Modelling and Analysis of Socio-Technical System of Systems

Russell Lock
School of Computer Science
University of St Andrews
St Andrews, Fife, UK
rl@cs.st-and.ac.uk

Ian Sommerville
School of Computer Science
University of St Andrews
St Andrews, Fife, UK
ifs@cs.st-and.ac.uk

Abstract— **This paper proposes a novel approach to System of Systems modelling based on the specification of system capabilities. The approach is designed to help end users graphically identify and analyse the hazards and associated risks that can arise in complex socio-technical System of Systems, with particular emphasis on the role of system dependencies. Through a case study this paper shows how the technique can identify the vulnerabilities that may arise within a given System of System configuration; and explore the resilience of a given system when considering evolution and unexpected circumstances.**

Keywords: SoS; Modelling; Socio-technical; Hazard Analysis

I. INTRODUCTION

System of System (SoS) engineering represents a rapidly evolving research area designed to deal with a number of pressing issues within traditional systems development:

- Increasing system complexity.
- The integration of legacy applications / processes.
- The drive for greater business agility (the creation of short term ad-hoc collaborations of systems in particular)
- The growth of trans-organisational systems which operate through the utilisation of multiple independent agencies.

SoS engineering provides one approach to dealing with these issues, albeit at the cost of some degree of control and potentially increased system vulnerability. This paper argues that many high level vulnerabilities can be managed through appropriate SoS modelling, analysis and discussion. For the purposes of this paper we define a SoS as:

> *A collection of systems both technical and socio-technical which pool their abilities to present a more complex system, whilst retaining their individual autonomy.*

The main defining characteristics of SoS are that they are not necessarily subject to centralised control, relying instead on their own internal organisational hierarchy to manage operations, and in many cases their own evolution. This in particular can introduce vulnerabilities within a given SoS as changes to systems affect those dependent on them. Within an ad-hoc SoS it may be possible for individual systems to find a deficiency with regard to their SoS, and bring in additional member systems without necessarily discussing the issue with other SoS members. As such vulnerability is introduced through a lack of visibility between systems.

The main body of systems engineering focuses primarily on managing development within traditional organisational boundaries, with considerable emphasis on centralised control of management and evolution. While techniques such as outsourcing have become a common phenomenon within systems engineering to deal with increasing system complexity and cost, they rely on one party controlling the interaction, and are an insufficient abstraction alone to cover the way in which complex systems such as the NHS operate.

In addition much of the existing SoS engineering research relies on organisations cooperating in the construction of a SoS [1],[2], something which this paper argues cannot be taken for granted when involving systems with overlapping and potentially competing interests. Part of this problem stems from the emergence of SoS, initially from the domain of military systems [3],[4], where the rigidity of control and common purpose simplifies issues.

Many SoS will be ad-hoc, potentially temporary amalgamations of systems which may not be aware of each other despite contributing towards common SoS goals. In these situations a heavyweight analysis using a more traditional means of vulnerability analysis would be complicated, costly, and depend upon highly trained investigators. The technique put forward in this paper is designed to enable the end users themselves to identify, analyse and discuss issues within a structured modelling framework that can be picked up with minimal training.

The structure of the paper is as follows. Section 2 discusses existing modelling techniques that are either used for SoS modelling or have a bearing on Socio-technical system analysis. Section 3 introduces a case study based on the NHS NPfiT program (National Program for Information Technology in the NHS), which is then used as a running example throughout the remaining sections of the paper. Section 4 shows how capabilities and dependencies can be used to provide sufficient structure to form the basis for vulnerability analysis. Section 5 shows how a cut down version of HAZOPs can be used to explore the potential hazards and risks associated with identified vulnerabilities.

Section 6 briefly explores the role of tooling within the approach. Finally section 7 highlights future areas of research and draws conclusions.

II. SoS Modelling background

The modelling of SoS architectures is still in its infancy. Traditional approaches alone are insufficient in this area, partly due to the way in which such systems are being developed, or in many cases simply appearing in ad-hoc form. Traditional systems engineering methods for example rarely lead to systems containing different competing capabilities, or the level of autonomy in terms of evolution routinely seen in SoS development. Although work in the area of SoS modelling can be traced back to the late 1990s [5] the focus has been predominantly on understanding the interactions between technical systems. This represents a logical attempt to extend software component based methodologies and modelling tools such as UML to describe wider technical system interactions within SoS [6].

