UNIVERSITY OF Southampton
School of Electronics and Computer Science

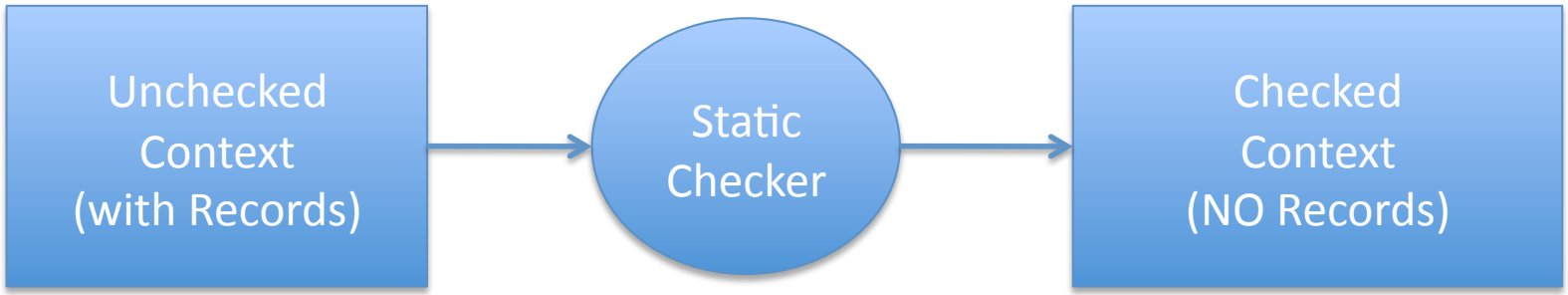
Records

Vitaly Savicks, Colin Snook & Michael Butler

Introduction

- Structured Types
  - Based on Butler & Evans Proposal (FM'06)
- Fields - projection functions
- Subtype Record
- Extended Record
- Feasibility (surjective) *(next release)*
- Closed (injective) *(next release)*
  - update and make functions *(next release)*

Translation



UNIVERSITY OF Southampton
School of Electronics and Computer Science

Translation (carrier record)

```
RECORD DECLARATIONS
  R
  FIELDS
    f1      type        T1
    f2      type        T2
    …
  END
```

unchecked context

checked context

```
SETS          R
CONSTANTS     f1, f2,…
AXIOMS
  R.f1.type1 :        f1 ∈ R ⟶ T1
  R.f2.type1 :        f2 ∈ R ⟶ T2
    …
  R.feasibility :     f1 ⊗ f2 ∈ R ⤖ T1 × T2
```

UNIVERSITY OF Southampton
School of Electronics and Computer Science

Translation (sub record)

```
RECORD DECLARATIONS
  S
  SUPERTYPES
    R
  FIELDS
    f3    type       T3
  END
```

```
CONSTANTS   S, f3
AXIOMS
  S.type1 :          S ⊆ R
  S.f3.type1 :       f3 ∈ S ⟶ T3
  S.feasibility :    f3 ⊗ f1 ⊗ f2 ∈ S ⤖ T3 × T1 × T2
```

Translation (extended record)

RECORD EXTENSIONS
 R
FIELDS
 …
END

nothing!!

Translation (extended record)

```
RECORD EXTENSIONS
 R
FIELDS
    f4      type       T4
END
```

```
CONSTANTS     f4
AXIOMS
  R.f4.type1 :           f4 ∈ R ⟶ T4
  R.feasibility.ext1 :   f4 ⊗ f1 ⊗ f2 ∈ R ⤖ T4 × T1 × T2
```

Using Open Records

```
VARIABLES
  v

INVARIANTS
  inv1   :   v ∈ R
```

```
e1   ≙
STATUS
 ordinary
ANY
 r
WHERE
 g1   :   f1(r) = f1(v)
 g2   :   f2(r) = t2
THEN
 a1   :   v ≔  r
END
```

feasibility axiom means this will not block

Closed records

- Open records
  - + can be developed in refinements, but…
  - – Awkward to update
  - – Cannot be animated
- hence – option to Close records
  - – injective
  - – update and make functions
  - – (cannot close extended records)

