A White-box Perspective on Self-Adaptation and Self-Awareness
(with a focus on Reflective Russian Dolls)

Alberto Lluch Lafuente
alberto.lluch@imtlucca.it

Roberto Bruni, University of Pisa
Andrea Corradini, University of Pisa
Fabio Gadducci, University of Pisa
Andrea Vandin, IMT Lucca
why
adaptation?
ensembles

“...open-ended, highly parallel, massively distributed systems...

...made of self-aware, self-adaptive, [...] autonomic components.”

- ascens-ist.eu
Robot Swarms

Cloud Computing
everything and the kitchen sink

E-vehicles
the goal of ascens

“...build ensembles in a way that combines software engineering, formal methods and autonomic, adaptive, self-aware systems.”

- ascens-ist.eu
what is adaptation?
On the Definition of Adaptivity*

In recent years, the notion of an adaptive system has come to play an important role in the theories of communication and control. Nevertheless, it is difficult to find a satisfactory explanation, much less precise definition, of this notion in the literature.\textsuperscript{1-4} Much of the vagueness surrounding the notion of adaptivity is attributable to the lack of clear differentiation between the external manifestations of adaptive behavior on the one hand, and the internal mechanism by which it is achieved on the other. To subsume both under a single concise definition has proved to be an elusive objective, since it is very difficult—perhaps impossible—to find a way of characterizing in concrete terms the large variety of ways in which adaptive behavior can be realized.

- Lotfi A. Zadeh, Proceedings of the IEEE, 1963
adaptation

“... changing something (itself, others, the environment) so that it would be more suitable or fit for some purpose than it would have otherwise been”

Self-adaptive software evaluates its own behavior and changes behavior when the evaluation indicates that [...] better functionality or performance is possible.”

adaptation in software systems

“Self-adaptive software modifies its own behavior in response to changes in its operating environment.”

adaptation in software systems

“A software system is self-adaptive if it modifies its behaviour as a reaction to a change in itself or its context to maintain/improve goal achievement”
what's the problem with software?

Most programs...
- ...“change behaviour” (e.g. conditional branching);
- ...“to react to external perturbations” (e.g. inputs);
- ...“to reach goals” (e.g. functional requirements).

How do we distinguish...
- ...an adaptive software from a non adaptive one?
- ...adaptation logic from application logic?
is this adaptation?

...  
if f(x) then  
  do_this;
else  
  do_that;
endif  
...

...
is this adaptation?

“if the hill is too steep then assemble with other robots and try again”
is this adaptation?

```java
if (tooSteep || ... ){
    setLight(green);
    setState(Anti_Phototaxis);
}
```
is obstacle avoidance adaptive?

“Obstacle avoidance may count as adaptive behaviour if […] obstacles appear rarely. […]”

If the ‘normal’ environment is […] obstacle-rich, then avoidance becomes […] “normal” behaviour rather than an adaptation.”

black-box adaptation

In other fields (e.g. Biology, Control Theory)

- systems have a “default” behaviour & goal;
- adaptation reacts to “external perturbations”.
white-box adaptation
5W1H (six honest men)

(1) Why? (e.g. robustness? performance? goals?)
(2) When? (e.g. reactively or proactively?)
(3) Where? (e.g. which architectural level?)
(4) What? (e.g. which artifacts?)
(5) Who? (e.g. autonomic managers? humans?)
(6) How? (e.g. which actions, in which order?)

- R. Laddag. “Active software”. Int. Workshop on Self-Adaptive Software. 11-26, 2000
this talk...

...focuses on the “HOW”

“A software system is self-adaptive if it modifies its behaviour as a reaction to a change in itself or its context to maintain/improve goal achievement”
Conceptual white-box definitions of adaptation...
what is a program?

“program = control + data”

what is a program?

Data can be used to

- store information;
- influence the control;
- both.
what is an adaptable program?

\[ \text{program} = \text{control} + \text{data} \]
what is an adaptable program?

