The Quest for Ubiquity:
A Roadmap for Software and Systems Traceability Research

O. Gotel[1], J. Cleland-Huang[2], J. Huffman Hayes[3], A. Zisman[4], A. Egyed[5], P. Grünbacher[5], G. Antoniol[6]

[1]Independent Researcher, New York City, USA (olly@gotel.net); [2]De Paul University, Chicago, USA (jhuang@cs.depaul.edu); [3]University of Kentucky, Kentucky, USA (hayes@cs.uky.edu); [4]City University London, London, UK (a.zisman@soi.city.ac.uk); [5]Johannes Kepler University Linz, Austria ({alexander.egyed; paul.gruenbacher}@jku.at); [6]École Polytechnique de Montréal, Québec, Canada (antoniol@ieee.org)

***Abstract*—Traceability underlies many important software and systems engineering activities, such as change impact analysis and regression testing. Despite important research advances, as in the automated creation and maintenance of trace links, traceability implementation and use is still not pervasive in industry. A community of traceability researchers and practitioners has been collaborating to understand the hurdles to making traceability ubiquitous. Over a series of years, workshops have been held to elicit and enhance research challenges and related tasks to address these shortcomings. A continuing discussion of the community has resulted in the research roadmap of this paper. We present a brief view of the state of the art in traceability, the grand challenge for traceability and future directions for the field.**

***Keywords*—Center of Excellence for Software Traceability (CoEST); Grand Challenge of Traceability (GCT); roadmap**

I. Introduction

The first use of the term "traceability" within the software and systems engineering community is difficult to pinpoint with certainty. What is certain, however, is that the ability "to trace" was already recognized as an integral, supporting activity come the "documented" dawn of software engineering. One of the papers of the pioneering 1968 NATO conference examined the requirements for an effective methodology of computer system design and praised three projects for the emphasis they placed on making "the system that they are designing contain explicit traces of the design process" [47].

Over the subsequent decades, traceability has emerged as a research area in its own right, spurring the formation of the Traceability in Emerging Forms of Software Engineering (TEFSE) workshop series in 2002 and the international Center of Excellence for Software Traceability (CoEST) in 2006. Traceability is a regular subject of publications in mainstream engineering conferences and journals, and has also provided a focus for multiple doctoral theses.

What is clear is that there are a thriving number of researchers and practitioners now working in the area of traceability. As we enter the decade in which fifty years will have passed since the NATO conference, it is time to assess where we are and direct where we have yet to go. One of the objectives of the CoEST has been to provide a strategic and coherent research agenda for the area, encouraging a level of maturity whereby the research contributions can be defined and measured and lead to a community vision.

To trace forward to a vision of traceability requires some imagination. As a result of brainstorming efforts, CoEST members agreed upon a vision of a future in which the cost of traceability would have effectively disappeared as a primary concern; up-to-date traceability would be achieved and employed as a by-product of other development activities. This vision led to the formation of eight challenges for traceability, including a grand challenge of *Ubiquity*. This vision and the traceability challenges can be found on-line [21]. To move toward the vision, it is now essential to provide signposts to navigate the challenges and to show paths that could lead there. This is the role of the roadmap[1].

Moving toward any vision requires a starting point. Sections II and III provide a selective review of the state of the practice and the direction of successful research efforts. The CoEST has drafted a Glossary of Traceability Terms and a synopsis of Traceability Fundamentals that can be read in conjunction with this work [9]. The community process for developing the roadmap is described in Section IV and the traceability challenges are reproduced for context. Section V outlines how the roadmap can be used to direct research while Section VI discusses its evaluation and evolution.

II. The Status of Traceability in Practice

There is an indisputable need for an updated survey on traceability practice across industries and projects. Without a recent resource, it is difficult to make claims about the current coverage of the practice, the impact of the latest success stories or the outstanding problems. We therefore point to data that is available about the stakeholders most likely to implement traceability at present and outline their typical rationale. We describe the guidance that is generally available when designing a traceability process and highlight the issue of knowledge dissemination. Finally, we examine an important driver for practice, return on investment.

A. Stakeholder Adoption in Practice

"Traceability" is not a term that is recognized by all practitioners. For example, in one study of a large IT

[1] While the roadmap was primarily constructed from a software engineering perspective, we believe that there is broader applicability in systems engineering (e.g., software-intensive systems) where the design focus is increasingly dominated by software.

978-1-4673-2785-5/12/$31.00 © 2012 IEEE

RE 2012, Chicago, Illinois, USA

consulting company in The Netherlands [8], five out of eight project managers with experience of different sized projects, typically banking and administrative, reported that they did not know about traceability. However, just because the concept is not recognized or shared, it does not mean that some form of traceability is not being achieved on projects through formal or informal efforts. It can depend upon the domain and nature of the software systems being developed as to whether traceability has a profile within an organization. For example, a more recent survey of ten practitioners suggested that the term "traceability" is one that is more likely to be recognized within certain domains (e.g., transportation) and contexts of development (e.g., regulatory and contractual) than in others [34]. This is also anticipated within safety-critical projects, where there is an obligation to provide evidence of system safety in the form of traces.

Traceability is implemented (in theory) within those software development projects whose organizations have been appraised at Level 2 of the Capability Maturity Model Integration (CMMI) or higher, given that maintaining bidirectional traceability of requirements and work products is a practice of the requirements management process area. Traceability should also be manifested in those organizations registered as ISO 9001 certified, since a requirement of this particular Quality System Model is that a process is established to identify and trace products. Traceability is achieved within regulated domains, where compliance and certification stipulations are demanded of their software, though it is not always clear to what levels. Examples include: the U.S. Food and Drug Administration standard that stipulates that traceability analysis be used to verify that the software design implements all the specified software requirements, that all aspects of the design are traceable to software requirements, and that all code is linked to established specifications and test procedures [51]; the U.S. Federal Aviation Administration standard that stipulates that "software developers must be able to demonstrate traceability of designs against requirements" [42]; and the stipulation of a "tracing system" for the "activities necessary to ensure that safety is designed into software that is acquired or developed by NASA" (National Aeronautics Space Administration) [37].

What is evident is that the few reports on practice that are available are often focused upon particular environments, notably large organizations and those in which traceability is mandated by standards, leading to a call for experience reports within more "untypical contexts" [38].

B. Practical Guidance

While practitioner-oriented Body of Knowledge (BOK) guides (e.g., for Business Analysis, Project Management and Software Engineering) and software process improvement models (e.g., CMMI and ISO/IEC 15504/SPICE) describe the need for traceability in general terms, and while standards have routinely demanded traceability for quality purposes, explicit guidance and assessment on specific practices to use is scarce. There is no definitive source of information on traceability processes and procedures, and there is no turnkey solution upon which practitioners can draw. The advice that does exist includes high-level pointers to best practices within industry whitepapers (e.g., [4]), the traceability considerations of broader requirements management within practitioner-oriented textbooks (e.g., [5]) and steps to follow when making traceability tooling decisions (e.g., [20]).

Some organizations build knowledge repositories to aid in the evolution and reuse of their traceability practices, but their structure, content and value remain a matter of anecdote for the wider community. There is little systematic sharing of lessons across organizations, and so little opportunity to build upon the successes or failures of others. There are many practical reasons for this, such as: the time it would consume to document lessons learned for wider consumption and the perceived lack of benefit to the organization concerned, the difficulty of extracting the traceability specifics from data that may be proprietary and sensitive, and the understandable reluctance to give away lessons if traceability provides for a competitive edge.

