THE STANDARD ELEVATOR INFORMATION SCHEMA – A CASE-STUDY

JONATHAN BEEBE
Jonathan Beebe Ltd
jonathan@jonathanbeebe.com

ABSTRACT— This short paper describes the Standard Elevator Information Schema. The schema, which is open, published and freely available for research and commercial use, aims to provide consistency and improved communication in a rich and complex information domain. It supports diverse activities including design, simulation, real-time control, monitoring and reporting of the operation of passenger elevators. The schema's compatibility with software and networking standards and the technical and architectural advantages of its use are demonstrated in a case-study of a prototype distributed elevator monitoring system employing web-services.

Key Words: : Elevator, Information Schema, Web-service, XML, Remote monitoring

1. INTRODUCTION

The Standard Elevator Information Schema (SEIS) is a mechanism for communicating static information, such as configuration details, dynamic status and events such as calls and car trips, between all manner of systems and users of passenger elevators. It comprises a set of definitions of complex and simple data types and the structures in which they may be used. It is also a formal statement of the information model that is already used, to some extent intuitively, by many of the systems which control, simulate and monitor elevator operation.

2. THE STANDARD ELEVATOR INFORMATION SCHEMA - CONTEXT

The value of this standard is that it formalizes the types of information that are commonly accepted as relevant to elevator operation, so that regardless of the physical medium used to communicate it, the information being sent for example, from an elevator controller is always meaningful to the monitoring station where it is received. It provides an important foundation of knowledge which applications can re-use, leaving developers free to concentrate their efforts on the specific details of their applications - optimizing the control and operation of elevators!

The idea of a standard language for communicating elevator information is not in itself new and several standards have evolved since the early days of applying computers to elevator control. Several commercial products also exist though are not in the public domain. The development of a standard for elevator information was instigated as part of the author's PhD thesis [1]. These ideas were contributed to the REM90 [2] standard, published in 1990, though this still lacked the technology specification through which such information could be specified formally. However, such a standard has only become commercially viable and easy to implement following recent developments in the software industry based on XML technologies.

The variety and complexity of data in the domain of elevator operation and the rate at which it must be processed has, until recently, presented physical and commercial limitations on what can be achieved outside the confines of a research.

Fortunately we now have access to vastly improved processing power and communication bandwidth coupled with sophisticated software particularly related to the Internet, much of which is open, in the public domain and freely available. We have an environment in which it is not only possible, but a real incentive, to create a self-describing and self-validating standard which supports the full scope of the elevator information domain.

The information domain of elevator operation is unusual - it is data-rich and combines

1) a wide variety of types of information - demand as passenger calls, elevator car events and usually, the interaction of a group of elevators working in consort

with

2) a high rate of data generation - many different devices are operating independently, typically with scanning rates of 10 samples per second

and in

3) a configuration which is highly distributed with a high degree of concurrency.

If a standard for describing such a domain is to be effective, it is important that it should be defined by a schema whose elements are atomic and independent of one another. The schema can then build more complex structures by additionally defining the relationships between those elements.

Thus it is that the Standard Elevator Information Schema uses the XML Schema Definition language (XSD) [3] which is itself written in XML, to define simple data types and complex data types as sequenced aggregations of both the simple data types and other complex data types. Constraints are also defined, such as the minimum and maximum number of elements that may occur in a list, ranges of allowed values and the formatting of text strings. These constraints may be used in validating any data purporting to conform to the schema and software libraries are readily available which automate the interpretation of the constraints, allowing such validation to be incorporated easily into any program that handles the data.

Unfortunately, there is not scope in this short paper to attempt a critical comparison, but the existence of other standards, both open and proprietary, must be acknowledged. However, the drive for these is generally different to that of the SEIS and they tend to be concerned more with communication of status and control signals between hardware component parts of elevator systems over a specific network infrastructure.

Meanwhile, XML based schema have been developed to describe other information domains, and the scope of the SEIS has been deliberately constrained so as not to overlap with these.

