Architecturing Future-proof Distribution of Traffic and Travel Information Using TPEG

J. Vlčinský*
Eago systems, spol. s r. o., Ostrava, Czech Republic
** Corresponding author: jan.vlcinsky@cad-programs.com*

ABSTRACT: The article summarizes key recommendations for creating a traffic and travel information distribution architecture, which will function properly and without changes for longer time. Recommendations are based on a work proposing an architecture over a distributing traffic and travel information over digital radio broadcasting and Internet using TPEG formats and protocols. The whole process of architecturing is briefly described starting with the planning scope, expected results and steps to achieve them; and the process of requirement engineering. For the requirements a list of user needs, created for intelligent transportation systems (ITS) by project KAREN, was found to be very useful. Lessons learned from existing traffic information distribution systems are also discussed as they were another source of requirements. Before describing the architecture, unique design concepts of TPEG itself are mentioned, namely the possibility to seamlessly convert a TPEG message from binary format to XML and back; promising location referencing methods incl. AGORA C; and the ability to be backward compatible even after enhancing the TPEG message to the server for new or updated TPEG applications. All those TPEG design concepts allowed the development of very modular and flexible architecture for distribution. The architecture itself is then briefly outlined by means of selected set of use cases, describing core functionality of reception and distribution; and by a sample deployment model, depicting the possible physical structure of a future working system. Finally, based on the created architecture, recommendations are stated. The system must be designed as a pure distribution channel, changes expected in the future are identified, suitable requirement engineering methods are proposed and the need for monitoring and auditing tooling is highlighted. Open issue is practical availability of the dynamic location referencing method which shall be evaluated in the near future.

KEY WORDS: Architecture, TPEG, DAB, Internet, traffic and travel information

1 INTRODUCTION

The architecture of an information distribution system affects many parties: the information provider, distributor and many information consumers. In project ITS 2005 (Svítek, 2006a), the following risk is identified: “ITS architecture does not include all functions needed for supporting current European policy of road transportation and for ITS services (mainly information services), currently being offered by the public, as well as private, sector. Architecture does not include functions necessary for new products and services, which will be developed in the future.” It is highly desirable therefore to create such an architecture, which provides the needed functionality and also works in a long term view.

1.1 Aim, methodology and results

This article aims to summarize key recommendations for creating a traffic information distribution architecture, which will function properly and without changes for a longer time. Recommendations are based on a detailed architecture design for distributing traffic information using the TPEG protocol as described in (Vlčinský 2008a). First the methodology for designing the architecture is mentioned and then the requirements engineering is described. Before describing the resulting architecture, unique features of TPEG are discussed, as they strongly affected and simplified the task. The resulting architecture is then described and finally key findings are summarized.

1.2 What is TPEG and DAB

In our research, pilot testing of distributing traffic and travel information via digital radio, as well as over the Internet, is planned. Before testing and building the system itself, the architecture had to be defined.

TPEG is a new standardized technology for the encoding and distribution of Traffic and Travel Information, providing various modular toolkit solving applications (e.g. Road Traffic Information, Public Transport Information, Parking Information), transmission methods (e.g. Internet or digital broadcasting technologies, like DAB or DMB), location referencing methods (pre-coded like TMC location tables used in existing RDS TMC or "on the fly" like AGORA-C) and devices (vehicle navigation systems, web browsers, mobile devices).

DAB is a digital radio technology for broadcasting radio stations providing primary service (radio stations) and also data service, usable for distribution of traffic information using TPEG technology.

1.3 Architecture – planning the mission

Architecture has long term impact – most of the appreciated functionality, as well as the unwanted problems, originate here. The whole development process should continually narrow uncertainties and with architecture, there was a big portion of that. The following questions had to be clarified:

What is really meant by architecture? It is said that when asking 10 architects for a definition of architecture, one gets 10 different definitions. There was no use to add another one and fortunately, the answer could be found in outputs of project CONVERGE (Jesty, 1998), which clarifies abstraction levels of architecture and offers good guidance for architecturing ITS systems.

