INVOLVING STAKEHOLDERS IN ITS ARCHITECTURE CREATION

Peter H Jesty
Peter Jesty Consulting Ltd, Warwick Lodge, Towton, Tadcaster, LS24 9PB, UK
Tel: +44 (0)1937 833640, Email: phj@peterjesty.com
Richard A P Bossom
System Design Services, Siemens Traffic Controls, Sopers Lane, Poole, BH17 7ER, UK.
Tel: +44 (0)1202 782216, Email: richard.bossom@siemens.com

SUMMARY

Part of the planning process for the deployment of integrated ITS services should be the creation of an ITS Architecture. This architecture will contain the solutions to the problems and aspirations of the various stakeholders. This paper describes a process, based on experience, for use by ITS architects to capture the Stakeholders Aspirations and then create the corresponding User Needs. It highlights the distinction between a problem and its solutions, and also provides advice for how to write, structure and use the User Needs.

INTRODUCTION

The deployment of integrated ITS services around the World is increasing because they provide a cheap and environmentally friendly way of making efficient use of the infrastructure for the transport of goods and people in a user friendly manner. Such deployments need planning, and part of this plan should be an ITS Architecture that provides a top-level view, and a systematic mechanism for integrating these systems and services. However, none of this makes any sense unless the ITS deployments provide the applications and services that the stakeholders need – it is important at this stage to notice that we have not ended this sentence with " … that the stakeholders want".

This paper describes a process for identifying what the stakeholders need. It is based on the work done by the KAREN project as it produced the first version of the European ITS Framework Architecture, combined with the experience of the authors in applying the results to a number of real scenarios.

WHO ARE THE STAKEHOLDERS?

A Stakeholder is any person that affects, or is affected by, either directly or indirectly, the system under consideration. They can be categorised into four generic groups as follows:

- *Want ITS* – These are the problem owners. Their problems may be concerned with traffic and/or transport, and they may be authorities that need to improve the transport environment of their political masters, network owners, operators, or travellers that wish to improve their travelling experience.

- *Make ITS* – These are solution providers. They comprise the component manufacturers and the system integrators.

- *Use ITS* – These are the travellers, system operators and service providers that will come into direct contact with the systems and use them to solve the problems.

- *Rule ITS* – These are the authorities that provide the legislative framework within which the solutions will be created and used, they also include the creators of Standards.

Whilst it may not be necessary to meet with representatives of all the Stakeholders, e.g. the legislative framework should be well known and understood, it is necessary to make sure that they are all identified. However, it is important for all the key *Want ITS* and *Use ITS* people, or their representatives, to be able to provide an input at an early stage in the creation of an ITS architecture. These are the people who will pass judgment on the final systems and the services that they provide, and the sooner their needs are identified the more likely it is that the ITS deployment will be successful.

Finding those that are relevant to a particular deployment can be achieved through discussions with the owner(s) of the transport infrastructure where the ITS will be deployed. They should know the identities of all the stakeholders and be able to advise on when and how they should be approached.

CAPTURING THE STAKEHOLDERS' NEEDS

The project leader, or ITS Architecture champion, needs to arrange one or more meetings between the Architecture Team and the stakeholders to identify their problems, as they understand them, and to obtain their aspirations for the future. This is not likely to be any easy task because people are normally very bad at identifying their real problems (see below). Some may even use the little knowledge that they do have to ask for features that will actually be of little benefit to them.

Experience has shown that an effective way to capture the real needs of the stakeholders, as distinct from what they think they want, is to have one member of the Architecture Team leading the meeting as a facilitator, whilst another makes notes of what is said. The stakeholders should be encouraged to describe their concerns and aspirations in their own words and to leave the technical interpretation to the ITS architects. The facilitator will need to be prepared to make suggestions, and possibly even give a presentation on ITS applications and services that might be relevant to the area.

