will first of all have a "present-value" property (which is associated with the actual space temperature read from the physical input). Other properties describe the sensor, minimum and maximum values of the input, resolution and engineering units of the value, and indicate the reliability status of the sensor. The definition of each object type indicates via a "conformance code" whether a given property is required or optional, read-only, or required to be writable.

BACnet presently defines 25 different object types. They include simple object types such as binary input and output, analog input and output, multistate inputs and outputs, as well as a number of more complex (yet still generic) types related to scheduling, trending, alarming, and life safety capabilities.

Any given building automation device may have zero, one, or many objects of each object type with the exception of the "Device" object, which must be present in every device. This object is used to present or control various characteristics of the device and, among other things, contains an enumeration of all other objects existing in the device. The properties "object-identifier" (unique to each object in a given BACnet device), "object-name" and "object-type" have to be present in every object. Nearly two hundred standard properties, and their use in each of the standard object types, are currently defined.

The BACnet object model can be easily extended to include new objects or properties as needed. This can be done by any implementer without obtaining anyone's approval and such new capabilities will not interfere with similar extensions made by others provided the implementer makes use of its "vendor ID," freely available from ASHRAE.

While objects provide an abstract representation of the "network-visible" portion of a building automation device, BACnet *services* provide messages for accessing and manipulating this information as well as providing additional functionality. Communication follows a client/server model. BACnet currently defines 40 application services which are grouped into five categories: Alarm and Event, File Access, Object Access, Remote Device Management, and Virtual Terminal, although these latter services have largely been supplanted by web-based tools.

Among the Object Access services are ReadProperty (the only service mandatory for all devices), WriteProperty, ReadPropertyMultiple and WritePropertyMultiple, which collectively can read or manipulate any individual or group of property values.

BACnet provides three distinct, but complementary, methods for handling "events," including those considered important enough to be designated as "alarms." The first is called "Intrinsic Reporting" and makes use of parameters embedded in individual objects. Intrinsic reporting makes use of standardized event type algorithms (nine are currently defined, such as "OUT_OF_RANGE," "CHANGE_OF_STATE," etc.) but applies them rigidly to specified properties of the standard objects.

"Algorithmic Change Reporting" makes use of the same algorithms but allows them to be more broadly applied to any property of any object. The parameters associated with the selected algorithm (e.g., high limit, low limit, deadband, time delay, etc.) are contained in an Event Enrollment object, rather than "intrinsically" in the referenced object, thus allowing different algorithms to be applied, if needed, to the same property. Both intrinsic and algorithmic change reporting can make use of a Notification Class object which contains information on how event notifications, either confirmed (acknowledged) or unconfirmed (unacknowledged) are to be distributed. This combination of capabilities allows for extremely powerful alarm and event recognition and distribution: notifications can be tailored to different recipients at different times of the day or week, assigned varying priorities, and so on. A life safety alarm, for example, could be directed to specific workstations during the workday but cause a dial-out procedure to be invoked after working hours or on the weekend.

The third type of reporting is called "Change of Value" (COV). It causes a COV notification to be sent when a par-

ticular property changes by a predefined amount or, at the discretion of the COV server, at predefined time intervals. Clients may use the SubscribeCOV service to register for notifications of the default properties of standard objects, or the SubscribeCOVProperty service to request COV notification for any property of any object with any desired COV increment. "Unsubscribed" COV notifications, usually broadcast, provide a mechanism to distribute the current value of globally significant information, such as a site's outside air temperature or occupancy status, at a repetition rate determined by the server.

Of the eleven BACnet services dedicated to alarm and event handling, only the AcknowledgeAlarm service is aimed exclusively at the human operator: it provides a means to convey to an alarm-originating device that a human has actually seen, and is responding to, an alarm event. The originating device may use the receipt of such an acknowledgment, or the lack thereof within a specified time interval, to invoke other application-specific logic to deal with the alarm condition, such as initiating a precautionary system shutdown, performing a dial-out notification to some additional recipients, and so forth.

Remote device management services include Who-Is/I-Am and Who-Has/I-Have for dynamically discovering the network addressing information of peer devices and particular objects by way of their object names and/or object identifiers. Other services in this category allow for time synchronization, reinitialization of devices, and the suppression of spurious communications due to hardware or software malfunctions.

The BACnet application layer also directly supports other services relevant to BA tasks such as the prioritized writing of start/stop commands, setpoint changes, time of day scheduling, and trend log processing.

While the BACnet standard defines a sizable set of services, only a subset is necessary for most devices. Requiring them all to be implemented would unnecessarily increase complexity and cost without providing any particular benefit. In order to be able to concisely describe the capabilities offered by, or required in, a particular device, the concept of BIBBs (BACnet Interoperability Building Blocks) was introduced to the standard[18] in 2000. A BIBB describes a particular functional capability in one of five interoperability areas: data sharing, alarm and event management, scheduling, trending and device and network management.

BIBBs come in client/server pairs (designated A and B), allowing the precise specification of whether a given device functions as the initiator of a service request, the responder to a service request, or both. A BIBB may also require the presence of one or more objects, or that specific properties be supported. For example, the BIBB "Trending-Viewing and Modifying Trends Internal-B" requires that the server side of the ReadRange-Service be implemented and that a Trend Log object be provided.

To ease the work of specifiers, several standard BACnet device profiles have been defined. Each profile is a collection of BIBBs that is intended to map to commonly available BA equipment: operator workstations; building controllers; advanced application controllers; application specific controllers; smart actuators; and smart sensors. The BIBBs were selected to serve as a baseline for the given type of device. In order to claim conformance to a given profile, a manufacturer must offer at least the capabilities contained in the profile—but is free to add any additional functionality that is appropriate to the intended application of the device. Details about the portions of BACnet that are implemented in a device are documented in its protocol implementation conformance statement (PICS). This includes the precise set of services implemented in client or server role, proprietary and optional objects and properties, supported network media, and support for the dynamic creation and deletion of objects, among other things.

Interoperability testing and certification programs have been pursued by both the BMA and BIG-EU. The BMA's testing program began in 2002. BIG-EU followed two years later with its own test lab. Both testing programs have been harmonized and test results are expected to be mutually recognized.

The main focus of both groups has been to develop suitable software tools to test BACnet products and the procedures that will be used for specific kinds of devices. The procedures are mostly based on the companion testing standard to BACnet [40]. In an effort to go beyond simply verifying that a device has implemented its BACnet capabilities correctly and to actually improve "interoperability," a BTL working group was established that has developed a set of guidelines for implementers to help them avoid problems discovered in the course of early testing or the "interoperability workshops" that the BMA has sponsored since 2000.