“adaptable program = control + data

control data + the rest”
what is an adaptable program?

adaptation is the run-time modification of control data
a program is...

“...adaptable if it has a distinguished collection of CD that can be modified at runtime.”

“...adaptive if its control data is modified at runtime for some computation.”

“...self-adaptive if it modifies its own control data at runtime.”
are these definitions...

... useful to disambiguate “adaptivity”?

... general enough & consistent with the literature?

... useful for designing & analysing adaptive systems?
on ambiguity

The choice of CD is (of course) arbitrary:
The same system can be adaptive in different ways!

but...

...the responsibility of declaring
"what is the adaptive behaviour"
is passed from the observer to the designer.
is this adaptation?

... if (tooSteep || ... ) {
  setLight(green);
  setState(Anti_Phototaxis);
}
..
design & analysis: control data & effectors
design & analysis: MAPE-K
design & analysis:
adaptation towers
design & analysis: architectural styles

do we want this?
Identify some control data such that its modification coincides with adaptation (as understood by the authors)
some examples

- Aspect-Oriented Computing [Salvaneschi et al. 2011]
- Service Component Ensemble Language [De Nicola et al. 2012]
- Specification-carrying programs [Pavlovic 2000]
- Context-Oriented Computing [Ghezzi et al. 2011]
- IBM's MAPE-K [Horn 2001]
- Run-time architectural models [Oreizy et al. 1999]
- FORMS [Weyns et al. 2012]
- Model-based adaptation [Zhang&Cheng 2006]
- Coordination contexts [Andrade&Fiadeiro 2002]
- Non-deterministic interactions [Broy et al. 2009]
- Adaptation patterns [Cabri et al. 2011]
- Reflective Russian Dolls [Meseguer&Talcott 2002]
- ...
reflective russian dolls
Semantic Models for Distributed Object Reflection

José Meseguer\textsuperscript{1} and Carolyn Talcott\textsuperscript{2}

\textsuperscript{1} Computer Science Department, University of Illinois at Urbana-Champaign, Urbana IL 61801
\textsuperscript{2} Computer Science Laboratory, SRI International, Menlo Park, CA 94025

Abstract. A generic formal model of distributed object reflection is proposed, that combines logical reflection with a structuring of distributed objects as nested configurations of metaobject that can control subobjects under them. The model provides mathematical models for a good number of existing models of distributed reflection and of reflective middleware. To illustrate the ideas, we show in some detail how two important models of distributed actor reflection can be naturally obtained as special cases of our generic model, and discuss how several recent models of reflective middleware can be likewise formalized as instances of our model.

logical reflection
reflection tower

\[ \mathcal{R} \vdash t \rightarrow t' \]
reflection tower

\[ U \vdash \langle \overline{R}, \overline{t} \rangle \rightarrow \langle \overline{R}, \overline{t}' \rangle \]

\[ R \vdash t \rightarrow t' \]
reflection tower

$\mathcal{U} \vdash \langle \overline{\mathcal{U}}, \langle \overline{\mathcal{R}}, \overline{t} \rangle \rangle \rightarrow \langle \overline{\mathcal{U}}, \langle \overline{\mathcal{R}}, \overline{t}' \rangle \rangle$

$\mathcal{U} \vdash \langle \overline{\mathcal{R}}, \overline{t} \rangle \rightarrow \langle \overline{\mathcal{R}}, \overline{t}' \rangle$

