# Documenting Theories
## *Working Group Results*

Dag I.K. Sjøberg

**Attendees.** Marcus Ciolkowski, Tore Dybå, Frank Houdek, Andreas Jedlitschka, Barbara Kitchenham, Dietmar Pfahl, Dag Sjøberg (chair), Sira Vegas and Laurie Williams

## 1 The Need for Theories

There are many arguments in favour of theory use, such as structuring, conciseness, precision, parsimony, abstraction, generalisation, conceptualisation and communication [1,9,11]. Such arguments have been voiced in the software engineering community as well [2,3,8,10]. Theory provides explanations and understanding in terms of basic concepts and underlying mechanisms, which constitute an important counterpart to knowledge of passing trends and their manifestations. When developing better software engineering technology for long-lived industrial needs, building theory is a means to go beyond the mere observation of phenomena, and to try to understand *why* and *how* these phenomena occur.

The issue of this workshop was how empirically-based theories in software engineering should be documented. Since some work in this area had already been reported by some of the attendees in the paper [5], the highlights of that paper, including Table 1, were presented using a projector. (The paper was sent by email to the attendees after the workshop.) A graph was also presented that showed which controlled experiments in software engineering used which theories in what ways.

It was suggested that it would be useful with information about which theories were used in other study types as well, not only experiments. Since such systematic reviews are formidable jobs, it was suggested to first focus on certain sub-areas of software engineering. One attendee said he would start collecting the theories used within requirements analysis.

## 2 Schemes for Documenting Theories

Table 1 shows the scheme used to document theories in the systematic review reported in [5]. The general feeling of the attendees was that this scheme seemed useful.

It was then shown how a web site in the information systems field collects and documents theories [7]. The scheme used is shown in Table 2. It was discussed whether one would like a similar web page within software engineering. The attendees were positive, but some people in the plenary session were critical when the summary from this workshop was reported afterwards. It was argued that there was a risk that one would collect many useless theories. Addressing this issue would require a careful inclusion acceptance procedure. Table 3 shows the guidelines that are used

V. Basili et al. (Eds.): Empirical Software Engineering Issues, LNCS 4336, pp. 111 – 114, 2007.

**Table 1.** Attributes of theory used in [5]

| |
|---|
| **Metadata** |
| - *Name*. The name given for the theory by the authors of the reviewed article, or by us if no explicit name is given. |
| - *References*. The literary references given for the theory. |
| - *Terminology* (*theory, model, none*). Indicates explicit use of the word 'theory', 'model' and their derivatives in referring to the theory. |
| - *Reference discipline*. The discipline(s) of an article's literary sources to the theory. |
| - *Topic*. The topic of the article in which the theory is used. |
| **Structural components – Generic (adapted from [4])** |
| - *Means of representation* (*words, tables, diagrams, mathematics, logic*). The way the theory is presented. |
| - *Constructs and relationships*. Examples of main constructs and relationships found for the theory. |
| - *Boundary conditions*. Indications given of the theory's boundary conditions. |
| **Structural components – Contingent on theory type (adapted from [4])** |
| - *Causal explanations*. Indications that the theory gives statements of relationships among phenomena that show causal reasoning (not covering law or probabilistic reasoning alone). |
| - *Predictions*. Indications that the theory gives statements of relationships between constructs in such a form that their operationalisations can be tested empirically. |
| - *Prescriptive statements*. Indications that the theory gives statements that specify how people can accomplish something in practice, *e.g.,* construct an artefact or develop a strategy. |
| **Theory role:** |
| - *design*: the experiment's research questions and hypotheses are justified or motivated by the theory |
| - *post hoc explanation*: the theory is used after the experiment to explain observed phenomena |
| - *tested*: the theory is tested by the experiment – *derivation*: derivation of theory, *instance*: instance of theory |
| - *modified*: the theory is enhanced, refined, conditionalised, *etc.* based on the experimental findings |
| - *proposed*: a major part of the theory is proposed by the author(s) in the current reviewed article, and the theory is used in one of the preceding roles or is based on treatment-outcome relations of the experiment |
| - *basis*: the theory is used as a basis for other theory used in one of the preceding roles. |
| **Calls for Theory:** Records any calls for theory or comments on the lack of theory being problematic. |

for contributors to the IS theory web site. This seems like a good starting point for guidelines to contributors to a theory web site for software engineering. Hence, Simula Research Laboratory has began building a similar site for SE theories and has initiated collaboration with those managing the IS theory web site.

