

# Requirements Engineering: A Tube-map

Vito Veneziano

School of Computer Science  
University of Hertfordshire  
College Lane, Hatfield (Herts)  
AL10 9AB  
United Kingdom

## Abstract

In this paper, a diagrammatic representation of the main processes occurring in Requirements Engineering has been introduced. Adopted style has been derived by the notorious London Tube Map: it allows practitioners, academics and all stakeholders to fully appreciate the complex set of iterations as they occur during any software development project.

Some benefits have been discussed: non technical stakeholders can be provided with a simple and yet effective tool to communicate among each other; requirements engineers and analysts can easily track their activities; academics can use the map for teaching purposes as well as to remind themselves what practical help they can provide against the expected outcomes RE community would look for.

Potential developments of the map are its transformation into a system integrator, in order to facilitate practitioners in managing the fragmentation of the discipline; and the possibility of the map to become a virtual “gate” to any knowledge repository which emerges within any development project.

## 1. Introduction

About ten years ago, Nuseibeh and Easterbrook [7] published a paper, which quickly became an important benchmark to both practitioners and academics active in the field of Requirements Engineering (RE).

In that paper, a number of topics, aiming at providing their readers with “an overview of current research in RE, presented in terms of the main activities that constitute the field” had been listed and addressed. Authors’ fully sharable assumption was that “while these activities are described independently and in a particular order, in practice, they are actually interleaved, iterative, and may span the entire software systems development life cycle”.

Detailing of such activities, together with the adoption of a simple dictionary (memorable is their short and yet effective definition of process, technique and method), the critical assessment of

available literature at that time and their providing an exhaustive list of all major factors to be handled by requirements engineers contributed in making the paper be a great opportunity to better appreciate the troublesome complexity as well as the crucial importance RE plays in software design and development.

By the end of their paper, a final section was specifically allocated to what they interpreted as roadmap building, which mainly had consisted of listing “the important developments in RE during the last decade, and give our predictions about what will be important in RE research for the coming decade”.

Trends and predictions were proven to be accurately defined and incredibly useful for further research and developments (an interesting review is available in [2]), but the way the roadmap had been defined seemed somehow self-limitative and misleading, as if the mountain, representing the overview of “the main activities that constitute the field” (which engaged the vast majority of that paper, and possibly still holds most of its value), delivered a mouse relegated to the “research” bit rather than the full body of knowledge and practice RE is made up of.

One would have little to object if their paper had been entitled: “Research in RE: A Roadmap”, but unfortunately (or fortunately?) that was not the case: and, to say the truth, indeed Nuseibeh and Easterbrook have provided us with the description of another more holistic, less explicit and still evident roadmap, which focuses on research-related issues, but also and mainly on RE as a whole.

In this essay, a visual representation of a wide set of activities as well as of the processes, methods and techniques by means of which these activities are conducted in RE has been derived. Such representation will be used not only in order to include most of the above mentioned roadmap and to provide practitioners and academics with a more straightforward referencing tool within their specific contexts, but also to suggest two further goals: the development of a system integrator software solution, aiming at reducing the overall multi-dimensional fragmentation currently detectable in the field of RE overall; and the exploitation of the map as a “virtual gate” to access the underlying knowledge repository each project would need to build and manage alongside its life-cycle.

For our representational purposes, the London Tube-map layout style has been adopted and adapted. Tube lines represent the main processes occurring in RE: the elicitation process, the specification process (which somehow contains the modelling process), the validation process, the usability engineering process, and the overall project management process. Each station represents one or more activities run accordingly with their own established methods. Lines get across “zones”, or “areas”, which represent an overview of the main “types” of processes occurring in RE, whether they are needed for grounding (the outer zone), managerial (the middle zone) or modelling (the inner zone) purposes pursued during the project lifecycle.

Like a tube network in a large city, RE demands and leads stakeholders (software analysts and engineers above all, but also those who design and develop tools and techniques for aiding and enhancing RE activities, as well as all “non-techie” stakeholders) to explore several “places” of a given, broader problem domain. Each process has its own scope of influence, goals, methods and techniques: we’ll discuss how using a map should facilitate stakeholders’ job in managing requirements whatever project they are involved with, but also those who hold a responsibility in teaching, researching and improving the practice.