However, it has a deliberate intention to limit the scope of the SEIS specifically to the domain of elevator operation to avoid the risk of conflict or competition with other standards due to overlap. For example, there is no attempt to cover geospatial information such as building location. Instead, because of its clearly bounded scope, the SEIS will complement and integrate easily with such standards as already exist or may be planned for future publication.

2.1 XML

Basing this standard on XML technology means that it can benefit immediately from a number of third party editor, graphical presentation, reader and validation tools and libraries that are already publicly available. Also, the content of messages conforming to such a standard is guaranteed to be compatible with a wide range of electronic media and does not require some new and proprietary format or protocol for it to be communicated freely over and between a wide variety of heterogeneous technologies. Because the standard is text based it is also easily read by humans when displayed or printed in its native form.

2.1 Scope

The SEIS is a complete information model, fully describing the great variety of information that characterizes elevator operation. It covers four fundamentally different types of information:

1) **Static data** - the configuration of an elevator (e.g. floors served, number of entrances at a floor, rated speed, etc) or group (e.g. the number and designations of floors).

2) **Dynamic data** - the current status an elevator (e.g. floor position, committed direction of travel, registered calls, etc) or group (e.g. registered hall calls).

3) **Event** - Changes of state in an elevator (e.g. doors become open, out of service, etc) or group (e.g. hall call registered, etc). Events are time/date stamped. An exception is a special type of event which indicates unusual behavior, though not necessarily a failure.

4) **Demand** - a profile of passenger demand over a number of sample periods.

These data types are described in detail elsewhere [4], [5].

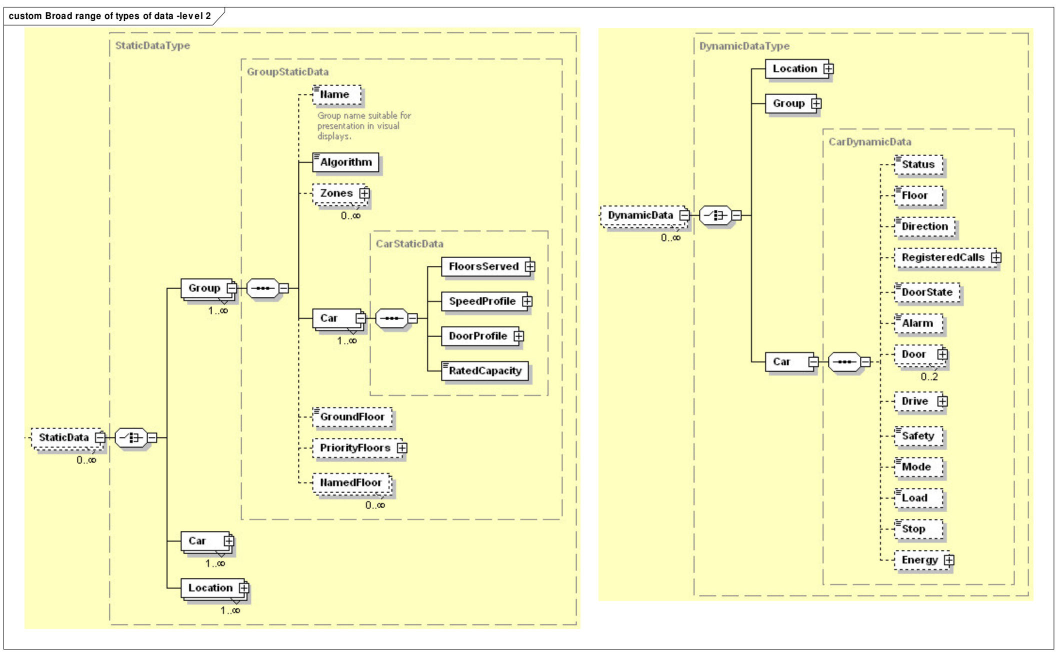

Figure 1. Example Static and Dynamic Data element composition

A wide range of applications is envisaged for the SEIS which includes, but extends far beyond, incorporation into elevator control system products and their peripheral devices (eg call stations and displays). Other possible applications include: elevator simulation, monitoring and reporting, traffic analysis, algorithm design and evaluation, failure identification and alerting, preventive maintenance.

In fact, simply as a formal model for documenting elevator design and operation, the schema can be used on its own without being employed as part of software program.

