

Standardised Model Data Exchange for Dispersed Systems Engineering Design Teams

David Harris

Systems Engineering and Evaluation Centre, University of South Australia

Email: david.harris@unisa.edu.au

Keywords Systems Engineering; Data Exchange; Teams.

Abstract The engineering of systems has always been a multi-disciplinary activity carried out by teams of engineers operating within different domains of expertise. The paper discusses work in developing standardised approaches to systems engineering tool data exchange and developments which will support globally dispersed teams. The paper begins with a brief overview of the domain of systems engineering and the issues involved in exchanging Systems Engineering (SE) models. It discusses the SEDRES Project, a European project which developed an appropriate SE data model and transitioned it into the ISO, where it has become Application Protocol AP-233, within the ISO 10303 STEP environment. Other issues were raised within the SEDRES Project, and the results of experiments into the effectiveness of the system are discussed.

1 SYSTEMS ENGINEERING

As Alice in Wonderland's hookah-smoking caterpillar may have said "When I use the words 'Systems Engineering' they mean exactly what I want them to mean, no more, no less". Almost every text on the topic has its own definition, and there is so little agreement that the final issue of one of the principal Systems Engineering standards, ANSI/EIA 632 [1] uses the title "The Engineering of Systems" and abandons attempts to define Systems Engineering. However, to give some bounds to this paper, the following definition, from the Australian Dept of Defence, will be used:-

Systems Engineering is the design, realisation and implementation of Engineering Systems, where such a system is an integrated set of hardware and software elements which, when operated together, perform a useful process of which the resulting output is greater than the output of the individual parts

The SE Process is described in several standards. One of the most used is ANSI/EIA 632 but there are others, most deriving from early military specifications, in particular MIL STD 499 [9]. An overview of the SE process is shown in Figure 1.

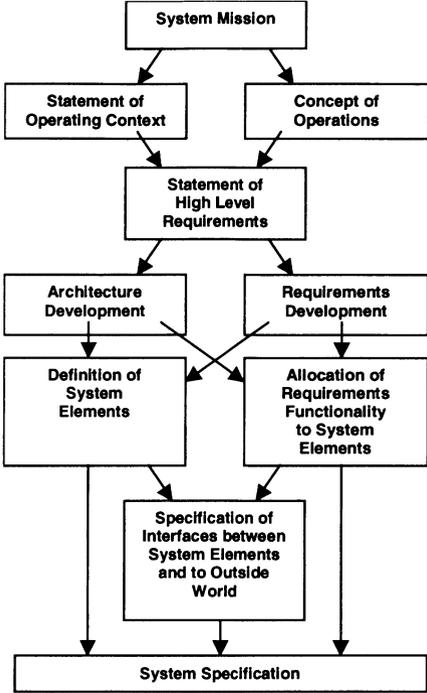


Figure 1 The Systems Engineering Process

Systems Engineering addresses complexity. It takes a holistic view of the system, viewing it from the top down and from the bottom up, leading to an integration of elements to achieve an intended, overall result. Taking a holistic view tends to fly in the face of the Cartesian method of decomposition of large problems into smaller, more easily solvable ones. The problem is that decomposition can lose sight of the connectivity and mutual influence of the elements.

On the other hand, problems must be brought down in size so that, at the lowest level, they can be within the scope of what can be held within one human brain. The methods of systems engineering, then, attempt to reconcile the need for decomposition while retaining the decomposed elements within a structure where their mutual effects remain at the forefront. In fact, integration is one of the major functions of the systems engineering activity.

2 SE DATA EXCHANGE NEEDS

The world of engineering is being dramatically affected by the growth of information technology, and SE is no exception. In fact, SE is affected in two ways - in the systems themselves and in system creation. It is now very difficult to think of a system of any complexity which does not include large amounts of software which control every phase of its operation. There is a tendency in the software community to see large software suites as “systems” in their own right. The view of the more engineering oriented Systems Engineer is that, at the very least, software programs require a (hardware) computer to run them. There is considerable tension over these issues between the two communities.