## **2. From a roadmap to a tube-map**

An overview of Requirements Engineering could likely end up with a discouraging outcome: the perception of a field being too fragmented and complex to ever support a unitary view. Despite this, people cope with the huge diversity of each activity’s intrinsic nature by relying on the availability of standard procedures, process models, requirements management tools (RMTs) and professional expertise in order to capture, analyse, validate and overall manage requirements.

Requirements themselves do not exist as transcendental entities in their own hyper-uranium: they are rather the product of decisions made on the basis of several factors: which technology is to be adopted, what environmental and cultural issues have to be addressed, what economical constraints must be taken into account, what is the knowledge and the suitability of given procedures within a certain problem domain, etc.

This implies that not only the object of RE activities (say, a software system to be developed), but also RE as such can be seen as a finite state system where transitions (out of metaphor, RE processes) can occur between a finite number of intermediary states (outcomes, which in turn could be representing other processes or activities) in order to achieve some other expected states (the final outcomes, such as an agreed specification document, a testing plan, a strategy to detect evolving requirements, up to the very end of the project, the so called “post mortem” examination and reflection in Software Engineering).

Another simple reason for developing a real roadmap, i.e. the actual visualisation of interrelated ordered processes, which practitioners need to follow whilst performing their activities, is that customers (in their capability of stakeholders) demand to understand what they are requested to be involved in: whilst it is not reasonable to explain in detail RE activities and scopes, customers should be granted the possibility to know what’s going on, at what stage the project is at a certain time, and how they could and should contribute in order to progress.

For these purposes, diagrammatic modelling is valuable as facilitating communication and comprehension are concerned: Larkin and Simon seminal paper [6] can just be seen as an obvious and still due reference.

Other authors have spent significant efforts in using UML (and especially activity diagrams) to model not only the systems Software Engineering aims at designing and developing, but also to represent Software Engineering as such (as well as other complex processes, like business processes), i.e. the array of professional activities throughout which the software (or any other product) is delivered [1, 3, 9]. Out of the many benefits, it is no surprise that facilitating an understanding and creating a visual diagrammatic representation of the process itself can be counted in.

Such a benefit becomes even more valuable in RE, not only because, as mentioned above, it consists of a well-balanced mix of technical and non-technical set of activities to be performed by a variety of stakeholders who need to understand the process they are contributing to generate, but also because of its own intrinsic nature, as RE can be seen like a process that recursively occurs across different engineering and management practices at different level and stages, and still within its own peculiar modalities. Modelling RE via UML can be seen, at least, as a good way to demonstrate the modelling power of the tools and the techniques RE is mostly made up of.

This paper, and the tube map it hereby includes, pursues a different goal.

Although they can facilitate understanding and communication between and towards non technical users and stakeholders, UML diagrams are mainly technical instruments aiming at providing or supporting a formal description of the system to be designed, and, in last instance, they reckon on the programmer more than on the customer.

A tube map is not a technical instrument, strictly speaking, even if practitioners can obviously find it valuable as well as the “non-techies” for planning and tracking their activities: it aims at facilitating an immediate identification/recognition of certain states of a complex system, the events and processes they are related to, and at sharing this information with others.

The style for designing such a tube map has been derived and adapted by the notorious London tube map.

To our knowledge, this is the first attempt to focus on Requirements Engineering and represent it by means of a visual format derived by the London tube map. London tube map diagrammatic version (i.e., similar to the version we all know nowadays) was first designed by Harry Beck in 1931 [5]: earlier versions (in the late XIX century) represented the tube underground layout isomorphic to the street layout over-ground, therefore sacrificing readability for sake of realism, although it was Fred.H. Stingemore, between 1925 and 1932, who first introduced some distortion into the original geographical layout in order to make maps more readable.

Beck’s 1933 tube map, as reported in Figure 1, is obviously the parent of modern maps, the way we all know them today.

Figure 1. Henry Beck's 1933 London Tube map.

