SpecEdit

An IDE for TLA+

1. Riwan Cuinat, Ciprian Teodorov and Joël Champeau.
SpecEdit: Projectional Editing for TLA+ Specifications

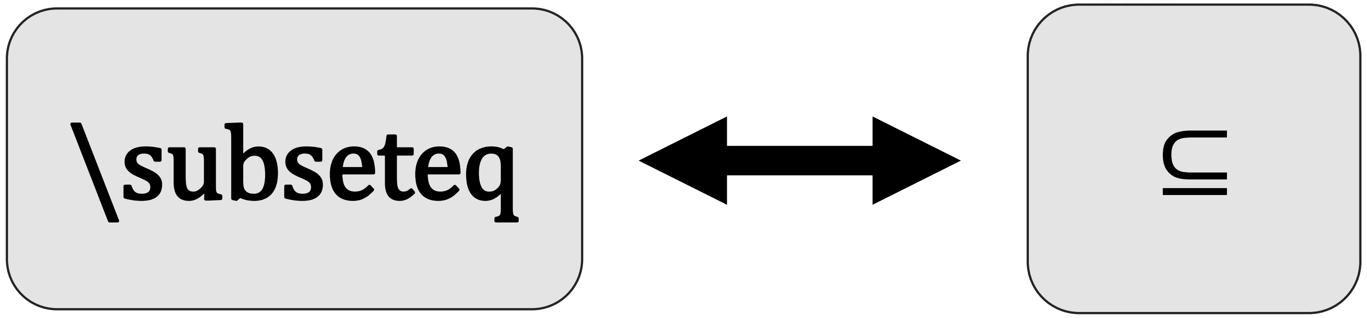
Riwan Cuinat
Ciprian Teodorov
Joël Champeau

31/08/2020

ENSTA BRETAGNE

Overview

One of the good things about TLA+ is that if [...][you don't] understand what a TLA+ construct means, [...][you] can look it up in a math book. Math books don't write math in ASCII, they use standard mathematical symbols.



Introduction

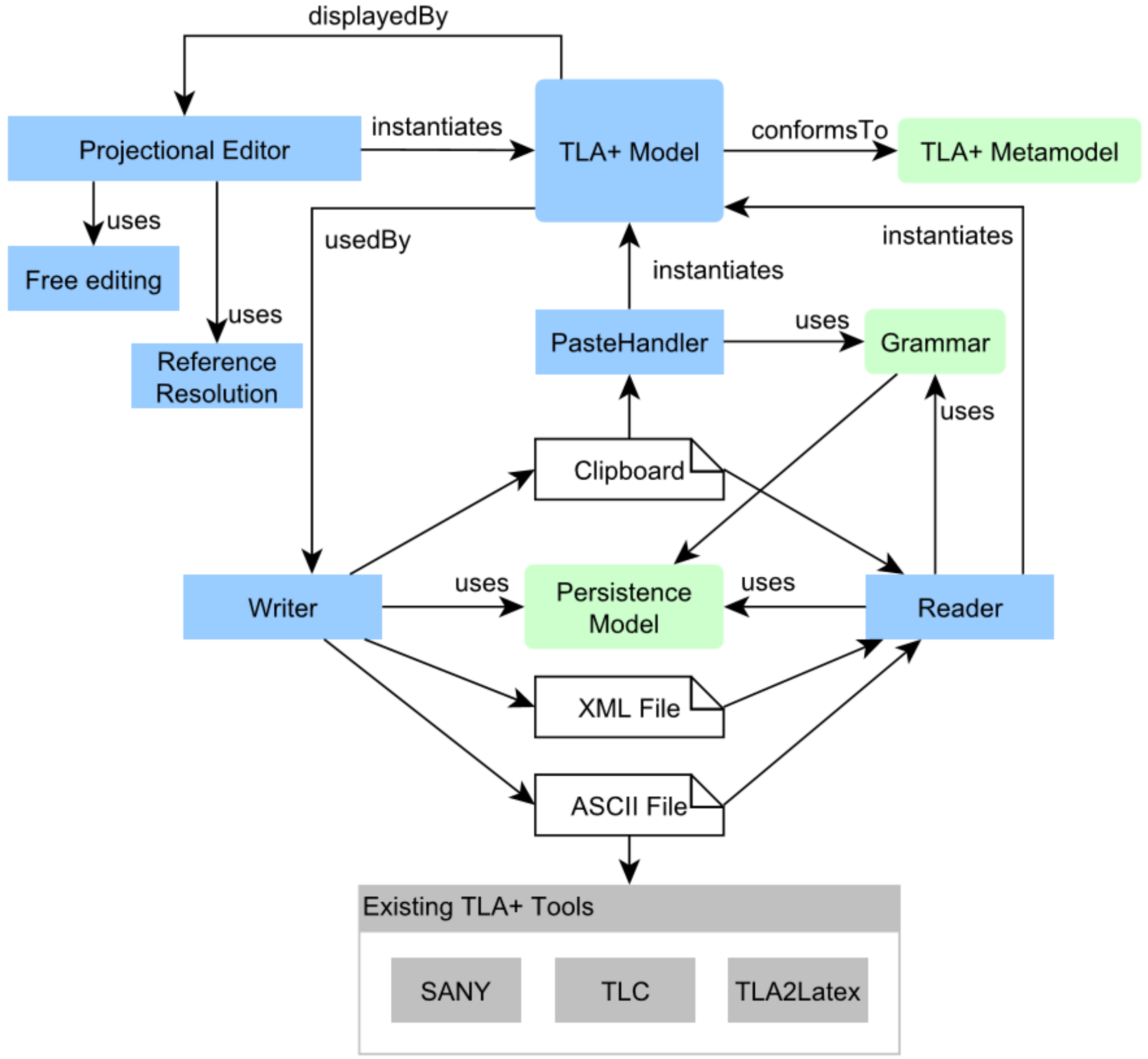
- ❑ Integrated Development Environment (IDE)
- ❑ TLA+ specification language
- ❑ TLA+ Toolbox
- ❑ Syntax duality