At some stage it is necessary to get the stakeholders into small groups so that each person has a proper chance to air his or her views, and to get involved. This can either be done by having breakout groups from a larger meeting (additional members of the Architecture Team may be required to help at this point), or by holding a number of small meetings at mutually convenient locations. The stakeholders should be encouraged to describe the problems that they wish to solve, and the services that they want, in a wide-ranging manner and regardless of any deployment issues. It is far better for the Architecture Team to come away with a lot of irrelevant information, than for a critical issue to be overlooked.

During this initial meeting(s) the ITS architects need to capture two types of information.

- The transport and travel problems – these will eventually need to be clearly identified as problems that lie within the scope of the proposed ITS architecture.

- The stakeholders' aspirations for the future – these will eventually need to be stated in a technology independent and functional manner.

By capturing the stakeholders' aspirations in this manner, the Architecture Team will be able to propose solutions that are not necessarily dependant on one supplier, and that are likely to remain valid for many years.

Problems and Solutions

Whilst it might be thought that the difference between a problem and a solution is obvious, in practice very few people seem to be able to get them correct all the time (1). Consider the following two examples, which are both real.

> The inhabitants of a village had identified the speed of the vehicles going through the village as being a major problem, and had asked for some traffic calming measures to ensure that it kept to the notified speed limit. Analysis of the actual speed of the traffic showed that, in fact, almost all the through traffic was travelling at, or near, the notified speed limit; thus illegal vehicle speed was not itself a problem. Further investigation identified a real problem, namely that children needed to be able to cross the road to get to their school bus without difficulty. Analysis of this problem showed that it was not the speed of the traffic that was causing the difficulty, but the volume (i.e. there were too few gaps in the queues of vehicles). Before a solution was agreed, a major new road network was opened a few kilometres away; this reduced the amount of traffic passing through the village, and thus alleviated the problem. However, to keep the villagers happy, the notified speed limit was also reduced.
>
> A large city had decided to create an application that would demonstrate the effectiveness of applying integrated traffic management measures on a major route within the city boundaries. In order to achieve this it was first necessary to get the agreement of all the separate travel and traffic related operators to supply their data to the planned new central system. After several acrimonious meetings, during which the various operators refused to supply their hard earned data in case it might be used for the benefit of a competitor, the project was abandoned. During a post-mortem it soon became apparent that no one had ever stated the traffic or transport problem for which the new system was to be a solution. The operators had therefore never been shown the possible advantages of contributing to the project.

In the first example the villagers had classified the need to enforce the speed limit as the problem, and thus their proposed solution would not have been effective, since speed is in fact only a symptom, or factor, in the road environment. The real problem of needing to enable the children to cross the road the road without difficulty had at least two solutions. The first was to arrange for the school bus to stop on the other side of the road, thus avoiding the need to cross. The second was to reduce the amount of traffic, which is what was done, albeit serendipitously. The solution of reducing the notified speed limit makes little contribution to this problem, if anything it reduces the availability of gaps in the queues of traffic! However, it does contribute to the problem of reducing the likelihood, and consequential severity, of possible accidents.

In the second example the project managers had concentrated on creating a solution without first having identified, or stated, the problem that it was to solve. As a result the meetings with the operators had no clear objectives and reduced to unstructured arguments.

Problems

A problem is the result or goal we must aim for. All too often problems are stated in vague and imprecise ways, e.g. "improved customer service". Projects without clear goals will not achieve their goals clearly (1). Problems need to be stated unambiguously so that it is obvious to everybody what is required, and equally obvious when you have got it.

You should also avoid mentioning solutions in your goal statements. If you can think of several possible alternative specifications for achieving what you want, then what you are specifying are solutions (1).

Solutions

A solution is a possible course of action to achieve the goal; it is a means to achieve the result that we want. Most solutions will have both positive and negative affects on the problem. A solution worth considering is one whose positive contribution to your requirements outweighs the negative ones.