Architectural frameworks such as MODAF [7] with its 6 different viewpoints (Technical, Strategic, Operational, Service, System & acquisition) and over 30 different individual view diagram types represents an in depth modelling approach which requires significant resources, in terms of both personnel and time. Despite being put forward as a potential SoS modelling technique it focuses on technical rather than socio-technical concerns. It can be argued that approaches such as MODAF rely on a level of specification that may not be attainable in evolving SoS. It is important to note that the definition of the term capability within MODAF differs from that of this paper in that MODAF defines a capability as something you may not be able to currently achieve, but can aim towards in the future; whereas our approach instead focuses on what organisations can currently achieve using the resources at their disposal.

TOGAF [8] is another such enterprise framework which can be used in the design and implementation of SoS. However given its emphasis on the construction of SoS it is of little use when considering existing or ad-hoc evolution of SoS configurations. The approach this paper puts forward is therefore one complementary to the techniques that may be used in the construction of a SoS.

In terms of supporting tools for SoS evolution, products such as HP Universal CMDB are an attempt to enable system administrators to maintain a coherent view of the dependencies present in technical systems within a larger SoS. However, in doing so they only support technical systems implementing their own companies' software. Setting aside the need to describe SoS's combining heterogeneous products, although they are capable of significant depth current tools fall down in supporting the human / organisational side of the SoS.

In particular this research aims to address discussions on a number of key factors for the successful deployment and management of SoS:

- The risks associated with a given SoS configuration.
- Seeking to gauge the volatility of the SoS.
- Supporting the design and ongoing evolution of the SoS.

We have shown in the past that techniques such as Responsibility modelling are useful abstractions for discussing socio-technical issues within individual systems [9],[10],[11]. Responsibility modelling is a graphical modelling technique designed to allow end users to model and explore many of the high level risks within their socio-technical systems without recourse to expensive risk management specialists. As such responsibility modelling is designed to focus on a relatively abstract model of the socio-technical system, containing the minimum information required from end users to aid in problem identification and analysis. It is this type of pared down modelling that we put forward for SoS; within SoS much of the more detailed information about how systems operate would not be known to those constructing and maintaining the wider SoS. However, we cannot simply apply our existing notation to SoS modelling. It is not appropriate to use the unifying concept of a responsibility to describe SoS primarily because:

- A system could be entirely technical, making responsibility of the system an inappropriate concept (The idea of a technical component being truly responsible in the human sense is controversial and considered untenable within this paper).
- It is plausible that an individual system within an SoS may not be considered responsible for a given task. The concept of SoS does not state that those participating systems have to be aware of, or accountable for the "bigger picture".
- Individual systems within a wider SoS are not necessarily transparent in how they operate, preventing responsibility from being applied to anything other than the system itself.

This paper argues that the term capability is a more appropriate abstraction than responsibility when considering system interactions. For the purposes of this paper the definition of a capability is:

The ability to provide some expertise relevant to the wider needs of an SoS

A capability may be independent, or may be depended upon by other systems or the capabilities they put forward into a SoS. A capability does not necessarily imply accountability or usage, and may be subject to operational and environmental constraints. For example:

- System X is capable of providing web support
- System Y is capable of maintaining a website

There are some research papers exploring the use of capabilities within systems engineering including [12], however the authors here focus on applying capabilities to Requirements Engineering for lengthy system developments, rather than ongoing risk analysis within an SoS by a systems users.

To summarise the position of this paper we advocate the use of Responsibility modelling within systems for risk analysis (where more detailed information is available), and Capability modelling between systems of an SoS (where information is likely to be more scarce). The approach outlined in this paper provides a graphical notation focusing on; capabilities, and their associated dependencies and vulnerabilities. It does not compete directly with existing modelling notations given its emphasis on ongoing SoS management from a user perspective, but instead seeks to complement approaches, including responsibility modelling that focus on risk management within individual socio-technical systems. The following section is split to consider each of these, alongside a running example from an existing case study based on the NHS Spine.

III. Case Study Overview

The NPfiT project is one of the most ambitious and complex system procurement programmes ever attempted. It is a project that has been running for a number of years, in which time new systems have been added, others made obsolete. It encompasses the replacement of hundreds of systems individual to GPs, Hospitals and Pharmacies with connected provisioning to allow information dissemination and sharing across a new dedicated NHS network (the Spine). The scope of the remit is such that many of the systems being developed would within other contexts be considered large scale complex systems in themselves.