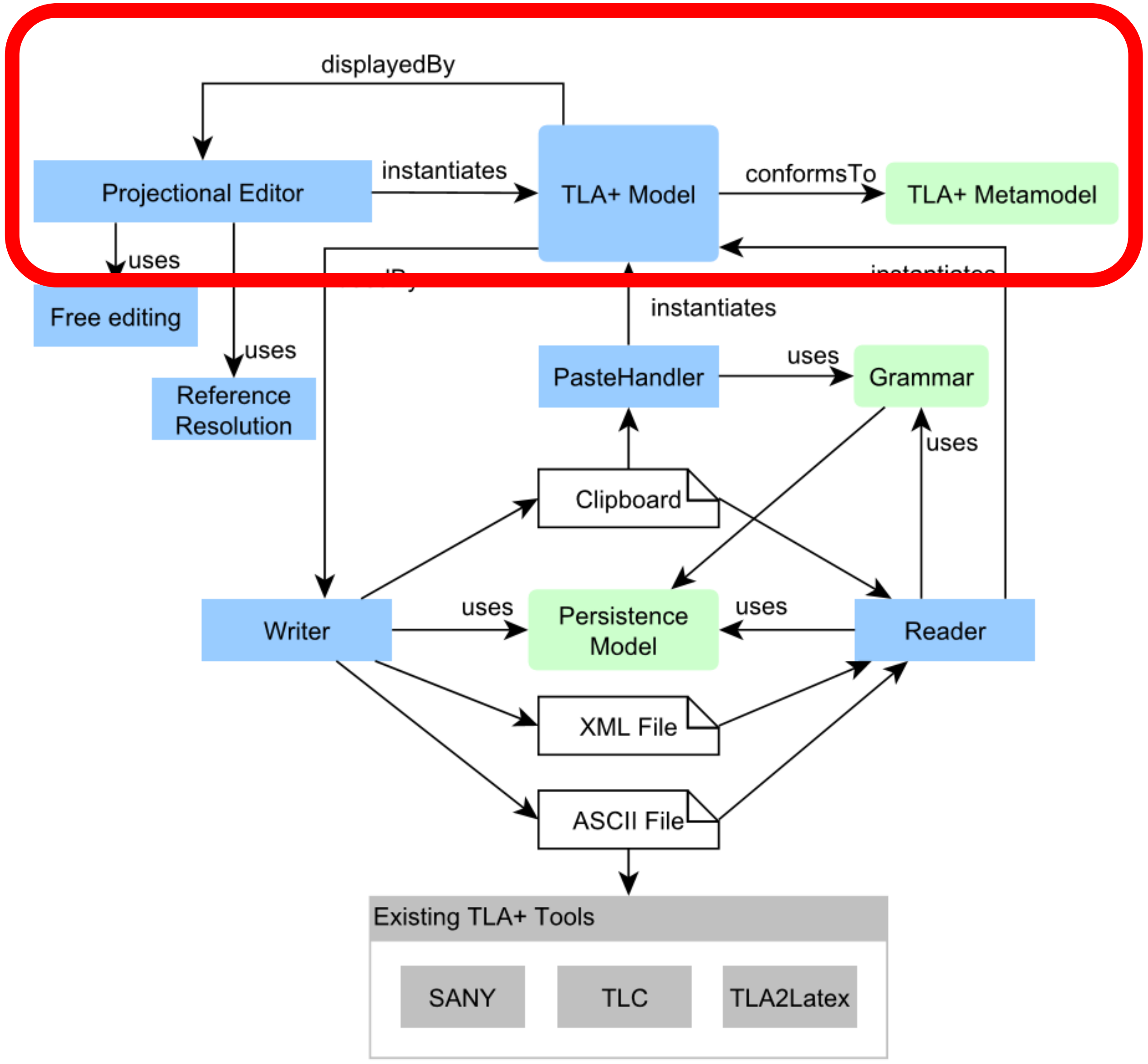
Is it possible to hide TLA+'s syntax duality in a viable bilingual Integrated Development Environment (IDE) to reduce the mental efforts of system engineers?