There have been some notable exceptions in terms of knowledge sharing, and one of the earliest descriptions of the requirements tracing activities employed by an organization are those outlined by Hayes [25]. Later examples include: DaimlerChrysler Research, who report on the value of a well-defined requirements management information model to guide their process [53]; emphasis on the role of management involvement and support, coupled with visible metrics, to perform traceability on a complex defense project [3]; and Alcatel-Lucent's Wireless Business Group, who share their traceability framework and describe how it can be used to seek a cost-effective traceability strategy [29]. However, this information tends to be fragmented across publications and has not been consolidated for the industrial community in any systematic way. Furthermore, such case studies usually give insight into limited aspects of the traceability process within particular domains, most conspicuously telecommunications, defense, automotive, aerospace, avionics and medical.

C. Return On Investment

One of the earliest surveys of requirements traceability practice distinguished between low-end traceability users and high-end users [43], the former regarding traceability as a mandate and employing simple schemes, and the latter regarding it as an important activity of a quality process and employing richer schemes. This survey served to reveal a spectrum of practices and views on the value of traceability, including its role in achieving competitive advantage.

Despite listing the potential benefits of implementing traceability, a recent article in a practice-oriented journal claimed that it still remains a challenge because "many organizations struggle to understand the importance of traceability" [27]. This indicates little change from an earlier study of nine software projects, small to multinational in scope, which reported that engineers did not always understand the return on investment from doing traceability [2]. More bleakly, the study of the Dutch company referred to in II.A also reported that many project managers might choose not to trace when working on a fixed price contract if it adds costs, also because the primary benefits are often realized after product delivery [8].

The cost-benefits of traceability, and the point at which these are incurred or attained, are critical issues for practitioners at both the strategic and operational levels. These issues are being examined in the latest industry reports, but quantitative data is needed upon which practitioners can base decisions. A positive and qualitative picture of traceability value, however, is reported from a survey within Teradyne's Semiconductor Test Division, where its role in identifying missing requirements, instilling confidence in requirements and test documents, and in conveying rationale is described [40]. But, this value is largely attained by those practitioners creating project documentation and not necessarily sustained through the life of a project. Arkley reported in 2005 that: "the goal of achieving traceability throughout the development process is seldom achieved in an industrial environment" [2]. Whether traceability remains a concern that is mostly restricted to the early phase of development in many organizations is not apparent; an understanding and assessment of its through-life value is clearly a prerequisite to its longevity.

III. Highlights of Traceability Research

Both university and industry-based researchers have conducted surveys with practitioners to understand the traceability stakeholders, processes and problems experienced across projects and organizations (e.g., [19, 34, 43]), and such empirical studies have helped to inform research directions. This paper does not aim to provide a comprehensive survey of traceability research; for that, the reader is referred to a recent publication [54].

In this paper, we note that a significant proportion of the research attention has been directed toward information models for traceability or toward the automated creation and maintenance of trace links, and both research areas have been augmented by the development of supporting tools. Less research has been focused upon the stakeholders' requirements for traceability and on measuring the satisfaction of traceability in actual use. As recent attention turns toward the economics of traceability, a sign of concern for its return on investment, the community is beginning to look beyond just the technical aspects of the traceability problem. It is within this broader socio-technical-economic context that future research efforts need to be grounded.

A. Traceability Information Models

One of the most significant contributions of the traceability research community has arguably been in the area of modeling. The role and requirements of a traceability information model (referred to interchangeably as a meta-model, reference model or scheme) was outlined in the early work of Ramesh and Edwards [44], who described the need for: "the development of a model that represents and provides the semantics of various traceability linkages or relationships between requirements and the system components". Traceability information models form an important part of any traceability strategy. They delineate the intended trace artifacts and the permissible trace links, and ensure that traces can be retrieved to answer anticipated traceability-enabled queries. While this research has impacted practice in specific domains [46], many of the proposed modeling concepts have been considered too complex for actual use in others [45], leading to the call to promote simpler models in practice [33]. Research that clarifies how such models can be selected, adapted and employed, as part of an end-to-end traceability process, is equally necessary. Advances in this area are now coming from the model-driven development community [54].

The focus upon the definition and use of traceability information models has influenced the functionality of leading requirements management tools. For instance, the Requirements Management Plan is a standard work product within the IBM Rational toolset that defines traceable artifacts and link types, and is expressed as a class diagram in the UML [4]. This focus upon traceability information models has also contributed to research into the semantics of the artifacts to be traced and their trace links, as reviewed in recent work [48], making rich and intelligent forms of traceability feasible [15]. Such modeling advances have stimulated a wealth of research on trace automation. It has also provided several classifications for trace links based upon the types of artifacts (e.g., [26, 41, 49]) or trace usability (e.g., verification [45] and impact analysis [52]).

B. Automated Trace Creation and Maintenance

Researchers have made many advances by developing and/or adopting a variety of techniques for acquiring and maintaining traceability links in either fully, or partially, automated ways. Early work [1] examined the use of the vector space model and a probabilistic model to recover links between pages of documentation manuals and code, and compared its effectiveness to that of simple string searches. Latent semantic indexing (LSI) was subsequently applied to the same datasets [35]. Other researchers proceeded to apply information retrieval (IR) techniques (e.g., [11, 14, 22]) and achieved a similar result: the methods retrieved almost all of the true links (in the 90% range for recall) and yet also retrieved many false positives (with precision in the low 10–20% range, with occasional exceptions).

This led to the pursuit of ways to improve precision. A number of techniques were applied: thesaurus, phrasing, filtering, golden keywords, goal-centric tracing and lexical affinities (e.g., [22, 30, 31, 39, 50, 55]). Rule-based tracing was applied to create different types of trace links in software artifacts based upon their semantics and the grammatical roles of their words [49]. This work was extended to support artifacts created during the development of product line systems [26]. A traceability recovery tool based on LSI [14] was applied and, by introducing categorization, reached a precision of 25% with 90% recall. Phrasing was applied, permitting researchers to obtain improvements of almost 20% precision for one dataset, when examining the top 5% of the returned candidate links [55].

In the past two years, the CoEST community has made major advances in developing an instrumented environment and associated benchmarking standards for comparatively evaluating techniques [13, 28], which is important for future advancements and technology transfer. Early transfer successes have included the integration of trace retrieval techniques in tools for SAIC and Siemens, and the release of RETRO (Requirements Tracing on Target) as an open-source product under NASA [23]. Nevertheless, far less attention has been paid to the artifacts that are the subject of traceability, and the promise here lies in the ability to also handle unstructured, informal data in various media.

C. *Traceability Economics*

The existence of traceability improves software development and maintenance [32]. However, in the absence of full automation, traces are costly which counters these savings. Studies suggest that traces captured by developers familiar with a system are cheaper and of better quality than traces recovered by less familiar developers [17], but familiarity diminishes over time (e.g., if people forget or move on) and so traces need to be captured early to avoid this. This adds the problem of maintenance – traces captured early need to be maintained while the code evolves. More research is needed to understand this trade off.

Not all traces are used and useful. Early trace capture needs to err on the side of recovering many/all traces. Trace recovery has the benefit that it can be tailored to the traces that are actually needed, but a better understanding of its different uses is needed. Which traces are more important or more likely to be used? Can the users tolerate a degree of trace incompleteness and/or incorrectness? Can the Pareto rule be applied, meaning that most trace benefits come at the expense of a small part of the trace cost? These questions highlight the important role of value-based reasoning on traceability. We know that the effort for capturing traceability at the level of classes is three-fold cheaper than at the level of methods [17], but we do not yet understand how granularity differences affect the uses of traceability.