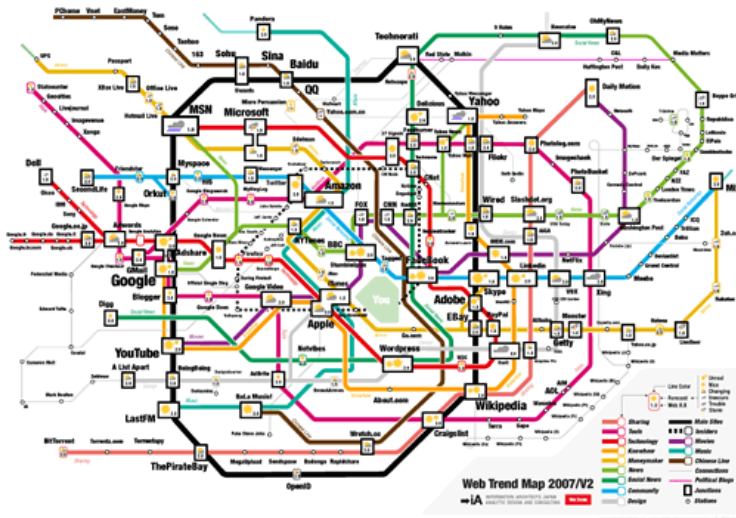


(From <http://homepage.ntlworld.com/clivebillson/tube/tube.html>, all rights reserved to their owners).

The London tube map has become, over the years, not only an icon for the capital of England and the United Kingdom: its fashionable and highly readable visual format had been re-used in a number of different fields.

One of the most famous applications has been the Internet. InformationArchitects.jp [w2] has been publishing for a number of years what they called the “Web Trend Map”, as in Figure 2.

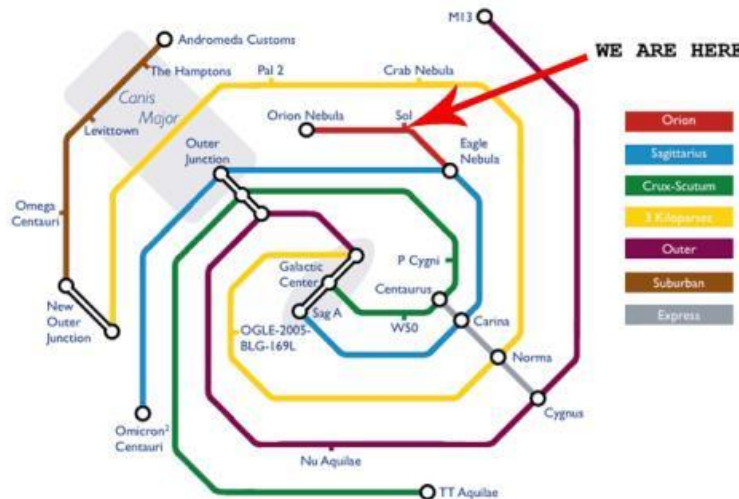
Figure 2. InformationArchitects.jp (2007) Web trend Map v.2



(<http://informationarchitects.jp/ia-trendmap-2007v2/>) © 2009 Information Architects, Inc.: all rights reserved to their owners)

Another interesting tube map applied to a completely different context has been designed by Harvard scientist Dr Samuel Arbesman, displaying our galaxy, the Milky Way [w4], in what he defined a “familiar” manner, as represented in Figure 3.