The scope and variety of such applications means that the SEIS can be used advantageously in a broad range of contexts including research, product development, maintenance, commercial management and building design and architecture.

2.3 Availability

The SEIS is freely available on the Internet [4] to be used as widely as possible, without charge, by anyone who can benefit from it. The only condition of use is that the copyright of the author be acknowledged in any application where it is employed.

The Schema is textual (an XML document) and the "source code" may be modified without restriction, though feedback in the form of comments and suggested enhancements is welcome. It can be downloaded and is then immediately available for use.

3. A CASE STUDY – DISTRIBUTED REMOTE ELEVATOR MONITORING

The second part of the paper presents a case study of the commercial application of the Standard Elevator Information Schema. The subject of this case-study is a project which provides remote monitoring of many widely distributed elevators together with centralized data collection and storage. It was conducted on behalf of the government of a densely populated major city, responsible for managing many hundreds of buildings containing elevators of all descriptions, manufactured and maintained by a multitude of different suppliers and contractors.

The solution discussed in this paper uses standard, open Internet technology and infrastructure and provides secure controlled access, from potentially anywhere in the world, to reporting and analysis facilities.

Many of the international elevator manufacturers have developed internet-enabled remote monitoring systems capable of displaying elevator operation and raising error alerts. However, this commercial user required a single, standard solution that would interface with any manufacturer's equipment and with the potential for integration with the other computerized systems, such as databases of installations and maintenance contracts management, for which they are responsible.

The project was established to build and evaluate a prototype which could be tested against operational elevators. This prototype demonstrates a number of benefits including the opportunity to operate the central station in a secure and fully serviced environment but to provide access to reports and activity monitoring anywhere in the world.

3.1 Objectives

The objectives of this case study were:

1) To build a prototype elevator monitoring system capable of reporting the operation and status of many disparate installations from a variety of manufacturers and suppliers located in a wide distribution of buildings.

2) To establish a standard protocol for communicating information relating to elevator operation that any elevator manufacturer could support which is easily implemented and affordable. This approach should reassure manufacturers that no proprietary or commercially sensitive information would be exposed against their wishes since they are responsible for the generation of all information that is communicated and only they would know how that information is derived.

3) To develop a robust, generic monitoring outstation capable of secure data-collection and transmission.

4) To develop a performant central repository for receiving and storing information from remote elevators that is scalable to support many hundreds of locations all reporting concurrently. Also providing an environment that supports remote analysis and reporting of stored information.

The economic model for this approach becomes viable for both the supplier and user when such a protocol is standardized and adopted internationally whereby:

1) The manufacturer (supplier) has a single product to develop - the outstation (which could, at the manufacturer's discretion, be integrated into the controller equipment). As a result, the manufacturer is assured that no patented or proprietary or commercially sensitive information regarding controller design or operation is exposed.

2) The user has a single application to operate and support, running on completely standard hardware and software environments.

