

The Third Semat Workshop: Approaching the First Architecture Spike

September 29 – October 1, 2010 Milan Italy

Prepared by Shihong Huang and Paul McMahon

The 3rd Semat workshop was held on September 29th – October 1st, 2010 in Milan, Italy. The workshop was attended by a group of people who are actively participating in Semat's work, including track leaders, track members, and industry people.

1 The Activities

The first day was a pre-meeting day dedicated to the individual track discussions and planning for track presentations. The second day focused on status reports from individual tracks that summarized what had been achieved since the 2nd workshop in Washington D.C., followed by discussions. The day began with a Troika opening speech recapping the Semat vision, the state of the initiative, and the objectives of the Milan's meeting. The day ended with Semat governance discussions. The third (the last day) of the workshop addressed issues raised the day before, the Semat China status report, the next architecture spike, and laid plans for next Semat meeting venues.

2 Presentations

The workshop commenced with Ivar's presentation "Semat Milan Introduction". People work for Semat voluntarily – because they see the potential value of Semat in the software community. The five key ideas of the Semat solution are:

- 1) A method is a composition of practices
- 2) The kernel consists of two things: a) the universals, which form the cornerstones of software engineering, and b) the kernel language used to describe methods, practices, and the elements of practices
- 3) The primary users of methods and practices are project participants (developers, testers, project leads, etc.)
- 4) Methods need theory – our work must stand on a solid theoretical basis. Methods are composed of practices. Practices are described in terms of universals and elements such as activities and work products; all are formalized into a language and is the beginning of such a theory.
- 5) Enactable, Executable, and Operational: Significant improved software engineering performance and cost reduction through common ground and alpha states.

The relationship between Semat and earlier work such as SPEM, OPF, EPF, UP, SWEBOK and CMMI was discussed and how these earlier work will play a significant role in Semat.

Each track lead presented the progress that has been achieved and the future work.

2.1 The Requirement Track

The Requirements track was created to provide the context and scope of Semat. It identifies the use cases to be supported by Semat and it prioritizes them. The results of the track are to describe some use cases, to provide a common glossary for the kernel language, and to facilitate and support other tracks.

The development of the requirements took a route through two independent actions: use case definitions and domain model definition. In the domain model considerable work has been done to define the context of the use of the language and beyond.

The presentation recapped the work that had been done since Zurich meeting (see 2nd Semat Workshop Report), and work completed since Washington D.C. meeting. Based on Zurich brainstorming scenarios, requirements track use cases, Semat Vision statement, and Kernel language Domain Definition, the track identified five use cases that are priorities for the kernel language – Define Practice Definition, Established well-formed Practice, Plan Based on Method, Compose Method, and Evaluate practice/method. For the architecture spike, the work has been focused on Define Practice Definition and associated Domain model. The Define Practice Definition use case basic path was tested and enhanced by the architecture spike. A detailed illustration of how to establish a well-formed practice from existing method was presented. The steps included analyzing method, scoping the Practice, outlining the Practice, and completing the Practice Definition. The well-formed practice then can be combined with other practice definitions.

The next tasks are to refine Define Practice Definition use case by addressing issues raised, including missing steps and agreement upon Activity Spaces & association of Activities, how to deal with Roles, etc. For Domain Model & Class definition, the track will investigate issues raised by the architecture spike and aligning class definitions with the emerging Domain Model.

2.2 The Universal Track

The universals capture the common ground that is shared by all software engineering endeavors. As a whole, the set of universals must be small, but comprehensive. This criterion applies to the collection of the universal elements; together, they must capture the essence of software engineering, providing a map that supports the crucial practices, patterns and methods of software engineering teams. The universals track's role is to identify the universal elements of software engineering, which must be integrated into the Semat kernel.

The universal track has identified eight universals, i.e., Opportunity, Stakeholder Community, Requirement, Software System, Work, Team, Method, and Practice. The track also has identified 232 Practices (still growing). The Practices list is classified into six different categories. Seven Practices have been classified and ranked. The top two practices – Scrum and Iterative Development – were selected for further elaboration. As an example, SCRUM was presented to follow the Define Practice Definition steps as an illustration of how a practice can be mapped to the emerging Semat domain and kernel language models.

