

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Standard Elevator Information Schema: Its Origins, Features and Example Applications

Jonathan Beebe and Ahmad Hammoudeh

Abstract

A generational change is taking place in building transportation systems as manufacturers and maintenance companies begin to integrate their products and services with the technologies of smart buildings and smart cities. Frequently this integration relies on the Internet of Things and cloud services. The diverse and heterogeneous nature of such collaborations requires a common shared semantic understanding of the complex and dense information that may be generated by transportation systems in buildings. The Standard Elevator Information Schema (SEIS) provides this in a format which is both machine and human readable. The role of the schema is to provide the ‘vocabulary’ for these collaborations. At the same time the schema specifies the properties, relationships and validation rules that define the information model, which could form the foundation upon which all elements of building transportation control and monitoring functions are constructed. SEIS is published under the Collective Commons licence and is free to download and incorporate into any product with the objective of reaching the broadest audience. This chapter discusses the origins and features of SEIS and provides a varied set of example applications. Consideration is also given to the issues of cyber security and data protection.

Keywords: lift, elevator, XML, XSD, schema, information model, IoT, REST

1. Introduction

The end of the second decade of the twenty-first century has witnessed a growing diversity of devices and information processing which smart buildings and cities are introducing into vertical (now also ‘two-dimensional’, c.f. thyssenkrupp elevator MULTI [1], and maybe ‘three-dimensional’ in the future [2]) building transportation systems. A new generation of lift systems has been developed offering increased levels of personal response particularly as a result of supporting destination call registration. This has been accompanied by increasing sophistication and variety of human interfaces, for users in many roles. Also there has been a rapid expansion of cloud-based analysis services, using Artificial Intelligence, that are capable of managing and processing vast amounts of data. The Internet of Things (IoT) is a key enabler of this technology evolution, so a section is included providing an overview of the IoT with the further objective of clarifying this somewhat misused appellation.

The variety of information domains and related equipment and service suppliers involved in these developments makes it increasingly important to conform to an agreed standard vocabulary and associated set of rules for interchange of information – a schema – that is specific to the application domain i.e. transportation systems in buildings.

Indeed the design and configuration of such systems would be enhanced considerably if such a schema was to be implemented internally as a foundation information model as well as for external communications. The continuing practice adopted by manufacturers of implementing proprietary communications, while promoting vendor lock-in, is no longer viable when travellers, administrators, owners, technicians, etc. experience many points of contact with such a variety of systems from disparate suppliers – For example, users cannot be expected to select the application of the lift manufacturer or maintenance company in whose lifts they wish to travel to register a call from their personal mobile device! However, the need for a common shared information model extends far beyond this trivial example and is relevant to most aspects of the operation of building transportation systems, particularly when machines are communicating directly with other machines without human interpretation of the exchanged information.

The Standard Elevator Information Schema (and complimentary Escalator Schema) which defines the information model is a formal declaration of the elements, their properties, interrelationships and validation rules and is directly meaningful to both humans and machines ('things').

2. Lift system communications: current status and potential developments

The richness of information and the potential diversity of interactions with external systems is much greater for passenger lift systems than for escalators and walkways and so this section refers mainly to lifts. However, a truly smart building will integrate communications with all transportation systems wherever practicable – probably a subset of what is possible for the lifts. In this section we look at what is currently available (discussed in greater detail in CIBSE Guide-D [3]) whilst casting an eye to potential developments that are within sight.

2.1 Current state of development

At the time of writing, a new generation of lift technology is appearing that embodies capabilities to communicate beyond the lift motor room through an expanding range of applications and interfaces.¹ Previously the lift system user interface has been specifically aimed at the primary users, i.e. the passengers, via simple electrical devices such as call buttons, floor position indicators, hall direction indicators and gongs, etc. The introduction of computer technology and networked communication between devices has led to an increasing complexity of the hardware of lift control equipment and also its associated passenger controls (e.g. destination call stations). At the same time, the new technology has promoted more sophisticated software in the form of control algorithms as well as a proliferation

¹ Inclusion (or omission) of references to manufacturers' products is purely for example and does not represent endorsement (nor rejection) of those products.

of external systems which collect, process and distribute the resulting complex and rich information.

Lift manufacturers [4–8] and third-party suppliers [9–14], sometimes in collaboration with major companies providing software and services (such as Microsoft and IBM), are developing new products and services which interconnect many lift installations concurrently thereby considerably extending the types of user that can interact in some way with a lift installation. The following subsections briefly discuss the different classes of internal and external systems that may be interconnected, identify the user roles that could benefit as a result and outline how these benefits might be delivered to those users.

2.2 Data logging

Data logging is in essence the capability to create and record a stream of data describing the activity of the lifts and their current status. As such, it should not be considered independently of external applications which then analyse and present the data. From the earliest days of data logging, and particularly since it became viable to install logging as a permanent feature rather than a commitment of expensive equipment only justifiable for a few days, it was acknowledged that the volume and diversity of the logged data necessitated a repeatable and automated (i.e. software-based) analysis.

Most modern lift control systems include some form of data logging, although this is usually in a proprietary format and available only to the staff of the manufacturer for the purposes of test, installation and maintenance. For older installations, a number of generic gateway devices are available from independent suppliers, which sample commonly available electrical signals and buses and in some cases additional sensor devices (e.g. accelerometer) to generate a stream of logged data, but again in a proprietary format. However, as building owners are increasingly demanding access to logged data of their lift activity (since they are in any case the designated owners of that data under Data Protection legislation) it is becoming common practice to include a requirement in the specification of new installations and refurbishments for some form of gateway that allows the connection of third-party logging systems [15] where the owner specifies the data format.

Data logging operates at several different levels, where access is restricted to persons possessing different skill sets and purposes:

Maintenance staff on site – to interrogate and alter system parameters and systems technicians able to update software and trouble-shoot. This data might include door timing, parking and priority floors and zone definitions, etc. More recently, device-specific operational ‘signatures’ (e.g. temperature, vibration, door operation counts, etc.) have been logged, which are used to identify abnormalities and drift due to wear and failure in specific components. There is an emerging trend of linking to advanced on-line web services that offer automated analysis and ‘learning by example’, enabling lift manufacturers to evolve their service offering and to drive preemptive maintenance scheduling. This analysis is of particular benefit to centralised maintenance managers and technicians who arrive on site better informed and provisioned with spares.

Owners/operators – to access performance data allowing analysis of how well the lift installation is managing the passenger demands, and duration of periods when lifts are unavailable etc. This data might include the number of stops, number of landing calls and their origins, car calls, system response times and an analysis of the demand patterns in terms of landing calls. Interest is growing in monitoring

and reporting the energy consumption of building transportation systems. Remote display (possibly in the lobby or building manager's office of the relevant building) of centrally held information is often also offered.

Maintenance management off-site – to review the operation of the lift, check any irregular behaviour and to plan a service schedule matching the duty cycle of each lift. Also to provide a 'forensic record', helping to establish the reason for a lift failure. Such data would include: fault and warning indications such as the safety circuit being interrupted prematurely, doors failing to open, abnormal journey duration, excessive re-levelling operations, door lock failures, machine and machine-space temperatures, etc. A number of major manufacturers [5–8] and a growing number of independent suppliers [9–14] offer remote monitoring services and display of lift status with real-time visual display of car and door activity plus registered calls.