In the unusual, and undesirable, event that it is decided that a particular solution must be used, then that solution becomes a requirement and should be stated as such. Ideally though, problems belong to the problem owners, i.e. those who *Want ITS*, and they are responsible for formulating their problems in terms of goals. Problem solvers, i.e. the Architecture Team, are responsible for finding appropriate solutions to these problems. They will establish the building blocks from which the eventual solutions can be built by those who *Make ITS* (1).

Stakeholders' Aspirations and Agreement

The result of this process will be a document written by the Architecture Team that lists the Stakeholders' Aspirations as a set of problems or goals for which they need solutions over the next planning period. The document should not contain any solutions unless they are part of the requirements (see above).

This document needs to be agreed by all the stakeholders before any further work takes place. Since it will contain all the goals that need to be satisfied by the resulting ITS Architecture and thence the actual deployment, it is essential to ensure that the creation process starts from the correct point, and in the correct direction.

WRITING THE USER NEEDS

Once agreement has been reached over the content of the Stakeholders' Aspirations, the Architecture Team can start to consider the high-level solutions to achieve them. The first stage is to express the Aspirations as a set of User Needs, the creation of which is a technical process that should proceed in a structured manner. The term "User Needs" can be defined as a set of high-level, user oriented, system specifications (2). With hindsight they should really be called "Stakeholder Needs", but we will continue to use the term "User Needs" in this paper to avoid confusion in the minds of those readers who are already familiar with it. Each part of the solution should be written in a stylised form as follows:

The system shall *perform an action*

Let us now consider the reason for this particular structure; first "the system shall". Each part of the solution will make a contribution towards satisfying the desired goals stated in the definition of the problem. By using the phase "the system shall" it is clear exactly what part of the ITS needs to do in order to provide this contribution. If the phase is omitted it may not be clear whether the action should be performed by the system, by the user, or by the user with assistance from the system (in which case what is the assistance that is being given). Consider the sentence:

"Know the reasons why a given travel plan has to be changed."

Is this a command or a statement? If it is a statement, who or what is the subject? If the subject is a traveller then what type of information has to be given, a descriptive statement of what has happened (for information only), or sufficient detail to enable an alternative route to be planned? If the subject is "the system", only, then what is it going to do with this knowledge? If the subject were the person who is awaiting the arrival of the traveller then, whilst this information may be convenient, a revised arrival time would be more useful. Whilst such discussions might be considered "academic", the fact that there are several possible interpretations that may be placed upon this sentence means that several possible interpretations will be put upon it. How many of them will satisfy the aspirations of the stakeholders?

Writing specifications at any level is fraught with difficulty. Not only do they have to be sufficient, but also they must not be ambiguous. The Architecture Team will be obliged to communicate with the stakeholders in their natural language, and yet all natural languages have nuances, or subtleties of meaning – why else is literature studied as an academic subject? Formal mathematical specification languages are now available but, even if the Architecture Team can use one, the stakeholders will almost certainly not understand it and thus not be able to confirm that the resulting statements correspond to their wishes (see below).

It is for these reasons that we strongly recommend that the User Needs be created using the stylised form of "The system shall *perform an action"*. The subject is "the system" and therefore the action must be stated in a manner that describes what the system shall do in order to satisfy the aspirations of the stakeholders. Thus the editor of the User Needs must continuously be thinking of what the stakeholders want (the problem) and how that can be expressed as a series of simple actions (the solution).

Properties of the User Needs

In addition to being written in the stylised form of "The system shall *perform an action*" each User Need should possess the following properties. It should be:

- *Unambiguous* – it must be absolutely clear from the text of "*perform an action*" what it is that is being said. Thus, for example, whenever the word 'information' is used it must be obvious from the context what should be contained within that information, otherwise examples should be given.

- *Technology Independent* – an ITS Architecture should not contain anything that might constrain how an ITS deployment might be implemented, unless this is essential. Since the architecture will have a lifetime of at least five years, and technology is changing faster than that, the only way to keep the architecture valid throughout its lifetime is to write it in a manner that avoids mentioning the use of any specific technology. Thus User Needs should only describe the functions and properties of the solutions, and not the manner of their implementation.

