An Analysis of the Requirements Traceability Problem

Orlena C. Z. Gotel & Anthony C. W. Finkelstein

Imperial College of Science, Technology & Medicine
Department of Computing, 180 Queen's Gate
London SW7 2BZ  (oczg; acwf@doc.ic.ac.uk)

Abstract

In this paper, we investigate and discuss the underlying nature of the requirements traceability problem. Our work is based on empirical studies, involving over 100 practitioners, and an evaluation of current support. We introduce the distinction between pre-requirements specification (pre-RS) traceability and post-requirements specification (post-RS) traceability, to demonstrate why an all-encompassing solution to the problem is unlikely, and to provide a framework through which to understand its multifaceted nature. We report how the majority of the problems attributed to poor requirements traceability are due to inadequate pre-RS traceability and show the fundamental need for improvements here. In the remainder of the paper, we present an analysis of the main barriers confronting such improvements in practice, identify relevant areas in which advances have been (or can be) made, and make recommendations for research.

[Keywords: requirements traceability, pre-requirements specification traceability, post-requirements specification traceability, requirements engineering practice, requirements traceability tools.]

1:Introduction

Requirements traceability (RT) is recognised as a concern in an increasing number of standards and guidelines for requirements engineering (RE) [12]. This concern is reflected by the various systems that have been developed and a growing research interest in the area [25]. Despite many advances, RT remains a widely reported problem area by industry. We attribute this to inadequate problem analysis.

Definitions of "requirements traceability" are discussed in detail later, though we provide the following for orientation:

• **Requirements traceability** refers to the ability to describe and follow the life of a requirement, in both a forwards and backwards direction.

• **Pre-RS traceability** refers to those aspects of a requirement's life prior to inclusion in the RS.

• **Post-RS traceability** refers to those aspects of a requirement's life that result from inclusion in the RS.

In this paper, we analyse the RT problem in detail. We describe our empirical investigations in Section 2, review the current support in Section 3, examine the underlying causes of the problem in Section 4, and present a framework for addressing these in Section 5. Section 6 lists the main problems facing pre-RS traceability improvements. Section 7 identifies how some of these can be tackled, and Section 8 discusses that aspect of the composite problem which is the subject of our ongoing research agenda at Imperial College.

2:Research method

Numerous data gathering techniques were used to define and analyse the RT problem. The empirical exercises took place over 1 year and involved more than 100 practitioners. Their work areas covered all aspects of development, maintenance and management; their experience ranged from 9 months to 30 years; and the projects they were involved with varied in number, type, and size. A detailed specification for RT support was produced in parallel. Here, introspection helped to identify requirements to support both this activity and its traceability.

2.1:Literature & tool reviews

The literature was surveyed, and over 100 commercial tools and research products were reviewed, to gather viewpoints regarding: what RT is; why it is needed; what problems it involves; and to locate relevant research and development.

2.2:Focus groups

5 semi-structured sessions were conducted. These involved 37 practitioners spread across 5 sites of a U.K. company. Each session lasted 1 hour, was audio taped, and transcribed. The data were used to: consolidate the above; discover how RT problems are overcome (if at all); get suggestions for improvements; and to inform questionnaire design.

2.3:Questionnaires & follow up interviews

A 2-stage questionnaire was used. Stage 1 was designed to rapidly gather data from a wide population of practitioners and to target those from whom more detail could be gathered. 80 were distributed and 69% returned. Stage 2 was tailored to the primary working areas and experiences of individual practitioners, using a reusable pool of questions. 39 were distributed and 85% returned. These provided a deeper understanding of the problems and preliminary requirements to address them. 2 informal interview sessions were subsequently conducted with the respondents. Each lasted 1.5 hours and were used to corroborate their answers, appraise their validity, extract additional information, and to check preliminary analysis.

2.4:Observation & participation

Data were also gathered following the observation of, and some participation in, a variety of RE exercises. For instance, Rapid Application Development workshops were observed,
where comprehensive notes were taken, and any informal documents produced were collected. Our analysis compared such artifacts with the workshop's end products.

3:Current support for requirements traceability

It has been noted that most tools do not cover RT [46], and that few support the RT requirements enforced by DOD STD-2167A [58]. Our survey further indicates that those which do, employ basically identical techniques. They differ mainly in cosmetics, and in the time, effort, and manual intervention they require to achieve RT. Both the type and extent of support provided depends on the underlying assumptions they embed about RT and the particular problems they focus on. However, they often suffer problems due to poor integration and inflexibility. These shortcomings are reflected by the preferred use of general-purpose tools in practice [38].

