Introduction

- Architects draw detailed plans before a brick is laid or a nail is hammered. Programmers and software engineers don't. Can the be why houses seldom collapse and programs often crash?
- Blueprints help architects ensure that what they are planning to build will work. "Working" means more than not collapsing; it means serving the required purpose. Architects and their clients use blueprints to understand what they are going to build before they start building it.
- But few programmers write even a rough sketch of what their programs will do before they start coding.
- Specifications: To designers of complex systems, the need for formal specifications should be as obvious as the need for blueprints of a skyscraper. But few software developers write specifications because they have little time to learn how on the job, and they are unlikely to have learned in school. Some graduate schools teach courses on specification languages, but few teach how to use specification in practice. It's hard to draw blueprints for a skyscraper without ever having drawn one for a toolshed.

[Leslie Lamport, Turing Award Winner, 2013]

Specifications (and formal methods) used to be relegated to safety critical systems like nuclear power, avionics and medical devices. Increasingly, a variety of industrial strength formal methods (e.g. TLA⁺ [4], Event-B [1], and many others) are now being used by Microsoft, Amazon, Facebook and Dropbox.

Significance & Contributions

Unit-B [3] is a new framework for specifying and modelling systems that must satisfy both safety and liveness properties. Compared to Event-B, Unit-B brings in record types and complete welldefinedness. In comparison to TLA⁺, Unit-B adds type checking, well-definedness checking and quantification over infinite sets.

Unit-B Web makes the Literate Unit-B prover available on the web. Unit-B Web leverages the automated predicate prover to two purposes:

- Teaching: can be used in classroom for demonstrations, or in evaluation in the form of online quizzes.
- **Online Proof Environment**, making specifications more accessible to casual users. It also serves as a "proof of concept" for a web IDE for the full modelling capabilities of Unit-B.

Unit-B Web's technology stack: • Syntax: based on $PT_EX • Web$: JavaScript, JSON, Yesod / Haskell • Prover: Haskell, Z3

The Magic of Specifications and Type Systems

elow are two screenshots of the Unit-B Web tool, showcasing its type checking and ell-definedness checking capabilities. Unit-B Web rover Examples +	Some formulas, such checking helps us iden over the proof of me this as an error; Unit TLA ⁺ is an untyped as {3, {7}}. In a s homogeneity rules on system such as Unit- subtyping.
Cover Examples ↓ Examples ↓ ories ts ts stants • goal:1:17: type error: arguments of 'union' do not match its signature: signature: [\set [_a]] -> \set [_a] left summents (mk-set (mk-set 3)) type \set [\set [\Int]]	over the proof of me this as an error; Unit TLA ⁺ is an untyped as {3, {7}}. In a s homogeneity rules of system such as Unit- subtyping.
bries ts stants + Examples ↓ Examples ↓ Examples ↓ Itel argument: Itel argument: Itel argument: Itel argument: Itel argument: Itel argument: Itel argument:	as {3, {7}}. In a s homogeneity rules ou system such as Unit- subtyping.
stants stants + stants - stants - stants - - - - - - - - - - - - - - - - <	Subtyping.
cl1 x : \set[\set[\Int]] - right argument: (mk-set x) type \set [\set [\set [\Int]]]	Further, type variable e.g. using the same of numbers.
cl2 i:\Int –	W
Imptions + $sets$ Im1 $x = \backslash \{ \backslash \{7 \rangle \} \backslash \}$ - Im1 $x = \backslash \{ \neg \} \backslash \}$ - Im1 $x = \backslash \{ \neg \} \backslash \}$ - Im1 $x = \backslash \{ \neg \} \backslash \}$ - Im1 $x = \backslash \{ \neg \} \backslash \}$ - Im1 $x = \backslash \{ \neg \} \backslash \}$ - Im1 $x = \backslash \{ \neg \} \backslash \}$ - Im1 $x = \backslash \{ \neg \} \backslash \}$ Im1 $x = \backslash \{ \neg \} \backslash \}$	WD checking [2] catc cannot catch, such a
$\begin{array}{c} (3 \ i, x \ i \ i, j \ i \ j \ i \ j \ i \ j \ i \ j \ j \$	Unit-B's WD-calculu Let us consider the f <i>C</i> , and <i>D</i> (which we
Unit-B Web Snapshot	The following calcul formula is rejected),
Theories Ill-defined	$A \wedge B \wedge (C \vee$
Functions, Sets Image: Sets Constants Image: Sets decl1 f: \Int \pfun \Int Image: Sets f: Sets f: Sets f: Sets f: Sets	$= \{associativity\} \\ A \land (C \lor D) \land \\ = \{distributivity\} \}$
decl2 x: \Int $-$	$((A \land C) \lor (A))$
asm1f \in \qset{x}{x \le 5}{x} \tfun \lnt $ f \in \{x: x \le 5: x\} \to \mathbb{Z}$ (asm1) $x \le 6$ (asm2)	
asm2 x \le 6 $- f.x \leq 6$	
Goal f.x \le 6	[1] Jean-Raymond Abrial. <i>Modeling in Event-B - Syste</i> Cambridge University Press,
	[2] Ádám Darvas, Farhad Mehta Efficient well-definedness cha In Automated Reasoning, 4t August 12-15, 2008, Procee
Clear Clear	
Figure 2: An ill-defined predicate — x is not in the domain of f	[3] Simon Hudon, Thai Son Ho The Unit-B method: refinen <i>Software & Systems Modeli</i>