What are the requirements for the future system? As described later, an existing dictionary of user needs from project KAREN and known problems in similar systems were a great source of inspiration.

What steps shall be followed to get the task done*?* After scoping the architecture, Business Process Model (BPM) was created to investigate the context of the system itself (Vlčinský, 2008d). Then, according to Unified Process described in (Arlow 2007), analytical model was built using UML: actors defined behavior of the outer world, use cases described the behavior of the system itself and a class model clarified terms and their relationships. Finally, a deployment model was developed to illustrate a possible physical architecture. Details of used methodology are described in (Vlčinský, 2008b).

2 REQUIREMENTS - NAMING THE OBVIOUS

Neither forgetting someone's needs nor fulfilling wrong ones pleases anyone. Our requirement engineering was based on three sources – user needs for ITS defined in the project KAREN; known problems in similar distribution systems and our will to build an architecture which can last and function for a longer time.

2.1 Shopping list of user needs from KAREN project

Having a list of possible user needs is a big help in requirement engineering, as it concentrates experience and lessons learned from many experts and projects. Project KAREN, defining methodology for ITS architecture in EU, provides such a dictionary (Jesty, 2006). Similarly, project ITS 2005 (Svítek, 2006a), applies the KAREN approach in the Czech Republic, defines methodology and contains many real examples of ITS architecture.

The KAREN dictionary contains about 500 specific needs grouped into more than 40 groups. For the designed architecture 33 of them were selected and are described in (Vlčinský, 2008c).

The general impression is of the dictionary being mature, thorough and promoting an open approach to everyone. To name some examples:

- The Framework Architecture shall not constrain its functionality to be implemented by specific local organizations.
- The Framework Architecture shall support interaction between services provided by private and public bodies.
- The system shall provide emergency, or urgent, information to all road users free of charge.
- The system shall be able to require payment for non-emergency, or non-urgent, information.

2.2 Learning from old and existing problems

Running any distribution system implies involving more different parties who have to perform well to ensure overall performance. Besides a clear definition of responsibilities, some tooling and design concepts might simplify resolving possible problems.

Monitoring shall automatically detect failures and outages and notify responsible persons as it shortens resolution time.

Auditing tools shall provide easy access to information on interfaces between different parties. This is typically achieved by logging data on import/export interfaces; however, providing access to this information via GUI simplifies the task. Digging the information manually from log files is not only cumbersome, but also requires a more privileged and less available resource – a system administrator. The ideal solution is if GUI access is given to all participating parties.

Dependency on one shared map data set, as seen with any pre-coded solution (RDS TMC), but also used in the National Traffic Information Centre (JSDI) in the Czech Republic, can cause some very difficult problems. A scenario whereby all participants update their map data sets in their systems at once is hard to follow and is likely to cause problems with longer resolution times. The biggest problems are if data sets must be exactly the same. A better situation is with TMC location referencing, as it allows at least partial functionality if data sets differ by one version. The best solution would be using some reliable implementation of "on the fly" location referencing. TPEG offers this option using AGORA-C, but implementing is not an easy task. Our architecture defines constraints on functionality with different data sets, however, real implementation has yet to be designed and decided on.

2.3 Trying to anticipate future changes and survive them

Architects are dreaming of architecture, which would last forever, or at least long enough. Reality and bad dreams come with the architecture destroyer – changes. As time goes on the situation around user needs, expectations and technologies changes and the old architecture slowly becomes less and less functional. To fight the changes, the following techniques can be used.

Prohibit any change. As long as architects do not have the power of dictators, this technique does not work much.

Know the changes in advance. The changes which are likely to occur are updating map data sets and implementing new TPEG applications on input and output.