3.1:Basic techniques

Numerous techniques have been used for providing RT, including: cross referencing schemes [16]; keyphrase dependencies [28]; templates [27]; RT matrices [9]; matrix sequences [4]; hypertext [32]; integration documents [36]; assumption-based truth maintenance networks [53]; and constraint networks [2]. These differ in the quantity and diversity of information they can trace between, in the number of interconnections they can control between information, and in the extent to which they can maintain RT when faced with ongoing changes to requirements.

Additionally, some form of RT can result from using certain languages, models, and methods for development. This is particularly exemplified by: the Requirements Statement Language [10]; process entity-relationship models [24]; the Planning and Design Methodology [42]; formal methods [8]; and Quality Function Deployment [59]. The quality of the resulting RT, however, depends on the rigid adherence to pre-specified procedures and notations for development.

3.2:Automated support

Many commercial tools and research products support RT, primarily because they embody manual or automated forms of the above techniques. We highlight some representative examples below and provide further details in Table 1.

General-purpose tools include: hypertext editors; word processors; spreadsheets; database systems; etc. They can be hand-configured to allow previously manual and paper-based RT tasks to be carried out on-line. This generally involves defining cross references and specifying their update criteria.

Special-purpose tools support dedicated activities related to RE and some achieve restricted RT. For example: the KJ-editor provides traceability between ideas and requirements [56]; and the T tool provides traceability between requirements and test cases [54]. Support is implicit in tool use, by automation of any mundane tasks needed to provide RT, and guidance is limited.

Workbenches contain a collection of the above to support coherent sets of activities. Less restricted RT can be achieved, but the quality depends on the focal workbench activity. Those in which RT and RT management are focal (such as the Automated Requirements Traceability System [21] and the Requirements Traceability and Management System [41]), are what we refer to as RT workbenches. They are typically centred around a database management system, and have tools to document, parse, organise, edit, interlink, change, and manage requirements. Other upper-CASE workbenches which assist RE activities frequently provide some support. This is either from explicit RT components (e.g., a Coupling Module in AGÉ [34]), or from having carried out other activities using its tools (e.g., the Requirements Apprentice [51]). Those workbenches which accept requirements documents, from which to drive the design and implementation, commonly provide coarse-grained RT between requirements and their realisation.

Environments, which integrate tools for all aspects of development, can enable RT throughout a project's life. The basis for integration defines how RT is established: a common language (e.g., Input/Output Requirements Language in Technology for the Automated Generation of Systems [54]); a common structure (e.g., relations of an Entity-Relation-Attribute Model in Genesis [48]); a common method (e.g., Information Engineering Method in the Information Engineering Facility [57]); or (where the tools are combined to support multiplicity) a specialised RT tool or repository structure (e.g., Teamwork/RqT [5]). Those which incorporate third-party tools use powerful repositories and database management systems to relate their products (e.g., the Digital CASE Environment [54]).

4:Why there is still a traceability problem

To date, techniques have been thrown at the RT problem without any thorough investigation of what this problem is. Despite a growth in specialised tools, and inflated claims of RT functionality from tool vendors, their use is not as widespread in practice as the importance of RT would suggest. RT problems are cited where they are used. Following investigations with practitioners, we have found that the RT problem is not perceived to be uniform, and attribute its persistence to diverse definitions and a number of fundamental conflicts.

4.1:Lack of common definition

Definitions of "requirements traceability", either by practitioners or in the literature, were found to be either:

• Purpose-driven (defined in terms of what it should do): "...the ability to adhere to the business position, project scope and key requirements that have been signed off" [Focus group practitioner].

• Solution-driven (defined in terms of how it should do): "...the ability of tracing from one entity to another based on given semantic relations" [47].

• Information-driven (emphasising traceable information): "...the ability to link between functions, data, requirements and any text in the statement of requirements that refers to them" [Focus group practitioner].

• Direction-driven (emphasising traceability direction): "...the ability to follow a specific item at input of a phase of the software lifecycle to a specific item at the output of that phase" [15].

Each definition differs in emphasis and delimits scope. No single one covers all concerns. This has implications for the
Development and use of tools to support RT: how can RT be coherently and consistently provided if each individual has his or her own understanding as to what RT is?