Architects, transportation system designers and researchers – may use logged data to specify new installations or refurbishments for planning and subsequent assessment of traffic handling capacity and suitability of control policies matching the particular demands of a building. The data to support this would include the number and timing of landing calls (particularly in the case of destination calls), car calls, floor position, direction commitments and door operations of lifts, passenger loading, the allocation of lifts to respond to calls, etc. If data collection is fully comprehensive, it may be possible to replay lift operation into a traffic simulation program, closing the loop between traffic simulation for planning and what is happening in the actual installation [16]. A less data-intensive approach is discussed later in this chapter (Section 7, Example 4).

2.3 Remote control

Remote control concerns a very carefully controlled flow of data in the opposite direction to data-logging (i.e. from the external environment to the lift controllers and associated displays). The safety of passengers must first be thoroughly assessed for risks in the context in which the control is to be used – for example simply including a remote control to return all lifts to an entry floor of a hospital might put patient lives at risk.

Maintenance management off-site – A car or landing call may be entered to check that a lift is in service. No form of direct remote control, other than that available locally to passengers, is allowed as this could jeopardise passenger safety, for example if the security of the system were to be compromised through a cyber attack.

Intending passengers – With the increasing use of destination control traffic control systems a more personalised lift service may be provided to passengers. Call-stations are no longer restricted to lobby locations. For example, applications have been developed to allow calls to be registered from personal mobile devices enabling improved planning of call allocations and reducing queues for call registration, these are often linked to building access control [17, 18]. Requests need not be limited to human passenger traffic – a recent press-release [19] announced the development of a robot delivery agent that could distribute supplies and accompany visitors within a building (e.g. hospital, hotel, office), which is able to register calls for lift travel between floors.

Travelling passengers – Public information displays and announcements may be provided to passengers [20–22] notifying both waiting and travelling passengers of the services, facilities and events on the floors which they may select as a destination, or to which they are already travelling. A second type of user interface may be required which provides a management system allowing both general and

targeted information (e.g. based on destination floors, etc.) to be prepared and updated before presentation to one or more groups of users (e.g. in a multi-tenanted building).

Machine to machine – In future it is likely that remote applications will be able to update, within strictly defined limits, a limited range of high-level parameters (e.g. parking and priority floors, zones, energy consumption targets, etc.) in the lift controllers thereby linking smart cities and buildings to influence lift performance. Such remote applications would use artificial intelligence to ‘learn’ the optimal parameter settings based on external conditions (e.g. weather, calendar, traffic, etc.). It is even possible that other more general sources (e.g. anonymised social media discussions) could be interpreted automatically, enabling members of the public freely to provide information which could lead to modification of the operation of the services that they use, though probably without the deliberate intention of doing so.

3. Overview of IoT

Similar to many booming computer technologies, the concept of the Internet of Things (IoT) has attracted a lot of hype and when it comes to what IoT is, stakeholders have a variety of interpretations. Such interpretations often are biased towards the interests of those stakeholders who wish to emphasise their own assets of IoT [23]. In the same vein, lift manufacturers are announcing products which claim to use IoT technology. However, IoT is more than a network of internet-connected devices. It entails compliance with a set of essential features. So what is IoT?

The International Telecommunication Union [24] defined IoT as:

‘A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies’.

For a general overview, Atzori et al. [25] groups the defining features of IoT into three main categories called ‘visions’. IoT is the intersection of those three visions:

- ‘Things-oriented’ vision – focuses on the things’ identity and functionality
- ‘Internet-oriented’ vision – emphasises the role of the network infrastructure
- ‘Semantics-oriented’ vision – focuses on systematic approaches towards representing, organising and storing, searching and exchanging the things-generated information.

Taking a slightly more abstract approach and distancing the network infrastructure, Raggett [26] of the W3C organisation looks at the challenges and the risk of fragmentation of the IoT and proposes a ‘Web of Things’

- Things standing for physical and abstract entities
- Applications decoupled from underlying protocols
- Shared semantics and rich metadata

The ‘Applications’ here will be specific to a business domain or discipline and the ‘Shared Semantics’ need to be described by a formal definition, such as a UML domain model [27].

In a detailed discussion of IoT in the building transportation industry [28], the key conclusion was that the successful integration of building transportation systems (particularly lifts) into the IoT requires a standard definition of the semantics of any information exchanged between inter-operating devices (the ‘things’). It is the third, ‘Semantics-oriented’ vision where the greatest value can be added that is specific to the domain of lift operation because doing so is key to fulfilling the requirement of the IoT in ‘enabling advanced services’.

We can conclude that integrating lifts into the IoT is not simply a matter of establishing communications between devices over the internet, nor is it achieved just by using Internet messaging protocols. It is both of these characteristics plus the ability to run applications that exchange information in a way that is meaningful to all the things involved in the exchange i.e. there is a shared definition of the semantics.

A feature of the current ‘human Internet’, which is now taken for granted and which needs to be replicated for the IoT, is that of search and discovery facilities. In support of this, services in the IoT must be discoverable so that devices which have never communicated with them before can use these services and vice-versa that services can discover devices. The significance of this ability is illustrated by imagining the ensuing disruption if it were necessary, when installing a new lift system, to make manual updates to all the systems with which the lifts might eventually communicate.

4. Standards

4.1 Generic standards

Successful interoperation of systems from different suppliers is dependent on the faithful adherence to standards at all levels. In particular, the IoT relies on many standards and standard services e.g. Wi-Fi, IPv6 addressing (required to address every individual connected thing uniquely), http and web-service protocols, time-of-day services, load-balancing, cloud computing, etc.

As ETSI says on its website [29].

‘Smart objects produce large volumes of data. This data needs to be managed, processed, transferred and stored securely.

The use of standards

- ensures interoperable and cost-effective solutions
- opens up opportunities in new areas
- allows the market to reach its full potential

The more things are connected, the greater the security risk. So security standards are also needed to protect the individuals, businesses and governments which will use the IoT’.

4.2 Domain-specific standards

Interoperating applications that are specific to a business domain must be able to understand the domain-specific information contained in any data that is shared via the network. This is because there is a machine (rather than a human who can make inferences and interpretations) in both the generating and consuming roles. A lexicon or dictionary of the relevant terms and the rules for their use, usually in the form of a schema, is required to support such applications.

4.2.1 Existing standards specific to the lift systems domain

Some standards specific to lift systems do exist, and continue to be enhanced, describe –

the active state of the dynamic components of a group of lifts:

- CANopen-Lift [30] and
- BACnet, although at least two different implementations of this exist:
 - 1) BSR/ASHRAE 2016 [31] and
 - 2) National Standards Committee of the People's Republic of China [15].

and the component parts:

- BIM (proposed, not published)

However, there is a real need for an agreed and adopted formal schema to describe the day-to-day operational characteristics of a system of lifts, which is independent of the underlying electrical connectivity and network protocols. Such independence is a key requirement for successful integration of lifts into the IoT.

4.2.2 Standard elevator information schema

The Standard Elevator Information Schema (SEIS) [32] (and complimentary Escalator schema) was developed in 2003 [33], from earlier work, and uses standard software development tools and methods to model the information and operation of a lift system. Whilst it certainly describes the data normally associated with remote monitoring (car and landing call registration and cancellation, car movement, floor-position, and door state, etc.) the SEIS is much more comprehensive, including elements of processed information such as demand profiles, car journey plans and energy consumption. This level of detail enables the development of sophisticated applications and services, against a common standard, which therefore do not have to concern themselves with, for example, the intricacies of individual car control. The current state of development of IoT applications for lifts makes it thoroughly appropriate that such a standard should now be widely adopted throughout the Building Services industry.