The most recent addition to BACnet are proposed annexes that describe the use of XML and Web services for the integration of BACS with other management-level enterprise systems (BACnet/WS). BACnet/WS will be protocol neutral, and thus equally applicable to non-BACnet systems (although a comprehensive mapping between BACnet and BACnet/WS services is included in the draft standard).

Fig. 7 shows an example of a possible BACnet/IP configuration that illustrates the use of a Web server for both a graphical user interface and Web services along with a workstation that contains a traditional BACnet client application.

B. LonWorks

The LonWorks[19] system has been originally designed by Echelon Corp. as an event-triggered control network system. The system (described in [41]) consists of the LonTalk communication protocol, a dedicated controller (Neuron Chip)

[18]The means previously defined by the standard was a set of (numerical) hierarchical conformance classes, which could be augmented with additional collections of communication capabilities specific to particular (additional) applications. The conformance classes proved to be unworkable, however, because they failed to account for the initiator/responder role of the devices and did not have sufficient granularity to describe existing equipment.

[19]"LON" stands for "local operating network," a play of words on the term "local area network."

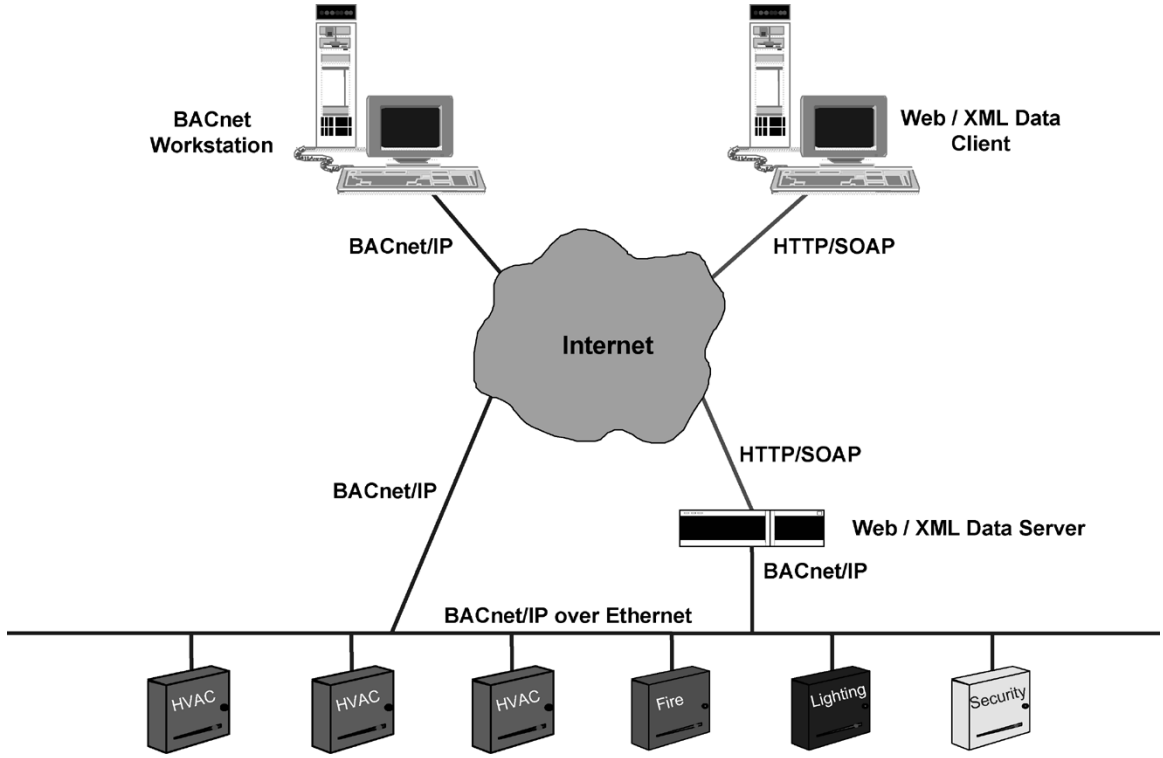

Fig. 7. BACnet/IP configuration example.

and a network management tool. In 1999 the LonTalk protocol was published as a formal standard, ANSI/EIA-709 ([42], revised 2002 [43]). While it has already been included in a European prestandard [18], it is planned to be published as a separate European standard in 2005. Below, the term EIA-709 is used to refer to the standardized communication protocol.

EIA-709 supports a variety of different communication media and different wiring topologies. Since it was designed as a generic control network, many protocol parameters are free to choose for the designer. To achieve interoperability (see below), a number of communication *channel* profiles were defined. These still include a variety of twisted-pair (TP), powerline (PL), and fiber optic (FO) channels. RF (radio frequency) solutions are available as well, albeit no standard interoperability profile exists for these. The most popular channel for building automation purposes is the 78.1 kb/s free topology TP profile (FT-10), which allows physical segments of up to 500 m using low-cost TP cable. A variant providing link power (LP-10) is also available. Often the 1.25 Mb/s bus topology TP (TP-1250) is used as a backbone to connect the lower speed FT-10 buses. FO is sometimes used for backbones as well.

For the TP medium, a unique medium access mechanism labeled *predictive p-persistent CSMA* is used. Its key mechanism is that when confirmed multicast services are used, a certain prediction on the future network load (i.e., the confirmations to be expected) can be made. The length of the arbitration phase is modified accordingly. Thus, the rise of the collision ratio with increasing load is mitigated. This helps to ensure an acceptable minimum packet rate even under heavy load, unlike in Ethernet-style networks using CSMA/CD, where the network load has to be kept well below 50%. At the start of the arbitration phase priority time slots are available for urgent messages. The mechanism, its properties and effectiveness are further discussed in [44]–[46].

More recently, building backbones turn from TP-1250 to IP tunneling mechanisms. Standardized in ANSI/EIA-852 [47] (also known as *LonWorks/IP*), IP tunneling is readily supported as a standard channel for EIA-709. Both tunneling routers and fully IP-based LonWorks/IP nodes are used. Channel configuration data including channel membership are managed by a central configuration server on the IP channel.

The entire routable address space of an EIA-709 network is referred to as the *domain.* Domains are identified by an ID whose length can be chosen up to 48 bit corresponding to requirements (as short as possible, since it is included in every frame; as long as necessary to avoid logical interference, especially on open media). A domain can hold up to 255 *subnets* with a maximum of 127 nodes each. Hence, up to 32 385 nodes can be addressed within a single domain. A subnet will usually correspond to a physical channel, although it is both possible for multiple physical channels to be linked into a subnet by bridges or repeaters as well as for multiple subnets to coexist on the same physical segment. Routing is performed between different subnets only. In particular, domain boundaries can be crossed by proxy nodes only (which transfer the information on the application layer). Subnets are usually arranged in a tree hierarchy as shown in Fig. 8.

