

Title	Verifying Specifications with Proof Scores
Author(s)	FUTATSUGI, Kokichi
Citation	
Issue Date	2005-09-21
Type	Presentation
Text version	publisher
URL	http://hdl.handle.net/10119/8321
Rights	
Description	1st VERITE : JAIST/TRUST-AIST/CVS joint workshop on VERification TEchnologyでの発表資料, 開催 : 2005年9月21日 ~ 22日, 開催場所 : 金沢市文化ホール 3F

Verifying Specifications with Proof Scores

FUTATSUGI, Kokichi
二木 厚吉

JAIST
Japan Advanced Institute of Science and Technology
Japan

**(this talk is based on our research results
with many persons' contributions)**

I am going to talk about...

- **Our perception of current situation of formal methods**
- **Introducing Proof Score Approaches and its realization in CafeOBJ**
 - ◆ **how to write formal specifications and verify properties of them with proof scores in CafeOBJ (hopefully with simple demonstration)**
- **What kinds of formal models are used for writing formal specifications/proof-scores in CafeOBJ**
- **Current achievements of the proof score approach**

Application areas of formal methods (FM)

1. Analysis and verification of developed program codes (**post-coding**)
 - model checking has brought many successes in code verification but ...
2. Analysis and verification of requirements, specifications, designs before coding (**pre-coding**) or without coding/programming

Successful application of formal methods to the area of requirements, specifications, designs (pre-coding**) can bring drastic effects for system developments, but it is not well exploited and/or practiced yet**

Difficulties in req., spec, design area

- **High level req., spec., design are inherently partial and evolutionary**
- **Usually there is no established formal (mathematical) model for the problem**
- **It is not easy to be convinced that some important property holds for req., spec., design**

Interactive developments with analyses/verifications are inevitable!

Our perception of the current situation of FM

- Verification with formal specifications still have a potential to improve the practices in upstream (**pre-coding**) of software production processes
- Model checking has brought a big success but still has limitations
 - ◆ It is basically “model checking” for program codes
 - initially for **post-coding**; applied at designs/specs later
 - ◆ Infinite state to finite state transformation can be unnatural and difficult
- Established (interactive) theorem provers are not necessary well accepted to software engineers
 - ◆ especially in requirement/spec/design (**pre-coding**) phase

Our approach

- **Reasonable blend of user and machine capabilities, intuition and rigor, high-level planning and tedious formal calculation**
 - ◆ **fully automated proofs are not necessary**
good for human beings to perceive logical structures of real systems



Proof Score Approach

Proof Score Approach

- **Requirement/specification engineers are expected to construct proof scores together with formal specifications**
- **proof scores are instructions such that when executed (or "played") and everything evaluates as expected, then the desired property is convinced to be hold (or proved)**

Specifications and Proof Scores in CafeOBJ

- **Specifications are only algebraic equational specifications**
- **Proof score is a sequence of reduction (simplification) commands for reducing expressions (usually boolean) to its normal form in some situations**
 - ◆ **situations: a set of equations (axioms) with some bindings (a set of name->object relationships)**
 - ◆ **proof score also contains CafeOBJ codes which build an appropriate situation in which expressions are reduced**

A simple example of proof score in CafeOBJ

**The definitions of two factorial functions
and the proof scores for verifying that the
two can compute the same function using
induction**

[Demonstration]

Introducing CafeOBJ

- **CafeOBJ is an algebraic formal specification language**
- **CafeOBJ is a formal language for writing formal models and reasoning about them with rewritings/reductions (ACIZ-rewritings)**
- **CafeOBJ is a successor of OBJ and developed by an international team headed by KF for last 10-15 years**

Related ongoing Language Development Projects

- **Maude** Language of SRI/UIUC is another project for following up the OBJ language
- **CASL** language of European researchers is an attempt of developing a common algebraic specification language
 - ◆ Two volumes of LNCS are already published