There has been considerable controversy regarding the way in which systems have been procured and paid for within the NHS, including a well publicised special issue of the Journal of information technology [13] during 2007. One of the general issues relating to NPfiT was that of the lack of socio-technical focus, given that the systems would provide generic replacement facilities to those currently operating proprietary local systems customised to their individual needs [14],[15]. The notations used within this paper are designed not to impose an increasing artificial separation between technical systems and their operators, and instead explores on the effect human operators have as part of a combined socio-technical system.

The systems within the programme represent a large scale SoS in that all the systems are relatively autonomous (though many require the capability to intercommunicate with others, or to make use of fixed or shared resources). The systems themselves are being developed by a number of different companies, and will be deployed within subsets of the NHS, as well as organisations outside its immediate control including GPs and pharmacies. Keeping track of the system at a high level is something of a challenge here due to the number of organisations involved, and the complexity of operations within them. Ultimately however each of the systems has a purpose, and makes available certain capabilities to the wider system. Each of the systems themselves are also dependent on each other, many of them with respect to the TMS (Transaction and Messaging Spine) which provides the communications backbone to many of the systems. It is infeasible to understand the internal structure of all of these systems in order to manage high level risks. A considerable number of vulnerabilities can in fact be mitigated with dependency information from a relatively abstract level. In addition, the recognition of system capabilities, especially in a situation such as this where the system itself is continually changing, can be used to quickly identify options for increased system dependability in the form of fallback options, redundancy etc. The examples used throughout the paper to illustrate how the approach works are drawn from this domain.

IV. Capability Specification

These are the capabilities that a given system puts into a SoS. They can be broken down by type and maturity, though indication of capability maturity does not necessarily mean it is in fact in use / not in use within a given SoS at a given point in time. Within the notation the current use within an SoS of a capability may be denoted by a heavier line than that used for unused capabilities. In this way the modelling can be used to provide a snapshot of the current operation of a SoS. As well as the provision of a capability to a given system, different capabilities and indeed the systems themselves may rely upon each other through high level dependencies.

Capability Maturity

- Current

  A current capability is one which is considered both maintained and available for use within a reasonable time-frame (information on which would be stored within the meta-data for a given capability). It does not imply that a given capability is static in terms of evolution, or that it will always be available, merely that it meets the minimum standards set out to achieve the goals of the SoS.

- Legacy

  A legacy capability is one which has formed an integral part of the SoS in the past but is no longer the preferred method of performing an operation. It may not be maintained to the same levels, if at all, as that of a current capability. Technical systems in particular may retain the capability to perform tasks past the time which they are required to do so. Organisational procedures are decommissioned, but

the training and knowledge held by those involved is retained into the future in a context dependently decreasing manner. There are numerous examples of this type of behaviour in the real world:

- The re-activation of Vulcan bomber refuelling for the Falklands conflict of 1981.
- The retaining by many shops of "zip-zap" manual credit card imprinters as an ultimate fall-back position despite the fact that they have not been in widespread use for decades [16].
- From within university administration, in many departments the position of Head of School is transitory, changing at regular periods, however in the absence of a given Head of School realistically it may make more sense to ask a previous Head of School for advice than not to seek advice at all.

Legacy capability use is not without its risks or indeed costs, but could potentially increase the dependability of a given system. Legacy capabilities have specific training requirements for personnel if they are to be retained in a meaningful way for any length of time. For technical systems there may be sizeable risks related to both throughput and compatibility. However, these capabilities represent the history of the SoS, something which needs to be documented and may be called upon when required.

- Development

  Development capabilities denote the future of the SoS in that they document those capabilities it will have in the future (how far in the future is something you would expect to be denoted within the capabilities meta-data). They may introduce new hazards into a given SoS, as well as remove old ones.

Capability types

Capability types are broken down based on their characteristics into the following categories:

- Technical (denoted T in the notation)

  Technical capabilities allow systems to expose artefacts to the wider system. These could include:

  - Applications / Services
  - OS / Platforms
  - Networks
  - Physical resources such as pc's

- Socio-technical Resources (denoted ST in the notation)

  Socio-technical capabilities provide resources involving both technical and organisational elements. For example a capability to process expense claims could involve both an accounting person for authorisation and checking, and a technical component to provide long term storage of data.