- *Testable* – since it will form the basis of the ITS Architecture, it must be written in a manner such that it is possible to check that each User Need is represented in some manner in the final ITS Architecture. Thus, for example, "The system shall provide a better service" is not a very useful User Need: it is first necessary to agree on the definition of "better" in the context under investigation, and to then include the measurable parameters that have been chosen in a set of suitably worded User Needs. On some occasions it can make sense to write a User Need in the form of a test that must be performed successfully on the system as part an actual deployment.

- *Singular* – multiple User Needs should be split into singular User Needs wherever possible. This is of particular importance for optional features to ensure that they do become truly optional and, where applicable, independent of each other. An ITS Architecture is a tool for planning ITS deployments, and it may not be desirable to implement all the services, or in their entirety, in one single development. Keeping each User Need separate provides maximum flexibility for the users of the resulting ITS Architecture, without them having to ask for the manifestation of part of a User Need. If there is a valid desire to reduce the overall number of User Needs then they should be combined by writing them in a more generic manner, though this will make the resulting ITS Architecture less specific. Thus a User Need of the form "The system shall manage road traffic in such a way that congestion (travel time) may be reduced" can be expanded later, when necessary, without committing to specific congestion reducing techniques at this stage (see below).

- *Traceable and Unique* – it must be possible to trace the manifestation of a User Need, through the ITS Architecture, and into the rest of the ITS development in both directions. Each User Need should therefore be given a unique number.

- *Allocated* – each User Need must be allocated to at least one category of stakeholder; otherwise it is not a User Need. This should be done explicitly.

The easiest way to present the User Needs so that they satisfy these properties is to use a table, as follows:

Ref. Nº	Description	Stakeholders		
		1st Stakeholder	2nd Stakeholder	3rd Stakeholder

Categorising User Needs

The above requirement that each User Needs should be singular normally results in many hundreds of User Needs being written. In order to keep them "under control" they should be grouped together under subject headings. The KAREN project used two methods of grouping the User Needs at the same time.

The first method was to use the Transport Information and Control Systems (TICS) Fundamental Services that had been defined by Working Group 1 of ISO Technical Committee 204 (3). These Fundamental Services were taken from the US National ITS Architecture and, at the time, it was expected that they would be the main point of reference. Since the Fundamental Services do not contain much detail, the KAREN project expanded on them and wrote its User Needs in the manner described above. However, when it tried to categorise these expanded User Needs according to the Fundamental Services it became clear that the result would not be very easy to use, e.g. a few Services had 100's of User Needs, many had very few, and some even had none at all because none of those Services were required in Europe.

The second method was to use a small number of subject headings according to the way that ITS services are often informally grouped. The resulting Groups are of three types, as follows:

- General (Group 1) – this group contains quality characteristics and properties that apply either to the Framework Architecture itself, or to systems and services developed from it.

- Supporting Services (Groups 2-5) – these four groups describe the services that support the operation of the principal ITS services. They cover Infrastructure Planning and Maintenance, Law Enforcement, Financial Transactions and Emergency Services.

- ITS Services (Groups 6-10) – these five groups describe the principal ITS Services themselves. They cover Travel Information and Guidance; Traffic, Incidents and Demand Management; Intelligent Vehicle Systems; Freight and Fleet Management; and Public Transport Management.

The 32 Fundamental Services were then allocated to their corresponding Groups 2 – 10, and then sub-divided into specific topics. The resulting hierarchical numbering system is as follows:

N.M.P.Q

Where:

- N – KAREN Group 1-10 as above
- M – ISO Fundamental Service
- P – Topic
- Q – Unique User Need number

Thus 7.1.7.4 is User Need Nº 4 within Traffic, Incidents and Demand Management (KAREN Group 7); Traffic Control (Fundamental Service); Speed Management (Topic).