On the other hand, the creation of all systems, including software-only systems - now requires the use of ever more sophisticated software tools. In many cases, the same tools are used in both fields. The kinds of tools in use reflect the activities involved in SE. They include:-

- System Modelling tools (System Dynamics Modelling, State Transition Tools)
- Requirements Management Tools (Text, Structured text)
- Functional modelling tools (Data flow generation, causal chain modelling)
- Behavioural modelling tools (event driven modelling, time-stepped modelling)
- Simulation tools and environments

There are many others. Interested readers are referred to the web-site of the International Council on Systems Engineering (INCOSE) [9,10]. As SE becomes more tool-centred, the potential for using network-based teams for the engineering of systems is increased. Teams are now frequently distributed over companies, over countries and even over the world. This is not, of course, unique to SE, but SE is entering this environment later than other engineering disciplines, and the rules of the game are still being worked out.

To indicate the data exchange situation faced by the SE team, Figure 1 illustrates the activities involved in a typical prime-contractor to sub-contractor exchange. In this illustration, the Prime works in environment A and releases a part of the total system design to be implemented by the Sub-contractor in environment B. The activities run concurrently, so that this is an implementation of concurrent SE.

1. A piece of the system model (created in Domain A) is identified for transfer to Domain B for development. The broad, outlined arrow indicates that this is a model transfer from Tool A to Tool B. This is contractual information crossing the boundary.
2. Note that the model changes form to reflect the fact that it is now in a different tool domain, with different model representation.
3. Work proceeds concurrently on the system model and the sub-system model. During this process there is a series of information exchanges. The broad arrows, crossing from tool domain to tool domain, carry tool information, but this is not model data (it could be, for example, screen dumps) and is not contractual but is used for explanation between the engineers. The narrow arrows do not penetrate the tool domains but use other media for human-to-human communication, again for explanation, clarification etc.

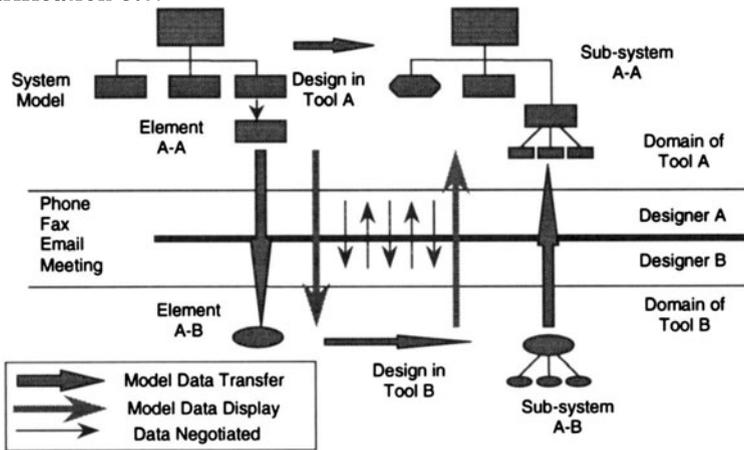


Figure 2 Model data transfer between collaborating network agents

4. At the end of the sub-system design it is shown as being transferred back from Tool B to Tool A, using the broad, outlined “contractual information” arrow. This is a model transfer. Note that the model elements change “shape” from the representation of Tool B (shown here as ellipses) to that of Tool A (shown as rectangles).

There is a lot which can be read into Figure 1, but for the purposes of this paper we will only make a few points:-

- The “thing” transferred is a piece of a system model. This is done on the assumption that this is the best way to transfer Designer A’s intentions to Designer B. Before network based computing, this would be done by written documents.