Every domain can host up to 256 multicast groups. Groups can include nodes from any subnet. Broadcasts can be directed to a single subnet or the entire domain. Each node carries a world-wide unique 48-bit identification, the *Node ID*. It can be used for addressing individual nodes for management and configuration purposes, while regular unicast

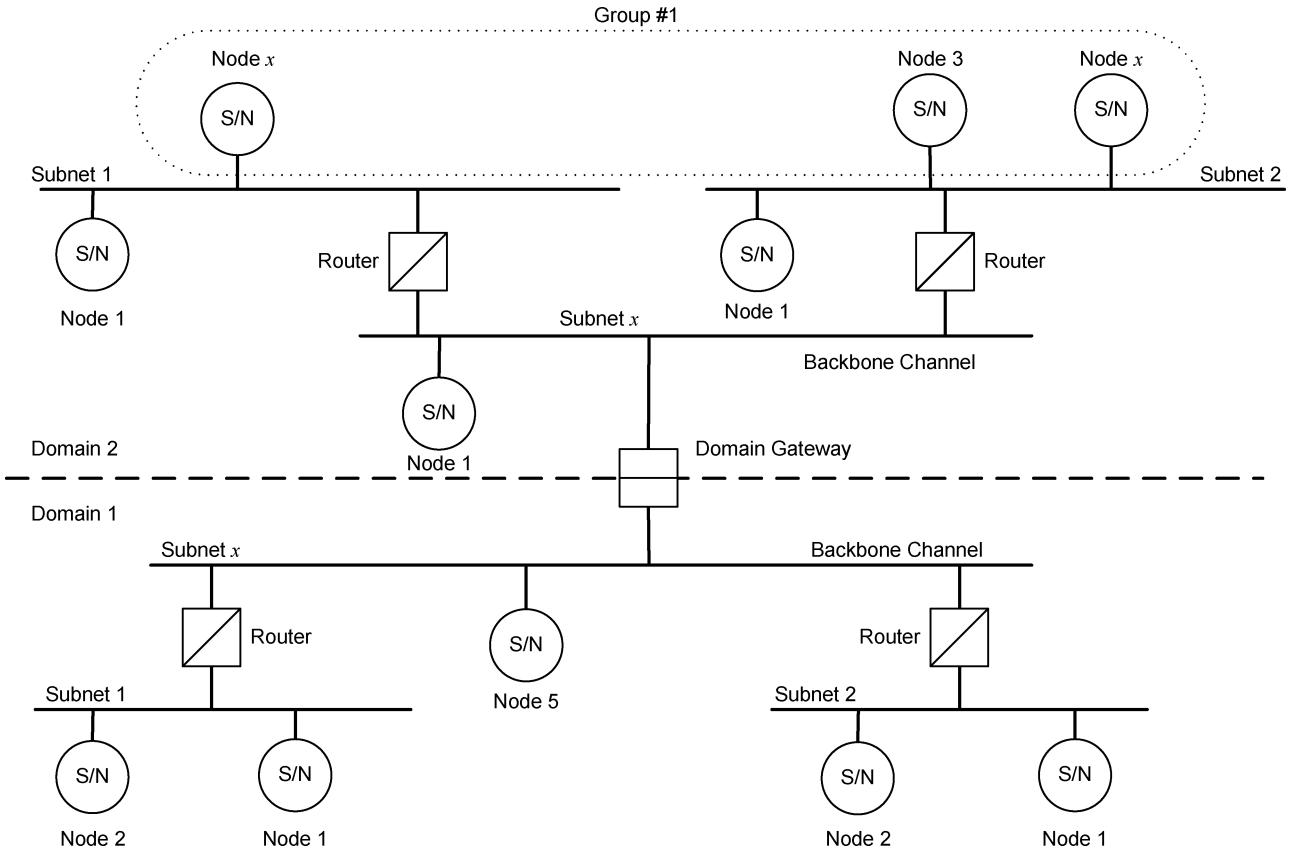

Fig. 8. Logical segmentation in EIA-709.

communication is handled through logical subnet and node addresses.

For both unicast and multicast, a reliable transmission mode (*acknowledged*) with end-to-end acknowledgments can be selected. In addition to the "one-shot" *unacknowledged* mode, an *unacknowledged-repeated* mode is provided, where every transmission is automatically repeated a fixed number of times. When acknowledged multicast is used, groups are limited to a maximum of 64 members each. Otherwise, they can contain an arbitrary number of nodes from the domain.

For acknowledged transmissions, a challenge-response authentication mechanism is provided. The challenge consists of enciphering a 64-bit random number using a 48-bit shared secret. The usefulness of this mechanism is limited since the algorithm is not published, 48-bit keys are not considered to be strong enough for attacks on high-bandwidth channels and the integrity of the message is not protected.[20] LonWorks/IP allows calculating a secure message authentication code for each encapsulated message (MD5 digest with a 128-bit key). Although this mechanism also does not encrypt the transmitted data, it protects the system from the injection of tampered messages.

The EIA-709 application layer allows generic application-specific messaging, but offers particular support for the propagation of network variables. Network variables are bound via 14-bit unique identifiers (*selectors*). The management and diagnostic services include querying the content type of the network variables (self-identification), the node status, querying and updating addressing information and network variable bindings, reading and writing memory, device identification and configuring routers.

Network nodes can be based on a chip from the Neuron series by Echelon [49] or other embedded controllers like the LC3020 controller by Loytec [50]. A typical network node architecture is shown in Fig. 9. The controller executes the seven OSI protocol layers and the application program, which interfaces with sensors and actuators connected through the I/O interface. A derivative of ANSI C called Neuron C is used to program the Neuron chips, whereas standard ANSI C can be used to program controllers like the LC3020.

Both provide implicit language support for network variables. Network variables are represented as standard C variables with the unique property that a data packet is automatically created and transmitted whenever the value of the C variable changes. Likewise, the value of the C variable will automatically be updated whenever a data packet has been received from the network.

A variety of installation and management tools are available for EIA-709 networks. The wide majority, however, is based on the LonWorks Network Operating System (LNS) management middleware by Echelon. Besides APIs for commissioning, testing, and maintaining, LNS provides a common project database, avoiding vendor lock-in of these valuable data. For configuration of vendor-specific parameters LNS provides a plug-in interface.

[20]Reference [48] compares security features in BACnet, LonWorks and EIB/KNX and proposes improvements for LonWorks based on smart cards.