5. Origins and features of the standard elevator information schema

5.1 Origins

The origins of SEIS can be traced back to a refurbishment project in 1980 of several groups of lifts in a classical (11 storey) building in central London – Bush House. The traffic demand was non-standard and continued throughout 24 hours of every day of the week. At the time there were virtually no computer controlled lifts anywhere in the world, and this provided the opportunity to design computer software from first principles for both individual car and group supervisory controllers.

5.1.1 Interface abstraction

Working from the available published literature [34], the Bush House central controller design was developed around an abstract model of the primary elements

of information that are essential to the control of a group of lifts. Each control element is linked to its external environment – electrical sensors, relays, indicators and other control computers – through an interface software layer which communicates via very short (less than 6 characters) textual event messages. Even delay timers are implemented as messages that the controller sends to itself, which are held for a specified period before being delivered. This architecture has a number of significant advantages:

1. The interface messages are directly intelligible to both humans and computers.
2. It was very easy to simulate messages in order to test the operation of the central control software thoroughly before it was installed with real equipment.
3. External devices such as a dynamic graphical status display and systems for performance monitoring, problem detection and data-logging could be driven directly from the interface messages which were exactly the events the controller itself was seeing. The project delivered all of these external devices.

The great benefit of the interface abstraction layer is that the controller is not predicated on a specific configuration of equipment nor a set of electrical connections to the hardware of the lifts. The design made it very straight forward to port the controller to a different building where components from other lift manufacturers might be used. An example of this abstraction is the use of a 64-bit absolute position encoder providing sub-millimetre resolution of the lift position in the shaft. From the point of view of the lift drive control, it is important to know very accurately the position of landing levels, slow-down points, etc., the thousands of positions in between however, are ignored. On the other hand, from the perspective of the logical control element of the lift, it is most important to know the floor position of the lift, including when the lift is travelling between floors and taking into account the direction of travel when a slow-down point is reached – hence the ‘Next Possible Stopping Floor’ of the lift was defined (now `<CarDynamicData><Floor>` in SEIS).

5.1.2 Event driven

A controller only receives a message when a change of state occurs, so it is truly ‘event driven’ and it interprets each received event in the context of the source of the event message and the current state of other known information. The source of the message needs to know nothing about the functionality that lies behind the interface, allowing great flexibility for the controller designers to evolve and improve their design. This ‘encapsulation’ is a fundamental characteristic of the object orientated programming paradigm [35].

5.2 Publication of a standard schema

However, it was not until the introduction of functionality into Internet web pages, supported by tools and technologies based on XML, that it became practicable to publish a formal statement of the full set of terms and relationships necessary to define lift operation. The format of this publication is a single XML schema file written in XML Schema Definition (XSD) language that is at once readable by humans and a wide range of computer operating environments. It contains specifications of:

- simple types, which can include validation rules for permissible numeric value ranges and text formatting
- complex types as heterogeneous collections of simple types
- aggregation relationships with optionality and multiplicity restriction rules, and every definition can be elaborated with textual documentation for the benefit of human readers.

So it was that in 2003 the knowledge from the earlier work was translated into an XSD – the Standard Elevator Information Schema – by Beebe and made publicly available [32] under the Collective Commons licence. It is hoped that through this mode of publication, discussion of the relative merits of any alternative schemas, should they be developed, would be limited to technical concerns thereby avoiding commercial and proprietorial considerations. The reference works of the day [36, 37] were analysed to identify significant nouns and their definitions that might form the atomic elements of the schema to ensure a wide-reaching scope. At the same time terms that were not directly relevant to lift operation were carefully excluded in order to avoid duplication or conflicts with schemas for other domains, currently available or potentially in the future. Based on the foundation elements and a more general survey of research across the field of lift traffic analysis, monitoring and simulation, further useful complex elements were identified and added to the schema e.g. demand profile, travel plan, door cycle, etc.

Two possible interaction modes were envisaged for SEIS at its inception (though a further mode was introduced with the application of REST transactions and more may be discovered as experience is gained):

- **Aggregated** – all the currently known information that is held under a specific node is communicated in a single instance message as an aggregation of elements. For example, this mode is used to drive a simple graphical status display example in this chapter (Section 7, Example 1).
- **Event** – only information which has changed is communicated as a time-stamped event message. This mode is appropriate when it is preferable to use communication bandwidth economically and the current state is maintained by the recipient of the message – see Event Data Log Example on the SEIS website [38].

This was the starting point for the root element types in the schema:²

5.3 Aggregated information

Aggregated information is subdivided, again to improve the efficiency of communications, into:

- **StaticData** – describing the unchanging parametric information of the lift systems.
- **DynamicData** – describing the current state of operational lifts.

² NB the text of the following subsections is graphically illustrated on the SEIS website – SEIS element names are in Courier New typeface [39]

5.3.1 Static data

For a lift car (`CarStaticData`) – the list of floors served by the specific lift, the number of car decks and door operators, the reference speed profile(s), door timing and energy profile(s).

For a group (`GroupStaticData`) – information such as the floors served by the group, named floors, pre designated priority floors, floors included in fixed zones plus a selected dispatcher algorithm name. Additionally, the group static data contains the collection of car static data for the lifts in that group.

For a building (`LocationStaticData`) – the location name along with the collection of group static data and/or car static data for the groups and any single cars at the location.

5.3.2 Dynamic data

`DynamicData` describes the current state of individual lifts and each group as a whole. The root element of each aggregation of dynamic data elements is time-stamped to specify the instant that it is valid in order to avoid inaccuracies due to possible communication delays. It should be noted that many of the dynamic data elements are specified as optional (multiplicity 0..1) to provide as much flexibility as possible to accommodate different installation configurations. Again, the dynamic data is separated into car-specific, group-specific and location-specific aggregations of elements of information:

CarDynamicData – includes floor position, committed (or next) direction of travel, door state, load, registered car and allocated landing calls. An optional additional element defines the `Route` which the car will follow in responding to its list of currently registered calls, which can be used for example to drive passenger information displays or to assist development and test activities.

GroupDynamicData – includes registered passenger landing calls (directional and/or destination) plus the collection of car dynamic data for the lifts in that group. Additionally, an optional `DemandProfile` for the group covering a number of data collection periods.

LocationDynamicData – is currently, simply the location name along with the collection of group dynamic data and/or car dynamic data for the groups and any single cars at the location. Additionally, an optional `DemandProfile` for the location covering a number of data collection periods.

5.4 Event information

Every event is time-stamped and includes the identifiers of the lift and/or group which generated the event. Optionally, the event may also include a destination identity which specifies the target object of the event (i.e. rather than the destination of the event message). Events may be communicated individually as a stream of updates or batched, providing flexible and efficient options for the design of data logging and status reporting systems. The time-stamp ensures that the sequence of events and the time they occurred are retained even if a communication link is temporarily lost or deliberately terminated.

A predefined set of event types, discovered from a thorough review of available literature plus experience of designing controller and management systems, is included in SEIS and this set may be enhanced as further experience is gained. It is therefore a requirement of SEIS that it is backwards compatible. Each predefined event type may have special properties, derived from SEIS type definitions, which are specific to that event.

ExceptionEventType – (describing a specific event instance) may contain other events thereby creating a context for the root event. This may represent the occurrence of an error, warning or simply some noteworthy information.

Recent developments of applications, which enable pre-emptive or predictive maintenance, use operating ‘signatures’ of devices to identify deviations from normal behaviour as an early indication of potential failure. Currently, the addition of an optional Signature property to the `LogEventType` is under consideration. Such a Signature data-type would need to be flexible and include the possibility of textual representation of a sequence of binary or analogue data values and an associated sampling interval.