3.2 Implementation

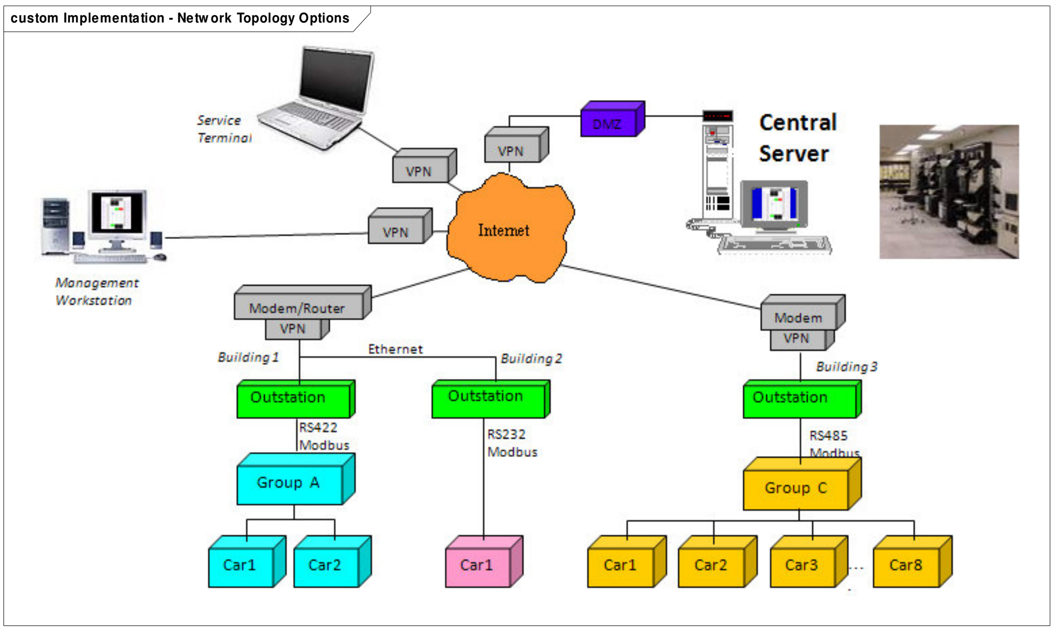

Figure 2. Network Topology

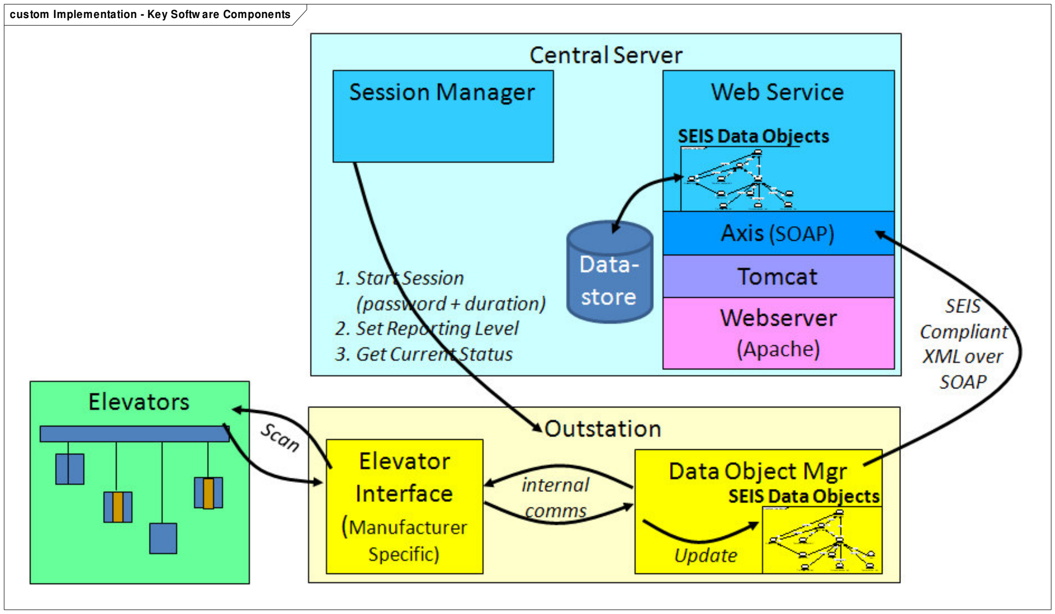

Figure 3. Software Architecture

The principal features of the prototype solution architecture are:

- Standard remote outstations that interface with proprietary elevator controller technologies and which allow an information model consisting of SEIS compliant objects to be populated to represent the current status of each group of elevators. The outstation interprets changes of state in the data objects to generate SEIS event messages which are transmitted to a central server site.

- The central server interprets the incoming event messages to re-construct the information model of the elevators in each building as a set of SEIS data-objects on which it can then perform analysis and archiving of the key features of elevator operation.

This architecture therefore demonstrates SEIS both as the definition of the two information models (outstation and server) and also as the schema for the event messages. In fact there is a third role for SEIS as the schema for the data-store in which the current status of each group of elevators is persisted.

Several different technology infrastructures were evaluated as part of the case study and each evaluation was based, as far as possible, on the SEIS schema. This paper concentrates on the solution that utilized web-services and was implemented using open standards and open software components.