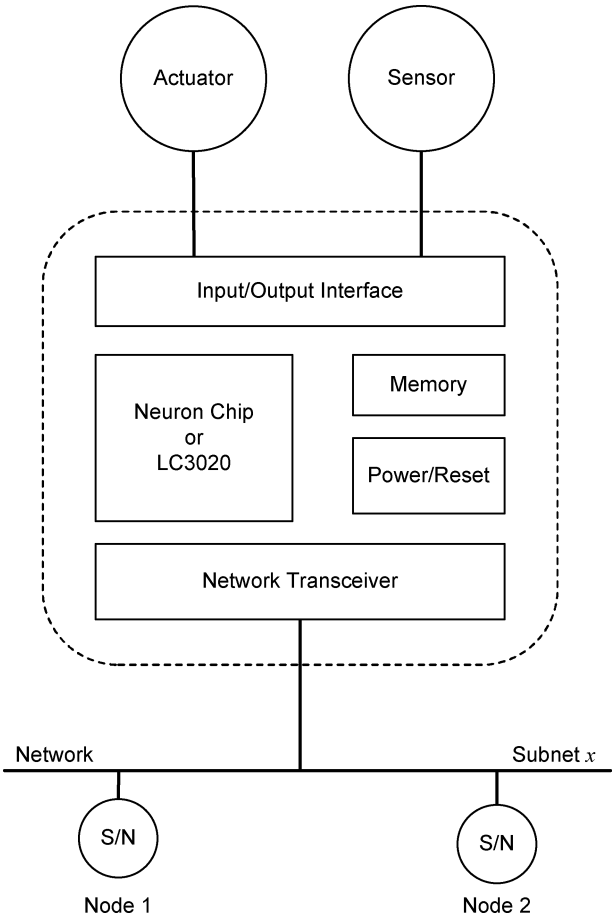

Fig. 9. Typical EIA-709 node architecture.

For performance analysis and troubleshooting, various protocol analyzers are available, including remote logging over IP networks. Modern network infrastructure components also have built-in statistics and diagnostics capabilities to allow remote monitoring and maintenance.

Some approaches exist for the automatic configuration of messaging relationships ("self-binding" or "auto-binding"). They are, however, confined to applications of limited complexity (single-vendor systems or very basic functionality only).

It should be noted that entirely nonopen systems can be (and are being) built using LonWorks technology. The LonMark Interoperability Association (now LonMark International) founded in 1994 [51] defines guidelines to manufacture and to integrate interoperable devices. These guidelines shall guarantee a smooth integration and operation of devices designed, produced, and installed by different manufacturers. They include LonTalk channel profiles, standard network variable types (SNVT) and functional profiles. A SNVT comprises syntactic as well as semantic information, like the associated engineering unit. Over 60 functional profiles have already been published. They relate to a number of application domains, most of them with a strong relation to building automation. Examples include "VAV Controller," "Constant Light Controller," "Scheduler," "Variable Speed Motor Drive" and "Occupancy Sensor." Although freely available, the LonMark guidelines and profiles are not part of any formal standard. Interoperability certification is provided on the basis of inspection of resource description files only, no laboratory tests are performed. In the most recent past, LonMark provides a self-certification tool, which LonMark members can use over the Web to certify their products.

C. EIB/KNX

The European Installation Bus (EIB) [52] is a fieldbus designed to enhance electrical installations in homes and buildings of all sizes by separating the transmission of control information from the traditional mains wiring. EIB is based on an open specification maintained until recently by EIB Association (EIBA). Key parts of it were included in [18] and [53]. In 2002, EIB was merged with Batibus and EHS (European Home System). The new KNX standard [54] seeks to combine their best aspects. The target of this merger was to create a single European *home and building electronic system* standard. Likewise, EIBA joined forces with the European Home Systems Association and Batibus Club International to form Konnex Association [55]. Still, the EIB system technology continues to exist unchanged as a set of profiles within KNX, frequently referred to as EIB/KNX.

[56] includes all the parts necessary from [54] for building products compatible at the bus interface, although some key elements like the interoperability model are still awaiting publication. Besides specification maintenance, Konnex Association is also responsible for promotional activities (including a university cooperation program) and the certification of test labs and training centers.

Regarding physical media, EIB already provided the choice of dedicated twisted-pair cabling and powerline transmission as well as a simple form of IP tunneling. RF communication and advanced IP tunneling were added under the KNX umbrella (albeit are not yet published within the context of [56]). The KNX specification also includes additional TP and PL variants which could be used for future devices.

The main EIB/KNX medium is the twisted-pair cabling variant now known as KNX TP1. The single twisted pair carries the signal as well as 29 V DC link power. Data is transmitted using a balanced base band signal with 9600 b/s. TP1 allows free topology wiring with up to 1000 m cable length per physical segment. Up to four segments can be concatenated using bridges (called *line repeaters*), forming a *line*. CAN-like, medium access on TP1 is controlled using CSMA with bit-wise arbitration on message priority and station address. Four priority levels are provided.

KNX RF uses a subband in the 868 MHz frequency band reserved for short-range devices (telecommand, -control, telemetry and alarms) by European regulatory bodies which is limited by a duty cycle requirement of less than one percent. Particular attention was given to minimizing hardware requirements. To this end, KNX RF does not only support bidirectional communication, but transmit-only devices as well. This reduces cost for simple sensors and switches without status indicators. KNX RF devices communicate peer-to-peer.

EIBnet/IP addresses tunneling over IP networks.[21] Its core framework supports discovery and self-description of

[21]EIBnet/IP supersedes "plain" EIBnet [57], which provided tunneling over Ethernet, and the legacy EIBlib/IP ("iETS") point-to-point IP tunneling protocol.