Figure 3. Milky Way Transit Authority

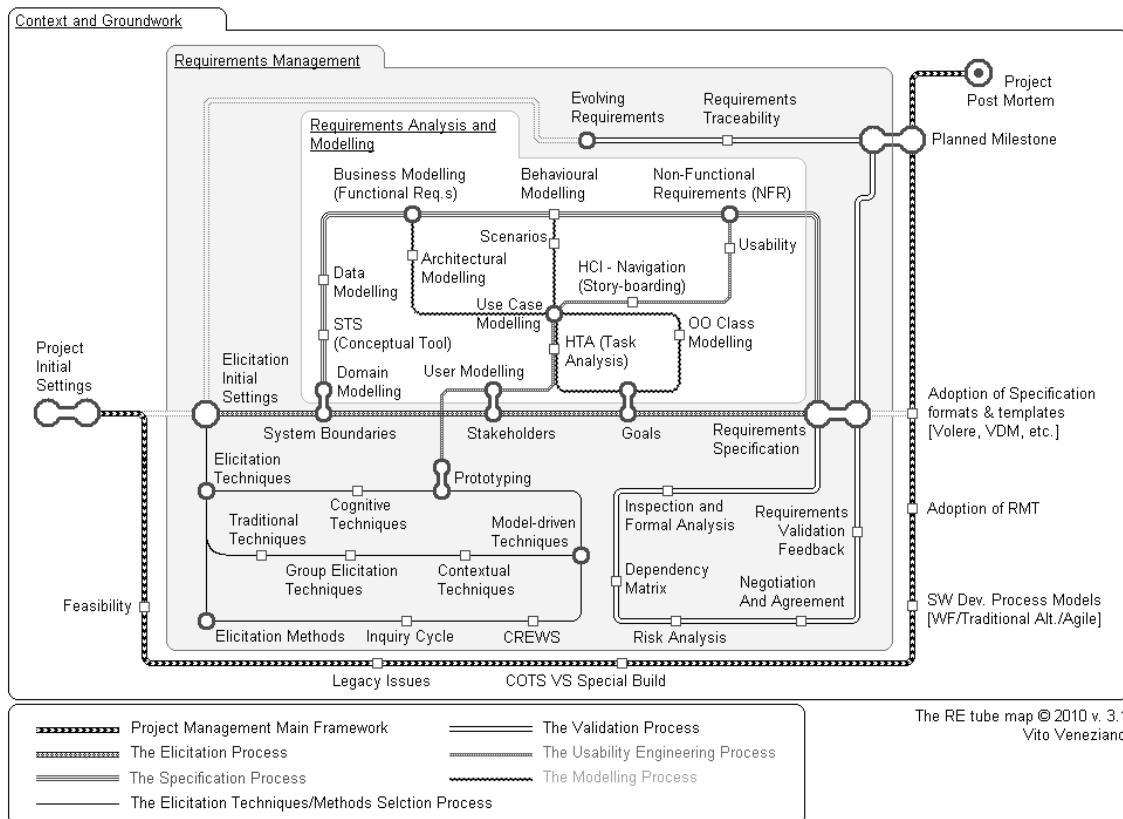


(<http://arbesman.net/milkyway/>) © 2009 Samuel Arbesman: all rights reserved to their owners)

Similar applications of the tube-map have been developed for modern music [w1] and locations of all the subways in the world [w3].

This representation format seems suiting perfectly RE, which processes are logically (and sometimes even practically) interleaved, as in the case of “prototyping”, which can be seen as a technique used to elicit requirements and the beginning of the so-called “Usability Engineering” process, as displayed in Figure 4.

Figure 4. The RE tube map.



A colour representation of the map is also available for downloading from [w5].

### 3. To whose benefit?

Many stakeholders within the RE process as well as professionals from other fields and with different goals may benefit from the above tube map.

Nevertheless, two actors have been given priority whilst developing the tube map: the RE practitioner and the academic.

The practitioner (usually a software engineer with an interest in system analysis and design) should find the map useful for at least two purposes: as a quick reference to the processes s/he is expected to manage; as a simple model upon which to plan and track her/his activities within a

certain software development project; and as a document upon which to support any communication with other stakeholders, such that they could appreciate the complexity and the scope of the project management process alongside software development.

Academics, on their side, might be well aware of the obstacles students experience when learning RE. The body of the discipline appears fragmented, complex, and, especially at undergraduate level, students might easily lose their focus on what they are studying, confused by a never-ending flat spin between some highly technical tasks, which include learning the number of different notations and diagrams used for development, and how and when to use them, and some other less technical and still challenging activities they need to acquire as crucial components of the requirements engineer's job, such as elicitation techniques, conflict negotiation and overall project management.

The RE tube map is likely to provide students with an easy, logically structured reference to the field, and a visual support for making their learning efforts more effective.