5.5 Message formats

We have already noted that SEIS is written in XML Schema Definition language (XSD) so XML might seem a logical choice for formatting messages that conform to the schema. The benefit of this is that received XML messages can be validated automatically against the schema using a number of third-party libraries and there are several editor tools that can generate validation classes directly from the schema in a variety of programming languages. Such tools and libraries can relieve software developers from the burden of significant code development and continuing maintenance effort.

However, an XML message format is not a requirement and similar auto-generation of validation code can be found for other formats, for example JSON. The advantage of JSON is that it is a much more efficient representation and removes a great deal of the bulk of tags that are required for XML. Even if XML is used, more compact messages can be achieved by translating SEIS’s long tag names into short acronyms. This could even be achieved in real time by using an XSL style-sheet (this is illustrated in Section 7. Example 3).

5.6 REST

SEIS is a schema that defines an information model – it does not include any specification of functionality and so there are no supporting function calls specific to the operation of passenger lifts. Instead, the same set of standard generic functions can be requested from any supported node of the model and these functions are not at all related to building transportation systems. Thus every supported node in the model must offer the same four functions:

- GET
- PUT
- POST
- DELETE

which might be implemented, for example, as a ‘RESTful’ [40] interface in a variety of programming technologies, or as a relational database with stored procedures, or similar. This underlines the technology-independent nature of SEIS. Nodes which have a multiplicity greater than 1 (i.e. lists) must support queries using the properties and referenced links of the node (this is illustrated in Section 7. Example 3). SEIS is independent of any network topology (e.g. proxies, gateways, firewalls, etc) so is scalable, which is another characteristic of REST.

6. Security

Threats to networked and other inter-connected systems are continually evolving. The adoption and growth of the IoT throughout the building services domain serves to increase the vulnerability of autonomous networked devices to attack from external hazards and malign enterprises. To ensure that remote systems and applications continue to be updated with protection against new threats as they appear and with minimum risk of introducing vulnerabilities, security measures should adhere to published standards and be developed using third-party tools, which are developed and maintained by experts. The SEIS schema itself does not include cyber-security measures, which is correct since it is concerned only with describing building transportation systems.

Connection to any external network or communication channel, whether confined to the building perimeter or extending beyond it raises critical cyber security issues and potential data-protection concerns, which are discussed in CIBSE Guide-D [3].

Securing the integrity of systems from both

- outside interference or control by unauthorised external agents and
- unauthorised access to information gathered and/or held by the system and the proprietary software code of the system itself

should be addressed directly and must be continually reviewed against newly emerging threats. This responsibility extends throughout all roles in a business and all its processes (i.e. staff induction, in-service training and departure; product design philosophy, customer/client interaction, service procedures, etc., etc.). It cannot simply be addressed by technology-based solutions.

7. Examples of potential SEIS applications

7.1 Example 1: dynamic graphical status display

A lift group controller supplies a SEIS conformant XML document, on demand, to a standard web browser. The browser invokes a locally held XML Style-sheet (XSL) transform to render the raw XML as an HTML graphical representation of the current status of the group (so, if desirable, the presentation might be rendered differently in each browser instance):

A static version of the display can be accessed from the SEIS website [41]. The displayed graphic (i.e. right-hand panel of **Figure 1**) is interactive so that car and landing calls may be registered by clicking on the 'button' images where calls are not currently registered (hovering over the button normally displays the text of the request that would be generated in the browser's status bar).

This example demonstrates how SEIS supports and promotes the Model-View-Controller (MVC) [42] software design pattern – SEIS defines the Model. The lift data is made independent of its presentation, so it is the choice of the client application (in this case a simple web browser) as to how to represent the data. If the client is a data logger then it may be that no transform is needed. The example graphic display is very crude and not easy to read, but a more sophisticated transform might produce a very attractive result. Of course, it is not necessary to display the XML source and XSL Stylesheet frames as they are simply there to illustrate this example.

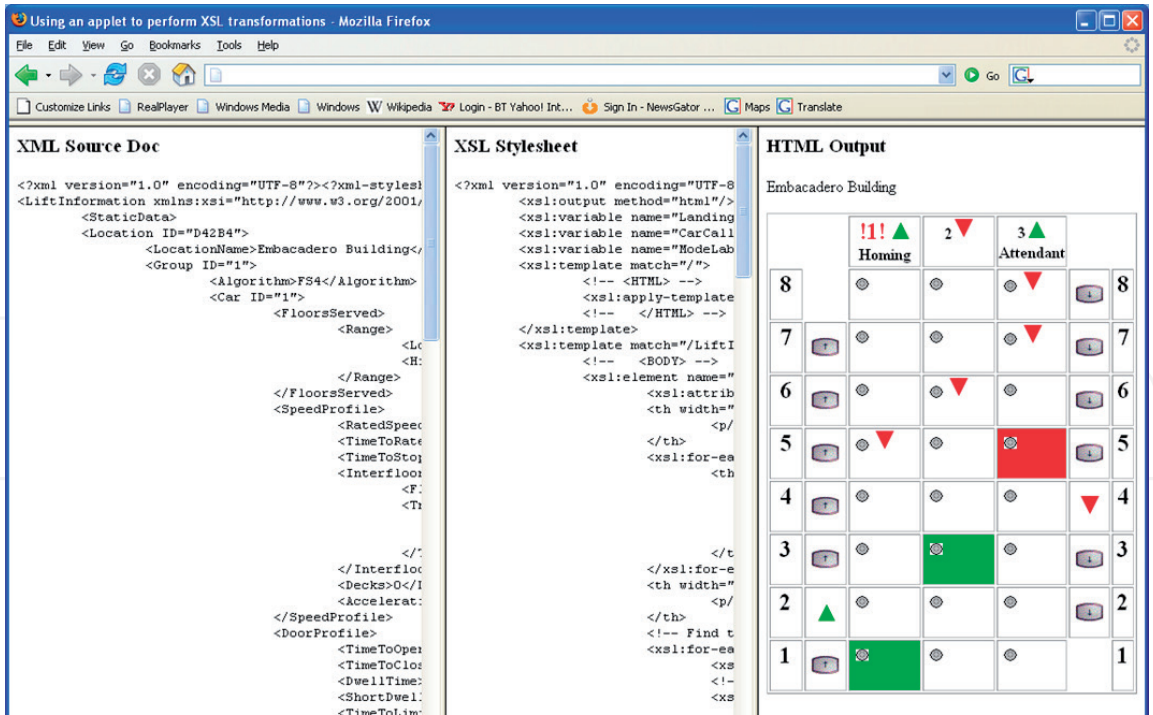


Figure 1.
XML source (left) is transformed by XSL (centre) renders HTML dynamic graphic status display (right).

7.2 Example 2: auto validation

This example shows how simply by referencing the SEIS schema (Figure 2, line 3 highlighted), an XML document [43] is validated. An error has been introduced (Figure 2, line 107 highlighted) into the document, assigning an invalid value to the Direction of Car 1's CarDynamicData, which is only valid if selected from the enumeration defined in the schema. The error is generated by the common class libraries that support XML. It has then been picked up and displayed to the user (Figure 2, bottom left circled), by the XML editor program in which the XML is being viewed. The resulting action will differ depending on the type of application which is receiving the erroneous XML, but the error detection is common.

