Similarity Analysis
for Process Modeling of Requirements Engineering

George Spanoudakis, Panos Constantopoulos, Miranta Kiprioti
Institute of Computer Science
Foundation for Research and Technology - Hellas
e-mail:{gs,panos,cypriot}@ics.forth.gr

Position Paper

Abstract

Similarity analysis of process models' traces could aid new enactments of these models and their revision. Both these tasks are important concerning process models describing the creative activities of requirements engineering, since they cannot always provide detailed guidance and may be incomplete. Similarity analysis detects analogies between traced enactments which reveal concrete examples on how, when and why certain steps in a process should be followed. Analogies also indicate commonalities in traces that may be missing from or violating the relevant models. These commonalities could be abstracted, by experienced software engineers, so as to refine or extend these models.

Due to their highly creative and human-oriented nature, activities of requirements engineering(e.g. acquisition and specification of requirements) can only be described through general process models that must be customized to specific needs of different projects[Mar93].

These models do not involve strict plans of actions[Rol94]. They are better described through hierarchies of decisions, which if adopted lead to actions transforming requirements specifications, and are associated with different stages in the evolution of these specifications(referred to as *situations* [Rol94]). Such models may not be always sufficient in providing detailed guidance[KTL+92]. They may also be incomplete and/or inaccurate, something indicated by violations during their enactment(i.e. execution by humans[Mad92, JPRS94]).

We propose the identification of analogies between traced enactments of such types of models as a means of improving the understanding of the processes they describe and revising them. Analogies may help software engineers realize concrete ways of operationalizing and/or decide between alternative steps of a model. Furthermore, general process models can be revised by abstracting from similarities of recorded traces.

Analogical Selection and Documentation Based on Examples. As discussed in [RP93] decision-oriented process models are operationalized by selecting decisions from a set of alternatives, associated with situations, which successfully match the current state of an evolving specification.

Depending on the generality of their associated situations, more than one alternative decisions may be available, in certain stages of a process. In such cases, detecting which of them had been selected in analogous concrete situations in the past could alleviate the selection problem.

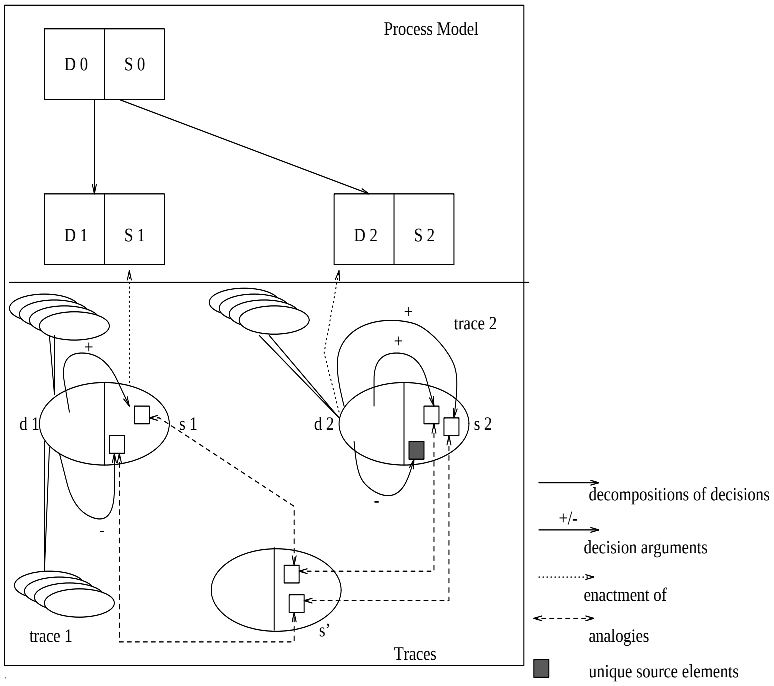

Figure 1: Analogies Between Current and Traced Situations

As discussed in [SC94a] the detection of analogies between two situations distinguishes three types of elements: the analogous elements(A) of the traced and the current situation, the unique elements(US) of the traced situation and, the unique elements(UT) of the current situation. Furthermore, elements in different situations are normally related with arguments favoring or disfavoring alternative decisions (cf. + and - arguments in figure 1). Selection can be based on heuristics combining these two roles of situation elements.