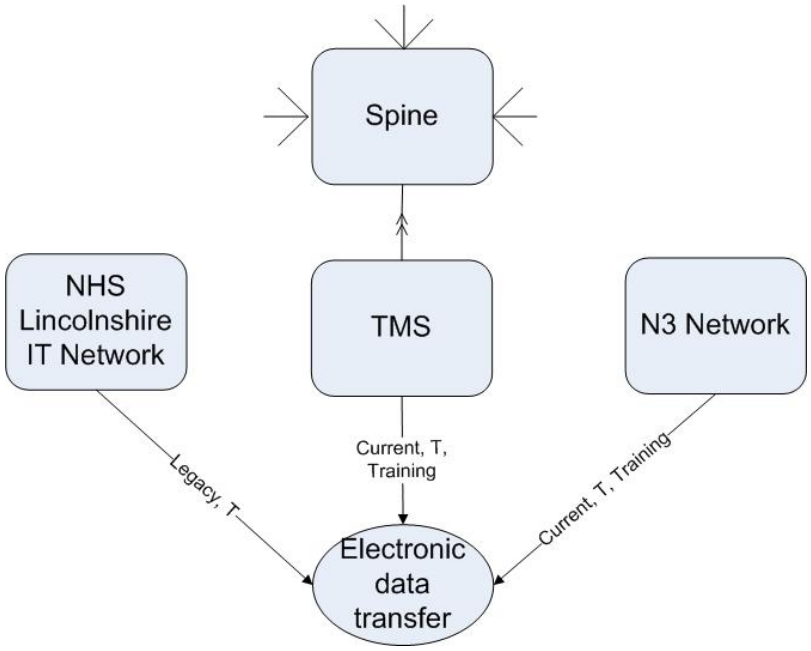

Figure 1: Network provisioning

- Manual (denoted M in the notation)

  Manual processes involve no technical systems. Decision making for example may rely entirely on a given human agent.

- Information Resources (denoted I in the notation)

  An information capability implies that information is provided by a given system, and may be utilised by other systems or individually exposed capabilities.

- Personnel Resources (denoted P in the notation)

  Some organisational units provide people for use in the provisioning of other systems or individual capabilities. For example a motor pool may provide drivers. This separates the person from system. A person has training, and can do a great many things but they are not a system. The training of personnel for given systems is often not performed by those actually using the system but by specific training personnel.

Dependent on the type of capability different types of vulnerability may occur. Entirely technical systems rarely display the same level of dependability when faced with unexpected circumstances that socio-technical or manual human based ones do, due to their deterministic nature. However technical systems do not suffer from fatigue and are more predictable in dangerous situations. Information resources are unusual in that unlike technical and socio-technical interactions an information resource cannot be exhausted.

Capabilities may be of limited lifespan if the system itself has a date for decommissioning etc. By collating this type of meta-data the modeller can seek to gauge the volatility of the SoS, and show its progression in terms of systems evolution.

Figure 1 illustrates the capability approach using an example from the NPfiT programme. A single legacy system has been used in the example, though many others exist in the wider SoS. It shows the capabilities (illustrated using ellipses) of three different communication network systems (illustrated using rectangles) in use within the NHS. The N3 network, and TMS delivered through the new NHS Spine project can both supply data transfer capabilities. The relationship between the spine and TMS is illustrated using a decomposition arrow. This allows systems to be decomposed into sub systems as necessary. In actuality the type of data transfer they support differs in that the new TMS system can deliver data in a more secure manner than N3, but the essence of the diagram is that there are two current viable ways of transmitting data. The NHS Lincolnshire IT Network is a parallel and separate system which is superseded by TMS. Where TMS and N3 provide training for use of their systems in addition to access itself the Lincolnshire system is no longer actively used, and is awaiting decommissioning or, in this case evolution into a new network (COIN). Based purely on capability it appears the dependability of the system rests on the two current systems and the possible use of the backup Lincolnshire system, though in the case of this system it would be geographically constrained. In order to understand how these systems contribute to the wider systems dependability it is necessary to consider what dependencies they have.