Be transparent and let others deal with the change. The transparency of the transmitting channel is one of the TPEG concepts and we have adopted this approach too. Responsibility for incoming (primary) data is up to the primary data provider. Simply stated, the architected system must prevent trying to take on too much responsibility and do only the minimum necessary. To keep the most of our system untouched by a specific format of processed data, each incoming message gets a generated predesigned metadata, which are then later used to control processing, filtering and distribution. *Metadata* contain information, like TPEG application, source system, message id, time of reception, creation and expiration, urgency, type and subtype (taxonomy), region etc.

Reject wording "Architectural change"; rename it to "System reconfiguration". If the system has to change, it has to change somewhere. The obvious place is a configuration file; plug-ins are also an option as a mean for implementing changed behavior. In the case of our system, plug-ins for converting incoming data to TPEG and for generating metadata for each message are expected.

3 TPEG AS THE UNIVERSAL SOLDIER

Key architectural concepts for TPEG itself are very close to the definition of a universal soldier: they easily change the form appropriate to the situation while not loosing anything of their identity; are expert on knowing where you are; and are able to evolve and grow without harming cooperation with older systems and colleagues.We found these TPEG concepts very handy for architecting our TPEG distribution system.

3.1 Morphing into different forms

TPEG defines two basic formats: binary and XML, which can be converted from one to another. This allows any TPEG message inside of the system to be in a single (XML) format regardless of the format used for final distribution. If the need arises for a new encoding schema (e.g. for other types of digital broadcasting), it is likely that the task will be solved by an additional output filter or converter not affecting the architecture of the whole system.

3.2 Location referencing

Knowing "where" is an essential part of any traffic related message. TPEG is designed from the very beginning with location referencing in mind and has undergone some development over the last years. Methods, using text and simple coordinates, were complemented by other methods using pre-coded referencing as well as dynamic ones.

The distribution architecture expects that consumers could receive messages with location referencing of their choice, if primary information provides enough information for that.

3.3 Adopting future applications

Binary format was developed with backward compatibility in mind so that old devices are able to consume enhanced TPEG messages not known at the date of device production. This goal is achieved by marking each part of the TPEG message in the way, allowing an older device to safely skip unknown sections of data. XML format allows the same.

New TPEG applications, which are likely to be completed in near future, are easy to adopt for distribution. In the case that primary information comes in TPEG format, the distribution system could theoretically distribute it even when not knowing the internal structure of the message.

4 RESULTING ARCHITECTURE

Answering a question “what is a car, daddy?” could be answered by pointing your finger to a picture of a car (physical structure) and explaining “the car goes here and there and does wrrrm, wrrrm” (functionality and behavior).

To describe resulting architecture, functionality will be depicted by means of core use cases and the physical structure will be illustrated by means of a deployment model. A detailed analytical model is described in (Vlčinský 2008e) and a deployment model in (Vlčinský 2008f).

4.1 Use cases describing core functionality

In 0, a key group of use cases participating in reception and distribution is shown. The ovals represent one use case, the rectangle around the ovals shows the border of the modeled system and the line joining the oval with an actor represents the interaction between the system and the world around.

The primary data provider is an actor, which sends source information to the system. The information must be in some agreed format and it is believed to be consolidated (correct and not duplicated).

Use case *Data reception* receives the data, if necessary calls use case *Convert data to TPEG* and hands control over to the use case *Message distribution*. Here, processing starts by generating metadata and if possible, additional location referencing is generated making the message ready for any sort of distribution channel.

Then, processing may fork depending on what distribution channels include the message in their subscription.

Use case *Broadcast TPEG over DAB* encodes the message to binary format, stores it in a collection of messages to be broadcast and keeps them broadcasting.

Use case *Send TPEG over Internet* is actively posting data to url (possibly web service) of the actor *Passive Internet consumer*.

Data can also be published, e.g. placing static XML files or an HTML report on a web server as shown by use case *Publish TPEG on Internet*. The actor *Time* depicts that the publishing is controlled by some scheduler.

Published data can be consumed (downloaded) by the actor *Active Internet consumer requesting data* as shown by use case *Active Internet consumer requesting data*.