4.2: Conflicting underlying problems

Each practitioner also had his or her own understanding as to the main cause of the RT problem. This finding is reflected in the literature, where it has been attributed to: coarse granularity of traceable entities [47]; immature integration technology [3]; hidden information [52]; and project longevity [42]. The problems that improved RT were expected to address were just as diverse, a finding also reflected in the literature: to support RS evolvability [30]; to enable safety analysis, audits, and change control [24]; to understand systems from multiple viewpoints [14]; and to permit flexible process modelling [20].

These findings demonstrate that: (a) the phrase "RT problem" is commonly used to umbrella many problems; and that (b) RT improvements are expected to yield the solution to further (and even ambitious or conflicting) problems. Complicating this was the observation that, underlying every situation in which RT is required in practice, different user, project, task, and informational requirements come into play. These cumulatively influence the problems experienced, which has further implications for any potential support: how can RT account for all these problems simultaneously?

5: A framework for addressing the problem

To provide a framework in which to locate and address the fundamental cause of RT problems, we first need to establish some shared and working definitions.

5.1: Defining requirements traceability

The definition most commonly cited in the literature is:

- "A software requirements specification is traceable if (i) the origin of each of its requirements is clear and if (ii) it facilitates the referencing of each requirement in future requirements."
development or enhancement documentation" (ANSI/IEEE Standard 830-1984) [26]. This definition specifically recommends backward traceability to all previous documents and forward traceability to all spawned documents. An alternative definition, derived from the word "trace" in the Oxford English Dictionary, is:

- The ability to "delineate" and "mark out" "perceptible signs of what has existed or happened" in the lifetime of a requirement to enable one to "pursue one's way along" this record.

Together, these suggest the following definition for RT:

- "Requirements traceability refers to the ability to describe and follow the life of a requirement, in both a forwards and backwards direction (i.e., from its origins, through its development and specification, to its subsequent deployment and use, and through all periods of on-going refinement and iteration in any of these phases)."

5.2: Pre-RS & post-RS traceability

Our investigations further suggest that RT is of 2 basic types:

- "Pre-RS traceability, which is concerned with those aspects of a requirement's life prior to its inclusion in the RS (requirement production)."
- "Post-RS traceability, which is concerned with those aspects of a requirement's life that result from its inclusion in the RS (requirement deployment)."

Figure 1 shows the typical setting of RT to illustrate these definitions. Note how requirements knowledge is distributed and merged in successive representations; note also the added complication of iteration and change propagation.

![Figure 1: Two basic types of requirements traceability](image)

Forwards and backwards RT are clearly essential. However, we emphasise the pre-RS and post-RS separation, because RT problems in practice were found to centre around a current lack of distinction here. Although both these types of RT are needed, it is crucial to understand their subtle differences, as each type imposes its own distinct requirements on potential support.

The main differences involve the information they deal with and the problems they can assist. Post-RS traceability depends on the ability to trace requirements from, and back to, a baseline (the RS), through a succession of artifacts in which they are distributed. Changes to the baseline need to be re-propagated through this chain. Pre-RS traceability depends on the ability to trace requirements from, and back to, their originating statement(s), through the process of requirements production and refinement, in which statements from diverse sources are eventually integrated into a single requirement in the RS. Changes in the process need to be re-worked into the RS. Changes to the RS need to be carried out with reference to this process, so they can be instigated and propagated from their source. This requires visibility of the subtle interrelationships that exist between requirements early on.

5.3: Support for pre-RS & post-RS traceability

Existing support mainly provides post-RS traceability. Any problems here are an artifact of informal development methods. These can be eliminated by formal development settings, which automatically transform an RS into an executable, and replay transformations following change [18]. In contrast, the issues that pre-RS traceability are to deal with are neither well understood nor fully supported. Post-RS traceability support is not suitable. This generally treats an RS as a black-box, with little to show that the requirements are in fact the end product of a complex and on-going process. Rigid commitment to categories for recording information also make it difficult to represent this process and to account for the dynamic nature of the sources and environment from which requirements are drawn. It has been argued that pre-RS traceability problems will remain, irrespective of formal treatment, as this aspect of a requirement’s life is inherently paradigm-independent [18].