	Unit-B vved Snapsnot	
Below are two screenshots c well-definedness checking cap	of the Unit-B Web tool, showcasing its type checking pabilities.	and Some formulas, such checking helps us ider
	Unit-B Web	this as an error; Unit TLA ⁺ is an untyped
Prover	Exa	mples ↓ as {3, {7}}. In a s homogeneity rules ou system such as Unit- subtyping
heories	9 goal:1:17:	Further type variable
Sets v	<pre>type error: arguments of 'union' do not match its signature: signature: [\set [_a],\set [_a]] -> \set [_a]</pre>	e.g. using the same
	<pre>type \set [\set [\Int]] right argument: (mk-set 3))</pre>	of numbers.
	<pre>type \set [\set [\Int]]]</pre>	
	using	We
asm1 $x = \setminus \{ \setminus \{7\} \setminus \}$	sets constants	WD checking [2] catc
ioal	$\begin{array}{c} x \subseteq \mathbb{P} .\mathbb{Z} \\ i \in \mathbb{Z} \end{array}$	cannot catch, such a
\{ \{ 3 \}, x \} \subseteq \{ \{ 3, 7 \} \} \2\land \gforall{i}{i \le i}{i \ge i-2}		Unit-B's WD-calculu
	$x = \{\{7\}\}$ (asm1)	
Clear Figure 1: A t	$\vdash \{\{3\}, x\} \subseteq \{\{3, 7\}\} \land \langle \forall j : j \le i : i \ge j - 2 \rangle$ type error — x is expected to be a set of numbers	Let us consider the form C , and D (which we
Clear Figure 1: A t	$\vdash \{\{3\}, x\} \subseteq \{\{3, 7\}\} \land \langle \forall j : j \leq i : i \geq j - 2 \rangle$ type error — x is expected to be a set of numbers Unit-B Web Snapshot	Let us consider the following calculation formula is rejected.
Clear Figure 1: A t	$\vdash \{\{3\}, x\} \subseteq \{\{3, 7\}\} \land \langle \forall j : j \leq i : i \geq j - 2 \rangle$ type error — x is expected to be a set of numbers Unit-B Web Snapshot	Let us consider the formula is rejected),
Clear Figure 1: A t Theories Functions, Sets	F {{3}, x} ⊆ {{3,7}} ∧ $\langle \forall j: j \le i: i \ge j - 2 \rangle$ type error — x is expected to be a set of numbers Unit-B Web Snapshot III-defined	The following calculation formula is rejected), $A \wedge B \wedge (C \vee C)$
Clear Figure 1: A t Figure s Functions, Sets	$[{3}, x] \subseteq \{{3, 7}\} \land \langle \forall j : j \leq i : i \geq j - 2 \rangle$ type error — x is expected to be a set of numbers Unit-B Web Snapshot Ill-defined using	Let us consider the formula is rejected), $A \wedge B \wedge (C \vee A = \{associativity\}$
Clear Figure 1: A t Figure 1: A t Functions, Sets Constants + decl1 f: \lnt \pfun \lnt	$[\{3\}, x\} \subseteq \{\{3, 7\}\} \land \langle \forall j : j \le i : i \ge j - 2 \rangle$ type error — x is expected to be a set of numbers Unit-B Web Snapshot Ill-defined Using functions, sets constants	Let us consider the formula is rejected), $A \wedge B \wedge (C \vee A \wedge B \wedge (C \vee A) \wedge (C \vee D) \wedge $
Clear Figure 1: A t Functions, Sets Constants + decl1 f: \Int \pfun \Int decl2 x: \Int	$[\{3\}, x\} \subseteq \{\{3, 7\}\} \land \langle \forall j : j \leq i : i \geq j - 2 \rangle$ type error — x is expected to be a set of numbers Unit-B Web Snapshot • Ill-defined • Ill-defined • Ill-defined • Ill-defined • Ill-defined	Let us consider the formula is rejected), $A \wedge B \wedge (C \vee A) = \{associativity\} \\ A \wedge (C \vee D) \wedge A = \{distributivity\} \\ ((A \wedge C) \vee (A) \} \}$
Clear Figure 1: A t Figure 1: A t Theories Functions, Sets Constants + decl1 f: \lnt \pfun \lnt decl2 x: \lnt Assumptions +	$ F = \{\{3\}, x\} \subseteq \{\{3, 7\}\} \land \langle \forall j : j \le i : i \ge j - 2 \rangle $ type error — x is expected to be a set of numbers Unit-B Web Snapshot Unit-B Ueb Snapshot Unit-B Ueb Snapshot Unit-B Ueb Snapshot III-defined IIII-defined III-defined IIII-defined IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	Let us consider the formula is rejected), The following calculation formula is rejected), $A \wedge B \wedge (C \lor A) = \{associativity\}$ $A \wedge (C \lor D) \wedge A$ $= \{distributivity\}$ $((A \wedge C) \lor (A))$
C Prove Clear Figure 1: A t Figure 1: A t Functions, Sets Constants + decl1 f: \Int \pfun \Int decl2 x: \Int Assumptions + asm1 f \in \qset{x}{x \le 5}{x} \tfun \Int	$\{\{3\}, x\} \subseteq \{\{3, 7\}\} \land \langle \forall j : j \leq i : i \geq j - 2 \rangle$ type error — x is expected to be a set of numbers Unit-B Web Snapshot • Ill-defined • Ill-defined • Ill-defined • $\begin{bmatrix} using \\ functions, sets \\ constants \\ f \in \mathbb{Z} \leftrightarrow \mathbb{Z} \\ x \in \mathbb{Z} \end{bmatrix}$ • $f \in \{x : x \leq 5 : x\} \rightarrow \mathbb{Z} \text{ (asm1)} \\ (asm2) \end{bmatrix}$	Let us consider the formula is rejected), $A \wedge B \wedge (C \vee A) = \{associativity\} \\ A \wedge (C \vee D) \wedge A \\ = \{distributivity\} \\ ((A \wedge C) \vee (A)) \}$