There is still no effective way to understand the value contribution of traces, the impact of creating and maintaining trace links at differing levels of quality and granularity, and dealing with trade-offs [16]. Research is only beginning to examine this issue, where value-based software engineering [7] provides prospect for balancing the costs and benefits, including value-based approaches to tracing [24]. A value-based approach has been proposed to determine what traces to capture and to use adaptive strategies to migrate between simple and more detailed forms of traceability as needs evolve [18], and it has been shown that the effectiveness of maintenance can be improved by varying the degree of granularity of the traceability information model [6]. It has been proposed that "good enough" traceability can be planned for by analyzing the failure to trace risk [12] and a reflexion model has been used to evolve sufficient traceability for developers to reason about key tasks [36].

IV. Roadmap Context and Processs

The primary objective of the CoEST is to provide leadership for traceability research, education and practice, enabled by identifying and tackling the challenges of implementing effective software and systems traceability (*www.coest.org*). Using seed funding provided by NASA, thirteen traceability researchers and practitioners first collaborated to formulate a draft of "The Grand Challenges of Traceability" in 2006 [10]. The National Science Foundation (NSF) then funded a follow-up symposium in 2007 where cutting edge research that addressed some of these challenges was presented. Over the subsequent years, the *Grand Challenges of Traceability* (GCT) has remained one of the central projects of the CoEST, and a completely revised version of the challenges has now emerged to bring rationale and cohesion to the material [21].

A. *The GCT Process and the GCT Report*

The revised GCT report was the result of a collaborative process involving nine members of the CoEST, including the authors of this paper, and sustained over a number of years. The GCT report lays out a vision for software and systems traceability in the future, a scenario that was formed by merging the separate visions of all the contributors. The report then lists seven traceability challenges that need to be addressed to realize the vision, as derived iteratively and by consensus. The eighth is labeled as the *Grand Challenge of Traceability* because making traceability *ubiquitous* was agreed to demand progress on the seven other challenges.

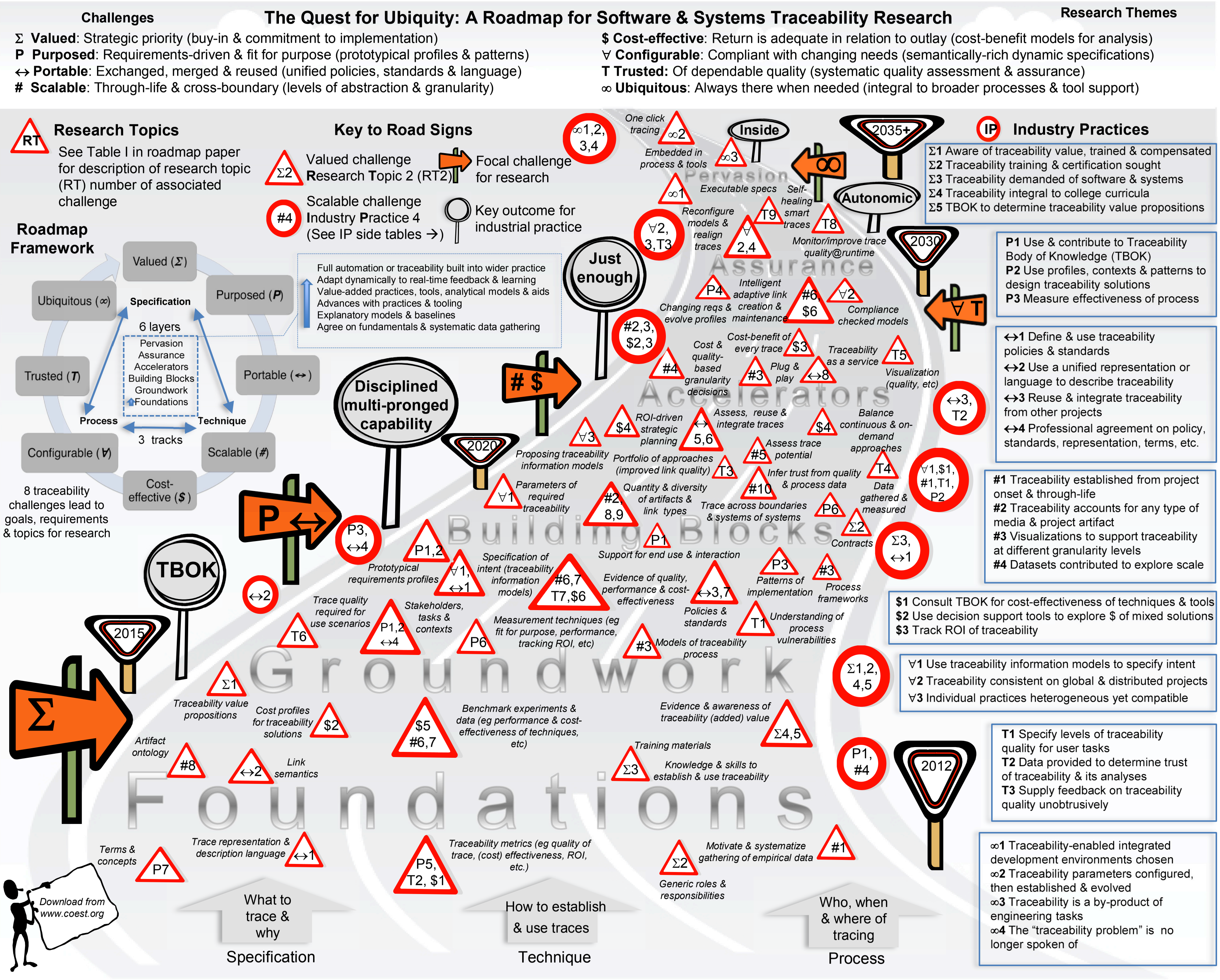
Each challenge presents a set of goals for research (thirty-five in total), which were decomposed into eighty-four requirements. Gap analysis was applied to the state of the art and the practice in satisfying these requirements, a synopsis of which is provided in the GCT report. As a result, fifty-two topics for research were identified. These research topics are traceable back to the challenges in the report. To assess progress with the research, twenty-nine positive adoption practices for industry were also determined. The challenges, goals and research topics are summarized in Table I. The list of industry practices is reproduced as part of Fig.1. To maintain traceability from the GCT report into the roadmap, each challenge has been assigned a unique symbol for reference and the key can be found at the top of Fig.1.