5.4: The need for improved pre-RS traceability

Only recently have these issues been acknowledged [17]. Our empirical findings intensify this concern: most of the problems attributed to poor RT were found to be due to the lack of (or inadequate) pre-RS traceability. Practitioners require techniques to record and trace information related to RS production and revision. Pre-RS traceability was also required to:

- Yield improvements in quality, as previously closed issues (even decisions about how to conduct the RE exercise itself), could be made explicit, possible to re-open, and possible to re-work (so assisting auditing [6], repeatability [29], etc.).
- Provide more economic leverage, as to use and maintain an RS in practice, it is often necessary to reconstruct an understanding of how it was produced (to compensate for invisibility [11]), which is currently error-prone and costly.

6: Problems confronting pre-RS traceability

Having identified insufficient pre-RS traceability as the main contributor to continuing RT problems, and shown how it is likely to be the only contributor in formal development settings, our investigations were re-focused to determine: what improvements in pre-RS traceability would involve; and how these could be realised. These indicated that the main barrier is due to an establish and end-use conflict. By this, we mean that the 2 main parties involved (i.e., those in a position to make it possible and those who require it to assist their work), have conflicting problems and needs (as shown in Figures 2 and 3).

6.1: Problems faced by the providers

- Perceived as an optional extra (and of low priority), so the allocation of time, staff, and resources is often insufficient.
- No allocation and management of the different roles that practitioners need to assume to: obtain and document the required information; organise it; and maintain it.
they cannot predefine their anticipated requirements for it. The pre-RS traceability must be sensitive to contextual needs, but support time and resources, Sufficient who wants it 6.3: Addressing both parties Each end-use context exhibits unique requirements, so • Reliance on personal contact, as there is always something • Inability to predefine how any access to information, and its potential information required, precludes predefinition. • The documentation of required information is no guarantee of its traceability. That which is structured, so it is traceable in many ways, provides no guarantee it will be up to date. • Poor feedback regarding best practice, and little dedicated support (be this clerical, procedural, or computer support), perpetuates the same problems and restricts advances.

Traceability depends on

Working practice
Sufficient resources, time and support
Ongoing cooperation and coordination

Awareness of information required to be traceable

Ability to obtain and document required information

Ability to organise and maintain required information for flexible traceability requirements of end-users (supporting change, restructuring, etc.)

Figure 2: Deconstructing the RT problem for provision

6.2: Problems imposed by the end-users
• A stereotypical end-user cannot be predefined. Their requirements will differ and often be inconsistent.
• The quantity, heterogeneity, and depth of detail of the potential information required, precludes predefinition.
• Inability to predefine how any access to information, and its subsequent presentation, will be required.
• Reliance on personal contact, as there is always something that is out of date, undocumented, inaccessible, or unusable.
• Each end-use context exhibits unique requirements, so problems will exist if end-users do not have the ability to filter and access the information pertaining to RS production that they require under different circumstances.

Traceability
Of what (information) & In what way (access to and presentation of information) depends on

Who wants it (user) Why/when they want it (task) Project characteristics

Figure 3: Deconstructing the RT problem for end-use

6.3: Addressing both parties
The challenge lies in satisfying both parties. For end-users, pre-RS traceability must be sensitive to contextual needs, but they cannot predefine their anticipated requirements for it. The providers must identify and document relevant information, in a (re)usable form (either as a by-product of other work or through more explicit support), but they cannot foresee and address all possible needs. Problems intensify when the same individuals assume both positions. The social nature of the activities involved suggests that technology alone will not provide a complete solution for pre-RS traceability.

7: Solutions to pre-RS traceability problems
An RS was produced to specify what is required to provide and make use of pre-RS traceability. The complexity of these requirements indicate that it would be premature to offer a comprehensive solution. It is a compound problem in need of improvements in many diverse areas. Here, we focus on those basic requirements for which some solutions already exist, and make recommendations for additional research.

7.1: Increasing awareness of information
Studies have revealed what project information is required by those involved in the different phases of development [35]. However, our investigations show that it is not possible to generalise, as both the amount and type required will remain subject to dispute. This issue is generally tackled by pre-specifying the types and structure of information required to assist focused activities, like the gIBIS argumentation scheme for design deliberation [7], but such schemes do not consider the wider informational requirements of all potential RE activities. RT models (as described in [50]), specifically aim to increase awareness of the various stakeholders' needs (primarily to inform the link types to maintain between different information), but use of such models will always be subjective. Problems like these could be assisted through the introduction of dedicated job roles (e.g., independent project documentalist, to augment and unify contributions, encourage objectivity, etc.).