- The model is represented differently in the two tools. It is almost certain that some information existing in the sending domain will not be representable in the receiving domain and will be distorted or lost.
- The boundary between Tool A and Tool B is a contractual boundary. Too free a flow of communication across this boundary can upset contractual arrangements and configuration management.

In a typical network environment this scenario will be repeated many times, and there will be interactions between them. The transfer between tools requires a translation between the representational models used by the tools. This transfer is performed by a data exchange interface and it can be seen that there will be clear benefits if there are standards for this model data transfer which can be used by interface developers.

3 STATE OF THE ART

3.1 What to transfer

Communicating SE information between team members appears to be more difficult than in other fields such as CAD/CAM. Why is this? Because:

- Communication techniques for SE data are immature. Model representations vary widely, there are no standard communication data models and many tools use proprietary databases. The environment is not conducive to robust data transfer.
- The use of data transfer is not well understood. There is often a view that transfer of data between tools is a “good thing” in itself without looking at its role in the complete SE process. The result tends to be technological focus on transfer of data rather than a focus on the functionality of the team members using the data.
- What the recipient needs is not product data but interface data. Viewed in the light of the whole process it can be seen that the reason for transferring a model is in order to provide the recipient with a clearer statement of the interface between the sender’s model and what the recipient is expected to provide. What the recipient needs is a clear set of interface requirements which may be provided by a model, or by other means.
- The transfer must achieve a sufficient level of shared meaning for the two parties to contribute successfully to the overall task. This requires the transfer of data, and the subsequent connection of this data into the world view of the recipient in such a way that a correct meaning is generated.

3.2 Usage Scenarios

The INCOSE working group on Tool Integration and Interoperability has produced a document detailing the way it sees the Systems Engineer of the future interacting with others in a fully Integrated Systems Engineering Environment (ISEE). This document - INCOSE Report on Scenarios for an ISEE [8] - sets out seven scenarios in which SE data exchange will be used, and details the kind of transactions which will be required. The scenarios identified are:-

Scenario 1: Tool Migration

One-off transfer of a tool database from one (legacy) tool to another (new) tool.

Scenario 2: Parent-Child Integration

This is the situation of prime to sub contractor relationship, which is described in Figure 2.

Scenario 3: Peer to Peer Integration

This scenario describes communication between people at the same level in the hierarchy of the system design network.

Scenario 4: Tool to Tool Traceability

Maintaining traceability between activities carried out in different tools by different people will become important in support of concurrent engineering.

Scenario 5: Data Transformation/Views

The objective here is to provide access to the developing system database in a form which is familiar to individual users. It is intended to provide “windows” onto the database, presenting a range of tailored views.

Scenario 6: Integrated CM Process across tools

In network-based design, configuration management is a major issue. This scenario addresses the activities required to achieve this.

Scenario 7: Navigation

To use the full capability of network-based design the individual engineer should be able to find needed information anywhere on the network. Network navigation activities are covered in this scenario.

3.3 User Requirements

At the highest level of requirements, any SE data exchange standard must:-

- Benefit the SE process.
- Support Integrated Product Teams and Configuration Management.

- Support the concept of a Logical Design Data Repository for network-based design.
- Provide a usable and efficient data exchange capability.
- Lead to International standard.
- Be supported across a number of systems engineering design tools.
- Obtain tool supplier involvement.

4. THE SEDRES PROJECT

4.1 Overview

The System Engineering Data Representation and Exchange Standardisation (SEDRES) Project [13] was an initiative of five major European aerospace companies and three universities, supported by a grant from the Commission of the European Community under the ESPRIT Scheme. Those involved are shown in Figure 3.