Clear Figure 1: A t Figure 1: A t functions, Sets Constants + decl1 f: \Int \pfun \Int decl2 x: \Int Assumptions + asm1 f \in \qset{x}{x \le 5}{x} \tfun \Int asm2 x \le 6	$\{3\}, x\} \subseteq \{\{3, 7\}\} \land \langle \forall j : j \le i : i \ge j - 2 \rangle$ type error — x is expected to be a set of numbers $Unit-B \text{ Web Snapshot}$ $using \qquad \qquad$	Let us consider the formula is rejected), $A \wedge B \wedge (C \vee A) = \{associativity\} \\ A \wedge (C \vee D) \wedge A \\ = \{distributivity\} \\ ((A \wedge C) \vee (A)) \}$
Clear Figure 1: A t Figure 1: A t Functions, Sets Constants + decl1 f: \lnt \pfun \lnt decl2 x: \lnt Assumptions + asm1 f \ln \qset{x}{x \le 5}{x} \tfun \lnt asm2 x \le 6 Goal f.x \le 6	$ \{3\}, x\} \subseteq \{\{3,7\}\} \land \langle \forall j : j \leq i : i \geq j - 2 \rangle $ type error — x is expected to be a set of numbers $ \textbf{Unit-B Web Snapshot} $ $ \textbf{Oill-defined} $ $ \textbf{Using} \\ functions, sets \\ constants \\ f \in \mathbb{Z} \leftrightarrow \mathbb{Z} \\ x \in \mathbb{Z} \\ \textbf{Oild} \\ Guide for a set for a $	Let us consider the formula is rejected), The following calculated formula is rejected), $A \wedge B \wedge (C \vee A) = \{associativity\} \\ A \wedge (C \vee D) \wedge A \\ = \{distributivity\} \\ ((A \wedge C) \vee (A)) \\ = [1] Jean-Raymond Abrial. Modeling in Event-B - System Cambridge University Press, A \\ = System $
C Prove Clear Figure 1: A t Figure 1: A t Functions, Sets Constants + decl1 f: \lnt \pfun \lnt decl2 x: \lnt Assumptions + asm1 f \ln \qset{x}{x \le 5}{x} \tfun \lnt asm2 x \le 6 Coal f.x \le 6	$ \left\{ \{3\}, x\} \subseteq \{\{3, 7\}\} \land \langle \forall j: j \leq i: i \geq j - 2 \rangle \right\} $ type error — x is expected to be a set of numbers $ \textbf{Unit-B Web Snapshot} $ $ \textbf{O Ill-defined} $ $ \textbf{using} \\ functions, sets \\ constants \\ f \in \mathbb{Z} \Rightarrow \mathbb{Z} \\ x \in \mathbb{Z} \\ \textbf{or stants} \\ f \in \{x: x \leq 5: x\} \rightarrow \mathbb{Z} \text{ (asm1)} \\ x \leq 6 \\ \textbf{or stants} \\ f.x \leq 6 \\ \textbf{or stants} \\ f.x \leq 6 \\ \textbf{or stants} \\ \textbf{or stants} \\ f.x \leq 6 \\ \textbf{or stants} \\ $	Let us consider the f C, and D (which we The following calculation formula is rejected), $A \wedge B \wedge (C \lor$ $= \{associativity\}$ $A \wedge (C \lor D) / (A \land C) \lor (A)$ $= \{distributivity\}$ $((A \wedge C) \lor (A)$ [1] Jean-Raymond Abrial. <i>Modeling in Event-B - Systee</i> Cambridge University Press, [2] Ádám Darvas, Farhad Mehtate Efficient well-definedness cher
Clear Figure 1: A t Figure 1: A t Functions, Sets Constants + decl1 f: \Int \pfun \Int decl2 x: \Int Assumptions + asm1 f \in \qset{x}{x \le 5}{x} \tfun \Int asm2 x \le 6 Goal f.x \le 6 2 Prove Clear	$ \{\{3\}, x\} \subseteq \{\{3, 7\}\} \land \langle \forall j: j \le i: i \ge j - 2 \rangle $ type error — x is expected to be a set of numbers $ \textbf{Unit-B Web Snapshot} $ $ \textbf{Ill-defined} $ $ \textbf{using} \\ functions, sets \\ constants \\ f \in \mathbb{Z} \leftrightarrow \mathbb{Z} \\ constants \\ f \in \mathbb{Z} \leftrightarrow \mathbb{Z} \\ f \in \{x: x \le 5: x\} \rightarrow \mathbb{Z} \text{ (asm1)} \\ x \le 6 \\ f.x \le 6 \end{cases} $ $ \textbf{http:/red.cse.yorku.ca:30} $	Let us consider the f C, and D (which we The following calcul- formula is rejected), $A \land B \land (C \lor$ $= \{associativity\}$ $A \land (C \lor D) \checkmark$ $= \{distributivity\}$ $((A \land C) \lor (A)$ [1] Jean-Raymond Abrial. <i>Modeling in Event-B - Syste</i> Cambridge University Press, [2] Ádám Darvas, Farhad Mehta Efficient well-definedness che In Automated Reasoning, 4t August 12-15, 2008, Proceed
Clear Figure 1: A t Figure 2: An	$\{\{3\}, x\} \subseteq \{\{3, 7\}\} \land \langle \forall j: j \leq i: i \geq j - 2 \rangle$ type error — x is expected to be a set of numbers Unit-B Web Snapshot • • • • • • • • • • • • • • • • • • •	Let us consider the f C, and D (which we The following calcul- formula is rejected), $A \land B \land (C \lor$ $= \{associativity\}$ $A \land (C \lor D) \checkmark$ $= \{distributivity\}$ $((A \land C) \lor (A)$ [1] Jean-Raymond Abrial. <i>Modeling in Event-B - Syste</i> Cambridge University Press, [2] Ádám Darvas, Farhad Mehta Efficient well-definedness che In Automated Reasoning, 4t August 12-15, 2008, Proceeder [3] Simon Hudon, Thai Son Hoa The Unit-B method: refiner Software & Systems Modelin