The central site acts as a "master" and allows remote "slave" outstations only to supply information during a specified, renewable time period. Up to 8 elevators may be connected to one remote outstation. The outstation software is structured to minimize the amount and complexity of code changes required to interface to a new type of elevator control equipment. These changes would usually be specified and implemented by the elevator manufacturer to preserve confidentiality of proprietary information, so it is important to make this as easy as possible to implement and maintain. The outstation consists of two software components -

1) elevator interface - which is specific to the control equipment and simply scans for changes of state in the incoming data

2) data interpretation and protocol handling software - which is common and independent of the control equipment or manufacturer, collects information and manages communication with the central site.

The data interpretation software converts scanned incoming data and populates and updates the SEIS conformant data objects, within the remote outstation unit. Changes in these data objects are then communicated as SEIS Events to the central monitoring site.

The Elevator Interface is written in C because that was felt to be a more appropriate language in a real-time environment and one with which elevator control software developers might be more familiar.

On the other hand, the Data Object Manager software is written in Java because of the availability of extensive open-source libraries for processing XML.

The prototype outstation was initially built on a Windows PC but has also been run under UNIX (Linux) to prove its portability.

The Central Server holds the position of master over the network of remote outstations. It establishes a session with each outstation and specifies which types of messages (or data object updates) it is interested in. Sessions are protected by passwords and communications from outstations without a session key are rejected. Once a session is established, the central server does not then need to poll the outstation redundantly just in case some new data has become available, but instead waits for the outstation to communicate events as and when they occur. The central server may also request the complete status of all or specified elevators connected to an outstation so that there is no risk of the server and outstation becoming un-synchronized. The outstation communicates both event and current status messages to a web-service, which is offered by the central server, using the SOAP protocol which is an open standard.

The web-service architecture is well-suited to support simple applications which allow users to query and analyze the centrally held data. However, the case study chose instead to demonstrate the ease with which a real-time presentation of the operation of a group of elevators can be displayed. Since this stored data is persisted as an XML document conforming to the SEIS schema, it is quite easy to transform into HTML and present via a variety of standard web browsers. The Dynamic Display tool shows "real-time" operation of a selected group of elevators in a graphical format, including floor-positions, car direction, door status, car load, car mode, in-service status, and hall calls and car calls.

During tests, this Dynamic Display was run on the central server itself. However, this is not a necessary condition because the server computer is a web-server and therefore the dynamic display may be run on any computer with a web-browser and suitable network connection to the server. Any number of these displays may be operated simultaneously, each displaying a different group of elevators.

3.3 Results

3.3.1 Performance

In the live test environments the software performed well.

Tests showed that the total delay for processing a single event was always less than 500ms. This figure includes the time taken to request data from the elevator controller and could be significantly reduced if the controller were to indicate changes of value/state when they occur rather than only "on-demand" as was the case for these test sites. In fact, the actual processing time for the XML components and round-trip network transmission was of the order of 120ms or less for each event message (the same order of magnitude as the sample period of many elevator control systems). The "change-of-value" mode of operation is much more likely to be the case when the outstation is developed by the elevator manufacturer (as proposed in the Design) since they will have knowledge and access to the internal workings of the controller equipment, which was not the case in the test environment for the prototype.

Time-stamping of events "at source", when they are detected removes the risk of inaccuracies that could result from variable network transmission times and server response times.

Lastly, there is potential for local processing in the outstation to concentrate data at source which could provide a significant further improvement in performance if necessary.

Full load tests still remain to be performed as the necessary resources were not available in the prototype environment.

3.3.2 Simple design based on proven technology

The results of the prototype tests proved that the design was in fact easy to implement. In the main, free, publicly available third party software components were used to build the web-server and a similarly free integrated development environment was used to automate the production of Java code to create the client and server elements of the web-service and the data objects. This leaves the software developer free to concentrate on the specifics of how to process the information that has been collected which is the key objective of this system. However, the option remains to purchase a variety of commercial implementations of these components with the benefit of maintenance and support contracts.