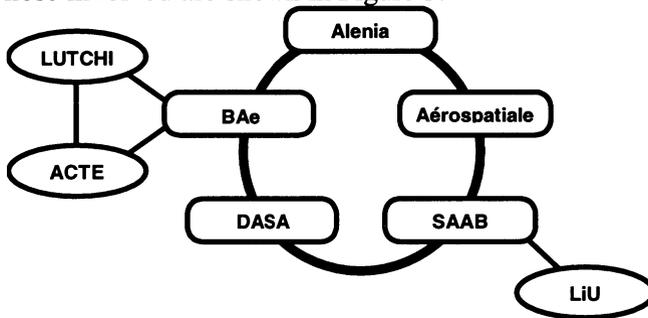


Figure 3 Participants in SEDRES Project

SEDRES addresses the issue of the representation of data held in System Engineering tools with the aim of standardising the exchange of this data [5,6]. SEDRES Project developed and demonstrated a data exchange capability, based on the STEP, ISO 10303, the Standard for the Exchange of Product Data [2,11].

The AP233 data model follows the STEP standard, providing a structure and a set of object classes onto which the elements typically found within a systems engineering tool are to be mapped. A major activity within the SEDRES project was to define a set of appropriate objects to cover the domain of SE tools. The STEP standard defines the methods by which data models will be structured and the way in which data transfer files will be constructed from these objects. The consequence is that an AP233 interface can accept a file (ie, it will not simply generate an error) from any STEP Application Protocol although of course, in most cases, it will not be able to construct a model in the receiving SE tool (Figure 4).

The representation of the model in the Source Tool is mapped, by the SEDRES export interface, onto the SEDRES Data Model, and thence into a standard flat ASCII file for transmission. In the STEP terminology, this file is known as a “Part 21” file. The Part 21 file is then transmitted to the receiving, or ‘sink’ tool, and is received by its SEDRES Import Interface. This maps the Part 21 file onto the SEDRES Data Model, and thence onto the internal representation of the receiving Tool.

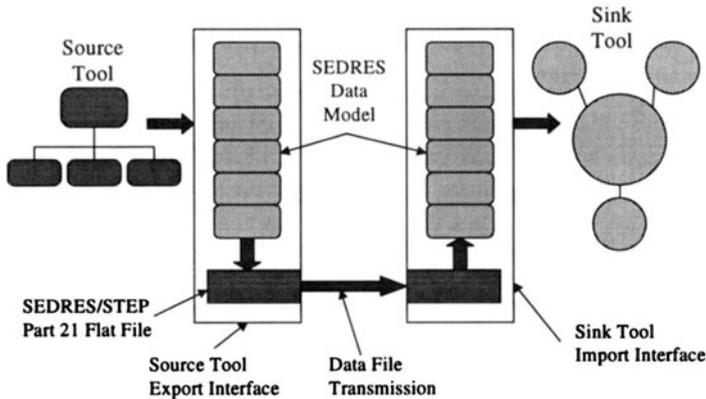


Figure 4 Model data exchange between dissimilar SE tools

Figure 4 illustrates the fact that the model representation in the two tools may be quite different, both in its internal, structural representation and in the way it is presented to the user, via its user interface.

4.2 Implementation

During the SEDRES Project a number of research-level interfaces were developed. Not all covered the same range of capabilities (Table 1).

Table 1. SEDRES exchange interface capabilities

Tool	Vendor	Import	Export	Req	Func	Behav	Config	Gph
LABSYS	AS		<input type="checkbox"/>	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>
CoRE	BAE	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Matrix-X	ISI		<input type="checkbox"/>		<input type="checkbox"/>			
SCADE	Verilog	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
StateMate	i-Logix		<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
StP	Aonix	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
TeamWork	Cayenne	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Workbench	SES	<input type="checkbox"/>			<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>

As can be seen, in some cases only an import or an export interface was developed. Others covered only some of the capabilities of Requirements

management, Function, Behaviour and Configuration Management. The “Graphics” column indicates that at least an attempt was made to transfer the screen placement of objects. In general, the interfaces demonstrated that the SEDRES data model could support the required level of data transfer, but only some of them were developed to the point where their effectiveness could be measured. For a fuller treatment of the accomplishments of this work readers are referred to [6]. Further information on the effectiveness of this data transfer is provided in [4].