Amin Bandali, Simon Hudon, Jonathan Ostroff Software Engineering Lab, EECS





Type Checking

as $\{x\} + 3 \leq 7$, are not meaningful. Type ntify and fix them instead of laboring needlessly eaningless formulas. TLA⁺ does not recognize t-B does (see Fig. 1).

logic which allows expressive formulas such simple type system such as that of Event-B, ut such a formula. The challenge in a typed -B is to allow such formulas. We do this using

es in Unit-B allow for polymorphic definitions, functions on sets of numbers and sets of sets

ell-definedness Checking

ches meaningless formulas that the type checker as division by zero or array out of bounds.

is is complete; while Event-B's is incomplete. following example with four propositions A, B, will specify shortly), where

> $A \Rightarrow WD(B)$ $B \Rightarrow WD(C)$ $B \Rightarrow WD(D)$

ation is not well-defined in Event-B (the red but it is well-defined in Unit-B:

D)	where
	$A: x \in dom.f$
В	$B: f.x \in dom.g$
	$C:g.(f.x)\leq 3$
$\land D)) \land B$	$D:7\leq g.(f.x)$

References

em and Software Engineering. 2010. a, and Arsenii Rudich. ecking. th International Joint Conference, IJCAR 2008, Sydney, Australia, edings, pages 100–115, 2008. ang, and Jonathan S. Ostroff. nent guided by progress concerns. ing, pages 1–26, 2015.

A+ Language and Tools for Hardware and Software Engineers.