B. *The Roadmapping Process and the Roadmap*

While ubiquitous traceability is labeled as the grand challenge in the revised GCT report, no ordering was given to the underlying challenges, making the course toward ubiquity problematic to plan for and measure. In isolation, there is no research agenda for the traceability community: there is no concept of a priority among the challenges, and no attempt to suggest a logical or temporal order in which the research topics should be tackled so as to build upon each other. The intention behind reformulating this material is to provide a plausible incremental structure for tackling the research topics and to suggest the focal challenges for the community in the near and longer term. Construction of the roadmap was a continuation of the community effort and the

TABLE I. SUMMARY OF TRACEABILITY CHALLENGES AND RESEARCH TOPICS (SYNTHESIZED FROM [21])

	TRACEABILITY CHALLENGE AND SUMMARY OF DERIVED GOALS	**RESEARCH THEME AND ASSOCIATED RESEARCH TOPICS (Note these research tasks are shown here in summarized form only – cross-references to [21] are provided)**
PURPOSED (P)	**[P]** *Traceability is fit-for-purpose & supports stakeholder needs (i.e., traceability is requirements-driven).* Develop prototypical stakeholder requirements for traceability, clearly defined & measurable for specific software & systems engineering tasks.	**[P]** *Define & instrument prototypical traceability profiles & patterns.* - Develop prototypical stakeholder requirements for traceability, including scenarios of use (RT1). - Develop a classification scheme to define the context of a traceability need (RT2). - Develop patterns for traceability implementations associated with traceability profiles & contexts (RT3). - Instrument a mechanism to both use & evolve this resource of profiles & contexts (RT4). - Propose & agree upon metrics to measure effectiveness in all areas of the traceability process (RT5). - Perform empirical studies of stakeholder types, use of traceability techniques, methods & tools (RT6). - Develop a Traceability Body of Knowledge (TBOK) (RT7).
COST-EFFECTIVE ($)	**[$]** *The return on investment (ROI) from using traceability is adequate in relation to the outlay of establishing it.* Develop techniques for computing the ROI of traceability in a project, understanding the impact of various traceability decisions at stages of the life-cycle upon both the costs & benefits of the traceability process.	**[$]** *Develop cost-benefit models for analyzing stakeholder requirements for traceability & associated solution options at a fine-grained level of detail.* - Agree upon metrics for measuring traceability cost-effectiveness (RT1). - Understand the typical cost profile of traceability outlay on a project (RT2). - Develop the means to associate a cost & a benefit profile for individual traces (RT3). - Create decision support tools & impact analysis tools for making traceability ROI decisions (RT4). - Develop benchmark studies for evaluating the cost-effectiveness of traceability techniques (RT5). - Decrease the costs & improve the effectiveness of traceability techniques (RT6).
CONFIGURABLE (∀)	**[∀]** *Traceability is established as specified, moment-to-moment, & accommodates changing stakeholder needs.* Develop techniques for dynamically generating & maintaining accurate & semantically rich traceability links that are configured according to the current needs of the project.	**[∀]** *Use dynamic, heterogeneous & semantically rich traceability information models (or similar specifications of the intended traceability) to guide the definition & provision of traceability.* - Provide techniques for defining traceability needs of a project (i.e., links, granularity, semantics) (RT1). - Leverage the conceptual traceability model to support change & compliance (RT2). - Propose traceability information models based upon stakeholders' & project-level requirements (RT3). - Reconfigure or re-purpose a pre-existing set of traces to accommodate changes in the definition of the conceptual traceability model (RT4).
TRUSTED (T)	**[T]** *All stakeholders have full confidence in the traceability, as it is created and maintained in the face of inconsistency, omissions & change; all stakeholders can & do depend upon the traceability provided.* Develop techniques for assessing & communicating the current state of traceability in a project, & develop self-adapting techniques so that quality is preserved in the face of change.	**[T]** *Perform systematic quality assessment & assurance of the traceability.* - Develop a traceability vulnerability model & techniques to reinforce reliability (RT1). - Formulate metrics for traceability quality assessment (RT2). - Improve the quality of both manual & automatically created & maintained trace links (RT3). - Provide ways of inferring trust in the traceability based upon trace history & expected use (RT4). - Create a visual dashboard for displaying & examining traceability quality attributes on a project (RT5). - Catalogue the quality required for supporting different end-user tasks within the TBOK (RT6). - Populate the TBOK with empirical evidence of the quality of traceability techniques & tools (RT7). - Advance the run-time monitoring of traceability quality with validated error detection models (RT8). - Apply concepts from autonomic computing to explore self-healing traceability techniques (RT9).
SCALABLE (#)	**[#]** *Varying types of artifact can be traced, at variable levels of granularity & in quantity, as the traceability extends through-life & across organizational & business boundaries.* Develop techniques for scaling up traceability techniques, & for supporting multi-grained traceability across a variety of artifact types & organizational boundaries.	**[#]** *Provide for levels of abstraction & granularity in traceability techniques, methods & tools, facilitated by improved trace visualizations, to handle large datasets & the longevity of these data.* - Create a shared repository of industrial datasets to support experimentation at levels of scale (RT1). - Develop scalable, extensible, & effective search, filtering & trace visualization mechanisms (RT2). - Develop customizable & componentized abstract model of the traceability process (RT3). - Develop a cost-benefit model to assess granularity decisions that impact scalability (RT4). - Provide techniques to evaluate & improve the traceability potential of various datasets (RT5). - Improve performance of real-time retrieval (RT6) & rendering of trace links to account for scale (RT7). - Integrate various artifact types & categories of media into traceability end-use (RT8). - Address scalability issues of tracing non-functional requirements (RT9) & systems of systems (RT10).
PORTABLE (↔)	**[↔]** *Traceability is exchanged, merged & reused across projects, organizations, domains, product lines & supporting tools.* Develop policies, standards, & formats for exchanging & integrating traceability information across projects & organizations.	**[↔]** *Agree upon universal policies, standards, & a unified representation or language for expressing traceability concepts.* - Develop a standard language for expressing traceability information models & traces (RT1). - Define granularity & semantics for the various types of trace links used in different domains (RT2). - Define policies, standards, infrastructure, processes & tools for tracing in distributed projects (RT3). - Examine the likely forms of cross-boundary traceability required in the future (RT4). - Provide techniques to assess existing traceability for reuse potential in other contexts of use (RT5). - Develop mechanisms to help extract, integrate & reuse traceability work products (RT6). - Apply techniques & standards from other distributed industries (e.g., the food industry) (RT7). - Re-conceptualize traceability as a service so that it can be procured & interchanged at will (RT8).
VALUED (Σ)	**[Σ]** *Traceability is a strategic priority valued by all; every stakeholder has a role to play & actively discharges his or her responsibilities.* Develop supporting techniques that cross the technical & business domains of a project so that the benefits of traceability are visible & accessible to all stakeholders.	**[Σ]** *Raise awareness of the value of traceability, to gain buy-in to education & training, & to get commitment to implementation.* - Develop techniques, methods & tools to leverage traceability value propositions on a project (RT1). - Define traceability development contracts for us in different projects & organizational settings (RT2). - Identify core knowledge areas & skills for traceability, & create effective pedagogical materials (RT3). - Develop software tools that deliver value for supporting software & systems engineering tasks (RT4). - Document evidence demonstrating the contribution of traceability to success rates & longevity (RT5).
UBIQUITOUS (∞)	**[∞]** *Traceability is always there, without ever having to think about getting it there, as it is built into the engineering process; traceability has effectively "disappeared without a trace."* Achieved only when traceability is established & sustained with near zero effort.	**[∞]** *To provide automation such that traceability is encompassed within broader software & systems engineering processes, & is integral to all tool support.* - Investigate automated ways to define the traceability strategy so that the traceability solution emerges as a natural byproduct of the system specification (RT1). - Total automation of high-quality traceability creation & trace maintenance (RT2). - Embed seamless traceability into software & systems engineering techniques, methods & tools (RT3).