4.3 SEDRES to AP-233

The SEDRES Data Model was primarily the work of Linköping University and was used as the basis of the 11 interfaces written for 8 tools (export and import interfaces count separately) during the SEDRES Project. During the currency of the Project a submission was made to the ISO for development of a new standard, based on the SEDRES data model, and falling within the family of STEP standards covered within ISO 10303. This was accepted by ISO as a New Work Item in 1996, with the proposed new standard given the title of Application Protocol AP-233.

4.4 SEDRES-2 and SEDM

The research work of the SEDRES Project is continued in two further projects, one in Europe and another in Australia.

SEDRES-2 [14] carries forward with most of the original partners, with additional new partners and with European grant support. Its aim is to consolidate the work of SEDRES, to carry out further evaluation of its contribution to the SE process, and to extend its uptake beyond the aerospace industries.

SEDM (Systems Engineering Design Methodologies) is an Australian project which uses the work of SEDRES and AP-233 to develop tool data exchange interfaces targeting the needs of its Australian partners. It is linked into SEDRES-2 through the provision of interfaces useable within that project as well as providing another evaluation facility based on database exchanges.

5. EFFECTIVENESS

In evaluating the “effectiveness” of tool-to-tool model data transfer between network-connected team members, we need to address, not the transfer itself, but the use to which it is put in furthering the design task. It

is also useful to compare it with the common current methodology of transfer of text-based requirements documents.

5.1 Requirements Transfer

Although there is currently some use of model data transfer, it is comparatively rare at present. Far more common is the transfer, from one domain to another, of a document setting a list of requirements for the “deliverable” to be supplied. The document may be transferred electronically and may include more than simple text (it may be structured). This does not affect its method of use, which is:-

- The sender builds a model of the system in the tool appropriate to this domain.
- The sender identifies an aspect of this model which is to be passed to another party for supply of an appropriate deliverable item.
- The sender prepares a document listing the requirements for that deliverable. The preparation of this document may be partially automated (it may be a report generated from the tool) but it is important that the requirements are well structured and provide a valid basis for the recipient to carry out the necessary work.
- The recipient will read the requirements document and gain an understanding of what is required of the deliverable. The recipient must re-generate the intent of the sender, or the “meaning” which the sender had in generating the requirements document.
- The recipient will use the meaning so generated to construct a model (by hand) in the tool in use and appropriate to the domain in which the deliverable is to be created.

We can see that the recipient must now derive the “meaning” of the sender by analysing the function of the model which has been sent, and now resides in the receiving tool.

5.2 Measures

In evaluating the effectiveness with respect to the process, a convenient start point is to take a high level view of any process innovation and see if can do the job “better, faster, cheaper”. We will change the order slightly.

Better implies that the received model forms a better basis for work on developing the deliverable than a model input by the engineer on the basis of an understanding generated from the text requirements. This subject has not, to the author’s knowledge, been investigated.

Cheaper seems to imply “faster”, with possibly some contribution from the quality of the model in the receiving tool (if it is, indeed, “better”).

Faster is measurable, and measurements were made in the SEDRES Project and reported in Britton et al [3]. This paper reports time savings in generation of the model in the receiving tool of up to 8-fold. However, it does not take into account other effects (model usability, time taken to comprehend the model etc). It appears that considerable gains are possible, but more work is needed to determine their magnitude.

6. SHARED MEANING

The objective in transferring information (requirements, tool models etc) between participants is to transfer meaning. The "engineer part" of the sending agent (an engineer using a tool) has a meaning - or an intent - within his mind which has been built into a construct within the tool.