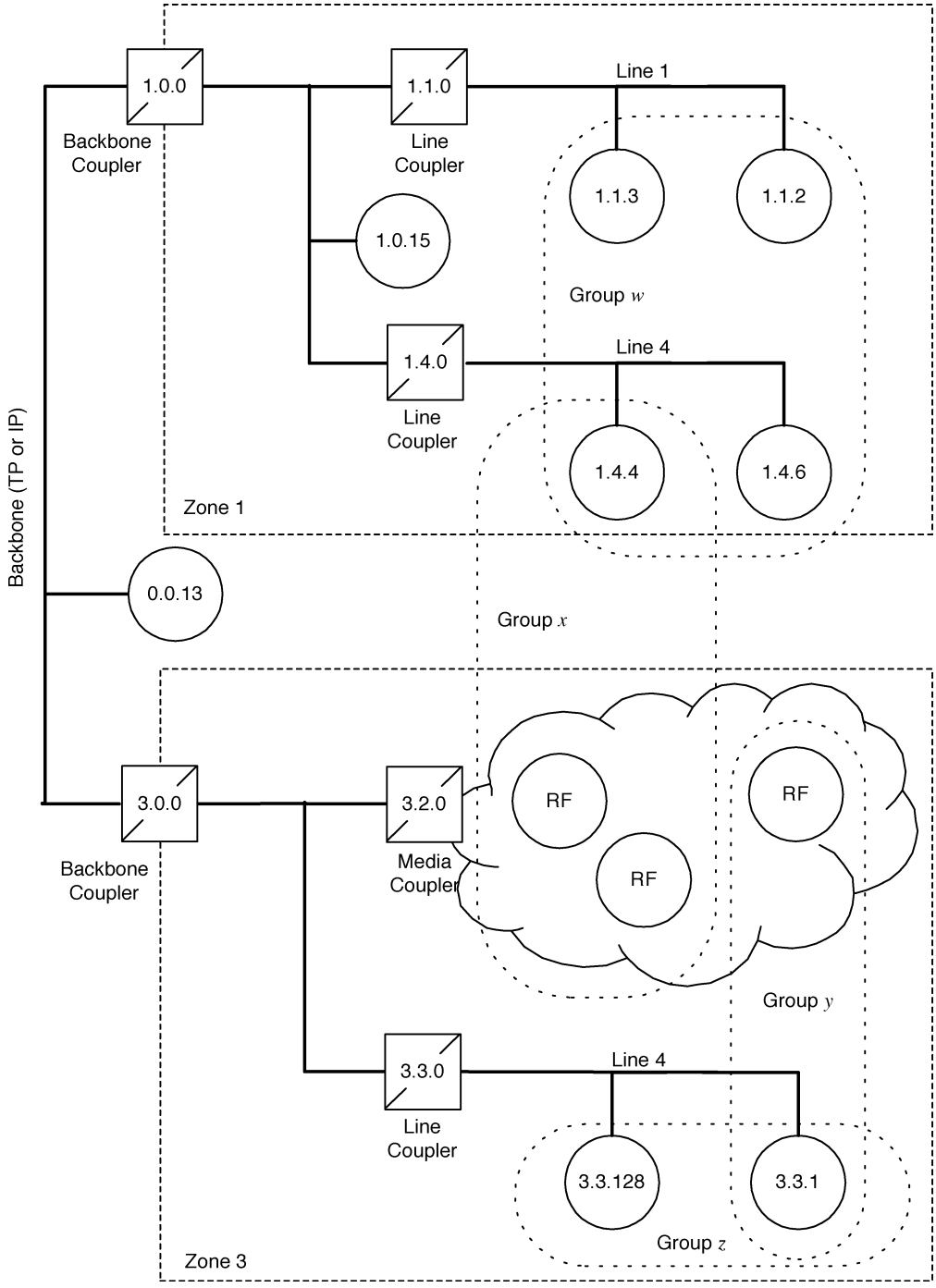

Fig. 10. EIB/KNX network topology.

EIBnet/IP devices. It currently accommodates the specialized "Service Protocols" Tunneling and Routing. Actually, both of them follow the tunneling principle as presented earlier, but differ in their primary application focus. EIBnet/IP Tunneling is to provide remote maintenance access to EIB/KNX installations in an easy-to-use manner and therefore restricted to point-to-point communication. EIBnet/IP Routing allows the use of an IP backbone to connect multiple EIB/KNX subinstallations. Routers using this protocol are designed to work "out-of-the-box" as far as possible. They communicate using UDP (User Datagram Protocol) multicast. Group management relies on IGMP (Internet Group Management Protocol). No central configuration server is necessary.

As outlined above, the basic building block of an EIB network is the *line*, which holds up to 254 devices in free topology. Following a three-level tree structure, sublines are connected by main lines via routers (termed *line couplers*) to form a zone. Zones can in turn be coupled by a backbone line, as illustrated in Fig. 10. Network partitions on open media are typically linked into the topology as a separate line or zone. IP tunneling is typically used for main lines and the backbone, with EIBnet/IP routers acting as couplers. Overall, the network can contain roughly 60 000 devices at maximum.

Every node in an EIB/KNX network is assigned an *individual address* which corresponds to its position within the

topological structure of the network (zone/line/device). This address is exclusively used for unicast communication. Reliable connections are possible. Multicast addressing is implemented in the data link layer. For this purpose, nodes are assigned additional nonunique MAC addresses (*group addresses*). The group addressing and propagation mechanism is thus extremely efficient. Yet acknowledgment are provided on layer 2 (i.e., within an electrical segment) only. The entire group answers at once, with negative acknowledgments overriding positive ones. Group addresses are routed through the whole network. Routers are preprogrammed with the necessary tables. Broadcasts always span the entire network.

EIB/KNX uses a shared variable model to express the functionality of individual nodes and combine them into a working system. Although this model uses state-based semantics, communication remains event-driven. Network-visible variables of a node application are referred to as *group objects*. They can be readable, writable or both (although the latter is discouraged to better keep track of communication dependencies). Each group of communication objects is assigned a unique group address. This address is used to handle all network traffic pertaining to the shared value in a peer-to-peer manner. Group membership is defined individually for each group object of a node, which can belong to multiple groups.

Usually, data sources will actively publish new values, although a query mechanism is provided as well. Since group addressing is used for these notifications, the publisher–subscriber model applies: the group address is all a node needs to know about its communication partners. Its multicast nature also means, however, that no authentication or authorization can take place this way.

Horizontal communication using shared variables between EIB/KNX nodes exclusively uses group addressing. Individual addressing is reserved for client-server style communication supporting vertical access. System management data like network binding information or the loaded application program are accessible through the properties of *system interface objects*. In addition, every device can provide any number of *application interface objects* related to the behavior of the user application. On the one hand, their properties can hold application parameters that are normally modified during setup time only. On the other hand, they can contain run-time values normally accessed through group objects.[22] Basic engineering functions like the assignment of individual addresses are handled by dedicated services.

The specification also encompasses standard system components, the most important being the bus coupling units (BCUs). BCUs provide an implementation of the complete network stack and application environment. They can host simple user applications, supporting the use of group objects in a way similar to local variables. Application modules can connect via a standardized 10-pin external interface (PEI), which can be configured in a number of ways. Simple application modules such as wall switches may use it for parallel digital I/O or ADC input. More complex user applications will have to use a separate microprocessor since the processing power of the MC68HC05 family microcontroller

[22]Mappings of these to BACnet objects are defined in the current BACnet standards.