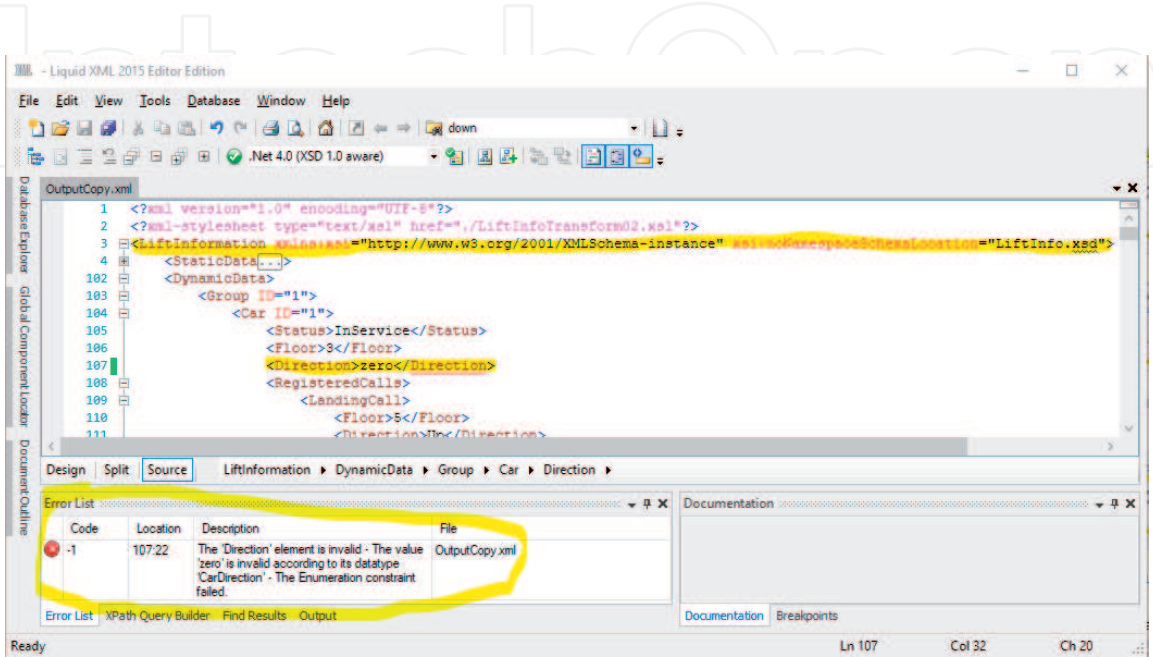


Figure 2.
XML document containing enumeration value error fails validation against SEIS schema.

This example demonstrates the support, which the schema provides for its own inevitable updates and enhancements that may be introduced over the lifetime of an application which depends on it. Maintaining the application validation software is greatly simplified (so is much less costly) because it is unlikely that each installed instance of the application will need to be upgraded on site.

7.3 Example 3: dispatcher interface

This example illustrates a group dispatcher published as a RESTful interface, conforming to SEIS and supporting the ‘publish and subscribe’ mode of interaction. The role of a dispatcher is to allocate passenger calls to the optimal lift in a group, as evaluated by an internal algorithm. The published interface is independent of the algorithm and it is the intention of the design that the dispatcher interface may be implemented in any installation configuration.

The aim of the example is to demonstrate how, by reference to a schema (i.e. SEIS) the dispatcher can operate simply through queries to and updates of the information without the need for a specific ‘allocate this call’ function. The benefits of this approach are that:

1. No assumption is made about the type of algorithm used by the dispatcher and therefore the inevitable debate is avoided about which parameters must be passed in such a call. The interface is compatible with any algorithm technique from dynamic sectoring to neural networks based on cost functions. In REST this is referred to as ‘separation of concerns’.
2. Furthermore, a simple ‘allocate’ function call is not appropriate in this application because any resulting allocation(s) must be communicated to several subscribed agents/devices, not just the initiator of the call. In fact there is not even a restriction on the number of dispatchers which may be operating concurrently for the same group of lifts.

Because the computing power and network bandwidth available to such an application, probably running in the lift motor room, are likely to be ‘constrained’ this example was implemented for the CoAP [44] protocol, which is specifically designed to minimise processing and communication demands. CoAP messages have a similar format to HTTP messages used by web-browsers to access a web server – each request includes:

- an ‘address’ (URI) which identifies the exact location of a particular resource that is the object of the request,
- optionally a ‘query’ string which acts as a filter for the data of the resource and
- optionally a data ‘payload’ which can carry data to or from the requested resource.

CoAP devices (endpoints) include a standard address ‘well-known’ which responds to a GET request by listing all the resources that the device supports so that, in this example, any lift can discover what nodes of the SEIS information model are supported by the dispatcher under the `<GroupDynamicData>` node as well as information on the algorithm and the group configuration under `<GroupStaticData>`. In this example we concentrate on the `<RegisteredCalls>``<LandingCalls>` sub-nodes within both the `<GroupDynamicData>`

and `<CarDynamicData>` nodes of a SEIS-conformant information model – these are published as CoAP resources.

Subscribing to a node is equivalent to making a GET request (with an ‘Observe’ flag) where the response containing the requested information is delayed until a change of state occurs in the node, and a response is re-sent with every subsequent change. Call stations (and mobile devices, etc. via secured apps on the Internet) subscribe to the `<GroupDynamicData><RegisteredCalls><LandingCalls>` node. Lift car controllers subscribe to the same node.

In this example messages are formatted in JSON, replacing the long XML element tag-names with short two-letter acronyms, to reduce message size. So for example `LandingCallType` data is represented as:

the query – `{"Fr=01","Dr="UP""}` and
the payload:
`{"Ss":0,"Df":0,"St":0.0,"Dn":0.0,"Ct":0.0,"Rk":0,"La":0,"Id":12345678}`

where:
Floor→Fr; Direction→Dr; Status→Ss; Destination→Df; StartTime→St;
Duration→Dn; AssignedTo→La; CallID→Id; etc.

The transformation to JSON and auto-validation (as shown in Example 2) can still be achieved using standard software development tools to generate code automatically, by making reference to a schema definition.

The following sequence describes the process of allocating landing calls to lifts:

1. When a passenger registers a landing call, a POST message containing `<Call-Registration>` data is directed to the `<GroupDynamicData><RegisteredCalls><LandingCalls>` node (**Figure 3**):
Address of node: `"GroupDynamicData/RegisteredCalls/LandingCall"`
Node query: `{ ["Fr=01", "Dr="UP""] }`
Payload: `{"Ss":0,"Df":10,"St":0.0,"Dn":0.0,"Ct":0.0,"Rk":0,"La":0}`
2. The dispatcher responds simply with a positive acknowledgement (ACK) to show that the POST request has been accepted. The response includes a unique reference (`CallID` - **Figure 3** dashed ellipse), so that the new data can be accessed in future, and which will be used to cross-reference future events associated with this landing call:
Response: `Code=2.01, Type=ACK, MID=142, Token=03: {"Content-Format":"text/plain"} Payload: 8 Bytes 34350805`
3. The dispatcher then performs what it is uniquely designed to do – finding the optimal lift to allocate to unallocated landing calls, at the same time possibly

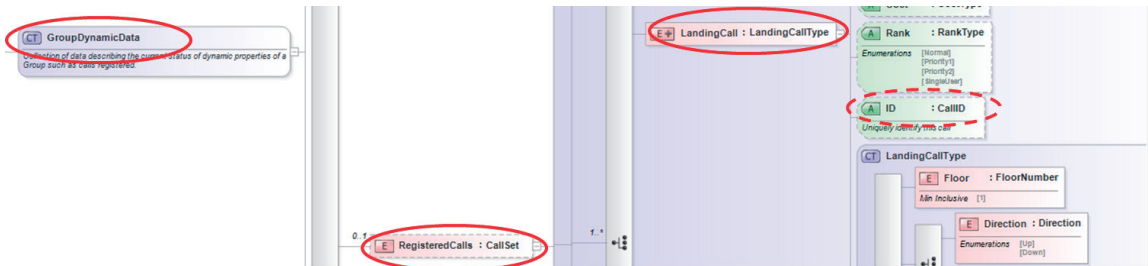


Figure 3.
`<GroupDynamicData><RegisteredCalls><LandingCall>`.

re-allocating other calls. The resulting allocations will be updated in the `<GroupDynamicData><RegisteredCalls><LandingCalls>` node.