Take away the engineer (with his brain) and the meaning disappears and what is left is structured data or text. In a tool-to-tool transfer, it is just this data which is transferred, and usually distorted on the way (data more than text, but text structures can suffer too). It becomes meaning again when the receiving engineer hooks it into his world-view within his brain and creates a meaning from it. The big issue (and the cause of so much heart-ache) is that these two meanings rarely coincide. They don't need to coincide exactly (in fact, they can't) but they need to coincide well enough to get the job done satisfactorily. The important issue is not the receipt of data between tools but the development of a sufficiently coincident meaning. This is further discussed in [7].

It is, of course, fundamentally important to the sharing of meaning that the model data is transferred correctly and that it builds a valid model within the receiving tool. To the extent that this does not occur, the two engineers will spend time unravelling the errors (ie, wasting time) rather than constructing a shared meaning. As a consequence, the work outlined above in obtaining agreement on an international standard for data transfer is critical. However, the next step in generating increased effectiveness is to address the shared meaning issue directly.

7. CONCLUSIONS

Data exchange standardisation for Systems Engineering is taking shape and will most likely be built on the STEP suite of standards, ISO 10303. At present, an appropriate standard has been under development within the ISO framework and will emerge as a Committee Draft in September, 2000. With this standard in place, it can be expected that commercially viable data

exchange interfaces can be built, providing the basis on which a body of users can grow.

With this technology in place, it will be necessary to address the creation of new system designs as a system in itself. With this view it can be seen that many of the issues are not technological but social, with an overlay of business issues. This is the next major area for research.

8. REFERENCES

- [1] ANSI/EIA Standard 632 1999, "Processes for Engineering a System" Electronic Industries Alliance, January 1999
- [2] AP233, 2000 http://www.sedres.com/ap233/sedres_iso_home.html, 2000.
- [3] Britton, J.H, Candy, L and Edmonds E.A, 1998, Software Support for Usability Measurement: An Application to Systems Engineering Data Exchange Development. Proceedings of HCI'98, People and Computers XIII. H. Johnson, L. Nigay, C. Roast.(eds). Springer-Verlag Berlin, 1998, pp 37-52, 1998. ISBN 3-540-76261-2
- [4] Harris, David D and Candy, Linda, 1999. " EVALUATION IN THE SEDRES PROJECT: Measuring the Effectiveness of Model Data Exchange between System Engineering Tools" Proceedings of the Annual Symposium of the International Council on Systems Engineering, INCOSE 99, Brighton, UK, July 1999.
- [5] Johnson, JFE, 1998, "The SEDRES Project (Systems Engineering Data Representation and Exchange Standardisation); extending STEP from structural definition to product functionality. Proceedings of the Annual Symposium of the International Council on Systems Engineering, INCOSE 98, Los Angeles, USA, July 1998.
- [6] Johnson, JFE, 1999, "The Systems Engineering Data Exchange Standard SEDRES / AP-233: Relevance to Requirements Engineering & Capability Maturity". Proceedings of the Annual Symposium of the International Council on Systems Engineering, INCOSE 99, Brighton, UK, July 1999.
- [7] Harris, D D, 2000. "Supporting Human Communication in Network-based Systems Engineering", Proc European Conference on Systems Engineering (EuSEC), Munich, September 2000.
- [8] INCOSE 1997. TIIG Report "Tools Integration Interest Group Report: Scenarios leading towards a concept of operations for an integrated systems engineering environment", Proceedings, INCOSE '97 conference, Los Angeles, July 1997
- [9] INCOSE 2000. <http://www.incose.org>, 2000
- [10] INCOSE TIIWG, <http://www.incose.org/tiiwg/>, 2000
- [11] ISO, 1994, ISO 10303-1, Industrial automation systems and integration - Product Data Representation and Exchange - Part 1: Overview and fundamental principles, ISO, 1994
- [12] MIL STD 499, United States Department of Defense standard.
- [13] SEDRES, 2000a, <http://www.ida.liu.se/projects/sedres/>, 2000.
- [14] SEDRES, 2000b, <http://www.sedres.org>, 2000.