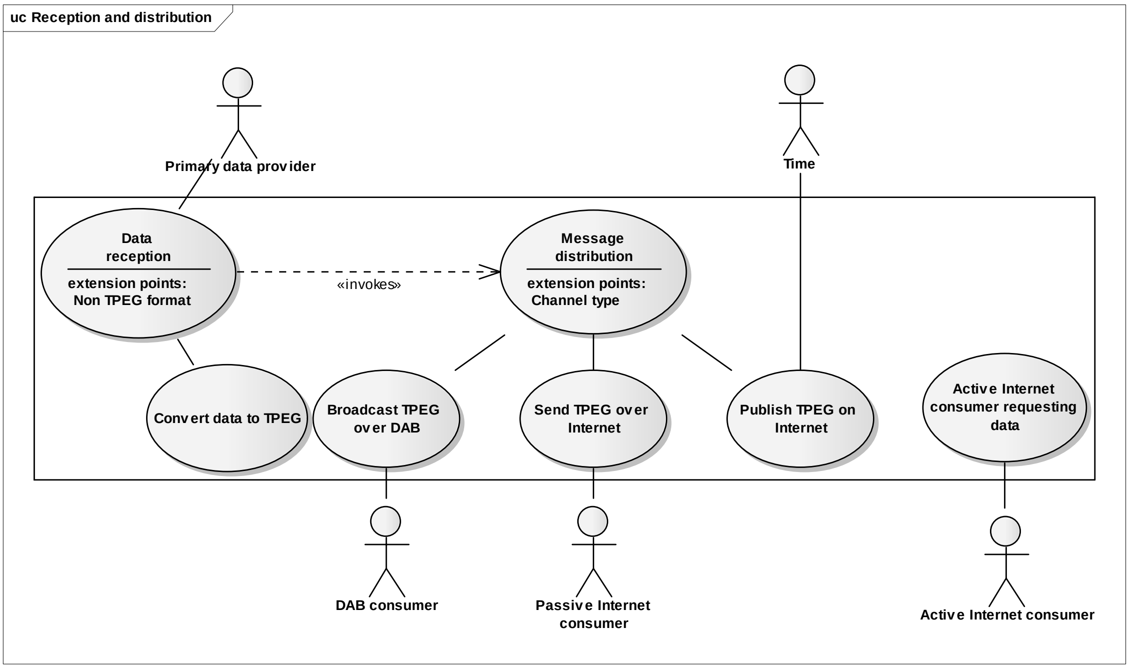

Figure 1:Key use cases participating in reception, processing and distribution

4.2 Deployment model illustrating possible physical structure

0 shows a possible physical architecture reflecting a proposed architecture. Note that, based on the same architecture, different deployments can be built. Take the deployment model as illustrative only.

The 3D box represents a physical device (server, encoder), rectangles inside depict the software component deployed on the device. The crossed arrows symbolize a shared medium (Internet, LAN or Terrestrial broadcasting) interconnecting all devices connected to them by a line. The rectangles around devices are used to show common logical name of that group.

JSDI server (National Traffic Information Center) sends via *DDR* (distribution interface) traffic information to *TPEG distribution* application. Similarly, *IDOS B2B* application provides Travel information (timetables).

TPEG distribution is internally using *Database* and publishes information on a *Web server* or *FTP server.* The w*eb service* on a *TPEG web server* uses data from *TPEG distribution* to respond to a web service request issued from *Application downloading data* placed on *Server of active consumer*. *Application downloading data* can, in the same manner, request data from the *Web server* or *FTP server.*

TPEG distribution can send traffic information to a *Web service – reception* placed on *Server of passive consumer.*

Digital radio broadcasting is via a *DAB encoder* and a *DAB transmitter* “to the air” where the end user with *DAB receiver* can consume the data.

Digital radio broadcasting is also monitored via *Check DAB receiver* and *TPEG checkpoint server*, which closes the loop by providing received data back to the *TPEG distribution* for auditing.