4. The update will trigger the dispatcher interface to issue GET response `CallAssignment` messages to any car that has subscribed to the node and also to the subscribed call stations (possibly via a cloud-based app in the case of a mobile device). The message includes the `CallID` reference so that if necessary the call station can be notified to inform the passenger that their call has been registered (`"Ss":1`) and, in the case of a destination call, which lift has been allocated to their call:

Response payload: `{"Ss":1,"Df":10,"St":40578.6,"Dn":0.0,"Ct":0.0,"Rk":0,"La":2,"Id":34350805}`

At the same time the resulting allocation will be notified to any subscribed lift and the allocated lift car will update its own `<CarDynamicData><RegisteredCalls><LandingCalls>` node, which signals acceptance of the allocation (see **Figure 4**).

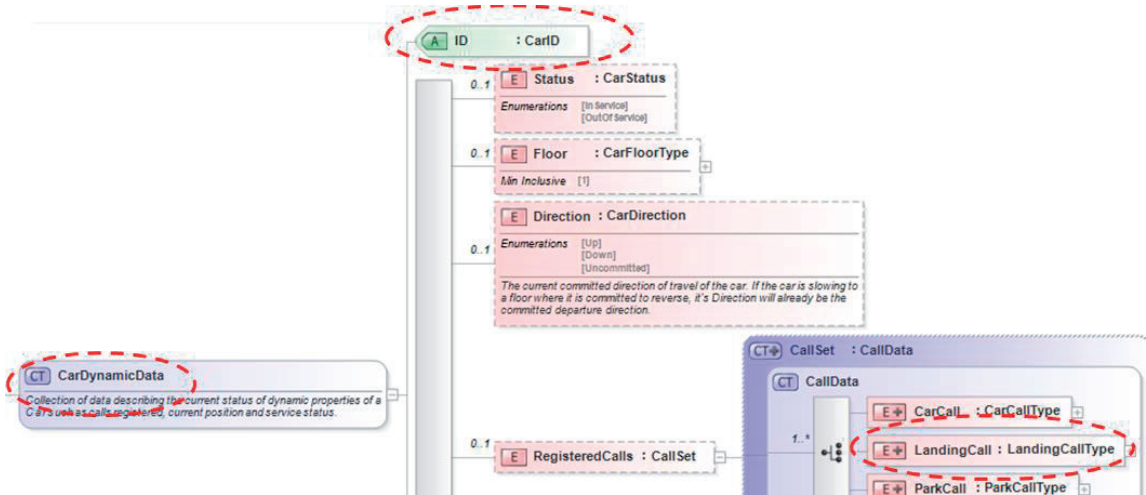


Figure 4.
`<CarDynamicData><RegisteredCalls><LandingCalls>`.

5. When the allocated lift arrives, to cancel the call it sends a `CallCancellation` (`"Ss":2`) UPDATE message to the dispatcher interface (in this case the system response time – `Dn` – was 20.5 s):

Address of node: `"GroupDynamicData/RegisteredCalls/LandingCall"`

Node query: `["Fr=01","Dr=UP","Id=34350805"]`

Payload: `{"Ss":2,"Df":10,"St":40578.6,"Dn":20.5,"La":2,"Id":34350805}`

This process works either with direction calls or destination calls and in fact supports configurations where both types of landing call station are used concurrently. This example is discussed in more detail in a separate publication [45].

7.4 Example 4: modelling and recording demand profiles

An architect designing a 10 storey office building wishes to simulate the operation of the lifts, particularly with respect to the expected traffic to and from the restaurant on the top floor during the lunch period of a working day. The building

has two entry/exit floors at the lowest levels – garage (floor 1) and a main terminal lobby (floor 2) for pedestrian entry/exit. The CIBSE Office lunch peak traffic template [46] is selected to define the traffic demand profile to drive the evaluation simulation. This template describes traffic in each of 12 five-minute intervals over the lunch hour in terms of the travelling population (actually the number of passenger arrivals) during the interval as a percentage of the total building population. The travelling population is divided by percentage into each of three categories – Incoming, Outgoing and Interfloor.

7.4.1 Selection of suitable traffic template

Even for a building with just a single main terminal floor, the lunch time template is considered to be very intense since it presents the features of both start and end of day peak flows, but in the presence of concurrent interfloor traffic. However, the building which is the subject of this design is not simple since it has two entry/exit levels. For the purposes of lift traffic analysis it is assumed that there is no resident population on the garage and lobby floors and furthermore, since the staff of the restaurant are not expected to travel during the busy lunch period, all three floors can be treated as if they are exits/entries of the building.

So, the specification by the template of traffic in terms of ‘incoming’ and ‘outgoing’ flows is convenient, as this example demonstrates, because the restaurant can be treated as a pseudo entry/exit floor. Consequently, for the building in question ‘outgoing’ traffic involves both up and down travelling passengers, and similarly for ‘incoming’ traffic.

During the lunch hour, the busiest period for the restaurant floor in this example, it is assumed that the “outgoing/incoming” traffic will be split 50% to/from the restaurant and 50% through the two real entry/exit floors where 10% passes through the garage and the remaining 40% via the lobby.

7.4.2 Populating the SEIS DemandProfile

With the parameters of the traffic template established, it is now possible to calculate the arrival rates, for each 5-minute interval, of passengers at the populated floors heading for an exit floor (outgoing) and similarly the arrival rates at the entry floors heading for the populated floors (incoming), lastly between populated floors (interfloor). We also have the means of estimating the destination floor of each arriving passenger and so the number of passengers arriving at a floor and sharing a destination floor is derived according to the above percentages for each type of traffic flow.

The result is a three-dimensional matrix of per-interval arrival and destination rates that is most easily generated via the passenger data editor of a lift traffic simulator [47], which then will be responsible for generating individual passenger journeys, based on its own internal statistical formula. Several simulation runs should be generated to ensure a valid sample size. Subsequently, the passenger data matrix should be loaded into a spreadsheet for easier manipulation. The matrix is most concisely and consistently expressed as a SEIS DemandProfile, (see **Figure 5**), which provides a crucial mechanism for linking the building under design with the reality of its operation.

The data in the spreadsheet can then be exported as XML in the form a DemandProfile as a permanent record of the design and analysis that has been conducted. A key element of the DemandProfile is the ResponseQuality (red ellipse in **Figure 5**), which links demand rate to response, thereby capturing a measure of the quality of service during the sample period. The particular value of this link is illustrated in this example.