Using the European ITS Framework Architecture User Needs

Selecting User Needs

An ITS Architecture is a tool for planning ITS deployments, and it may not be desirable to implement all the services, or in their entirety, in one single development. It is therefore possible to define each development by selecting an appropriate sub-set of the User Needs.

Expanding User Needs

If a User Needs has been written in a high-level generic manner, users of them may eventually need to make them more specific. This can be done quite easily by introducing a fifth level in the hierarchical numbering system and adding extra specific User Needs to expand the definition of a generic User Need.

Adding New User Needs

In the event that a new service, topic or User Need has to be added to the existing list then, whilst the process is simple and obvious, care must be taken if the new list has to be kept compatible with the European ITS Framework Architecture in general, and hence with the other architectures that have been developed from it. For this situation a set of rules has been devised for the configuration management of the European ITS Framework Architecture (4).

Stakeholders' Agreement

Whilst some Stakeholders, e.g. transport executives, will only be interested in the Stakeholder Aspirations, others, e.g. ITS engineers, will also be interested in the content of the User Needs. It is therefore a very good idea to ask these latter stakeholders to confirm that the list of User Needs represents an acceptable solution to the problems or goals that they had described in their Aspirations. Although there will be a strong temptation to start using the User Needs as the reference point for all future work, especially once these stakeholders get used to reading them, the Aspirations must not be forgotten. These are the original problems or goals desired of the new ITS services, and it will be against them that the success of these services will eventually be judged, especially by the transport executives.

WHAT HAPPENS NEXT?

Once the Stakeholders Aspirations and User Needs have been agreed the ITS Architecture creation process can begin in earnest. This is a technical process that uses the User Needs as a starting point. It can be speeded up by using the European ITS Framework Architecture to create the basic structure. Once this has been established the special Needs of this particular ITS deployment can be added. The resulting specific ITS Architecture can be used to produce specifications for the components and infrastructure needed by the ITS deployment. It should also contain sufficient information to permit the production of, say, organisational and cost benefit studies, deployment plans and risk analysis reports. All of these results can be presented to the stakeholders for their comment and acceptance, which they should do by reviewing them against their initial aspirations. This enables the future ITS deployments to be implemented with a high degree of confidence in their ability to deliver the expected services and benefits.

CONCLUSIONS

The involvement of stakeholders in any ITS deployment is vital to its success, since these are the people that will judge its success. This is achieved by getting stakeholders to identify their current problems and future desires, albeit with advice from the Architecture Team. This Team then specifies solutions in the form of User Needs in a stylised manner that details the required functionally and properties of the desired ITS deployment. Since they are also written in natural language the stakeholders can review them and pass comment, thus maintaining ownership and involvement. The writing of the User Needs in a technology independent and singular manner gives maximum flexibility for the system owners as they plan the deployment.

ACKNOWLEDGEMENTS

Some of the work described in this paper was funded by the KAREN support project of the Transport sector of the Telematics Application Programme of the Fourth Framework Programme, an RTD activity of the European Community, and the FRAME-S support action of the Information Society Technologies Programme of the Fifth Framework Programme of the European Community; the ideas contained therein do not necessarily represent the views of the Commission.

REFERENCES

(1) Tom Gilb, Principles of Software Engineering Management, Addison-Wesley, 1988, ISBN 0-201-19246-2.

(2) Peter H Jesty, et al., List of European ITS User Needs, Deliverable D2.02, TR 4108 KAREN support project of the Transport sector of the TELEMATICS APPLICATIONS Programme, Fourth Framework Programme (1994-98), 2000. {www.frame-online.net}

(3) ISO/TC204/WG1, Transport Information and Control Systems – Reference Model Architecture(s) for the TICS sector – Part 1: TICS Fundamental Services, 1998.

(4) Richard A P Bossom and Peter H Jesty, "Configuration Management in the European ITS Framework Architecture Environment", *Proceedings of the 10th World Congress on ITS*, 2003.