**Table 2.** Attributes of theories used in the IS Field dataset [7]

| |
|---|
| - Name of theory |
| - Acronym |
| - Alternate name(s) |
| - Main dependent construct(s)/factor(s)/variable(s) |
| - Main independent construct(s)/factor(s)/variable(s) |
| - Concise description of theory (max 500 words) |
| - Diagram/schematic of theory |
| - Originating author(s) |
| - Originating article(s) - capture in separate worksheet/document |
| - Originating area (economics, psychology, etc) |
| - Level of analysis (individual, group, firm, industry, etc.) |
| - Hyperlinks to WWW sites describing theory with brief description of link |
| - Links from this theory to other theories |
| - IS articles that use theory - capture in separate worksheet/document |
| - Contributor(s) |
| - Date of latest summary |

**Table 3.** Guidelines for contributors to the IS Field dataset [7]

| |
|---|
| 1. Contributors of new theory summaries should use the format of present summarized theories and electronically submit the text to the editors using this template. |
| 2. New summaries should be complete; summaries missing significant portions will not be accepted. |
| 3. Corrections or additions to existing theory summaries are welcomed, and should be justified to the editors. |
| 4. References should be in the standard MISQ reference format: (http://www.misq.org/roadmap/standards.html). |
| 5. All contributors agree that any information submitted to this site becomes the property of the site and is presented publicly under a GNU Free Documentation License. |
| 6. Contributions will be acknowledged by email, reviewed by the editors, and may be forwarded to the site advisors for comment. Accepted contributions will, in due course, be posted on the site with attribution to the contributor. |

**Table 4.** Attributes of theories in psychology collected at [6]

| |
|---|
| - Description |
| - Example |
| - So what? |
| - Using it |
| - Defending |
| - See also (other theories) |
| - References |

Another web site to look at is one that records theories in psychology [6]. Their format of theory description is relatively simple, see Table 4. This scheme can be compared with the scheme of the IS web site by looking at, for example, the cognitive dissonance theory, which is described at both sites. The workshop did not discuss in detail what kind of scheme would be appropriate in a possible theory web site for

software engineering, but the schemes shown in Tables 1, 2 and 4 are certainly good starting points.

## References

1. S.B. Bacharach. Organizational theories: Some criteria for evaluation. Academy of Management Review, 14(4):496–515, 1989.
2. V.R. Basili. Editorial. Empirical Software Engineering, 1(2), 1996.
3. A. Endres and D. Rombach. A Handbook of Software and Systems Engineering. Empirical Observations, Laws and Theories. Fraunhofer IESE Series on Software Engineering. Pearson Education Limited, 2003.
4. S. Gregor. The nature of theory in information systems. To appear in Management Information Systems Quarterly, 2006.
5. Jo E. Hannay, D.I.K. Sjøberg, T. Dybå, A Systematic Review of Theory Use in Software Engineering Experiments, accepted for publication in IEEE Transaction on Software Engineering, 2006.
6. http://changingminds.org/explanations/theories/theories.htm
7. http://www.istheory.yorku.ca/
8. B.A. Kitchenham, S.L. Pfleeger, L.M. Pickard, P.W. Jones, D.C. Hoaglin, K. El Emam, and J. Rosenberg. Preliminary guidelines for empirical research in software engineering. IEEE Transaction on Software Engineering, 28(8):721–734, August 2002.
9. J.W. Lucas. Theory-testing, generalization, and the problem of external validity. Sociological Theory, 21:236–253, 2003.
10. W.F. Tichy. Should computer scientist experiment more? 16 excuses to avoid experimentation. IEEE Computer, 31(5):32–40, May 1998.
11. D.G. Wagner. The growth of theories. In M. Foschi and E.J. Lawler, editors, Group Processes, pages 25–42. Nelson–Hall Publishers, Chicago, 1994.