3.3.3 Scalable

The design uses web-server technology which is inherently scalable and allows for multiple servers with load-balancing to be used when the number of outstations grows beyond the handling capacity of a single server (as demonstrated by heavily used sites such as on-line booksellers, banking, etc) confirming that the required performance levels of even the largest collection of buildings can be supported by this design.

As the number of networked buildings increases the management of the "static" information about the elevators located in them will become more time-consuming. For this reason the design, allows a remote outstation to communicate the configuration of the elevators to which it is connected. For example, number of elevators, their capacity and speed and floors served.

3.3.4 Secure

The architecture allows the central server to be operated in a protected environment (secure, air-conditioned, uninterruptible power-supply, etc) while allowing the important information to be presented to authorized staff wherever they work.

Consideration of a number of aspects of the security of the system is important. The prototype demonstrated:

1) outstation may only send messages when permitted by the server

2) server may only establish a session with the outstation by authenticating itself with a password.

3) hardware VPN devices establish encrypted point to point "tunnel" connections over the public Internet if required .

These measures ensure that the outstation will only communicate with a "bone fide" server and that those communications remain secret and indecipherable to "eavesdroppers". However, there remains flexibility to configure backup servers to be substituted in the event of a server or power supply failure.

3.3.5 Economic

Some economic considerations:

- **Hardware cost** - The outstation is replicated many times so it is important that it can be implemented economically. The prototype demonstrated that the outstation computer does not need to be powerful and could be a dedicated single board or embedded computer. The use of Java makes the software very portable without need even to re-compile it. So the computer can be chosen on environmental, commercial and manufacturing considerations.
- **Software cost** - Apart from the cost of the computer operating system (Microsoft Windows XP and Linux were both used), all other third-party software components both for development and run-time were free of charge.
- **Specialist skills** - Although the skills required to develop web-services are somewhat specialized, this is already a widely accepted technology.
- **Network infrastructure** - The communication network is completely standard and is designed to be able to operate securely through the standard infrastructure (including access to the public Internet) of the host building.

7 CONCLUSION

A prototype remote monitoring system has been built and deployed that is capable of supporting many hundreds of groups of elevators. The conclusion of the study is that the technology provides a performant, very accessible, scalable and secure environment based on an easy-to-use infrastructure which allows designers to concentrate on the application itself. The elevator information schema supports all the required information both for

1) observing the operation of a group of elevators and

2) cataloguing and analyzing long-term performance and traffic handling patterns.

Significantly, the presentation of a graphical display of elevator activity in real-time demonstrated that the performance of the prototype was compatible with the application and that the chosen technology was appropriate for the information-rich domain of elevator operation.

The viability of the design using an information model and communication messages founded on a standard schema (the Standard Elevator Information Schema) and its resulting performance is proven. Its flexibility provides assurance that it can both accommodate larger networks of elevators and also provide more functional monitoring as it becomes more widely accepted. The use of mainstream technologies makes it both affordable and highly suitable for integration with leading-edge building services monitoring systems as they become commercially available. At the same time, attention to the requirements for secure and private communications means it can be used with confidence in commercially sensitive situations.

This project is not simply about being able to communicate incidents from remote elevators to a central monitoring station but rather to accommodate building owners' needs to manage centrally a large number of buildings with a variety of equipment from various manufacturers.

REFERENCES

1. Beebe, J.R. *"Lift Management",* PhD. Thesis, University of Manchester Institute of Science and Technology, UK, 1980.
2. IAEE "*DISD: REM90 Remote Monitoring of Lifts, Escalators and Passenger Conveyors"*. ISBN 0-9513498-4-8, International Association of Elevator Engineers, London England, 1990.
3. Gao, S., Sperberg-McQueen, C. M. and Thompson, H. S., *W3C XML Schema Definition Language (XSDL) 1.1*, http://www.w3.org/TR/xmlschema11-1/,Worldwide Web Consortium, 2007,
4. Beebe, J.R., *Standard Elevator Information Schema*, http://www.std4lift.info/, 2003-8.
5. Beebe, J.R. "An XML-based standard for communicating elevator information", *Elevator Technology 15, Proceedings of Elevcon 2005*, *Beijing*, International Association of Elevator Engineers, 2005, pp1-10.