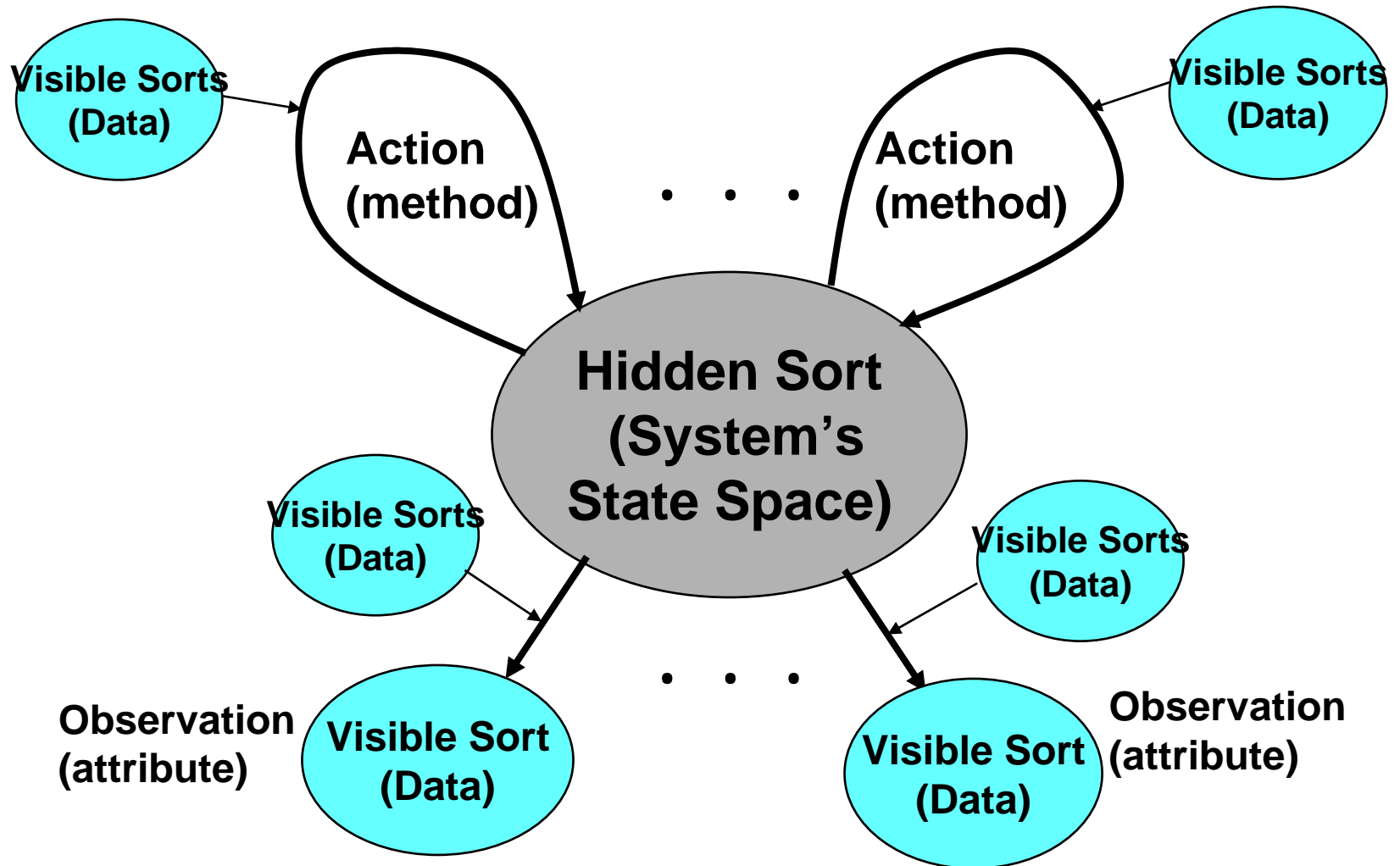
Two kinds of formal models in CafeOBJ

- **Abstract data types** with tight semantics
 - Initial algebra semantics
 - Induction based reasoning
- **Abstract machines (abstract process types)** with loose semantics
 - Coherent hidden algebra semantics
 - Co-induction based reasoning