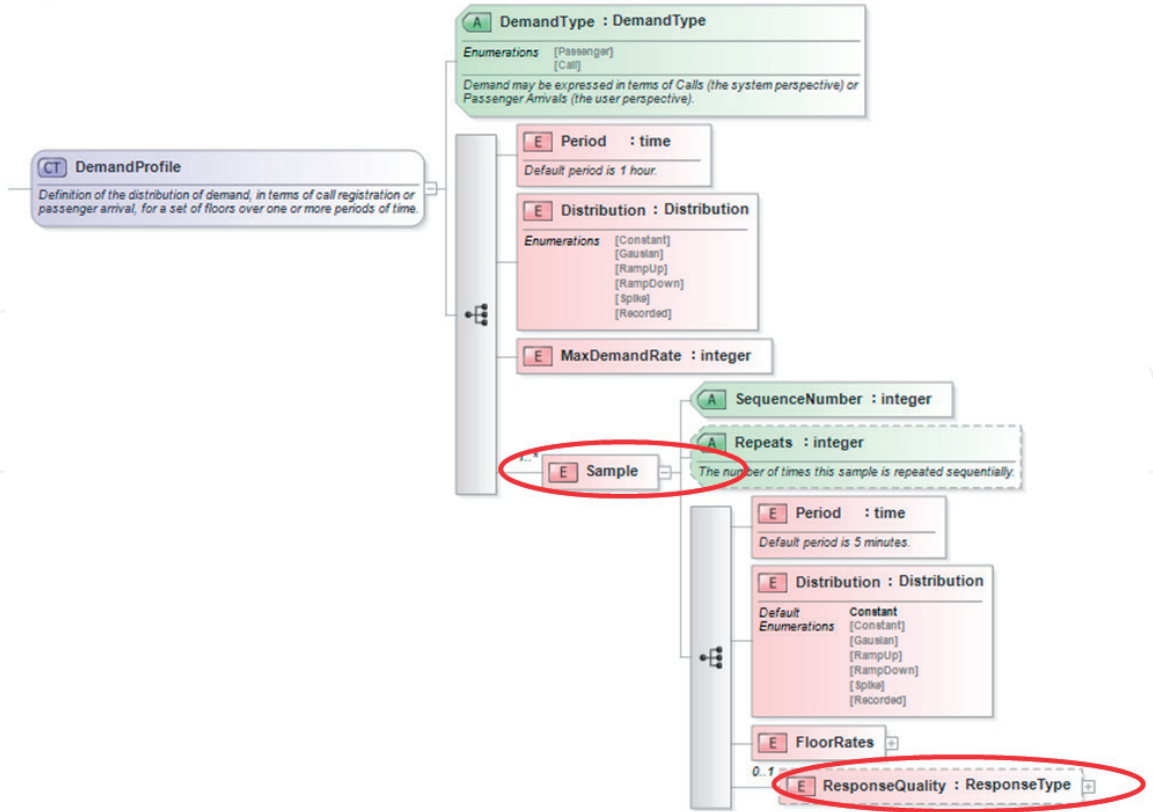


Figure 5.
Definition of a SEIS DemandProfile element.

7.4.3 Using demand profiles throughout the lifecycle of the building

During the building design phase the demand profile can be used to configure a lift traffic simulation, where the simulator will be responsible for generating individual passenger journeys, based on its own internal statistical formula, over each period of the demand profile. These simulations can then be used to assess the suitability of the proposed lift system – its handling capacity and its anticipated quality of service.

After the building is constructed and inhabited by the user population, data logging should be set up to record the real DemandProfiles as they occur, which should be compared with those of the design phase to ensure that a good match has been achieved. A match of design against recorded is more readily achieved for destination calls than for direction calls (where some interpretation of the data is required) but the demand profile is compatible with both (even a mix of) call types. Many demand profiles should be recorded to ensure a statistically valid set of results is used and the records should continue over the lifetime of the building to ensure that the characteristics of passenger demand have not changed and the ability of lifts to satisfy it has not deteriorated. If the match is not demonstrated the recorded results should be used to generate further simulations where the parameters of the lift system may be adjusted in the search for a more appropriate configuration. All demand profiles (planned and recorded) should be referenced and co-located in the Building Information Model (BIM level-3) [48] – once it has become standard practice to maintain such a resource for the management of a building and its assets throughout its operational lifecycle.

8. Conclusion

The origins and features of SEIS have been discussed in relation to the increasing interconnection and interaction of building transportation systems with

the buildings and cities which they support. The importance of conforming to a standard definition of the operation of building transportation systems cannot be overstressed and SEIS provides this in an open and accessible format while being compatible with a properly secured environment, allowing commercial confidence and intellectual property to be protected. The examples presented have illustrated how wide and varied is the range of potential applications for SEIS in this domain. In particular the example of a group dispatcher indicates the role that SEIS could play in supporting the interoperation of devices (potentially the ‘things’ in the Internet of Things) where different autonomous elements, which may not have interacted previously, must collaborate.

Author details

Jonathan Beebe* and Ahmad Hammoudeh

1 Jonathan Beebe Ltd, Bury, Lancashire, UK

2 Princess Sumaya University for Technology, Jordan

*Address all correspondence to: jonathan@jonathanbeebe.com

IntechOpen

© 2020 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] thyssenkrupp MULTI. Cable-Less Elevator That Moves Not Only Vertically But Also Horizontally [Internet]. Available from: <https://multi.thyssenkrupp-elevator.com/> [Accessed: 05 January 2020]
- [2] So A, Al-Sharif L, Hammoudeh A. Traffic analysis of a three-dimensional elevator system. *Building Services Engineering Research and Technology*. 2018;**39**(1):5-20
- [3] CIBSE. CIBSE Guide D. Transportation Systems in Buildings. CIBSE Pub. London. 2015/20; ISBN: 978-1-906846-64-0 (book); ISBN: 978-1-906846-64-7 (PDF); Revision expected 2020
- [4] thyssenkrupp. MAX Product Brochure. Available from: https://max.thyssenkrupp-elevator.com/assets/pdf/TK-Elevator-MAX-Brochure_EN.pdf [Accessed: 05 January 2020]
- [5] KONE. KONE e-Link Product Information [Internet]. Available from: <https://www.kone.co.uk/new-buildings/kone-people-flow-intelligence/monitoring-solutions/> [Accessed: 30 March 2019]
- [6] Schindler. Schindler Ahead Product Information [Internet]. 2016. Available from: <https://www.zdnet.com/article/huawei-talks-up-cloud-ambition-rides-iiot-elevators-with-schindler/> [Accessed: 30 March 2019]
- [7] OTIS. OTIS EMS Panorama Product Factsheet. Available from: http://otusazottcdaw1.cloudapp.net/site/ca-eng/OT_DL_Documents/OT_DL_DownloadCenter/Product%20Information%20EMS/EMS%20fact%20sheet.pdf [Accessed: 30 June 2019]
- [8] Mitsubishi. Mitsubishi Electric- Mel Eye Product Brochure. Available from: http://www.mitsubishielectric.com/elevator/products/basic/elevators/control_system/pdf/catalog.pdf [Accessed: 30 March 2019]
- [9] Kollmorgen. Kollmorgen Visual Lift Product Brochure. Available from: http://www.kollmorgen.de/uploads/tx_fwdownloads/VisualLift_EN.pdf [Accessed: 30 March 2019]
- [10] Escalade. iCON Product Information [Internet]. Available from: <http://www.iescalade.com/icon/> [Accessed: 30 December 2019]
- [11] Peters R, Nickerson J. Lift and escalator management systems: Requirements and implementation. In: *Proceedings of the 10th Symposium on Lift & Escalator Technologies*; September 2019. Northampton; 2019. pp. 103-112
- [12] Data Hoist. Analytics and IoT for All Elevators [Internet]. Available from: <http://datahoist.com/index.php/about-us/> [Accessed: 30 September 2019]
- [13] LiftNet-USA. Liftnet Online Brochure [Internet]. Available from: <https://www.liftnet.com/about/liftnet-online-brochure/> [Accessed: 30 December 2019]
- [14] LiftNet-Germany. Product Datasheet. Available from: https://liftnet.org/fileadmin/liftnet1/Docs/LIFTnet/LIFTnet2.0-Datenblatt_2018.1_E.pdf [Accessed: 04 January 2020]
- [15] PRC National Standards Committee of People's Republic of China. GB/T 24476-2017 - Specification for Internet of Things for Lifts, Escalators and Moving Walks. 2018. Available from: <https://www.chinesestandard.net/PDF/English.aspx/GBT24476-2017> [Accessed: 04 January 2020]