Academics themselves, whose research interest covers RE, can find difficult sometimes to calibrate their more theoretical investigations over and against practical expectancies the practitioner might be holding, with the risk they can end up with unneeded "blue sky" research: a map of the professional activities is likely to act as a polite reminder, and a help for the identification and the analysis of those issues "that really matter" in RE.

Last but not least, the tube map can be proven a valuable tool to let the "users", the domain experts, the managers, the decision makers, etc., familiarise with RE activities. It is fair to assume that those who, within a project and in any capability, need to be involved within elicitation, specification or validation, deserve the right to quickly gather information about what is going on, without relying on previous competences or paying the cost of a long learning process.

An informed stakeholder, aware of the complexity in RE, is likely to be more motivated and participative during requirements elicitation, more committed in making stable decisions and assuming her/his responsibilities about the system to be developed, and less prone to criticize the financial implications of undertaking a certain project within a professional perspective.

#### **4. How to read the tube-map**

The tube map has been built not only upon, but also around the software requirements engineering life-cycle. The "geography" of the map consists of three "areas" or "zones", each surrounding the other, and representing the "types" of processes occurring in RE.

The outer zone (namely, "Context and Groundwork") represents what Finkelstein [4] describes as preparation before a project starts: specifically, context would include "organisational



setting”, “contract and procurement procedures”, “personnel and staffing the requirements engineering process”; groundwork enlists “bounding” and “feasibility and risk”. Whilst not all of these activities have been displayed on the current version of the tube map, an attempt has been done to capture their being part of an overall “project management main framework”, which is represented by the only tube line crossing the zone.

The middle zone (“Requirements Management”) is self-explicative: it includes all processes, which “main” scope is management. Elicitation and Validation are mostly concerned with managing activities resulting into or from stakeholders’ interaction. Some processes, such as the “Selection of Elicitation techniques” or the “Elicitation Process” itself, include stages/activities which are part of or hold a direct relationship with processes belonging to the other zones. This means that “the same” activity can be identified, understood and performed under more than one different perspective.

The inner zone (“Requirements Analysis and Modelling”) includes processes (or part of them) which mainly converge onto the Requirements Specification outcome. Not only the “Specification Process” as a whole, but also two other major processes (which somehow spin off from the specification process) are mainly developed within this third zone: Usability Engineering and Modelling, respectively holding the responsibility for modelling the (main) non-functional and functional requirements for a given system/project. This third zone is populated by mainly “technical” processes, where requirements engineers need to apply specific professional techniques, usually derived by a software engineering formative background.

Still, it is worthy to note that the specification process has an important management component: this has been partially mirrored by having its final outcome (the actual process of creating the specification document) ending in the middle zone.

Even where stations names are just “nouns” (like “Stakeholders”), it is crucial to remember that they actually represent activities or processes having the noun as their object or focus (e.g. “Stakeholders’ identification”).

## **5. The tube map as a system integrator and a gate to knowledge repositories**

An intrinsic value of the tube map has been so far discussed: it mainly relies on its ability to immediately convey the “You are here” information to a number of stakeholders within a software development project.

But it should be noticed that this map holds a further, intriguing added value: as the potential system interface by means to control the processes it displays.

And, as every interface does, this map facilitates the navigation between different modules and even independent systems under a unique virtual umbrella: it becomes the opportunity to develop

a generic, high level “meta-system” by simply integrating and connecting existing tools and techniques that are, so far, available in the “practice plaza” like Leibnizian monads, i.e. individual, un-interacting, self-functional and self-referencing units.

It is worthy to underline that a need for integrating existing systems and techniques seems to be quite a common feeling among requirements engineers: just as an example, there is a stream in requirements literature, dedicated to the evaluation/comparison of tools [8], and this can be reasonably interpreted as a symptom of uneasiness, as if a fragmented market somehow contributes to the fragmentation of the overall body of RE, leaving its practitioners alone in the ocean of their job without a compass and, well... a map.

Integrating existing systems (be they tools, techniques or methods) is not the only opportunity for the tube map to become useful to the RE community: within each development project, all stakeholders contribute to the development of a specific knowledge repository, partially directly derived by activities performed, partially populated by collecting and re-using relevant references and sources available.