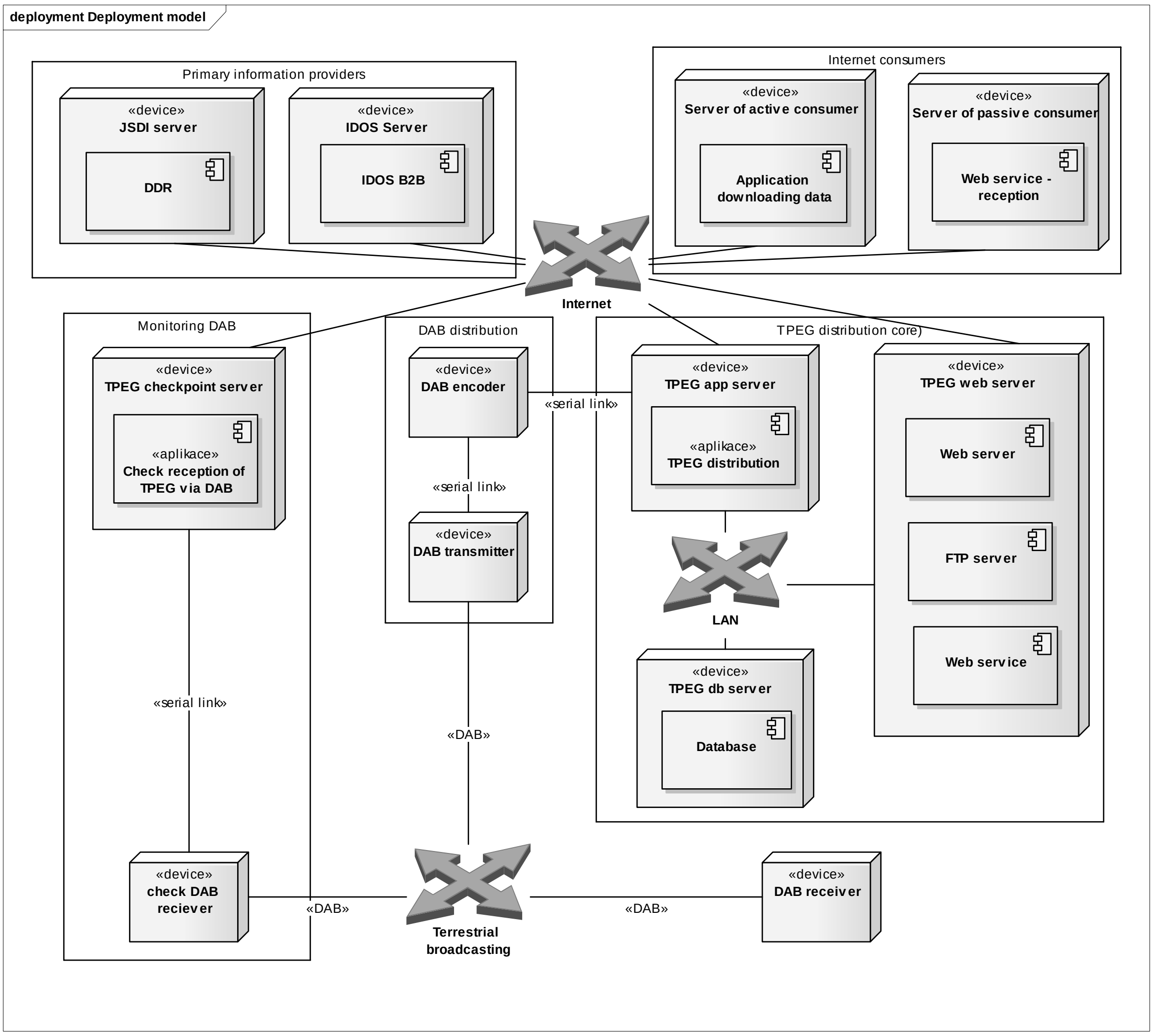

Figure 2: Deployment diagram of possible physical architecture

5 CONCLUSIONS

TPEG design concepts allow designing the architecture of a system for traffic information distribution, which is very likely to last unchanged as new TPEG applications and even distribution channels will appear. To achieve that goal, the distribution system must behave as a pure distribution channel and shall not try to take too much other responsibility.

