A Bootstrap Theory: The SEMAT Kernel Itself as Runnable Software

Iaakov Exman

laakov [at] jce.ac.il

June 2014

Copyright by Iaakov Exman (c)
Question

The Essence of Software Engineering?
Goal

The Theory Itself as *Runnable* Software
Agenda

Part ONE
1- The SEMAT Kernel: \textit{Gedanken Experiments}
2- Software \textit{Automation} Test
3- Distributed Software Test
4- Self-Evolving Software Test

Part TWO
5- A Bootstrap Theory
6- Kernel alphas: \textit{Top Level Ontology}
7- Discussion

June 2014
Copyright by Iaakov Exman (c)
The SEMAT Kernel

Fig. 1: Things to Work With

- Opportunity
- Stakeholders
- Requirements
- Software System
- Work
- Team
- Way of Working

Customer
Solution
Endeavor

Part ONE

Alphas
Is the KERNEL testable in a formal sense?

**Definition**

**Essence** = *Necessary & Sufficient* Concepts to develop \( p \)

\( \forall \text{Programs } p \)
Submit it to Thought Experiments in the tradition of physics and philosophy.

The philosophical challenge: How can we learn about reality just by thinking?
ALPHAS that fit Automation

Current Kernel Assumption: Some of the Alphas involve or are performed by people.

• **Stakeholders**: The people, groups, or organizations that affect or are affected by a software system;
• **Team**: The group of people actively engaged in the development, maintenance, delivery and support of a specific software system;
• **Work**: Activity involving mental or physical effort done in order to achieve a result.
ALPHAS’ Automation

→ Allow partial/complete automation: Software replaces people.

Deep Reasons:
• Reliability
• Formalization
**ALPHAS’ Automation**

**Satellite** sent to another planet. Self-testing periodically or by trigger events. **Bug** found in software sub-system.

**Scenario**

**Software Stakeholder:**
Decision mechanism whether to signal planet Earth.

**Opportunity:** Signal received in planet Earth.

Same **Alphas** – slightly revised – encompass **software automation**.
**Distributed Software**

**Scenario 3**

**Group of Satellites** sent to another planet.
Elect among themselves sub-set of communicators.
**Bug** found in software sub-system.

**Distributed Software Stakeholders:**
News spread among satellites.
Mutual consultation whether software replacement needed.
**Distributed teams:** Software & human workers solve the problem.
Self-Evolving Software

1- Group of Robots compose themselves
   e.g. to climb stairs.
   **Opportunity** provided by surrounding obstacles.

2- **Self-stabilizing systems** modify themselves
to restore desirable conditions of execution.
   **Software Stakeholders/Teams** are sub-systems
   of the target software.
Intermediate Conclusion

Extrapolating From Gedanken Experiments

ALPHAS indeed elastic enough

→ A plausible Essence candidate

…wish to go beyond plausibility
Bootstrap Theory

Similarity of Kernel to Software:

Kernel is:
• Hierarchical
• Actionable
• Extensible
• Runnable
Bootstrap Principle [for Software Theory]:

A *software theory* should itself be *runnable* software.
### Bootstrap Theory

<table>
<thead>
<tr>
<th>Consequences</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Testability</strong></td>
<td>beyond gedanken experiments</td>
</tr>
<tr>
<td><strong>Precision</strong></td>
<td>amenable to quantitative criteria</td>
</tr>
<tr>
<td><strong>Universality</strong></td>
<td>get limitations of acceptable inputs</td>
</tr>
</tbody>
</table>
Kernel ALPHAS: a top-level ontology

Essence Statement:

The SEMAT kernel alphas, the top abstraction level of the kernel hierarchy, can be viewed as a top-level ontology.
Kernel ALPHAS: top-level ontology
Completeness & Consistency

Axiom

“all requirements are fulfilled by modules of the software-system”

...
Discussion

*Alphas* refined?
e.g. opportunity provided by environment obstacles

*Software* is the only engineering discipline whose entities can be their *own theories.*
Summary: A testable *Theory*

1. **Conceptualize**

   The **essence** of each *software* system is its *TOP Ontology*.

2. **Runnable Software**

   *Necessary & Sufficient* Concepts to *bootstrap* a *runnable software*. by new developed & implemented tools