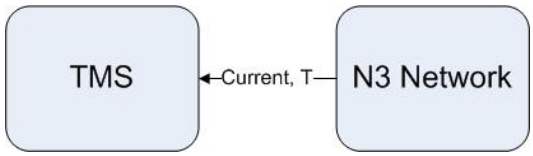

Figure 2: Network dependency

V. DEPENDENCIES

These relate to inter-system and environmental dependencies. If a given system provides a capability to the wider SoS it is possible that other capabilities will depend upon it. For example a system may provide a web service but will depend on another systems network to fulfil it. There are a number of identifiable types, which could be either uni-directional (a capability relies on the provisioning of another) or bi-directional (capabilities or systems relying on each other in order achieve a stable and useful SoS).

In Figure 2 a current network dependency exists between N3 and TMS. This means that TMS cannot be relied upon in the event of N3 failure. We also see that there is no such dependability linking N3 to the legacy Lincolnshire network and can thus assume they are independent in terms of operation.

This paper does not explore the use of scheduling dependencies, ie one capability being utilised before another or in parallel etc, as existing methods from the domain of workflow modelling, including UML event diagrams should be sufficient for use in SoS.

To summarise, an arrow pointing away from a system to a capability indicates the exposure of a capability by that system. Arrows pointing towards systems / capabilities show dependencies.

Figure 3 shows the way in which patient records are shared between sites. The preferred method relies on whether a given GP (General practitioners practice) is GP2GP system enabled or not. If it is then that GP can transfer records electronically using the secure data transfer capability of TMS. If it is not GP2GP enabled, although it has access to the same underlying network (N3) it cannot transfer records electronically (as doing so requires a secure connection), and so has to fall back on a legacy socio-technical process. In this case the socio-technical process involves using a technical computer system to print patient records, or transfer to a CD, and then use a postal envelope (known colloquially as Lloyd George envelopes). The articles are then posted. On arrival at the destination GP data is re-entered into the system. Note that although a given GP