7.2: Obtaining & recording information
Much progress has been made in the ability to obtain and record diverse types of RE information. For example: the history of requirements evolution (REMAP [49]); requirements trade-offs (KAPTUR [1]); explanations and justifications (XPLAIN [44]); a record of collaborative activities (Conversation Builder [33]); and multimedia information [45]. For comprehensive coverage, such tools could be amalgamated in an exploratory workbench (or requirements pre-processor), using suitable integration standards. With additional computer metaphors, so that more RE activities can be carried out on-line, more of this information could be produced as a by-product of main activities. Advances here can be informed through the use of ethnography, or ethnomethodology, to study and describe the details of requirements production, use, and manipulation.

7.3: Organising & maintaining information
To support iterative development, information requires flexibility of content and structure. Relevant work includes the use of: viewpoints as a structuring principle [19]; logical frameworks for modelling and analysing an RS to support gradual elaboration [13]; hypertext to provide explicit visibility of structure and maintain relations [22]; and change models [40]. More research is needed to deal with informal and
unstable information. Much could also be gained from: guidelines to reconceptualise requirements as modular viable systems; the object-oriented representation of self-monitoring multimedia objects; various rollback strategies for persistent repositories; and the creation of explicit job roles to cover the responsibilities of: (a) project librarian (to collect, clean-up, and distribute information); (b) repository manager (to coordinate, control, and maintain information integrity); and (c) RT facilitator (to provide and ensure continual RT).

7.4: Access & presentation of information
RT is predominantly hardwired, predefining what can be traced, and its presentation [21]. Developments in areas such as information retrieval, artificial intelligence, and human computer interaction, are often pertinent. Focused research, like that separating the representation of requirements from flexible presentation, offers potential [31]. Programmable multimedia workstations for end-users would also enable: graphical and textual traces; sophisticated visualisation, to assist activities like impact analysis (i.e., presenting requirements dynamically, using animation, links which light up, etc.); concurrent (global and local) traces; and engaging methods of interrogation. To account for the context of end-use, research is needed to provide flexible RT, where traces can dynamically mature to queries.

8: A research agenda
The current research and recommendations concentrate on throwing increasing amounts and types of information at the pre-RS traceability problem. When such information is generated through adherence to methods, models, or guidelines, it will vary in reliability, as these are rarely used as intended. Any manually-provided information will suffer from subjectivity and incompleteness, as it is difficult to be reflexive, notions of relevance differ, classification schemes are rarely shared, and equal commitment to detail is unlikely. Furthermore, there will always be occasions when the information required will either: not be there; be tailored to a different audience; or not be entirely suited to the purposes at hand.

8.1: Location & access of pre-RS sources
In our investigations, we found that practitioners regularly encountered the above situation. When they do, their fall-back strategy involves identifying and talking to those who can assist. A statistically significant finding was the agreement that the most useful pieces of pre-RS information were: (a) the ultimate source of a requirement; and (b) those involved in the activities which led to its inclusion and refinement in the RS. RT problems (to date), have been solely attacked with techniques that aim to supplant human contact with information. However, even when suitable information is available, the ability to augment this with face-to-face communication was found to be desirable, often essential, and even a fundamental working practice. It is the inability to do just this which we found to underlie many of the continued RT problems.

This finding implies that both *eager* and *lazy* generation of project information is required. By eager, we mean whilst engaged in aspects of RS production. Such information is well suited to the immediate needs of those involved and useful as a later reference point. With time, this static snapshot is less suited to additional needs, and difficult to interrogate. Information generated on need (i.e., lazily and by those originally responsible), can be provided with hindsight, and targeted to specific needs. Both are essential. Without reference to information recorded at the time, to regain context, it would become increasingly difficult to reproduce the required details.

8.2: Location & accessibility: the crux of the problem
Surprisingly, the inability to locate and access the sources of requirements and pre-RS work was the most commonly cited problem across all the practitioners in our investigations. This problem was also reported to be a major contributor to others:
- An out of date RS, as an RS evolves poorly when those originally responsible are not involved in its evolution, or where it is impossible to regain the original context.
- Slow realisation (and deterioration as a result) of change, as the most time-consuming and erroneous part is often the identification of those to involve and inform.
- Unproductive conflict resolution, decision making, and negotiation, as most tools supporting these activities do not help to identify or locate the essential participants.
- Poor collaboration, as the invisibility of changing work structures and responsibilities makes it difficult to transfer information amongst parties; integrate work; and assign work to those with relevant knowledge and experience.
- Difficulty in dealing with the consequences when individuals leave a project and with the integration of new individuals.
- Poor reuse of requirements, as reuse is mainly successful when those initially responsible for their production are either directly involved or readily accessible.