The Quest for Ubiquity: A Roadmap for Software & Systems Traceability Research

Challenges

Σ **Valued**: Strategic priority (buy-in & commitment to implementation)
P **Purposed**: Requirements-driven & fit for purpose (prototypical profiles & patterns)
↔ **Portable**: Exchanged, merged & reused (unified policies, standards & language)
# **Scalable**: Through-life & cross-boundary (levels of abstraction & granularity)
$ **Cost-effective**: Return is adequate in relation to outlay (cost-benefit models for analysis)
∀ **Configurable**: Compliant with changing needs (semantically-rich dynamic specifications)
T **Trusted:** Of dependable quality (systematic quality assessment & assurance)
∞ **Ubiquitous**: Always there when needed (integral to broader processes & tool support)

Research Themes

Research Topics

RT See Table I in roadmap paper for description of research topic (RT) number of associated challenge

Key to Road Signs

Σ2 Valued challenge **R**esearch **T**opic 2 (RT2)

Focal challenge for research

#4 Scalable challenge **I**ndustry **P**ractice 4 (See IP side tables →)

Key outcome for industrial practice

Roadmap Framework

8 traceability challenges lead to goals, requirements & topics for research

Full automation or traceability built into wider practice
Adapt dynamically to real-time feedback & learning
Value-added practices, tools, analytical models & aids
Advances with practices & tooling
Explanatory models & baselines
Agree on fundamentals & systematic data gathering

IP Industry Practices

Σ1 Aware of traceability value, trained & compensated
Σ2 Traceability training & certification sought
Σ3 Traceability demanded of software & systems
Σ4 Traceability integral to college curricula
Σ5 TBOK to determine traceability value propositions

P1 Use & contribute to Traceability Body of Knowledge (TBOK)
P2 Use profiles, contexts & patterns to design traceability solutions
P3 Measure effectiveness of process

↔1 Define & use traceability policies & standards
↔2 Use a unified representation or language to describe traceability
↔3 Reuse & integrate traceability from other projects
↔4 Professional agreement on policy, standards, representation, terms, etc.

#1 Traceability established from project onset & through-life
#2 Traceability accounts for any type of media & project artifact
#3 Visualizations to support traceability at different granularity levels
#4 Datasets contributed to explore scale

$1 Consult TBOK for cost-effectiveness of techniques & tools
$2 Use decision support tools to explore $ of mixed solutions
$3 Track ROI of traceability

∀1 Use traceability information models to specify intent
∀2 Traceability consistent on global & distributed projects
∀3 Individual practices heterogeneous yet compatible

T1 Specify levels of traceability quality for user tasks
T2 Data provided to determine trust of traceability & its analyses
T3 Supply feedback on traceability quality unobtrusively

∞1 Traceability-enabled integrated development environments chosen
∞2 Traceability parameters configured, then established & evolved
∞3 Traceability is a by-product of engineering tasks
∞4 The "traceability problem" is no longer spoken of

Valued (Σ)
Purposed (P)
Portable (↔)
Scalable (#)
Cost-effective ($)
Configurable (∀)
Trusted (T)
Ubiquitous (∞)
Specification
Process
Technique
6 layers
Pervasion
Assurance
Accelerators
Building Blocks
Groundwork
Foundations
3 tracks
TBOK
Disciplined multi-pronged capability
Just enough
Inside
Autonomic
2015
2020
2030
2035+
2012
Σ
P ↔
# $
∞
∀ T
Foundations
Groundwork
Building Blocks
Accelerators
Assurance
Pervasion
Terms & concepts
P7
Artifact ontology
#8
Traceability value propositions
Σ1
↔2
Link semantics
Cost profiles for traceability solutions
$2
Trace representation & description language
↔1
T6
Trace quality required for use scenarios
P1,2 ↔4
Stakeholders, tasks & contexts
P3, ↔4
Prototypical requirements profiles
P1,2
∀1, ↔1
Specification of intent (traceability information models)
P6
$5 #6,7
Benchmark experiments & data (eg performance & cost-effectiveness of techniques, etc)
P5, T2, $1
Traceability metrics (eg quality of trace, (cost) effectiveness, ROI, etc.)
∀1
Parameters of required traceability
#6,7 T7,$6
Measurement techniques (eg fit for purpose, performance, tracking ROI, etc)
∀3
Proposing traceability information models
#2, 8,9
Quantity & diversity of artifacts & link types
Evidence of quality, performance & cost-effectiveness
#3
Models of traceability process
$4
ROI-driven strategic planning
↔ 5,6
Portfolio of approaches (improved link quality)
T3
P1
Support for end use & interaction
↔3,7
Policies & standards
Σ3
Training materials
Knowledge & skills to establish & use traceability
Σ2
Generic roles & responsibilities
#2,3, $2,3
#4
Cost & quality-based granularity decisions
Assess, reuse & integrate traces
#5
Assess trace potential
#10
Infer trust from quality & process data
Trace across boundaries & systems of systems
P3
Patterns of implementation
T1
Understanding of process vulnerabilities
Σ4,5
Evidence & awareness of traceability (added) value
#1
Motivate & systematize gathering of empirical data
∞1,2, 3,4
∀2, 3,T3
P4
Changing reqs & evolve profiles
Intelligent adaptive link creation & maintenance
Cost-benefit of every trace
$3
#3
Plug & play
↔8
#6, $6
∀2
Compliance checked models
Traceability as a service
T5
Visualization (quality, etc)
$4
Balance continuous & on-demand approaches
T4
Data gathered & measured
P6
Σ2
Contracts
#3
Process frameworks
One click tracing
∞2
Embedded in process & tools
∞3
∞1
Executable specs
Reconfigure models & realign traces
∀ 2,4
T9
Self-healing smart traces
T8
Monitor/improve trace quality@runtime
↔3, T2
∀1,$1, #1,T1, P2
Σ3, ↔1
Σ1,2, 4,5
P1, #4
What to trace & why
Specification
How to establish & use traces
Technique
Who, when & where of tracing
Process
Download from www.coest.org

Figure 1. Traceability Roadmap.

result is shown in Fig.1. The process involved one author making initial suggestions and placements, then discussion and iteration until consensus was reached among the authors.

The roadmapping process began by examining the individual research topics to characterize the fundamental nature of the research required. This involved determining whether the research predominantly addressed the "who, why, what, when, where" or "how" of traceability, or some combination. This led to the concept of *research tracks*. The research tracks were separated into three categories dependent upon the primary focus of the research effort:

- *Specification*–What to trace and why.
- *Technique*–How to establish and use traces.
- *Process*–The who, when and where of doing the tracing.

These distinctions are also strategic, tactical and operational in nature. It was decided not to delineate a research track on traceability tooling since tools will need to be developed to support research efforts in each track. The research tracks constitute three paths on the horizontal axis of the roadmap. There is no temporal ordering along this axis.

The roadmapping process also involved determining where each research topic would fit in a logical progression (i.e., which research appeared to be a prerequisite or a co-requisite to other research). This sequence was then examined to see whether the topics clustered to give a distinct emphasis to the research at different points along the progression. This led to the concept of a *research layer*. The research layers mimic the idea of process maturity by giving a structure to the progression of the research efforts. While research on any topic could be conducted at any time, and be of variable duration, six layers of evolution were identified:

- *Foundations*–Agreement reached on the traceability fundamentals and systematic data gathering conducted.
- *Groundwork*–Explanatory models and baselines formed for traceability and its measurement.
- *Building Blocks*–Advances achieved with individual and integrated traceability practices and their tooling.
- *Accelerators*–Value-added traceability practices and tooling achieved, exploiting analytical models and aids.
- *Assurance*–Traceability adapted dynamically to account for real-time feedback and learning.
- *Pervasion*–Traceability built into the wider software and systems engineering practices and tooling.

The research layers constitute six stages on the vertical axis of the roadmap. While a temporal ordering is suggested between the layers, it is not suggested within each layer.