Changes, which are likely to happen in the future, are in the areas of new and changed input formats, new TPEG applications and new distribution channels. All these changes can be incorporated into the proposed architecture by means of plug-ins solving conversion of incoming data, generating metadata for each message and servicing sending data via a specific distribution channel.

Requirements engineering of the proposed architecture took the advantage of a dictionary of user needs created by the project KAREN and ITS 2005. It is generally highly recommendable to consult these resources for any future work in architecturing, requirements engineering and risk evaluation of ITS systems, as they are very comprehensive and may help prevent some omissions.

Tooling for monitoring and auditing of any information distribution system shall be required, as they shorten problem resolution time and increase the total quality of the provided service.

The currently used distribution systems suffer from using pre-coded location referencing methods like TMC location tables, shared map data sets etc. TPEG seems promising by providing dynamic location referencing called AGORA-C, however, there are still some challenges in the area of technical and commercial availability, which shall be solved, investigated and evaluated in future research.

Assuming dynamic location referencing availability, the proposed architecture has a mechanism for providing location references for consumers in any TPEG standardized location referencing method.

6 AKNOWLEDGEMENT

The study was supported by the Czech Ministry of Transport, within the research project R&D Nr. CG741-139-120 called “Use of new protocols and distribution channels for the distribution of traffic information”.

REFERENCES

Arlow, J.; Neustadt, I., 2007. *UML2 a unifikovaný proces vývoje aplikací Průvodce analýzou a návrhem objektově orientovaného softwaru.* 1st Ed. Brno: Computer Press, 2007. ISBN 978-80-251-1503-9.

Jesty, P. H., 2006. *List of European ITS User Needs* [online], <http://www.frame-online.net/USER NEEDS/D2.02 Database User Needs Issue 2 (for Browsing V3) - for Web Site.pdf> [cit. 2008-06-15].

Jesty, P. H., et al. 1998. *Guidelines for the Development and Assessment of Intelligent Transport System Architectures* [online], <http://www.frame-online.net/Karen_doc/Converge/ArchGuidelines.PDF>[cit.2008-06-15].

Svítek, M., et al. 2006a. *ITS v podmínkách dopravně-telekomunikačního prostředí ČR (projekt č. 802/210/108) zpráva za rok 2005* [online], <http://www.lt.fd.cvut.cz/its/rok_2005/dokumenty/technicka_zprava_its_2005.pdf>[cit.2008-06-15].

Svítek, M., et al. 2006b. *ITS v podmínkách dopravně-telekomunikačního prostředí ČR (projekt č. 802/210/108) Příloha 1: Metodika využívání a rozšiřování ITS architektury při praktickém návrhu projektu.* [online], < http://www.lt.fd.cvut.cz/its/rok_2005/dokumenty/its2005_priloha_1.pdf>[cit.2008-06-15], pp. 37.

Vlčinský, J., 2008a. *DAB a TPEG – Detailní návrh architektury – Základní zpráva.* [online], <http://www.tpeg.cz> in section Dokumenty ke stažení: Závěrečné zprávy aktivit.

Vlčinský, J., 2008b. *DAB a TPEG – Detailní návrh architektury – Metodika.*
Vlčinský, J., 2008c. *DAB a TPEG – Detailní návrh architektury – Požadavky KAREN.*
Vlčinský, J., 2008d. *DAB a TPEG – Detailní návrh architektury – BPM.*
Vlčinský, J., 2008e. *DAB a TPEG – Detailní návrh architektury – Analytický model.*
Vlčinský, J., 2008f. *DAB a TPEG – Detailní návrh architektury – Model nasazení.*