For instance, if all the A and the US elements favor and disfavor a particular decision, respectively, this decision could be reasonably suggested for adoption regarding the current situation(e.g. situations s' and s2 in figure 1). Also if all the US elements favor and all the A elements disfavor a decision, this decision could be reasonably suggested for rejection. Furthermore, alternative decisions can be associated with a similarity measure reflecting the aptness of the analogy between their own traced and the current situation [SC94b], which may further aid the software engineer selecting from them.

Analogies between traced and current situations provide concrete examples on how a decision could be operationalized. These are the specific actions in the recorded trace, which were perfomed upon the A elements of the traced situation (cf. figure 2), as a consequence of the adopted decision.

Analogies can be also used for documenting decisions, which is a prerequisite for understanding them in the future.

The analogous counterparts of the elements in a traced situation, which are involved in the documentation of its associated decision are likely to be involved in the documentation of the same decision in the current situation, in exactly the same manner.

Due to the absence of generally valid criteria determining the validity of such analogical conjectures [SC94b], it is up to the software engineer to confirm the suggested documentation as well as to provide his/her own additional arguments.

Analogical Improvement of Process Models. As the number of the available traces increases it is possible to utilize analogies between their subparts in refining and/or extending a process model.

In our view, refinements correspond to adding details to descriptions of situations and decisions of a process model. Such details can be abstracted from analogies between specific enactments of decisions. In reference to figure 2, the detection of analogies between the marked subtraces of the traces recorded as enactments of the process paths 1 and 2, could be used for refining the decision-situation pairs A and B, respectively.

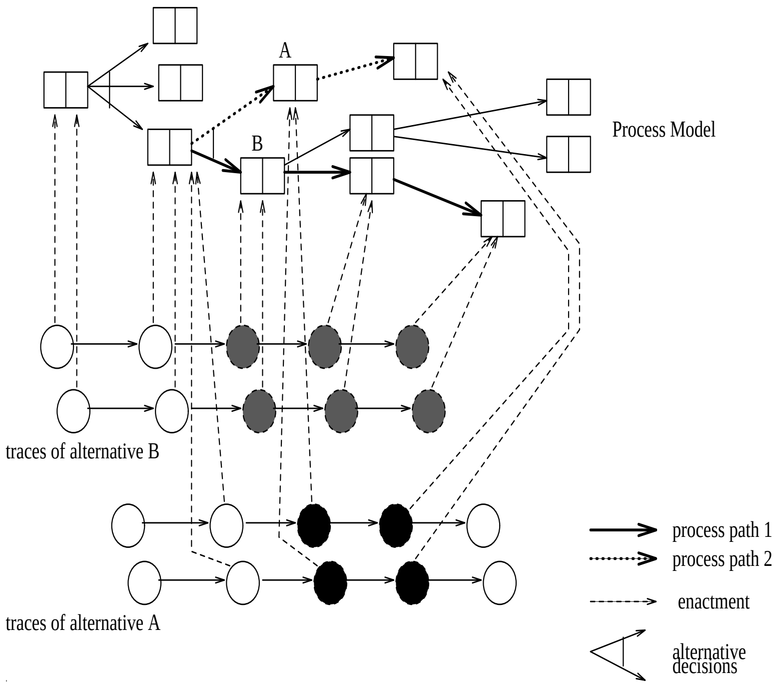

Figure 2: Traces of Common Paths of Process Models

The possible ways of abstracting from analogies depend on the representation of situations and decisions at the trace and the process model level [MK90]. Given particular representation schemes for these two levels, it will be necessary to identify different possible abstraction operators, associate them with specific kinds of analogies between the involved traces and let some human, presummably expert in revising the process models, to decide to apply these operators if the relevant analogies are detected.

The second type of enrichment is the introduction of new decision/situation pairs in a process model by recording violations of this model and detecting analogies between them.

Violations occur when all the available alternative decisions or some of the actions operationalizing a decision(i.e. *original decision*) in a process model are not selected in a particular enactment. They are recognized by the execution of an action not predicted by the process model, in reference to some situation. Violations correspond to finite sequences of actions, transforming the product of the process. These actions generate new situations which are matched against the situations associated with decisions in the model. If successful, this matching may make software engineer reender the model by selecting a suggested decision (i.e. *terminating decision*), thus notifying the end of the violation.