$\mathcal{R} \vdash t \rightarrow t'$
reflection tower

\[ U \vdash \ldots \]
\[ U \vdash \langle \overline{U}, \langle \overline{R}, \overline{t} \rangle \rangle \rightarrow \langle \overline{U}, \langle \overline{R}, \overline{t}' \rangle \rangle \]
\[ U \vdash \langle \overline{R}, \overline{t} \rangle \rightarrow \langle \overline{R}, \overline{t}' \rangle \]
\[ R \vdash \overline{t} \rightarrow \overline{t}' \]
adaptation tower

\[ \mathcal{R}_\emptyset + \mathcal{CD}_\emptyset \vdash t \rightarrow t' \]
adaption tower

\[ U + R_1 + CD_1 \vdash (\overline{R_0 + CD_0}, \overline{t}) \rightarrow (\overline{R_0 + CD'_0}, \overline{t'}) \]

\[ R_0 + CD_0 \vdash t \rightarrow t' \]
adaptation tower

\[ \vdash \quad \ldots \]
\[ \vdash \quad U + R_2 + CD_2 \]
\[ \vdash \quad U + R_1 + CD_1 \]
\[ \vdash \quad R_\emptyset + CD_\emptyset \]
\[ \vdash \quad \langle R_\emptyset + CD_\emptyset, t \rangle \rightarrow \langle R_\emptyset + CD'\emptyset, t' \rangle \]
\[ \vdash \quad t \rightarrow t' \]
what is Maude?

“A high-level, declarative language that supports both equational and rewriting logic computation”

– maude.cs.uiuc.edu
modular, functional, rule-based, etc.

```
mod m(x) is
    import m1, m2,...
    sort s1, s2,...
    ops f, g,...
    eq f(...) = ...
    rl l => r
endm
```
Maude is also a semantic framework

models of concurrent computation: equational programming, lambda calculi, Petri nets, process algebras (CCS and pi-calculus), actors, operational semantics of languages (via K, SOS): Java, C, Python, Haskell, agent languages, active networks languages, hardware description languages, logical framework and metatool: linear logic, translations between theorem provers, type systems, open calculus of constructions, tile logic distributed architectures and components: UML, OCL, MOF, Service architectures and middlewares, open distributed processing, models, specification and analysis of communication protocols: active networks, wireless sensor networks, firewire leader election protocol, modeling and analysis of security protocols: cryptographic protocol specification language CAPSL, MSR, security specification formalism, Maude-NPA, real-time, biological, probabilistic systems: real-time maude, pathway logic...
Maude is tool supported

- An Interpreter;
- Debuggers (declarative/interactive);
- Model checkers (e.g. LTL);
- Confluence & Termination checkers;
- A theorem prover;
- ...
maude supports logical reflection

- META-LEVEL is a module (the universal theory) where
  - modules, terms, rules, equations, sorts, etc. are data.
  - matching, rule application, rewriting, etc. are functions.

- META-LEVEL is a module so...
  - it can be treated as data again, and again...
  - enabling the tower of reflection.

- Meta-programming applications
  - transformation of modules;
  - analysis of modules;
  - ...
  - adaptation.
nested configurations
message filtering & replication
black-box adaptation
white-box adaptation
white-box adaptation

(state) (rules)

(manager)

(white-box component)

(state) (rules)

(interpreter)

(black-box component)
“Self-Awareness means that the system is aware of its self states and behaviors.”

reflective russian dolls
reflective russian dolls
mobility
“formal” autonomic managers

model checking, logical inference
concluding remarks
White box-adaptation:
- Focus on "how" rather than "why";
- Adaptation logic decided by the designers, rather than observers;

"Control Data" white box criteria:
- "adaptable program = control + {control data + other data}";
- Wide spectrum of control data: from "parameters" to "programs";

Reflective Russian Dolls:
- Support formal techniques for adaptation and awareness;
- Rely on logical reflection and wrapping techniques;
main references


- R. Bruni, A. Corradini, F. Gadducci, A. Lluch Lafuente, A. Vandin, **Modelling and analyzing adaptive self-assembling strategies with Maude**, Pre-proceedings of the International Workshop on Rewriting Logic and its Applications (WRLA'12), 2012, draft available at http://eprints.imtlucca.it/1048/
Thanks!
alberto.lluch@imtlucca.it
linkedin.com/in/albertolluch