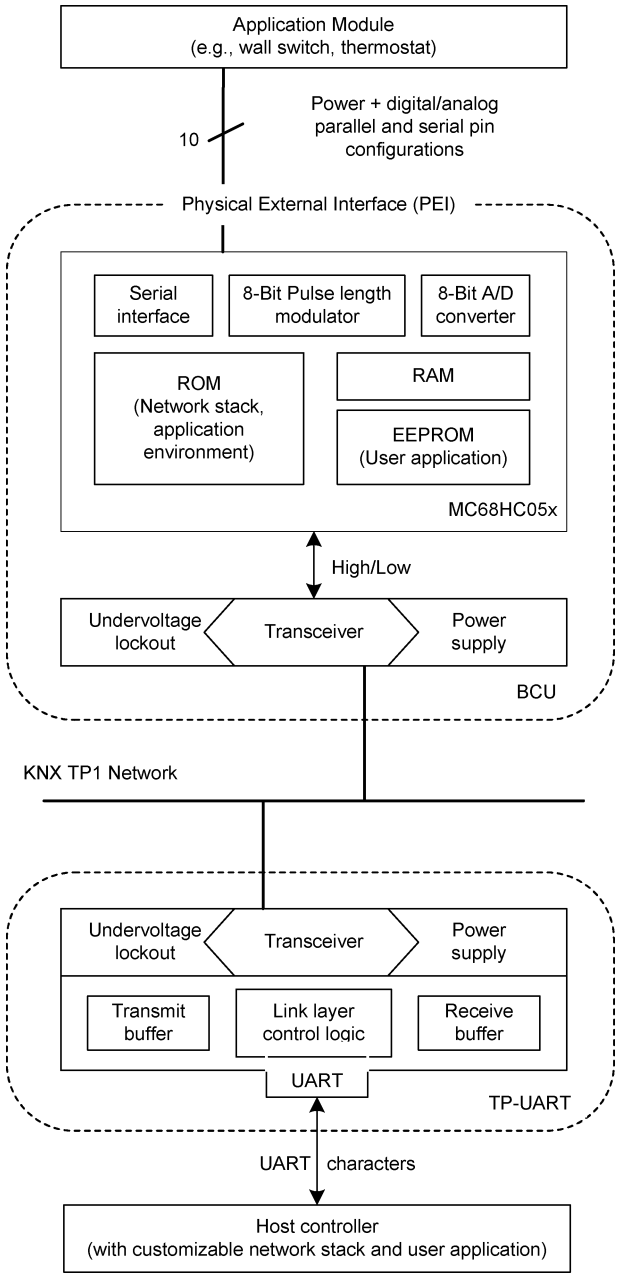

Fig. 11. Typical EIB/KNX node architectures.

employed in BCUs is limited. In this case, the application processor can use the PEI for high-level access to the network stack via a serial protocol. As an alternative, TP based device designs can opt for the so-called TP-UART IC. This IC handles most of the EIB/KNX data link layer. Unlike the transceiver ICs used in BCUs, it relieves the attached host controller from having to deal with network bit timings. These design options are illustrated in Fig. 11. System components are not included in [56].

For commissioning, diagnosis and maintenance of EIB/KNX installations, a single PC-based software tool called ETS (Engineering Tool Software) which can handle every certified EIB/KNX product is maintained by EIBA. KNX devices may support additional setup modes defined by the standard which do not require the use of ETS. A-Mode devices are preconfigured to automatically connect to each other (“plug and play”). In E-Mode, devices whose

group objects are to be bound together are designated by either pushing special buttons, assigning identical code numbers via DIP switches or codewheels, or via a handheld configuration device.

When using ETS, group objects are bound individually. The EIB Interworking Standard (EIS) merely defined a standardized bit-level representation for various types of shared variables. Functional blocks were provided for dimming and control of motorized blinds to ensure a base level of interchangeability. Such an approach is no longer viable with the E- and A-Modes. Therefore, numerous semantic data type and functional block definitions for various application domains are being added to the KNX specification. E- and A-Modes always link group objects and interface object properties at the granularity of these functional blocks (or channels of multiple blocks).

VI. Related Fields

Home automation or *domotics* is probably the field bearing the greatest resemblance to building automation. Yet it has significantly different characteristics. Instead of focusing on economic benefits, comfort and peace-of-mind are the key drivers. Systems are of considerably smaller scale, with emphasis on the integration of entertainment systems. Both equipment as well as commissioning and maintenance cost have to be kept very low. This favors robust "plug-and-play" systems with the capability of communicating via the mains or wireless. A large number of proprietary, centralized solutions is available. Networking and middleware standards of particular importance are X10 [58], EHS/KNX [54], and UPnP [59].

The possibility of BAS contributing to information in *facility management* data bases (e.g., for maintenance and cost allocation purposes) was already discussed. As an integral part of the building services, building automation systems also are part of the entities computer integrated architecture is concerned with. This includes the fields of *specification frameworks* (e.g., the Industry Foundation Classes, [60]) and *construction management*. The planning of BACS networks is essentially a part of the construction process. The use of compatible data models could ease tasks like heating and cooling load calculation, as it is the case for radio propagation today. *Building simulation* allows to assess the impact of BA control strategies even before systems are deployed by modeling the physical characteristics of a building. It also allows proactive instead of only reactive control strategies. Control can further be improved by retrieving values from the model where sensors are not available, for example in the middle of an office space [61].

For future directions, the fields of pervasive, ubiquitous and mobile computing come into play. Already now, smart phones and other mobile devices may be used instead of a light switch. The vision of pervasive and ubiquitous computing as expressed by [62] is that computers will be integrated into the environment—so that no one notices their presence—rather than having computers which are distinct objects. Sensor webs and microelectromechanical systems (MEMS) are related concepts. Eventually, building materials may become "smart" themselves, with for example wall paint itself being able to measure temperature and light level. Future buildings could also be aware of the tenants and their actions, taking appropriate measures for their comfort and safety (*sentient computing, situation modeling, user awareness*), possibly using an agent-based model as in e.g., [63].

On a more short-term perspective, technologies discussed in the context of *location-aware* systems could be of interest for BACS commissioning and maintenance, as they would allow to determine the location of nodes without reference to external databases.

VII. Conclusion and Outlook

The present paper provided a survey on building automation systems, directing attention to their communication systems. After a general overview on building services and the benefits provided by present-day BACS systems, the three-level functional model was introduced and shown how control networks can be embedded inside the automation pyramid. While the presented three-level model reduces the complexity of each individual level and keeps the levels lean and transparent, today's control networks are ready to span more than one level. As a result, the hierarchy flattens. The three-level functional model is mapped onto a two-tier network topology, with the functions of the former automation level being reassigned.