This “amount of knowledge” is difficult to organize and retrieve as needed, because of its intrinsic diversity and because of the variety of cultural and cognitive background people come with.

The tube map, appropriately imported into a knowledge management system as its leading navigational interface, would be likely to help any stakeholder to classify the knowledge outcomes of their RE processes in similar ways, hence improving communication and mutual comprehension within the team.

Whilst these usages of the tube map are probably not the only possible ones, their identification and better definition would greatly benefit from a broader feedback, both in terms of stakeholders viewpoints and in terms of analysis and speculation from academia.

## **6. Conclusions**

A tube map to diagrammatically display the whole set of processes occurring in RE has been introduced in Figure 4. Whilst the map itself is under continuous review and improvement, the idea of supporting a visual representation of the field, including its main professional and technical activities and outcomes, seems valid and promising.

The map supports and suggests the interaction occurring between several processes, such as elicitation, validation and specification (the tube lines), and their “typology” (the zones), by means of specific activities and or methods (the stations), alongside any software development project.

Some of the potential benefits, which have been listed above, need further analysis and longer efforts to be fully appreciated and exploited. In this paper, few hints have been anticipated towards future and hopefully fruitful research streams.

## 7. References

- [1] M. Beyer & W. Hesse, "Use of UML for software process modelling", pp. 311-312, in Bernhard Rumpe, Wolfgang Hesse (eds.), *Modellierung 2004, Proceedings zur Tagung, 23.-26. März 2004, Marburg, Proceedings*, Lecture Notes in Informatics, Volume 45 (2004), Gesellschaft für Informatik, Bonn
- [2] B.H.C. Cheng , J.M. Atlee, "Research Directions in Requirements Engineering", 2007 Future of Software Engineering, p.285-303, May 23-25, 2007
- [3] D. Jäger, A. Schleicher, B. Westfechtel, "Using UML for Software Process Modeling", Lecture Notes in Computer Science, Volume 1687/1999, pp. 91-108, Springer, 1999
- [4] A. Finkelstein, "Requirements Engineering: a review and research agenda," presented at 1<sup>st</sup> Asian-Pacific Software Engineering Conference, 1994.
- [5] K. Garland, "Mr Becks underground map: a history". Capital Transport Publishing: Harrow Weald, 1994.
- [6] J.H. Larkin & H.A. Simon, (1987). "Why a diagram is (sometimes) worth 10,000 words". *Cognitive Science*, 11, 65-99.
- [7] B.A. Nuseibeh and S.M. Easterbrook, "Requirements Engineering: A Roadmap", In A.C.W. Finkelstein (ed) "*The Future of Software Engineering*". (Companion volume to the proceedings of the 22nd International Conference on Software Engineering, ICSE'00). IEEE Computer Society Press.
- [8] A. Rainer, S. Beecham, V. Veneziano, C. Sanderson, "An investigation of the literature review as a method of tool evaluation", School of Computer Science, STCA, University of Hertfordshire, T.R. 485, March 2009.
- [9] N. Russell, W. van der Aalst, A. ter Hofstede and P. Wohed, "On the Suitability of UML 2.0 Activity Diagrams for Business Process Modelling", in *Proc. of The 3rd Asia-Pacific Conf. on Conceptual Modelling (APCCM 2006)*, Springer Verlag, 2006

## 8. Web references

[w1] Modern Music tube map:

[http://blogs.guardian.co.uk/culturevulture/archives/2006/02/03/going\\_underground.html](http://blogs.guardian.co.uk/culturevulture/archives/2006/02/03/going_underground.html)

[w2] Web Trend (Internet tube map): <http://informationarchitects.jp/ia-trendmap-2007v2/>

[w3] World Metros tube map: <http://www.flickr.com/photos/anniemole/313981428/sizes/l/>

[w4] Milky Way tube map: <http://arbesman.net/milkyway/>

[w5] RE tube map (colour): [http://homepages.feis.herts.ac.uk/~comqv/RE\\_tubemap\\_v3.1.png](http://homepages.feis.herts.ac.uk/~comqv/RE_tubemap_v3.1.png)