The topics were then positioned within a two-dimensional framework of research tracks and research layers to map out the "terrain". Related and overlapping topics emerging from the different challenges were fused, while some topics were decomposed further. The resulting topics are shown as triangular signs on the road of Fig.1. While a brief synopsis is given within the roadmap itself, these road signs should be read in conjunction with the cross-referenced topic descriptions in the right column of Table I.

Each traceability challenge was assigned a dominant research theme in the GCT report, as shown in Table I. The research layers were therefore examined to determine when the research theme for each challenge should have been largely addressed. Mapping the realization of the challenges to the research layers was then undertaken to suggest a priority for tackling them, as indicated by the directional signposts placed along the sides of the road in Fig.1. Note that the grand challenge realization aligns with the top layer.

Since it is envisioned that progress in industry practice will be made as the research community moves through the research layers and addresses the challenges, circular road signs were also placed alongside the road to suggest when we would expect positive adoption to occur. The key outcome for industrial practice with the progression through each layer was the final addition to the roadmap, as shown by the placement of the large circular signposts in Fig.1.

V. Navigating The Roadmap

The realization of ubiquitous traceability is probably over twenty years away. The roadmap is one way of organizing the research topics to help us progress toward this vision in a disciplined manner. While advances can be sought and achieved within any research layer at any time, there is value in targeting the priorities over the years ahead to set ambitious unified goals for the research community, and also to set practical milestones for industry. We suggest a way to use the roadmap below, and anchor this in the symbol of the the challenge (a key is provided in Fig.1), followed by the number of either a research topic (*RT*) or industry practice (*IP*). Therefore, *IP*P1 in the text refers to industry practice 1 of the purposed challenge, depicted with a circle sign containing P1 on the roadmap and described in Fig.1. *RT*↔1 refers to research topic 1 of the portable challenge, depicted with a triangle sign containing ↔1 and described in Table I.

A. Near-term: Foundations and Groundwork (<3 Years)

Traceability needs to be valued in practice if it is to advance within any one organization, within industry in general, and be a strategic priority (*IP*Σ1). Where practiced, there needs to be commitment to its implementation at all levels and training made available (*IP*Σ2,4). Practitioners should anticipate the emergence of a Traceability Body of Knowledge (TBOK) in the near-term to support this milestone (*IP*P1). Researchers should aim to achieve the challenge of traceability that is valued in the near-term and be on the road to making it purposed and portable:

- ***Specification***–The research community needs to move toward consensus on the traceability foundations. This comprises the terminology and concepts (*RT*P7), and agreement on identifying the artifacts we trace (*RT*#8) and on the nature of these traces (*RT*↔2). While there are shared schemes in other disciplines for representing and exchanging traceability data, such as the Global Trade Item Number and TraceCore in the food industry, this is yet to be the case in software and systems engineering (*RT*↔1,*IP*↔2,4). A better understanding of traceability stakeholders and their needs (*RT*P1,2) is paramount groundwork for building upon these foundations and predicting future needs (*RT*↔4). We need to understand what stakeholders value (*RT*Σ1) and what it costs to achieve this value to satisfy concerns (*RT*$2,*IP*Σ5).

- ***Technique***–Traceability metrics and measurement (*RT*T2,P5,$1) need to be a priority for research. Without such measures, there is little way to compare techniques and methods and ensure that we are making improvements as we build (*RT*P6). Benchmarks are fundamental to assessing progress and rely upon maturity with this topic (*RT*#6,7,$5).
- ***Process***–Case study and empirical data on how traceability is undertaken (*RT*#1), such as the required roles and responsibilities (*RT*Σ2), needs to be aggregated to provide a basis for identifying the skills demanded (*RT*Σ3) of the essential traceability process (*RT*#3). This is a precursor to developing suitable training materials (*RT*Σ3) and tools (*RT*Σ4). Collaboration with industry will be decisive at all levels, as will be finding a way to encourage the sharing of data (*RT*Σ5,*IP*#4).

B. Short-term: Building Blocks (<5 Years)

Traceability needs to be requirements-driven and demonstrably fit for purpose (*IP*P2,3,T1,∀1) if its value appreciation is to grow within industry. Also, it needs to be possible for practitioners to exchange and reuse the fruits of their labor to advance practices (*IP*↔1, Σ3). Practitioners should anticipate the availability of the resources they need within an evolving TBOK to undertake a disciplined and multi-pronged through-life approach to traceability within their environments (*IP*$1,#1). Researchers should aim to achieve purposed and portable traceability in the short-term and be on the road to making it cost-effective and scalable:

- ***Specification***–The research community will need to place building blocks upon its traceability requirements analysis to develop stakeholder profiles (*RT*P1) along with instruments to express the context of their need (*RT*P2) and the quality demands (*RT*T6). Concurrent research will need to investigate how traceability information models can be designed and parameterized to specify and address these needs (*RT*↔1,∀1).
- ***Technique***–Researchers will need to measure competing approaches comparatively to achieve measurable advances in all aspects of traceability implementation and use (*RT*T7,$6). With the knowledge and confidence provided, researchers will be able to focus upon mechanisms to mix and match approaches to achieve different cost and quality profiles (*RT*T3,#6,7). To build further, research will need to integrate support for diverse new media artifacts, and handle informal data and quality requirements within their techniques and methods (*RT*#2,8,9,P1).
- ***Process***–Domains, organizations, and projects within them are going to need to share the practices and products of traceability (*RT*#10) and recognize vulnerabilities (*RT*T1), so are going to require a blend of process frameworks, patterns, policies, standards and contracts for working (*RT*P3,↔3,7,#3,Σ2). Research can promote all aspects of professionalism and coordinate ongoing data gathering (*RT*P6).

C. Mid-term: Accelerators (<15 Years)

In practice, traceability needs to be through-life, cross-boundary and encompass all potentially traceable artifacts. Moreover, the return from its use needs to be measurably adequate in relation to outlay. Practitioners should anticipate the availability of the tools to design and implement just-enough traceability now that it can be properly purposed and shared (*IP*$2). They should also expect the ability to assess the credibility (*IP*T2). The TBOK will evolve to provide cost-benefit models for analyzing traceability requirements (*IP*$3) and solution options to deal with dimensions of scale and reuse (*IP*#2,3,↔3). Researchers should aim to achieve cost-effective and scalable traceability in the mid-term and be on the road to making it configurable and trusted:

- ***Specification***–To expedite progress, research will need to provide intelligent aids for traceability strategy makers to balance decisions about traceability requirements, granularity, cost and quality (*RT*$4,#4). The development of mechanisms to bootstrap traceability information models and strategies from project data will be further accelerators as project size and complexity grows (*RT*∀3).
- ***Technique***–Acceleration will occur with growing reuse capability (*RT*↔5,6) and the potential to assess the traceability of artifacts and projects (*RT*#5) both before decisions and by moment. Research will need to focus upon providing for dynamic heterogeneous approaches (*RT*#3) and on techniques to measure and maximize the value of every trace that is implemented and used (*RT*$3).
- ***Process***–The research community will need to explore sophisticated instrumentation that enables all traceability process attributes to be set, monitored and fine-tuned over the course of a project (*RT*$4,T5). Providing the means to assess the credibility of the traceability process and product in real-time (*RT*T4) will be crucial as automation begins to address the problems of scale and traceability becomes offered as a service (*RT*↔8).