The next tasks are to transform the set of universals to use the kernel language, invite comment and discussion, and Integrate into the Domain Model; identify more and different kind of universals, create an initial software kernel, more in-depth analysis of the universals found so far, and validate against important practices and methods.

2.3 The Assessment Track

The focus of the Assessment track has been on two products: the Semat Assessment Framework and the Assessment of the Architecture Spike. The purpose of the Semat Assessment Framework is to help Semat users self-assess their use of the Semat product. The initial version of the framework has also been employed internally by Semat working groups to help decide if we are ready to “go-live” with the Semat product. The Semat Assessment Framework includes three sections corresponding to the life cycle of a methodology: Design, Enactment and Improvement/Performance. The framework distinguishes two types of assessments: Capability and Performance. Assessments during design and enactment are both capability assessments, while assessments during improvement/performance can be both capability and performance assessments.

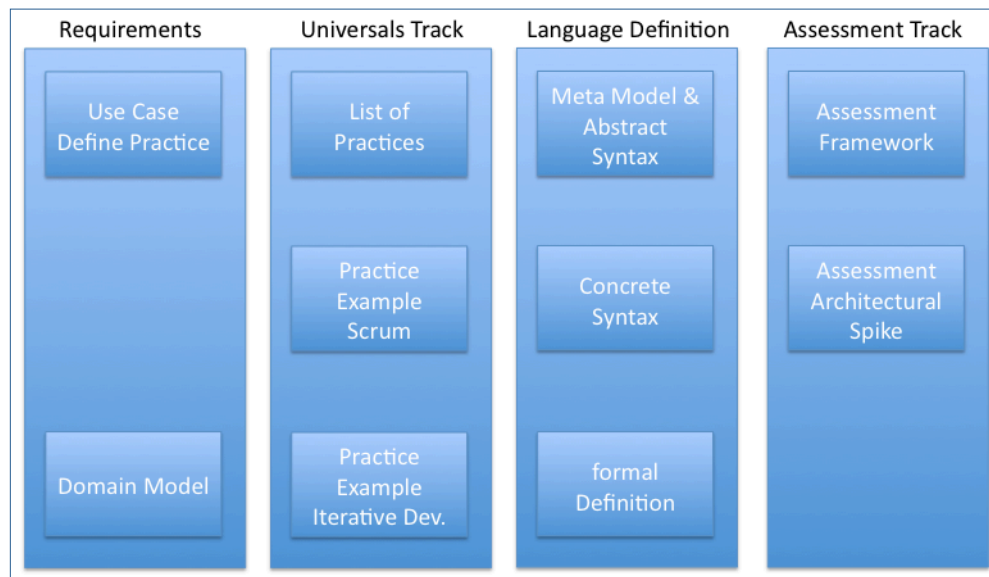
The Assessment of the Architecture Spike reviewed it from three perspectives: the criteria listed in the draft Assessment Framework, Semat Vision Statement goals, general assessment observations, and comments summary. Each issue raised through the assessment was identified as “current issue”, “issue resolved”, or “guidance required”. The assessment of the two architectural spikes provides the insight to what has been achieved and what is missing and needs change in the definitions. The general observation is that the Architectural Spike is viewed as a great success.

Some of the feedback to the Assessment track is the need to add empirical evaluation to the assessment work.

2.4 The Kernel Language Track

The kernel language is for defining universals, practices and methods. The kernel language report first presented an overview of the activities and results from different tracks (Requirements, Universals, Language Definition, and Assessment) from the language point of view. The language definition includes Meta Model (abstract syntax, well-formedness rules, operational semantics), concrete syntax, and formal definition.

The overall architecture and work that has been done from kernel language perspective is shown in the figure below.



The next tasks are updating abstract syntax based on concrete examples, composition of practices, generalization the model, and refining and polishing the specification.

The current version of kernel language include 21 notations to represent Alpha, Work Product, Activity Space, Area of Concern, Activity, Competency, Competency Level, Alpha State Machine Diagram etc.