Translation Closed Records

```
RECORD DECLARATIONS
  R
    closed
  FIELDS
    f1      type        T1
    f2      type        T2
    …
  END
```

```
SETS            R
CONSTANTS       f1, f2,…
AXIOMS
  R.f1.type1 :        f1 ∈ R ⟶ T1
  R.f2.type1 :        f2 ∈ R ⟶ T2
    …
  R.feasibility :     f1 ⊗ f2 ∈ R ⤖ T1 × T2
```

as open record …

Translation Closed Records

```
CONSTANTS
          make_R,    update_R_f1,   update_R_f2

AXIOMS
  ..
  R.closure :          f1 ⊗ f2 ∈ R ⤖ T1 × T2

  R.make.axm1 :        make_R ∈ T1 × T2 ⤔ R
  R.f1.make.axm2 :     ∀ t1⦂T1,t2⦂T2 · f1(make_R (t1 ↦ t2))=t1
  R.f2.make.axm2 :     ∀ t1⦂T1,t2⦂T2 · f2(make_R (t1 ↦ t2))=t2

  R.f1.update.axm1 :   update_R_f1 ∈ R × T1 ⟶ R
  R.f1.update.axm2 :   ∀r⦂R,t1⦂T1·
                           update_R_f1(r ↦ t1)=make_R(t1 ↦ f2(r))
```

(repeat these 2 update axioms for each field)

Using update

```
e1   ≙
STATUS
 ordinary
BEGIN
 a1   :   v ≔  update_R_f2(v ↦ t2)
END
```

Using make

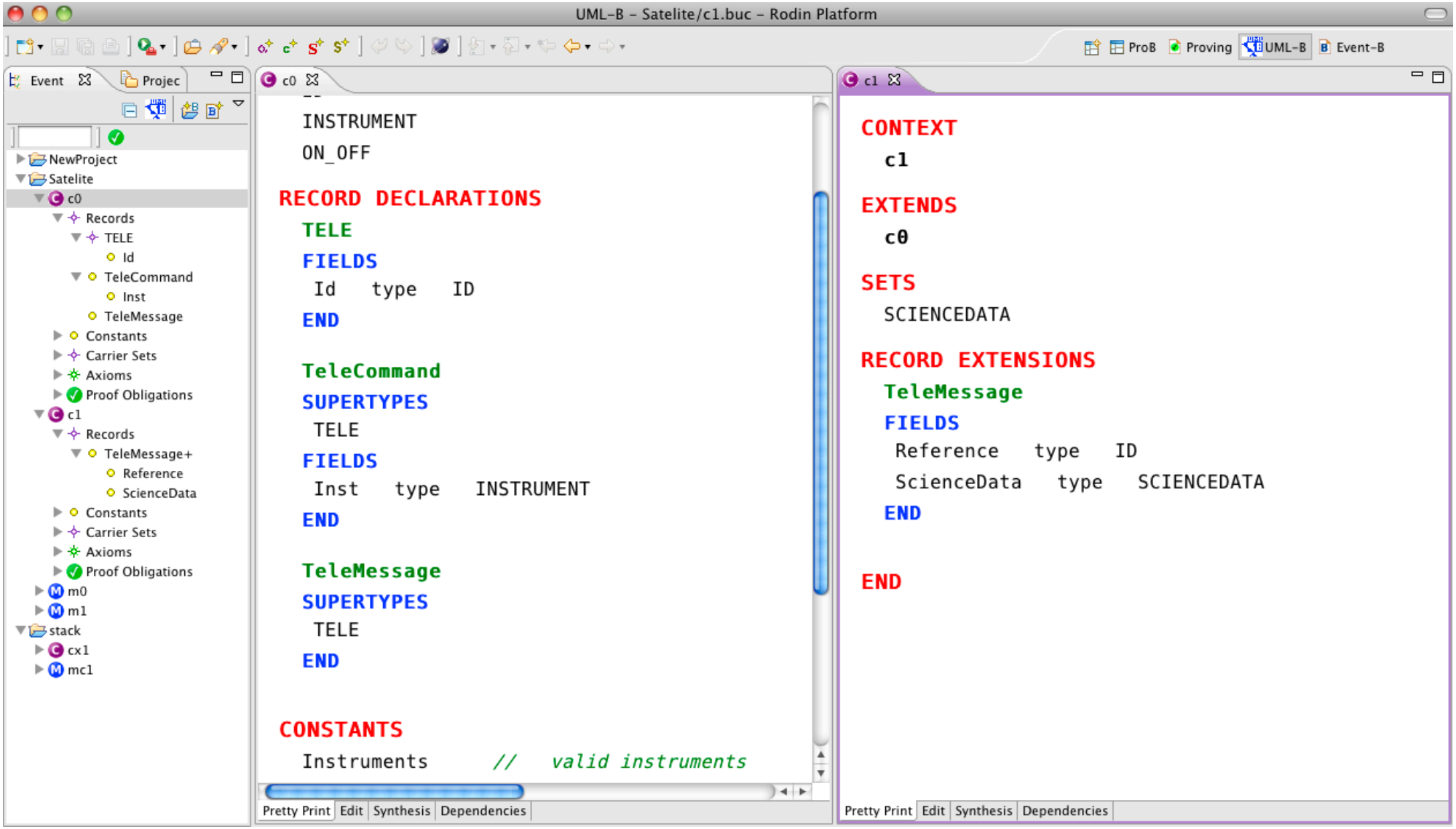
```
e2   ≙
STATUS
 ordinary
BEGIN
 a1   :   v ≔  make_R(t1 ↦ t2)
END
```