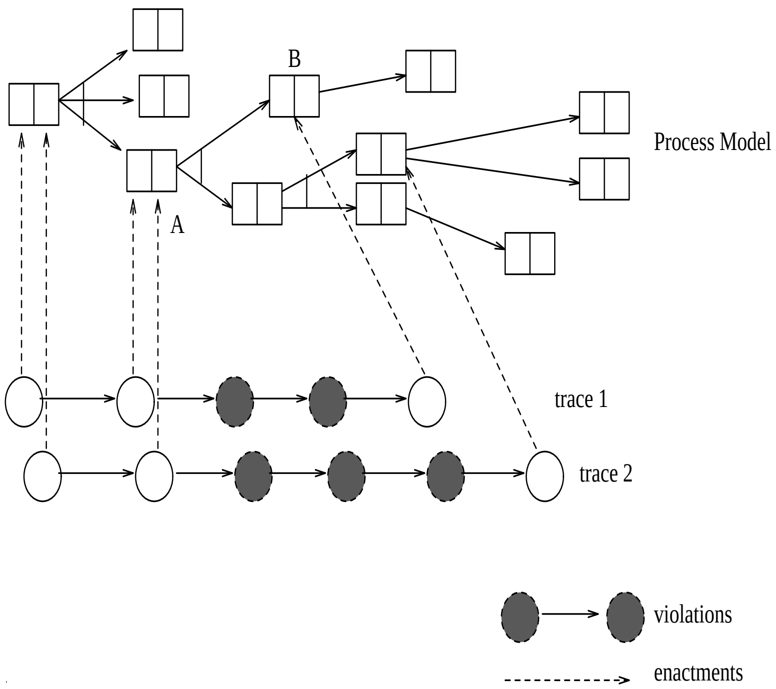

Figure 3: Violations of Common Parts of Process Models

Therefore, it is not only possible to monitor and record violations but also to relate them with original and terminating decisions in process model(e.g. decisions A and B for trace 1 in figure 3). Analogies between traced violations with the same original and terminating decisions can also be abstracted into general situations and thereby, into new decisions rationalizing the undertaken actions in a process model by a human expert. After the introduction of new decision/situation pairs, the violations caused them, become their own maintained traces.

Conclusion. Analogies between traced enactments of process models, could be exploited in selecting and documenting decisions of process models as well as refining and extending these models.

Several aspects of this approach need further research, including:

- relating decisions with specific elements in evolving specifications;
- representing traces in a way enabling the detection of analogies;
- formulating selection heuristics based on relations of specification elements to decisions and analogies between elements in traced and current situations; and,
- identification of abstraction operators for refining and extending process models, pertaining to specific schemes for representing process models and traces as well as to particular kinds of analogies between traces.

References

[JPRS94] M. Jarke, K. Pohl, C. Rolland, and J.-R .Shmitt. Experience-based Method Evaluation and Improvement: A Process Modeling Approach. In *IFIP WG 8.1 conf. CRIS 94, Maastricht, Netherlands*, September 1994.

[KTL+92] Herb Krasner, Jim Terrel, Adam Linehan, Paul Arnold, and William H.Ett. Lessons learned from a software Process Modelling System. *CACM*, 35(9), September 1992.

[Mad92] N.H. Madhavji. Environment Evolution: The prism model of changes. *IEEE Transactions on software engineering*, 18, May 1992.

[Mar93] Pentti Marttiin. Towards flexible process support with a CASE shell. In *metaPHOR project*. University of Iyvaskyla Finland, 1993.

[MK90] Ryszard S. Michalski and Yves Kodratoff. *Machine Learning: An AI Approach*, volume 3, chapter Research in Machine Learning: Recent Progress , Classification of Methods, and Future Directions, pages 2--29. Morgan Kaufmann, 1990.

[Rol94] Colette Rolland. A Contextual Approach for the Requirements Engineering Process. In *proc of 6th International Conference on Software Engineering and Knowledge Engineering.* Jurmala ,Latvia, June 1994.

[RP93] C. Rolland and N. Prakash. Reusable Process Chunks. In *Proc of 4th International Conference on Database and Expert Systems Applications.* DEXA93, Prague Slovakia, September 1993.

[SC94a] G. Spanoudakis and P. Constantopoulos. Measuring Similarity Between Software Artifacts. In *proc of 6th International Conference on Software Engineering and Knowledge Engineering.* Jurmala ,Latvia, June 1994.

[SC94b] G. Spanoudakis and P. Constantopoulos. Similarity for Analogical Software Reuse: A Computational Model. In *Proc of the 11th European Conference on Artificial Intelligence.* Amsterdam, The Netherlands, August 1994.