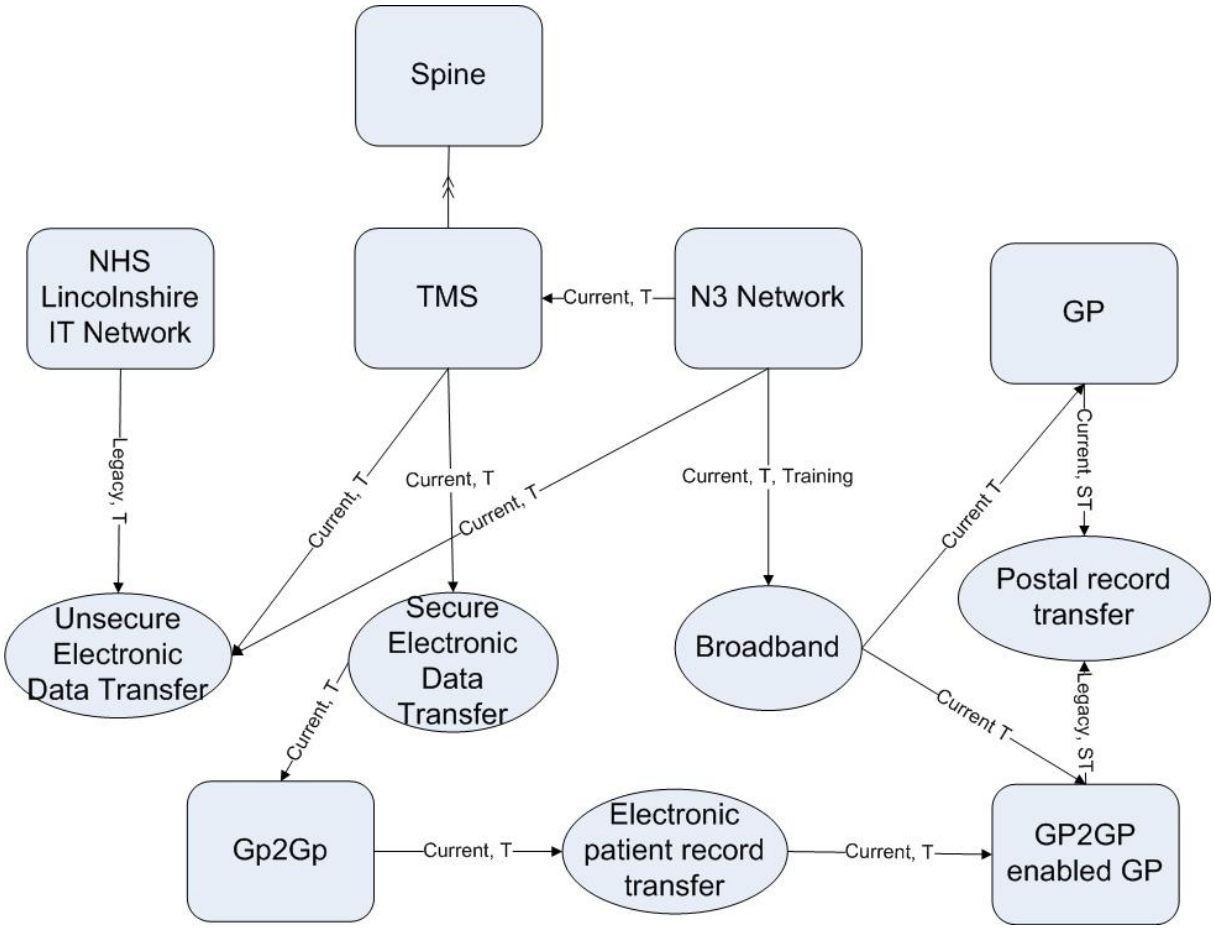

Figure 3: A wider example

may be GP2GP enabled it needs to maintain the legacy mechanism, even though not the preferred mode of operation, in order to communicate with other GPs who are not GP2GP enabled.

As you can see from the diagram a considerable amount of information can be conveyed using the minimum of capabilities and their associated dependencies, however additional meta data may be needed to interpret the diagrams. As a guide the following information may commonly be associated with systems and capabilities

- Organisational information relating to the structure in which the system is managed
- Constraints on use. These may indicate resource/personnel limitations, or environmental concerns.
- Training requirements
- Information relating to the lifespan of the capability. If a given capability is only provided as a legacy alternative a decommissioning date may already have been set
- Information regarding the steps necessary to reactivate legacy capabilities that are no longer in use may also be given.
- Links to external documentation relevant to the capability
- Personnel / Resources Interfaces. For example information relating to points of contact (if known).

Figure 4 shows the type of information that may be associated with a given system/capability within the running example. With the addition of suitable meta information the diagrams themselves can then be used discursively to determine the high level vulnerabilities within the SoS.

VI. VULNERABILITIES

Vulnerability analysis is useful to ascertain the impact of deviation from the norm within the socio-technical structures modelled within a given SoS. The distinction between failure and success is unlikely to be clear cut within a socio-technical system. As such metrics such as MTTF, MTTR etc are of limited use. In these situations it is more

appropriate to apply vulnerability analysis techniques similar to those used in Dependability Cases [17] or Safety Cases [18] for example, to illustrate the strength of the system from the perspective of its processes, training and management. Whilst applicable to both technical and socio-technical systems, Dependability / Safety Cases require expert construction, an unreasonable approach when considering a rapidly evolving SoS. Instead a more simplistic but flexible type of analysis is required for the type of SoS's envisioned (However, we believe Responsibility models could act as a pre-cursor to safety cases in more traditional system architectures; an idea explored further in section VIII). This paper applies an adapted HAZOPs approach similar to that used with our previous work on responsibility modelling; designed to achieve much of the assurance provided by standard HAZOPs but less sensitive to incomplete information, and through the use of more limited generic categories of hazard. It is important to note that by adapting HAZOPs in this way some rigour is inevitably lost (essentially non experts are now performing a part analysis instead of experts comprehensively completing an analysis) but by doing so the approach can be used in more rapidly evolving situations such as SoS. The following shows the information used to construct an individual HAZOPs clause:

- **Target**

  The provisioning of some capability or system

- **Keyword Hazard**
  - Early

    A system / capability which is made available earlier than required may waste resources.

  - Late

    A system / capability which is only made available later than required could have a number of different consequences. For example the wider SoS may use a fallback legacy system, or may merely delay until the system is ready.

  - Unavailable

    A system / capability which is unavailable could have considerable impact on a SoS if what has occurred is not quickly ascertained and potentially mitigated through the use of legacy capabilities

  - Incorrect

    If a system / capability supplies the wrong data to other systems that depend upon it cascade failures could occur across the SoS.

  - Insufficient

    The overloading of systems is a concern in SoS where the needs of the wider system are likely to change over time. The lack of centralised control may hamper the addition of provisioning to deal with increasing demands. One of the questions those modelling must therefore ask is what effect a lack of availability might have within the SoS.

