Z and Vers

Bart Massey
Portland State University
May 8, 2001
What is Z?

• Formal logic and set theory
• Finite and integer sorts
• Models world
• Specifies constraints
Modeling and Analysis

Real System \( \xrightarrow{\text{behavior}} \) Real Effects
\[ \downarrow \text{modeling} \]
Symbol System \( \xrightarrow{\text{abstract behavior}} \) Symbolic Effects

Bart Massey
Portland State University
May 8, 2001
Z Paragraphs

Z description is natural-language +

• Entities
• Axioms
• Schemata
Z Definitions

Set existence

[FISH]

\[ COIN ::= \text{penny} \mid \text{nickle} \mid \text{dime} \mid \text{quarter} \]

Functions and constraints

\[ \text{least\_change} : 0 .. 99 \rightarrow \text{bag COIN} \]
\[ \forall n : 0 .. 99 \bullet \text{elems(least\_change} n) \leq 9 \]
Z Schema

Like records: indicate state change

\[
\text{VendingMachine} \\
\text{change : bag COIN} \\
\text{elems(change) \leq 1000}
\]
Change in $Z$

Schema can show state change

\[ Sell \]
\[ \Delta VendingMachine \]
\[ payment?, price? : \mathbb{N} \]
\[ change' = change \text{ less} \]
\[ \text{least}\_change(payment? - price?) \]
\[ payment? \geq price? \]
Requirements Inspection

1. Clean up numbering
2. Strengthen vague requirements
3. Add missing requirements
4. Use clear, consistent terminology
Why Z For Vers?

- Consistency
- Completeness
- Precision
- Accuracy
- Terseness
- Verifiability
The Vers Environment

\[ [USER, DIR, FILE,FILENAME] \]

\[ fs : DIR \times FILENAME \rightarrow FILE \]

\[ cwd : USER \rightarrow DIR \]
Internals

\[ \text{VERSION} = = \mathbb{N}_1 \]
\[ \text{LOCK\_STATE} ::= \text{locked\_by}\langle\text{USER}\rangle | \text{unlocked} \]

+ Repository
The Repository

`Delta`  
version : VERSION  
file : FILE

`HISTORY == seq Delta`

`Record`  
lock_state : LOCK_STATE  
history : HISTORY

`Repository`  
repository : FILENAME \(\rightarrow\) Record
Most Recent Version

\[
mrv : \text{Record} \rightarrow \text{VERSION} \\
\forall r : \text{Record} \bullet \\
\quad \text{mr}r = (r.\text{history} \ 1).\text{version}
\]
Creating A New Record

\[\text{new\_record} : \text{USER} \times \text{FILE} \rightarrow \text{Record}\]

\[\forall u : \text{USER}; \ f : \text{FILE}; \ r : \text{Record}; \ d : \text{Delta}\]

\[\bullet\]

\[r = \text{new\_record}(u, f) \land\]

\[r.\text{lock\_state} = \text{unlocked} \land\]

\[r.\text{history} = \langle d \rangle \land\]

\[d.\text{version} = 1 \land\]

\[d.\text{file} = f\]
Changing The Repository

InitRepository

Repository

repository' = ∅

New

ΔRepository

filename? : FILENAME

user? : USER

repository' = repository ∪

{filename? ↦
  new_record(user?,
  fs(cwd user?, filename?))}

filename? ∉ dom(repository)
Conclusions

• Important to inspect Vers reqs
• Z formalization helps with reqs
• Reqs are hard