D. Long-term: Assurance (15+ Years)

Traceability needs to be compliant with (changing) needs in practice (*IP*∀3) and of dependable quality at all times, irrespective of scale, distribution and without a major cost impact (*IP*∀2). Practitioners should anticipate autonomic traceability that is subject to regular quality assessment and assurance (*IP*T3). The traceability will be near optimal for needs and the TBOK will direct practitioners to traceability optimization and trace assurance practices. Researchers should aim to achieve configurable and trusted traceability in the long-term and be on the road to making it ubiquitous:

- ***Specification***–Traceability will be guided by semantically rich and dynamic specifications of the intended traceability. Research will need to focus on assessing, adapting and assuring the specifications and aligning the results of these specifications at all times (*RT*P4,∀2,4).

- ***Technique***–The research spotlight will be on intelligence and adaptation (*RT*#6,$6). Research will need to advance support for smart traceability, producing only traces that have absolute autonomy (*RT*T9).
- ***Process***–With ongoing optimization and assurance of the traceability process and its products (*RT*∀2), research will need to concentrate on the intelligent and dynamic incorporation of process feedback at all levels (*RT*T8).

E. Final Destination: Pervasion (Year of the Vision)

Where traceability is always there when needed, configured as wanted and trusted fully, practitioners should expect traceability support to be "inside" all processes and tools and for traceability to be "inside" all projects (*IP*∞1, 2,3). The TBOK will have evolved into a manual of all the "remarkable things you can do with traceability" (*IP*∞4).

In realizing the grand challenge of traceability that is ubiquitous, the research community will need to focus upon ensuring that traceability specifications are fully executable (*RT*∞1) and that support for traceability is pervasive within broader software and systems engineering processes (*RT*∞3). Proven traceability techniques will need to be integral to all tool support and traceability should never be more than "one click" away when using this support (*RT*∞2).

F. Interim Paths

While the traceability roadmap covers a vast terrain and is designed to provide a broad structure and possible route for research in the area, it can also be used to define interim paths and research agendas. For instance, a near to short-term process-specific agenda may be to focus upon the roles and responsibilities for traceability, the required knowledge and skills, and the provision of requisite training material. A short to mid-term specification-specific agenda may be to focus upon traceability information models, a way to propose suitable models for different situations and a way to adapt these models over time. A mid to long-term technique-specific agenda may be to focus upon reusing traces in new contexts, adapting them intelligently, then providing traces themselves with the capacity they need to remain relevant (both individually and within multiple traceability networks) or to expire benignly. To mature as an area, there is a need for the community to pursue research that fits together to take us toward a shared vision of the final destination.

VI. DISCUSSION

This roadmap is the work of a subset of traceability researchers, some of whom have been responsible for implementing traceability in industrial or government settings. There is a need for feedback from additional researchers and practitioners. Going forward, the CoEST intends to solicit feedback on the priority of the research topics, via a triage-based approach, to gauge perceptions on the value of each topic and the likely difficulty of the research involved. This may reveal an alternative way of organizing the terrain so as to ensure that we focus upon those research topics that are perceived to be of high value and low difficulty in the near-term. The CoEST also intends to solicit feedback to assess how well we are advancing with the topics and challenges, so we can measure our progress along the road. One suggestion for the latter, arising from a TEFSE initiative, has been to classify all future contributions in the area using the identifiers of the GCT report, though this would have to take care to also account for traceability research less visibly embedded within other programs (e.g., software product lines and model-based development).

In summary, this paper presents a first attempt to map out the terrain for traceability research and to carve out a road across it. We offer this roadmap as a vehicle to bring some structure to what may otherwise remain fragmented research pursuits in a demanding field. We do not view the research terrain as fixed and we concede that the research topics presented here are but those that arose out of one of many potential visions of the traceability future. Just as mountains emerge and earth ruptures, advances in traceability research and changes in traceability practice may lead to novel destinations and the need for entirely new roads. What will remain critical if the research in this area is to mature, however, is for the community as a whole to start thinking about the use of such roadmaps and the role they could play.

ACKNOWLEDGMENTS

The authors thank the participants of the workshops in which the traceability challenges were explored (at NASA's IV&V facility in 2006 and in Kentucky in 2007), and also thank NASA and the NSF for funding these workshops (NASA grant #NNX06AD02G and NSF grant #0647443).

REFERENCES

[1] G. Antoniol, G. Canfora, G. Casazza, A. De Lucia, & E. Merlo, "Recovering Traceability Links between Code and Documentation," IEEE Trans SE, 28(10), 2002, p970–983.

[2] P. Arkley & S. Riddle, "Overcoming the Traceability Benefit Problem," Proc. 13th IEEE Int RE Conf, Paris, France, 2005, p385–389.

[3] D. Aubrey, "Controlling the HMS Program through Managing Requirements," Proc. 14th IEEE Int RE Conf, Minneapolis MN, USA, 2006, p227–232.

[4] T. Behrens, "Never 'Without a Trace': Practical Advice on Implementing Traceability," IBM developerWorks, 15 Feb 2007.

[5] B. Berenbach, D.J. Paulish, J. Kazmeier, & A. Rudorfer, Software and Systems Requirements Engineering. McGraw-Hill, 2009, Ch.7.

[6] A. Bianchi, G. Visaggio, & A.R. Fasolino, "An Exploratory Case Study of the Maintenance Effectiveness of Traceability Models," Proc. 8th Int Wkshop Prog Comp, Limerick, Ireland, 2000, p149–158.

[7] S. Biffl, A. Aurum, B.W. Boehm, H. Erdogmus, & P. Grünbacher, Value-based Software Engineering. Springer-Verlag, 2005.

[8] F. Blaauboer, K. Sikkel, & M.N. Aydin, "Deciding to Adopt Requirements Traceability in Practice," Proc. 19th Int Conf Adv Info Sys Eng, Trondheim, Norway, 2007, p294–308.

[9] J. Cleland-Huang, O. Gotel, & A. Zisman (Eds.), Software and Systems Traceability. Springer-Verlag, 2012.

[10] J. Cleland-Huang, J.H. Hayes, & A. Dekhtyar (Eds.), Center of Excellence for Traceability: Problem Statements and Grand Challenges (v0.1). COET-GCT-06-01-0.9, 10 Sept 2006.

[11] J. Cleland-Huang, R. Settimi, O. BenKhadra, E. Berezhanskaya, & S. Christina, "Goal-Centric Traceability for Managing Non-Functional Requirements," Proc. 27th Int Conf SE, St. Louis MO, USA, 2005, p362–371.

[12] J. Cleland-Huang, G. Zemont, & W. Lukasik, "A Heterogeneous Solution for Improving the Return on Investment of Requirements Traceability," Proc. 12th Int RE Conf, Kyoto, Japan, 2004, p230–239.

[13] J. Cleland-Huang *et al.*, "Grand Challenges, Benchmarks, and TraceLab: Developing Infrastructure for the Software Traceability Research Community," Proc. 6th Int TEFSE Wkshop, Hawaii, USA, 2011.

[14] A. De Lucia, F. Fasano, R. Oliveto, & G. Tortora, "Recovering traceability links in software artifact management systems using information retrieval methods," *ACM Trans SE Method*, 16(4), 2007.

[15] J. Dick, "Rich Traceability," Proc. 1st Int TEFSE Wkshop, Edinburgh, UK, 2002.

[16] A. Egyed, S. Biffl, M. Heindl, & P. Grünbacher, "Determining the Cost-Quality Trade-Off for Automated Software Traceability," Proc. 20th IEEE/ACM Int Conf. ASE, Long Beach CA, USA, 2005, p360–363.