The formal definition of the kernel language has 2 parts: Precise informal English definition (part 1), and Formal definition of the language (part 2). The necessary structures are represented as graphs, and changes and checks are given by transformation rules. The future work includes formalizing composition, operational semantics (enactment), and looking for prototyping tools.

Bertrand Meyer presented “Method and Notation: Reflections and Proposal”. He spoke that the purpose of modeling is to learn more about what is being modeled than we already know. He gave the English definition of a Practice, and then showed an example of a precise definition of Practice, Composite Practice, and steps taken to reach the formal description. During the 3rd day’s discussion, these steps were being mapped one-by-one to the tasks that are being and will be carried out in the kernel language track. For example, “natural language example” step can be mapped to scenarios of use cases; “natural language definitions” can be mapped to the description of the semantics definitions, description of the business rules, domain model classes and associations, whereas “formal definition” is being done with “seamlessness”.

3 Semat Working Structure

Based on the work has been done and the need of track coordination, a new track – Architecture track – is added to the existing track structure. The new Semat working structure includes 6+1 tracks:

Architecture (Spike) track: led by Paul McMahon, coordinates activities and focus of the other tracks, works with track leads and the troika to prioritize architecture spikes.

Requirements track: led by Dave Cuningham, works on Domain Model and Use cases.

Theory Track: led by Michael Goedicke, provides formal explanations of the concepts.

Kernel Language track: led by Michael Goedicke, works on meta-model, abstract syntax, concrete syntax, and formal definition of the language.

Universal Track: led by Ian Spence, works on the states of alphas, creates new universals, and identifies more practice examples to apply Semat output.

Assessment track: led by Paul McMahon, will continue refining and elaborating on the Semat Assessment Framework, and Assessment for future architecture spikes.

Definition Track: Led by Mira Kajko-Mattsson, tracks the concepts developed in other tracks and ensures that each has a definition that is clear, rigorous, acceptable to the industry, legally publishable and not contradictory nor unnecessarily duplicative with Semat or other terms. The track will also be responsible for releasing Semat related documents.

4 Semat Work Flow and Document Process

Issues were raised during the meeting regarding the common infrastructure, document version control and access, consistency among Semat terms and definitions, and coordination among different tracks. These issues were given substantial discussion during the 3rd day meeting. The actions to address these issues are: each track will keep their own work-in-progress documents, open to everyone in the Semat working group to view, but keep the discussion within the individual group. After reaching an agreement within the group, the released baselined documents will be version controlled. The Definition track will then check the consistency of terms and definitions among different documents before release to the public (subject to the feedback and approval by the troika if necessary). Origo (<http://www.origo.ethz.ch/>) will be used for version control.

5 Action Items and General Goals

The next architecture spike will be formalized by the Architecture Track. Action items of each track are mentioned in the respective sections above. Some of the general goals and deliverables include:

- Define universal state, assertion of alphas
- Define a set of universals and validate against set of practices
- Further refine kernel language to describe universals and compose practices
- Create a set of metrics to access process, product and people, and validate them
- Provide training material and menus for Semat output

6 Final remarks

Much progress has been made along all tracks since July 2010. The architecture spike was a success – it defined domain model, Defined Practice Definition use case; defined Kernel language abstract syntax, concrete syntax, notations; identified six kind of practices, eight universals, prioritized seven practices to be tested; produced Semat Assessment Framework and Assessment for Architecture Spike; used meta-model to describe two practices – Iterative Development and Scrum, applied Define Practice Definition to Scrum and Iterative Development; addressed Semat workflow, document version control, track collaboration and governance issues; laid out tasks before the next meeting.

Semat China Chapter is being established. New signatories, such as Beijing University, China Software Developer Network (over 10 million subscribers), and China System and Software Process Improvement Association – a major representative of China Software Industry in China – have endorsed Semat. Semat China will work closely with Semat international towards the specific needs in China. Three seminal Semat articles by the Troika and an introductory article by Shihong Huang will be published as a Semat series in the Programmer Magazine in China.

7 The Future Venue

Detailed information about the 4th Semat workshop will be posted on the Semat website (www.semat.org).