  - Inefficient

    Legacy applications may prove more inefficient in terms of both the time and resources required for operation. By examining the dependencies different systems / capabilities have it is possible to identify what would be affected by a slowdown in one part of the system

- **Risk**

  We define risk in this context as a combination of the probability of the hazard occurring and the severity of the hazard. While probabilistic measurement would give the best basis for comparison and analysis they are likely to be beyond the capabilities of untrained users to either generate consistently or reason about. Instead qualitative statements are preferable as categories specific to a given domain can be formulated and applied in a more consistent manner.

- **Consequences**

  Indicating what would or could occur next. These fields can be used to identify potential cascade failures. Part of this information is captured visually in that the dependencies between different capabilities implies certain impacts on failure, however the effect is not captured graphically and is instead captured under consequences.

- **Actions**

  To indicate any mitigating actions that could be taken if this were to arise. For example would a legacy fall-back be brought in to cover the gap in capability?

The HAZOPs style structure can then be applied to the individual capabilities / systems of the SoS. Dependent on the situation it may be possible to define some hazards applicable to an entire system, or it may require the specification of hazards to individual capabilities. For example in Figure 4 it does not make sense to list a HAZOPs clause stating the hazard associated with N3 being insufficient. Whilst at first glance this would appear to comment on the network availability, the system represents more than that one capability. A given socio-technical system encompasses people, processes etc. In this case in order to discuss vulnerabilities associated with the speed of the network for example it is necessary to write a HAZOPs clause for the appropriate broadband capability instead.

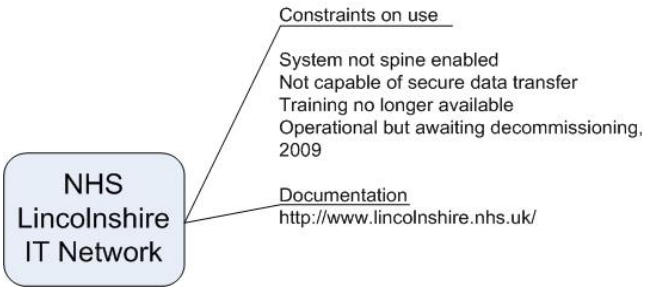

Figure 4: Meta Data example

Dependent on context and the type of capability explored the keywords used may vary. For example it makes little sense to discuss the vulnerabilities of network access through 'broadband' being available early; it is much more likely for vulnerabilities to be caused by 'broadband' being insufficient or unavailable.

Table 1 outlines some of the HAZOPs clauses produced in the modelling of the example NHS SoS. The following points are drawn from the data put forward:

- The failure of the spine TMS system would affect all spine TMS dependent systems, however it would not remove all network connectivity. In fact the main issue would be the loss of 'secure data transfer'. Although not explored in Figure 4 many of the systems involved in this procurement were developed piecemeal. For some applications it may be possible to resort to N3 unsecured communication. These could be shown as legacy capabilities on a diagram exploring the systems of the Spine, space unfortunately precludes the inclusion of this in the paper.
- The failure of N3 would in effect start a cascade failure across the NHS systems. It would affect those systems dependent upon it in the first instance, for example the provisioning of broadband access. It would then affect the TMS systems capability to supply a secure network on the Spine (through the identified dependency) leading to the potential failure of all systems reliant upon it. Cascade failures are straightforward to identify when using the hazard clause structure, as the affected systems are listed within the consequences of the clause. Supporting tools will in future be able to identify and track such changes merely by using identifiable tags to other capabilities / systems within the clauses.
- Superficially the final clause should indicate the possibility to resort to legacy capability usage, ie if the network is running slowly then use manual processes. This however is not listed due to the context and related metadata relevant to the case. In a situation such as this it is necessary to explore the time taken to transfer data using a degraded service as opposed to the alternative, in this case using the postal service. Even in quite extreme situations it is unlikely that the transfer of small amounts of data would be more quickly achieved in this way. Some legacy capabilities therefore are more likely to be used only in the event of catastrophic failure due to the time, resources etc involved.

VII. Development of Tools for Evaluation

The techniques outlined within this paper make use of meta information to convey additional details regarding constraints etc that are not visually associated with the diagrams. Tool support is required in order to link and manage the data involved in an efficient and above all, user friendly manner. The figures used within this paper were generated using a prototype tool based within Microsoft Visio, which is designed to allow users to quickly create templates and design interfaces to access meta information regarding both entities and relationships.