Implementation

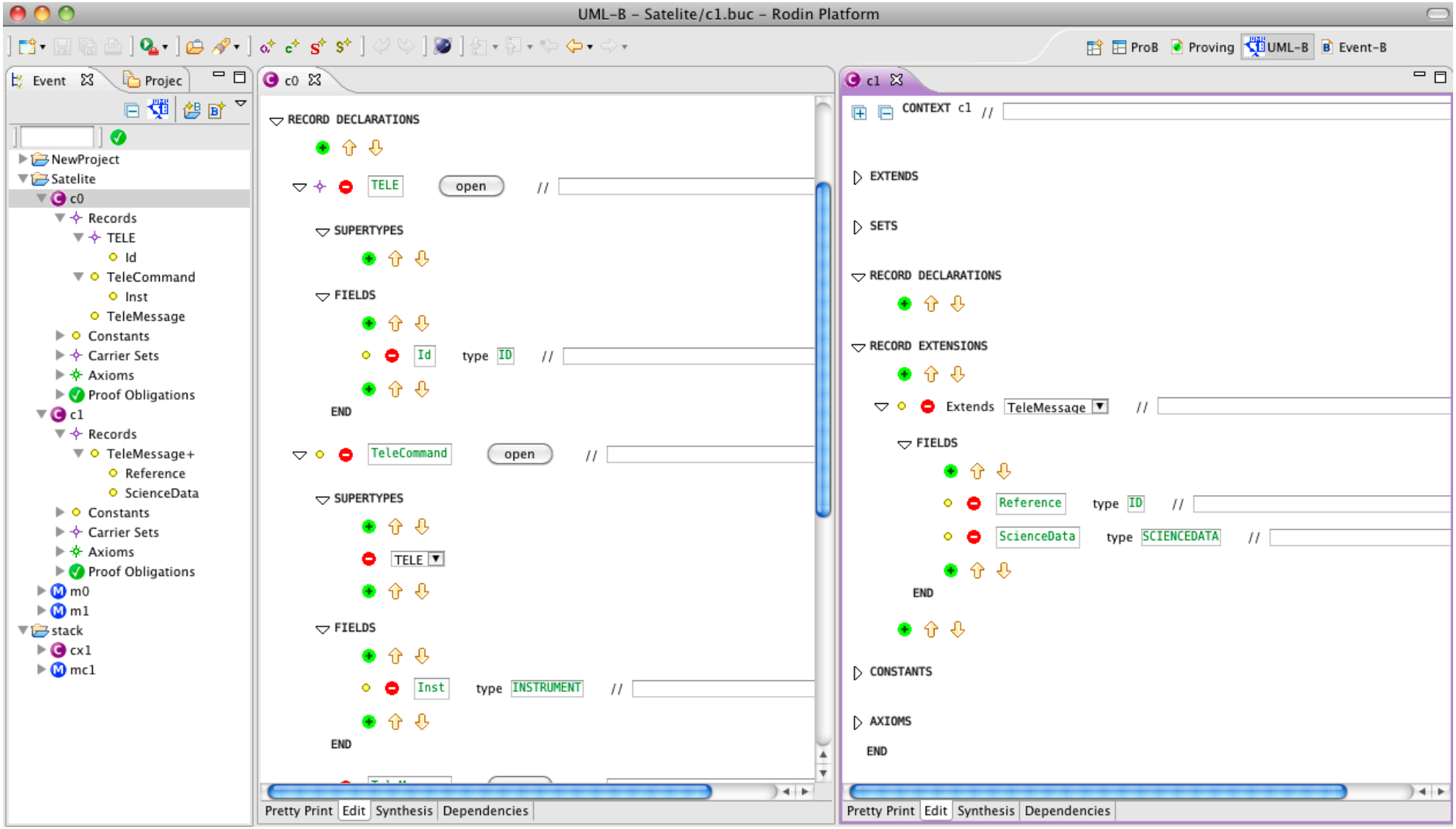
- Extensions to
  - Rodin DB
  - Form based-editor
  - Navigator
  - Static checker
  - Refactoring *(in next release)*
  - Pretty Printer *(in next release)*
  - Event-B EMF *(in next release)*

- Camille – proposals for extensibility

Pretty Print



Form-based Editor



Summary

- Structured (composite) Types
- Use variables to hold instances of records
- Subtype and Extend
- Add new fields to open records (in refinement)
- All field values are available (feasibility)
- Close records to get make and update functions

UNIVERSITY OF Southampton
School of Electronics and Computer Science

Questions ?