This problem was often reported to be due to politics, which prohibited any knowledge of, or access to, the original sources or requirements engineers. This can only be addressed by re-examining the policies of affected projects. The other reason behind this problem was reported to be the difficulty in keeping track of the original sources and subsequent traces of participation. The common approach, listing contributors to information in document fields, was felt insufficient. This cause of the problem can be tackled with suitable assistance.

Certain project characteristics were found to promote the occurrence of this problem. In projects consisting of individuals split into a number of teams, the location and access of sources was found to be either impossible, time consuming, or unreliable. This was due to: a lack of shared or project-wide commitment; information loss; inability to assess the overall state of work or knowledge; little cross-involvement; poor communication; minimal distribution of information; and changing notions of ownership, accountability, responsibility, and working structure (characteristics amongst those identified elsewhere as contributors to project failure). Characteristics that reduced its occurrence were found in projects consisting of few individuals, due to: a clear visibility of responsibilities and knowledge areas; clarity of working structures; team commitment and ownership; and individuals who acted as *common threads of involvement* (also contributors to success).

8.3: Related work
Many project management tools provide facilities to model organisational charts, role structures, work breakdown...
structures, work-flow, etc. Although these are often incorporated in CASE tools (e.g., the ProKit WORKBENCH [54]), they are not well suited to the location and access problem. They tend to be descriptive, prescriptive, or predictive, so used to model formal, static structures and predefined work plans. The drift between what is modelled, what took place, and what is the case in later project life, can be substantial. RS production and maintenance is a social accomplishment in which such structures are continuously created and recreated. Notions like ownership and responsibility are often only transient. The ability to locate relevant individuals therefore deteriorates as the volume and complexity of communication paths grow over time. Models which can reflect these dynamics and manage this complexity are critically needed.

Some recent models (e.g., DesignNet [37]), make initial attempts here, typically through the ability to restructure plans, and so forth. Models of the organisational environment in which a system is intended to operate are also relevant. These each tend to embed different views of an organisation. They focus on specific structures, such as the intentional structure [60] or the responsibility structure [55], so singularly lack an appreciation of the wider organisational context. Collective and more dynamic variants of these could help clarify the organisational structure of development projects. Process modelling research is of further interest here, as these models promise to provide a fuller understanding of the complete environment in which a system is developed (see [43]).

Following a comprehensive analysis of software errors, recommendations were made for modularising responsibility and promoting communication [39]. Our studies independently consolidate and particularise this: RT problems will persist when accurate responsibility cannot be located and these individuals cannot be accessed for the informal communication often necessary to deal with them. The remedy is to provide a continuously up to date picture which promotes and instigates these activities. Our current research is directed at exactly this.

9: Conclusions
We have illustrated the multifaceted nature of the so-called "requirements traceability problem" that many practitioners claim to experience. We have shown why little real progress has been made here, and how this can only be achieved if based on a thorough understanding of the actual problem. We have distinguished between pre-RS and post-RS traceability, demonstrated how advances in the former are needed and offer most opportunity, and made suggestions for progress here.

In conclusion, to achieve any order of magnitude improvement with the RT problem, there is a need to re-focus research efforts on pre-RS traceability. Of particular concern is the intrinsic need for the on-going ability to rapidly locate and access those involved in specifying and refining requirements, to facilitate their informal communication. Continuous and explicit modelling of the social infrastructure in which requirements are produced, specified, maintained, and used (reflecting all changes), is fundamental to this re-orientation.

Acknowledgements
Much of this work was carried out by the principle author, whilst at Oxford University, supported by a BT University Research Initiative. This author would also like to thank the States of Jersey for continued financial support. Both authors acknowledge the comments and assistance of colleagues, students, and anonymous referees. In particular: David Michael; Marina Jirotka; Matthew Bickerton; Joseph Goguen; Daniel Berry; Jeff Kramer; and Manny Lehman.

References