VIII. Future Work and Conclusions

In summary this paper has outlined an approach to SoS modelling based around a given systems capabilities and dependencies, to which hazard analysis can then be applied. In the future we will explore the effect trust has on the

Target	Hazard	Probability	Severity	Consequence	Actions
Spine TMS	Unavailable	Low	High	Failure of spine TMS enabled system which rely on communication	Fall back on N3 or alternate networks where possible. Resort to manual processes for spine enabled systems
Broadband	Unavailable	Low	Very High	Failure of all spine TMS communication through network dependency	Fall back on alternate networks where possible. Resort to manual processes for spine enabled systems
Spine TMS	Insufficient	Medium	Low	Spine enabled information transfers will slow	During development ensure graceful degradation of service quality

Table 1: HAZOPs Clauses

efficiency and complexity of systems. For example does a lack of trust lead to additional checks on input, and what effect does this have on the wider efficiency of the SoS.

We are also in the process of exploring how our existing modeling and analysis techniques could mesh with existing safety / dependability assurance processes including those of safety cases. While our approach was designed to allow non experts to produce useful diagrams, it is possible that these could then act as valuable input when producing professional safety cases. In doing so the safety case constructor could reduce the number and type of questions they would need to ask of stakeholders. The process would also give stakeholders a better understanding of the system in which they operate, acting as preparatory material to understanding the safety case itself.

REFERENCES

[1] J. Hammond. Will it work. Proc 5th International Symposium on Requirements Engineering, RE01, 2001. ISBN: 0-7695-1125-2

[2] G. Carlock, R.E. Fenton. System of Systems (SoS) Enterprise Systems Engineering for Information-Intensive Organizations. Journal of Systems Engineering, vol 4, no 4, 2001. John Wiley & Sons Inc. DOI: 10.1002

[3] P. Redmond. A System of Systems Interface Hazard Analysis Technique. Msc Thesis, Naval Postgraduate School Monterey California.

[4] S. Cook. On the acquisition of System of Systems. Proc INCOSE 2001.

[5] J.S Osmundson, T.V Huynh. A Systems Engineering Methodology for Analyzing System of Systems. System of System Engineering Centre of Excellence

[6] System Modelling Language (SysML) Specification v1.1, November 2008. Object Management Group

[7] MODAF Handbook, Technical Specification for MODAF. Ministry of Defence, 2005

[8] TOGAF Version 9 Enterprise Edition. The Open Group. February 2009 ISBN 9789087532307

[9] R. Lock, T. Storer, I. Sommerville, G. Baxter. Responsibility Modelling for Risk Analysis. ESREL 2009. pp 1103-1109. ISBN: 978-0-415-55509-8

[10] I. Sommerville, T. Storer, R. Lock, Responsibility Modelling for Civil Emergency Planning. Palgrave Macmillan's, Risk Management, awaiting publication 2009. DOI:10.1057/rm.2009.11

[11] I. Sommerville, R. Lock, T. Storer, J. Dobson. Deriving information requirements from responsibility models. Advanced Information Systems Engineering, 21st International Conference, CAiSE 2009. ISBN 978-3-642-02143-5

[12] R. Ravichandar, J.D. Arthur, S.A. Bohner. 40th Annual Hawaii International Conference on System Sciences HICSS'07, ISBN: 0-7695-2755-8

[13] Journal of Information Technology, Palgrave Macmillon. Volume 22, issue 3, september 2007

[14] C. Clegg, C Shepherd. 'The biggest computer programme in the world…ever!': Time for a change in mindset? Journal of Information Technology, Palgrave Macmillon. Volume 22, issue 3 pp. 212-221(10), september 2007. DOI 10.1057

[15] M. Peltu, K. Easen, C. Clegg. How a socio-technical approach can help NPfiT deliver better NHS care. Bayswater Institute, Socio-technical Group report May 2008

[16] D. Stearns. In Plastic We Trust: Dependability and the Visa Payment System. DIRCshop Conference 2006, Newcastle, UK

[17] P.G. Bishop, R.E. Bloomfield. A Methodology for Safety Case Development, Safety-critical Systems Symposium (SSS 98), Birmingham, UK, Feb, 1998

[18] R.E. Bloomfield, B. Littlewood, D. Wright. Confidence: Its Role in Dependability Cases for Risk Assessment, Proc International Conference on Dependable Systems and Networks (DSN-2007), pp338-346, 2007. ISBN: 0-7695-2855-4