Without question, communication systems based on open standards are continuously gaining importance. For this reason BACnet, LonWorks and EIB/KNX were presented in more detail. LonWorks and EIB/KNX are field-level centric solutions, while the controller-oriented approach of BACnet lends itself more to upper-level functionality. Especially in Europe, BACnet is frequently deployed in combination with other control networks covering the field level. They are accessed over a gateway, which provides their object representation to the BACnet system.

With growing power and integration of building systems, demands on communications security rise. However, when considering the ultimate goal of sentient, user-aware buildings it is time to work on appropriate security models respecting privacy concerns as well. For the next years, interoperability will remain an important issue. Functional blocks from different domains and systems have to converge, opening the way for scalable solutions. This topic is related to the goal of "plug-and-play" system components on the one hand and to reusable functional templates on the other hand to reduce engineering and administration efforts.

Any discussion regarding building automation systems and their implementation is not complete without addressing design issues. Today's systems are still being installed without formal specification or design. System integrators have to rely on perceived wisdom, experience and best practice. Thus, techniques for better prediction of performance and dependability are required along with automated tools to support this. Up to now, despite the existence of sophisticated management tools, complete analysis or precise modeling of the distributed application is still beyond reach.

These problems need to be addressed and leave ample place for future research activities.

ACKNOWLEDGMENT

The authors would like to thank H.-J. Langels (Siemens A&D ET) and D. Loy (LOYTEC) for valuable input on an earlier draft of this paper, and the anonymous referees for helpful comments and suggestions.

REFERENCES

[1] H. M. Newman, *Direct Digital Control of Building Systems: Theory and Practice*. New York: Wiley, 1994.
[2] H. Arkin and M. Paciuk, "Evaluating intelligent buildings according to level of service systems integration," *Autom. Construction*, vol. 6, no. 5–6, pp. 471–479, 1997.
[3] *Building Automation and Control Systems (BACS)—Part 2: Hardware*, ISO Std. 16 484-2, 2004.
[4] J. K. W. Wong, H. Li, and S. W. Wang, "Intelligent building research: a review," *Autom. Construction*, vol. 14, no. 1, pp. 143–159, 2005.
[5] D. Snoonian, "Smart buildings," *IEEE Spectr.*, vol. 40, no. 8, pp. 18–23, Aug. 2003.
[6] K. Daniels, *Advanced Building Systems: A Technical Guide for Architects and Engineers*. Basel, Switzerland: Birkhäuser, 2003.
[7] *Moderate Thermal Environments—Determination of the PMV and PPD indexes and Specification of the Conditions for Thermal Comfort*, ISO Std. 7730, 1994.
[8] *Thermal Environmental Conditions for Human Occupancy*, ANSI/ASHRAE Std. 55, 2004.
[9] H.-E. Endres, "Air quality measurement and management," in *Sensors in Intelligent Buildings*, O. Gassmann and H. Meixner, Eds. Weinheim, Germany: Wiley-VCH, 2001, vol. 2, pp. 85–101.
[10] M. Thuillard, P. Ryser, and G. Pfister, "Life safety and security systems," in *Sensors in Intelligent Buildings*, O. Gassmann and H. Meixner, Eds. Weinheim, Germany: Wiley-VCH, 2001, vol. 2, pp. 307–397.
[11] C. P. Underwood, *HVAC Control Systems: Modeling, Analysis and Design*. London, U.K.: Routledge, 1999.
[12] J. P. Thomesse, "Fieldbuses and interoperability," *Control Eng. Practice*, vol. 7, no. 1, pp. 81–94, 1999.
[13] J. Plönnings, M. Neugebauer, and K. Kabitzsch, "A traffic model for networked devices in the building automation," in *Proc. 2004 IEEE Int. Workshop Factory Communication Systems (WFCS 2004)*, pp. 137–146.
[14] E. Finch, "Is IP everywhere the way ahead for building automation," *Facilities*, vol. 19, no. 11/12, pp. 396–403, 2001.
[15] S. Soucek and T. Sauter, "Quality of service concerns in IP-based control systems," *IEEE Trans. Ind. Electron.*, vol. 51, no. 6, pp. 1249–1258, Dec. 2004.
[16] (2005). Open Services Gateway Initiative. [Online]. Available: http://www.osgi.org
[17] *Building Automation and Control Systems (BACS)—Part 5: Data Communication Protocol*, ISO Std. 16 484-5, 2003.
[18] *Data Communication for HVAC Applications—Field Net—Part 2: Protocols*, Eur. Pre-Standard 13 154-2, 1998.
[19] *Data Communication for HVAC Applications—Automation Net—Part 1: BACnet, Profibus, WorldFIP*, Eur. Pre-Standard 13 321-1, 1999.
[20] *Data Communication for HVAC Applications—Field Net—Part 1: Objects*, Eur. Pre-Standard 13 154-1, 2000.
[21] *Information Technology—Home Electronic System—Guidelines for Product Interoperability—Part 1: Introduction*, ISO/IEC Std. 18 012-1, 2004.
[22] P. Fischer, "Comparison of fieldbus systems in a room automation application," in *Proc. 4th IFAC Conf. Fieldbus Systems and Their Applications (FET'2001)*, 2001, pp. 155–160.
[23] (2005). ZigBee Alliance. [Online]. Available: http://www.zigbee.org
[24] *Heat Meters—Part 3: Data Exchange and Interfaces*, Eur. Std. 1434-3, 1997.
[25] T. Pfeifer, A. Micklei, and H. Hartenthaler, "Internet-integrated building control: leaving the lab-robust, scalable and secure," in *Proc. 26th Annu. IEEE Conf. Local Computer Networks (LCN 2001)*, 2001, pp. 306–315.
[26] M. Kunes and T. Sauter, "Fieldbus-internet connectivity: the SNMP approach," *IEEE Trans. Ind. Electron.*, vol. 48, no. 6, pp. 1248–1256, Dec. 2001.
[27] (2005). Extensible Markup Language (XML). [Online]. Available: http://www.w3.org/XML
[28] (2005). World Wide Web Consortium XML Protocol Working Group. [Online]. Available: http://www.w3.org/2000/xp/Group/
[29] (2005). Coactive Networks. [Online]. Available: http://www.coactive.com
[30] (2005). Tridium, Inc. [Online]. Available: http://www.tridium.com
[31] (2005). Automated Logic Corp. [Online]. Available: http://www.automatedlogic.com
[32] (2005). OPC Foundation. [Online]. Available: http://www.opcfoundation.org
[33] (2005). Open Building Information Exchange. [Online]. Available: http://www.obix.org
[34] S. T. Bushby, "BACnet: a standard communication infrastructure for intelligent buildings," *Autom. Construction*, vol. 6, no. 5–6, pp. 529–540, 1997.
[35] *BACnet—A Data Communication Protocol for Building Automation and Control Networks*, ANSI/ASHRAE Std. 135, 2004.
[36] (2005). ASHRAE SSPC 135 Web site. [Online]. Available: http://www.bacnet.org
[37] S. Hong and W. Song, "Study on the performance analysis of building automation network," in *Proc. 2003 IEEE Int. Symp. Industrial Electronics (ISIE '03)*, vol. 1, pp. 184–188.
[38] C. Tamboli and C. N. Manikopoulos, "Determination of the optimum packet length and buffer sizes for the industrial building automation and control networks," in *Proc. IEEE Int. Symp. Industrial Electronics (ISIE '95)*, vol. 2, 1995, pp. 831–836.
[39] (2005). Kiyon, Inc. [Online]. Available: http://www.kiyon.com
[40] *Method for Test for Conformance to BACnet*, ANSI/ASHRAE Std. 135.1, 2003.
[41] D. Loy, D. Dietrich, and H. Schweinzer, *Open Control Networks*. Norwell, MA: Kluwer, 2004.
[42] *Control Network Protocol Specification*, ANSI/EIA/CEA Std. 709.1, Rev. A, 1999.
[43] *Control Network Protocol Specification*, EIA/CEA Std. 709.1, Rev. B, 2002.
[44] P. Buchholz and J. Plönnings, "Analytical analysis of access-schemes of the CSMA type," in *Proc. 2004 IEEE Int. Workshop Factory Communication Systems (WFCS 2004)*, pp. 127–136.
[45] Z. Hanzálek and J. Čapek, "Channel backlog estimation in LonWorks," in *Fieldbus Technology: Industrial Network Standards for Real-Time Distributed Control*, N. P. Mahalik, Ed. Berlin, Germany: Springer-Verlag, 2003, pp. 487–500.
[46] M. Miśkowicz, "Analysis of the LonTalk/EIA-709.1 channel performance under soft real-time requirements," in *Proc. 2003 IEEE Int. Conf. Industrial Technology (ICIT 2003)*, vol. 2, 2003, pp. 705–708.
[47] *Tunneling Component Network Protocols Over Internet Protocol Channels*, ANSI/EIA/CEA Std. 852, 2002.
[48] C. Schwaiger and A. Treytl, "Smart card based security for fieldbus systems," in *Proc. IEEE Conf. Emerging Technologies and Factory Automation 2003 (ETFA'03)*, vol. 1, pp. 398–406.
[49] (2005). Echelon Corp. [Online]. Available: http://www.echelon.com
[50] (2005). LOYTEC Electronics. [Online]. Available: http://www.loytec.com
[51] (2005). LonMark International. [Online]. Available: http://www.lonmark.org
[52] W. Kastner and G. Neugschwandtner, "EIB: European Installation Bus," in *The Industrial Communication Technology Handbook*, R. Zurawski, Ed. Boca Raton: CRC, 2005, vol. 1, pp. 34-1–34-18.
[53] *CEBus-EIB Router Communications Protocol—The EIB Communications Protocol*, EIA/CEA Std. 776.5, 1999.
[54] *KNX Specifications, Version 1.1*, Konnex Association, Diegem, Belgium, 2004.
[55] (2005). Konnex Association. [Online]. Available: http://www.konnex.org
[56] *Home and Building Electronic Systems (HBES)*, Eur. Family of Std. 50 090.
[57] *Data Communication for HVAC Applications—Automation Net—Part 2: EIBnet*, Eur. Pre-Standard 13 321-2, 2000.
[58] (2005). X10 Powerline Carrier Technology. [Online]. Available: http://www.x10.com/support/technology1.htm
[59] (2005). UPnP Forum. [Online]. Available: http://www.upnp.org
[60] (2005). International Alliance for Interoperability. [Online]. Available: http://www.iai-international.org