- [16] Peters R, Smith R. Analysis of elevator performance and passenger demand with destination control. *Elevator Technology*. 2008;17. Available as reprint from: <https://www.peters-research.com/index.php/support/articles-and-papers/41-analysis-of-elevator-performance-and-passenger-demand-with-destination-control> [Accessed: 06 May 2020]
- [17] Schindler. Schindler myPORT Product Information [Internet]. Available from: <https://www.schindler.com/us/internet/en/mobility-solutions/products/destination-technology/myport-technology.html> [Accessed: 30 March 2019]
- [18] KONE. KONE Access Solutions Product Information [Internet]. Available from: <https://www.kone.co.uk/new-buildings/kone-people-flow-intelligence/access-solutions/> [Accessed: 30 March 2019]
- [19] thyssenkrupp. thyssenkrupp Launches Elevator Interface to Allow Seamless Multi-Level Robot Movement Throughout Hotels, Hospitals and Other Building [Internet]. Available from: <https://storage.thyssenkruppelevator.com/assets/downloads/news/pr/tkE-RoboticsIntervaceinTKE-10172019.pdf> [Accessed: 30 October 2019]
- [20] Gateway. The Magic Mirror which Opens Up the Lift Car to the Outside World Product Webpage [Internet]. Available from: http://gateway.luve.it/?page_id=2 [Accessed: 30 March 2019]
- [21] KONE. KONE Infoscreen Product Factsheet [Internet]. Available from: https://chad.kone.com/Images/factsheet-kone-infoscreen_tcm162-18655.pdf [Accessed: 30 March 2019]
- [22] OTIS. OTIS eView Factsheet [Internet]. 2016. Available from: <https://files.otis.com/otis/en/au/contentimages/otiseViewFactSheet.pdf> [Accessed: 30 March 2019]
- [23] Minerva R, Biru A, Rotondi D. Towards a definition of the internet of things (IoT). *IEEE Internet Initiative*. 2015;1:1-86
- [24] International Telecommunication Union. Overview of the Internet of Things. 2012. Available from: https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-Y.2060-201206-II!!PDF-E&type=items [Accessed: 17 January 2020]
- [25] Atzori L, Iera A, Morabito G. The internet of things: A survey. *Computer Networks*. 2010;54(15):2787-2805
- [26] Raggett D. Using Semantics and Rich Metadata to Bridge IoT Silos—W3C's Work on the Web of Things. 2015. Available from: <https://www.w3.org/2015/12/09-wot-m2m.pdf> [Accessed: 29 December 2019]
- [27] Rumbaugh J, Jacobson I, Booch G. *The Unified Modeling Language Reference Manual*. Reading, England: AddisonWesley; 1999
- [28] Beebe J. Integration of lift systems into the Internet of Things and the need for an open standard information model. In: *Proceedings of the 6th Symposium on Lift & Escalator Technologies*; 21-22 September 2016. Northampton; 2016. pp. 41-52
- [29] ETSI website. Internet of Things [Internet]. Available from: <http://www.etsi.org/technologies-clusters/technologies/internet-of-things> [Accessed: 29 December 2019]
- [30] CAN in Automation. CiA® 417 Series: CANopen Application Profile for Lift Control Systems [Internet]. 2018. Available from: <https://www.can-cia.org/can-knowledge/canopen/cia417/> [Accessed: 10 February 2019]
- [31] BSR/ASHRAE. Proposed Addendum to Standard 135-2016, BACnet® — A Data Communication Protocol for Building Automation and Control

- Networks (Second Public Review Draft). 2016. Available from: http://www.bacnet.org/Addenda/Add-135-2016bs-PPR2-ISC-Draft-3_chair_approved.pdf
- [32] Beebe J. Standard Elevator Information Schema [Internet]. Available from: <http://www.std4lift.info/> [Accessed: 29 December 2019]
- [33] Beebe J. An XML-based standard for communicating elevator information. In: Proceedings of the International Congress on Vertical Transportation Technologies (ELEVCON 2005). Beijing; 2005
- [34] Barney GC, Dos Santos SM. Lift Traffic Analysis, Design and Control. Stevenage, England: Peter Peregrinus; 1977
- [35] Wikipedia. Object Oriented Programming, Encapsulation [Internet]. Available from: https://en.wikipedia.org/wiki/Object-oriented_programming#Encapsulation [Accessed: 11 January 2020]
- [36] Barney G, Cooper D, Inglis J. Elevator & Escalator Micropedia. Incorporating microGuide-D. London: GBA Pub; 2009
- [37] Barney G, Al-Sharif L. Elevator Traffic Handbook: Theory and Practice. 2nd ed. Abingdon, England: Routledge; 2016
- [38] Beebe J. Standard Elevator Information Schema, Example Application - Event Data Log [Internet]. Available from: <http://www.std4lift.info/LiftInfoExampleEventDataLog.pdf> [Accessed: 10 January 2020]
- [39] Beebe J. Standard Elevator Information Schema - Lift Information Type [Internet]. Available from: <http://www.std4lift.info/HTMLSchema/index.html?url=liftinformationtype.html> [Accessed: 10 January 2020]
- [40] Wikipedia. Representational State Transfer [Internet]. Available from: https://en.wikipedia.org/wiki/Representational_state_transfer [Accessed: 11 January 2020]
- [41] Beebe J. Standard Elevator Information Schema - Simple Display [Internet]. Available from: <http://www.std4lift.info/SimpleDisplay/Output.xml> [Accessed: 14 January 2020]
- [42] Wikipedia. Model-View-Controller [Internet]. Available from: <https://en.wikipedia.org/wiki/Model-view-controller> [Accessed: 02 February 2020]
- [43] Beebe J. Standard Elevator Information Schema - Validation Failure [Internet]. Available from: <http://www.std4lift.info/SimpleDisplay/Output.error.xml> [Accessed: 14 January 2020]
- [44] CoAP. Constrained Application Protocol RFC 7252. Available from: <https://tools.ietf.org/html/rfc7252> [Accessed: 05 January 2020]
- [45] Beebe J, Towards A. Global traffic control (dispatcher) algorithm - requirements analysis. Transportation Systems In Buildings. 2018;2:1-18. DOI: 10.14234/tsib.v2i1.147
- [46] CIBSE. CIBSE Guide D. Transportation systems in buildings, Sec 4.7.3. CIBSE Pub. London. 2015/20; ISBN: 978-1-906846-64-0 (book); ISBN: 978-1-906846-64-7 (PDF)
- [47] Peters Research. Elevate™ Traffic Analysis and Simulation Software [Internet]. Available from: <https://www.peters-research.com/index.php/elevate/about-elevate> [Accessed: 14 January 2020]
- [48] Innovate UK and Infrastructure and Projects Authority. Creating a Digital Built Britain: What You Need to Know – BIM Maturity Levels [Internet]. Available from: <https://www.gov.uk/guidance/creating-a-digital-built-britain-what-you-need-to-know> [Accessed: 21 January 2020]