

A Brief Overview of CafeOBJ/ProofScore and Formal Methods

FUTATSUGI, Kokichi
二木 厚吉

JAIST
(Japan Advanced Institute of Science and Technology)

Application areas of formal methods (FM)

1. Analysis and verification of developed program codes (**post-coding**)
2. Analysis and verification of (models/specs of) domains, requirements, and designs before/without coding (**pre-coding or without coding**)

Successful application of formal methods to the area of (models/specifications of) domains, requirements, designs can bring drastic good effects for systems developments, but it is not well exploited and/or practiced yet.

specification = description of model

The current situation of FM

- Verification with formal specifications still have a potential to improve the practices in upstream (**pre-coding**) of systems development processes
- Model checking has brought a big success but still has limitations
 - It is basically “model checking” for program codes
 - Still mainly for **post-coding**
 - Infinite state to finite state transformation can be unnatural and difficult
- Established interactive theorem provers (Isabelle/HOL, Coq, PVS, etc.) are not necessary well accepted to software/systems engineers
 - especially in upstream (**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 problems/systems
 - **interactive understanding/description of real problem domains/requirements/designs is necessary**



Proof Score Approach

Proof Score Approach

- **Domain/requirement/design 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 hold**
 - **Proof by construction/development**
 - **Proof by reduction/computation/rewriting**

Development of proof scores in CafeOBJ

- ◆ **Many simple proof scores are written in OBJ language from 1980's; some of them are not trivial**
- ◆ **From around 1997 CafeOBJ researchers at JAIST use proof scores seriously for verifying specifications for various examples**
 - **From static to dynamic/reactive system**
 - **From ad hoc to more systematic proof scores**
 - **Introduction of OTS (Observational Transition System) was a most important step**
 - **Generate/Check Method is a most recent development**

Some achievements of CafeOBJ/OTS proof score approach

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

- **Some classical mutual exclusion algorithms**
- **Some real time algorithms**
e.g. Fischer's mutual exclusion protocol
- **Railway signaling systems**
- **Authentication protocol**
e.g. NSLPK, Otway-Rees, STS protocols
- **Practical sized e-commerce protocol of SET**
- **UML semantics (class diagram + OCL-assertions)**
- **Formal Fault Tree Analysis**
- **Secure workflow models, internal control**
- **International standard for automotive software**
- **Protocols for Cloud computing**

A little bit of CafeOBJ history

- **KF thought of the basic ideas of CafeOBJ after he participated OBJ project at SRI in 1983-1984, and several design and implementation attempts were done during 1985-1995**
- **The CafeOBJ development project is fully supported by IPA/MITI of Japanese Government from 1996.4 to 1998.3**
 - **Six Japanese Companies, Five Japanese Universities, Three Foreign Research Group participate CAFÉ project**
 - **A book entitled “CafeOBJ Report” was published in 1998 which defines the syntax and semantics of the CafeOBJ language**
- **Sufficiently reliable and usable CafeOBJ system was available at around the beginning of 1999.**
- **KAKEN-KIBAN(S) project on CafeOBJ in 2011-2016**
- **Several groups including KF’s group at JAIST are using CafeOBJ for developing formal methods for various application areas and/or for education of FM**