**Can provide unified specification style
both for static and dynamic systems**

OTS/CafeOBJ Behavioral/Observational Model



OTS is naturally used to model distributed concurrent systems in CafeOBJ

- **Typed data for specifying a system are represented as **visible sorts****
- **The state space of a system is represented by **a hidden sort****

*** Behavioral/Observational equivalence need not (or can not) appear in OTS by definition**

An simple example of OTS

A Bank Account Example

-- a most simplest example of OTS

[Demonstration]

Prerequisites for proof score writing in CafeOBJ (1)

- **Algebraic modeling:**
 - development of algebraic specifications**
 - ◆ **defining signature for a real problem**
 - ◆ **expressing the problem in equations**
 - **more exactly, if you want to prove some property of the spec, expressing the problem in reduction rules**

Prerequisites for proof score writing in CafeOBJ (2)

- **Equational logic, rewriting, and propositional calculus with complete rewriting calculus**
 - ◆ **equational reasoning**
 - **equivalence relation, equational calculus, ...**
 - ◆ **reduction/rewriting**
 - **termination, confluence, sufficiently completeness**
 - ◆ **propositional calculus with “xor” normal forms which has the complete rewriting calculus**

Prerequisites for proof score writing in CafeOBJ (3)

- **Proof by induction with case analyses and lemma discoveries**
 - ◆ **case splitting using key predicates in specifications**
 - ◆ **discovery of lemmas**
 - ◆ **decomposition of a goal predicate into an appropriate conjunctive form**

**These are the most difficult parts of
proof score writing**

Equational proof by reduction/rewriting

Why do we care about

equational reasoning by reduction ?

- It is simple and powerful and a good light weighted formal reasoning method
 - easy to understand and can be more acceptable for software engineers
- It supports transparent relation between specs and reasoning by reduction (**good traceability**)

Traceability in proof score approach with CafeOBJ

- **All reductions are done exactly using equations in specifications**
 - ◆ **this make it easy to detect necessary changes in specs for letting something happen (or not happen)**
- **Usually reductions are sufficiently fast, and encourage prompt interactions between user and system**

This is a quit unique feature of the proof score approach with CafeOBJ comparing to other verification method which often involves several formalisms/logics and translations between them

Current Achievements of OTS/CafeOBJ proof score approach

OTS/CafeOBJ approach has been applied to the following problems and found usable:

- **Some classical mutual exclusion algorithms**
- **Some real time algorithms**
e.g. Fischer's mutual exclusion protocol
- **Authentication protocol**
e.g. NSL, Otway-Rees, STS protocols
- **Practical sized e-commerce protocol of SET**
(some of proof score exceeds 60,000 lines;
specification is about 2,000 lines,
20-30 minutes for reduction of the proof score)
- **UML semantics (class diagram + OCL-assertions)**
- **Formal Fault Tree Analyses**
- **Secure workflow models**

Future Plan

- **Develop proof score writing environment**
 - ◆ **Standard platforms for programming environment can be naturally used (e.g. Eclipse Env.)**
Write specs and proof-scores as writing programs!
- **Automate case analysis and lemma discovery**
 - ◆ **Automation of inductive proof (Crème)**
 - **NSLPK and STS protocol verification is already done automatically**
 - ◆ **Incorporation of model checking technologies into proof score approach**
 - **Especially for finding counter examples**
- **Apply to the new areas**
 - ◆ **business and/or social system specs and analyses/verifications**
 - **Secure workflows/processes**
 - **E-commerce domain models**
 - ◆ **System Biology**

CafeOBJ Home Page

- **CafeOBJ official home page:**
<http://www.idl.jaist.ac.jp/cafeobj/>