[61] A. Mahdavi, "Simulation-based control of building systems operation," *Building Environ.*, vol. 36, no. 6, pp. 789–796, 2001.
[62] M. Weiser, "The computer for the 21st century," *Sci. Amer.*, vol. 265, no. 3, pp. 94–104, 1991.
[63] M. ZhengChun, "Intelligent buildings and intelligent agents—a human-centered framework for building controls," in *Proc. 41st SICE Annu. Conf. (SICE 2002)*, vol. 5, pp. 3151–3156.

Wolfgang Kastner received the Dipl.-Ing. and Dr. Techn. degrees in computer science from the Vienna University of Technology, Vienna, Austria, in 1992 and 1996, respectively.

Since 1992, he is with the Institute of Automation, Vienna, where he has been holding the position of Associate Professor of Computer Science since 2001. From 1992 to 1996, he was working on reliable transmission protocols for distributed real-time systems. Since 1997, his research interests are in the area of field area networks and automation systems, with a special focus on spontaneous networking and home and building automation networks.

Dr. Kastner has been a Member of the Program Committee of numerous IEEE conferences and workshops.

Georg Neugschwandtner received the Dipl.-Ing. degree in computer science from the Vienna University of Technology, Vienna, Austria, in 2004.

Since 2004, he has been an Assistant Lecturer at the Institute of Automation, Vienna. He is also working with Siemens A&D ET, Regensburg, Germany, on plug and play concepts for fieldbus systems in building automation. His further research interests include component-based software architectures and networked embedded systems.

Stefan Soucek (Member, IEEE) was born in Vienna, Austria, in 1974. He received the Dipl.-Ing. and doctorate degrees in electrical engineering from the Vienna University of Technology, Vienna, Austria, in 1999 and 2002, respectively.

From 1999 to 2000, he was a Research Assistant at the Institute of Computer Technology, Vienna. From 2000 to 2002, he joined the development team for LonWorks/IP routers and control network systems at Coactive Networks, Inc. Since 2002, he is Head of Development and Research at LOYTEC, Vienna. His special research interests are fieldbus control systems over IP networks, protocol design, and design methods in embedded systems. Currently, he is working on applications in the control network/IP area, focusing on EIA-852 and BACnet.

H. Michael Newman (Member, IEEE) was born in New York in 1942. He received the B.Eng. physics degree (with distinction) from Cornell University, Ithaca, NY, in 1965 and the M.S. degree in the same field in 1966.

Following ten years as a pilot and flight instructor, he returned to Cornell University to take charge of the installation and development of its computerized energy management and control system, which he has managed ever since. He formed and served as chairman of ASHRAE's BACnet committee from its inception in 1987 until 2000 and currently serves as the chairman of its XML Working Group. He has also represented the United States on ISO/TC 205/WG 3, Building Control System Design, since 1996.

Mr. Newman is an ASHRAE Fellow and a Senior Member of the Instrumentation, Systems and Automation Society.