[17] A. Egyed, F. Graf, & P. Grünbacher: Effort and Quality of Recovering Requirements-to-Code Traces: Two Exploratory Experiments. Proc. 18th IEEE Int. RE Conf., Sydney, Australia, 2010, p221–230

[18] A. Egyed, P. Grünbacher, M. Heindl, & S. Biffl, "Value-based requirements traceability: Lessons learned," Proc. 15th Int RE Conf, New Delhi, India, 2007, p115–118.

[19] O.C.Z. Gotel & A.C.W. Finkelstein, "An Analysis of the Requirements Traceability Problem," Proc. 1st Int Conf on RE, IEEE CS, Colorado Springs CO, USA, 1994, p94–101.

[20] O. Gotel & P. Mäder, "Acquiring Tool Support for Traceability," in [9] p43–68.

[21] O. Gotel *et al.*, The Grand Challenge of Traceability (v1.0). Center of Excellence for Software Traceability Tech Report #CoEST-2011-001, 14 June 2011. (Appears in [9] p343–409.)

[22] J.H. Hayes, A. Dekhtyar, & S. Sundaram, "Advancing Candidate Link Generation for Requirements Tracing: The Study of Methods," IEEE Trans SE, 32(1), 2006, p4–19.

[23] J. Hayes *et al.*, "REquirements TRacing On target (RETRO): improving software maintenance through traceability recovery," Innovations in Sys & Sw Eng, 3(3), 2007, p193–202.

[24] M. Heindl & S. Biffl, "A Case Study on Value-based Requirements Tracing," Proc. 10th ESEC-FSE, Lisbon, Portugal, 2005, p60–69.

[25] J. Huffman Hayes, "Risk Reduction Through Requirements Tracing," Proc. Software Quality Week 1990, SF, CA, May 1990, p1–25.

[26] W. Jirapanthong & A. Zisman, "XTraQue: Traceability for Product Line Systems," Sw & Sys Modeling, 8(1), 2009, p117–144.

[27] A. Kannenberg & H. Saiedian, "Why Software Requirements Traceability Remains a Challenge," CrossTalk: The Journal of Defense Software Engineering, July/Aug 2009, p14–19.

[28] E. Keenan et al., "TraceLab: An Experimental Workbench for Equipping Researchers to Innovate, Synthesize, and Comparatively Evaluate Traceability Solutions," (Tool Demo) Int Conf SE, Zurich, Switzerland, 2012.

[29] V. Kirova, N. Kirby, D. Kothari, & G. Childress, "Effective requirements traceability: Models, tools, and practices," Bell Labs Technical Journal, 12(4), 2008, p143–158.

[30] P. Lago, H. Muccini, & H. van Vliet, "A scoped approach to traceability management," Sys & Sw, 82(1), 2009, p168–182.

[31] Y.S. Maarek, D.M. Berry, & G.E. Kaiser, "An Information Retrieval Approach for Automatically Constructing Software Libraries," IEEE Trans SE, 17(8), 1991, p800–813.

[32] P. Mäder & A. Egyed, "Do software engineers benefit from source code navigation with traceability? - An experiment in software change management," Proc. 26th IEEE/ACM Int Conf ASE, Kansas, USA, 2011, p444–447.

[33] P. Mäder, O. Gotel, & I. Philippow, "Getting Back to Basics: Promoting the Use of a Traceability Information Model in Practice," Proc. 5th Int TEFSE Wkshop, Vancouver, Canada, 2009.

[34] P. Mäder, O. Gotel, & I. Philippow, "Motivation Matters in the Traceability Trenches," Proc. 17th IEEE Int RE Conf, Atlanta GA, USA, 2009, p143–148.

[35] A. Marcus & J.I. Maletic, "Recovering documentation-to-source-code traceability links using latent semantic indexing," Proc. 25th Int Conf SE, Portland OR, USA, 2003, p125–135.

[36] G.C. Murphy, D. Notkin, & K.J. Sullivan, "Software Reflexion Models: Bridging the Gap between Design and Implementation," IEEE Trans SE, 27(4), 2001, p364–380.

[37] NASA Office of Safety and Mission Assurance. Software Safety Standard NASA-STD-8719.13B w/Change 1. 8 July 2004. (NASA HQ, Washington, DC 20546.)

[38] C. Neumüller & P. Grünbacher, "Automating Software Traceability in very Small Companies: A Case Study and Lessons Learned," Proc 21st IEEE Int Conf ASE, Tokyo, Japan, 2006, p145–153.

[39] N. Niu & S. Easterbrook, "Extracting and Modeling Product Line Functional Requirements," Proc. 16th IEEE Int RE Conf, Barcelona, Spain, 2008, p155–164.

[40] M.C. Panis, "Successful Deployment of Requirements Traceability in a Commercial Engineering Organization…Really," Proc. 18th IEEE Int RE Conf, Sydney, Australia, 2010, p303–307.

[41] K. Pohl, Process-Centered Requirements Engineering. John Wiley Research Science Press, 1996.

[42] Radio Technical Commission for Aeronautics, Inc. (RTCA), DO-178B: Software Considerations in Airborne Systems and Equipment Certification. Issued 12-1-92, Prep. by SC-167, Supersedes DO-178A, Errata Issued 3-26-99.

[43] B. Ramesh, "Factors Influencing Requirements Traceability Practice," Comms. of the ACM, 41(12), 1998, p37–44.

[44] B. Ramesh & M. Edwards, "Issues in the Development of a Requirements Traceability Model," Proc. 1st IEEE Int Symp RE, San Diego CA, USA, 1993, p256–259.

[45] B. Ramesh & M. Jarke, "Towards Reference Models for Requirements Traceability," IEEE Trans SE, 27(1), 2001, p58–93.

[46] B. Ramesh, C. Stubbs, T. Powers, & M. Edwards, "Requirements Traceability: Theory and Practice," Annals SE, 3(1), 1997, p397–415.

[47] B. Randell, "Towards a methodology of computing system design," in P. Naur & B. Randell (Eds.), NATO Software Engineering Conference, 1968, Report on a Conference Sponsored by the, NATO Science Committee. Garmisch, Germany, p204–208 (7–11 Oct).

[48] G. Spanoudakis & A. Zisman, "Software Traceability: A Roadmap," in S.K. Chang (Ed.) Handbook of Software Eng. and Knowledge Eng., Vol.3: Recent Advancements, World Sci Pub Co., 2005.

[49] G. Spanoudakis, A. Zisman, E. Perez-Minana, & P. Krause, "Rule-Based Generation of Requirements Traceability Relations," Sys & Sw, 72(2), 2004, p105–127.

[50] S. Sundaram, J.H. Hayes, A. Dekhtyar, & E.A. Holbrook, "Assessing traceability of software engineering artifacts," RE Journal, 15(3), 2010, p313-335.

[51] U.S. Food and Drug Administration, General Principles of Software Validation; Final Guidance for Industry and FDA Staff. 11 Jan 2002.

[52] A. von Knethen, "Automatic Change Support based on a Trace Model," Proc. 1st Int TEFSE Wkshop, Edinburgh UK, 2002.

[53] M. Weber & J. Weisbrod, "Requirements Engineering in Automotive Development: Experiences and Challenges," IEEE Software, Jan/Feb 2003, p16–24.

[54] S. Winkler & J. von Pilgrim, "A survey of traceability in requirements engineering and model-driven development," Sw and Sys Modeling, 9(4), 2010, p529–565.

[55] X. Zou, R. Settimi, & J. Cleland-Huang, "Phrasing in Dynamic Requirements Trace Retrieval," 30th Int COMPSAC, 2006, p265–272.