Objectives and choices

- ❑ SpecEdit, an IDE:
  - ❑ Exposing only the mathematical syntax to the user,
  - ❑ Translating it to the ASCII version for backward compatibility.
- ❑ Technology: JetBrains MetaProgramming System (MPS)
- ❑ Advantages:
  - ❑ Projectional editing
  - ❑ Mathematical notation support
  - ❑ Adequate input mechanisms

Architecture



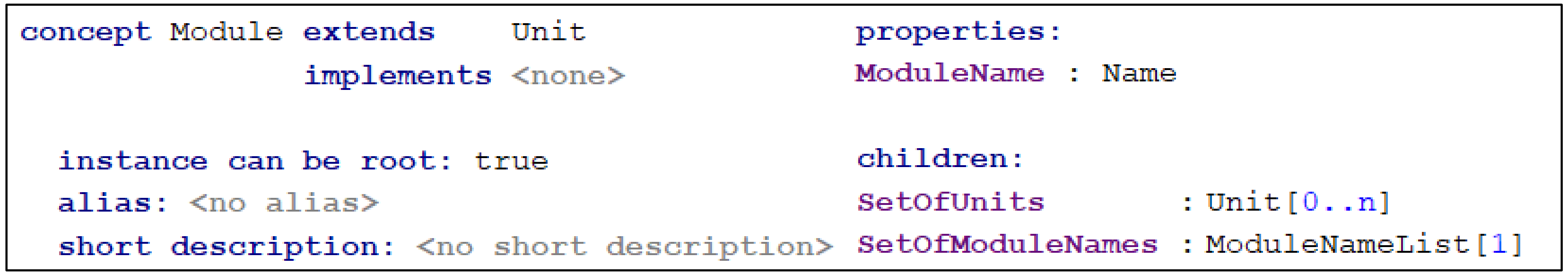


I - Creation of a projectional editor

MPS Concepts

- No grammar
- MPS Structure Language (Abstract Syntax)
- Need to convert the grammar into MPS Concepts
- Result: Metamodel with 110 interconnected Concepts

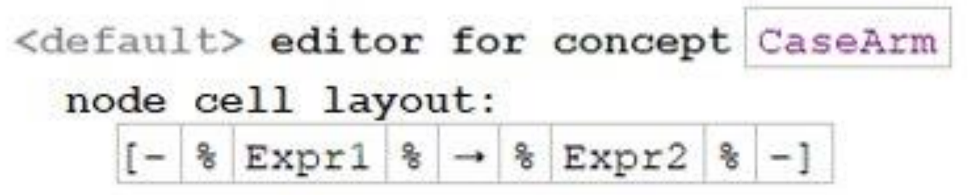
$$\begin{aligned} G.Module \ ::= \ & AtLeast4(\text{“-”}) \ \& \ tok(\text{“MODULE”}) \ \& \ Name \ \& \ AtLeast4(\text{“-”}) \\ \& \ & (Nil \mid (tok(\text{“EXTENDS”}) \ \& \ CommaList(Name))) \\ \& \ & (G.Unit)^{*} \\ \& \ & AtLeast4(\text{“=”}) \end{aligned}$$



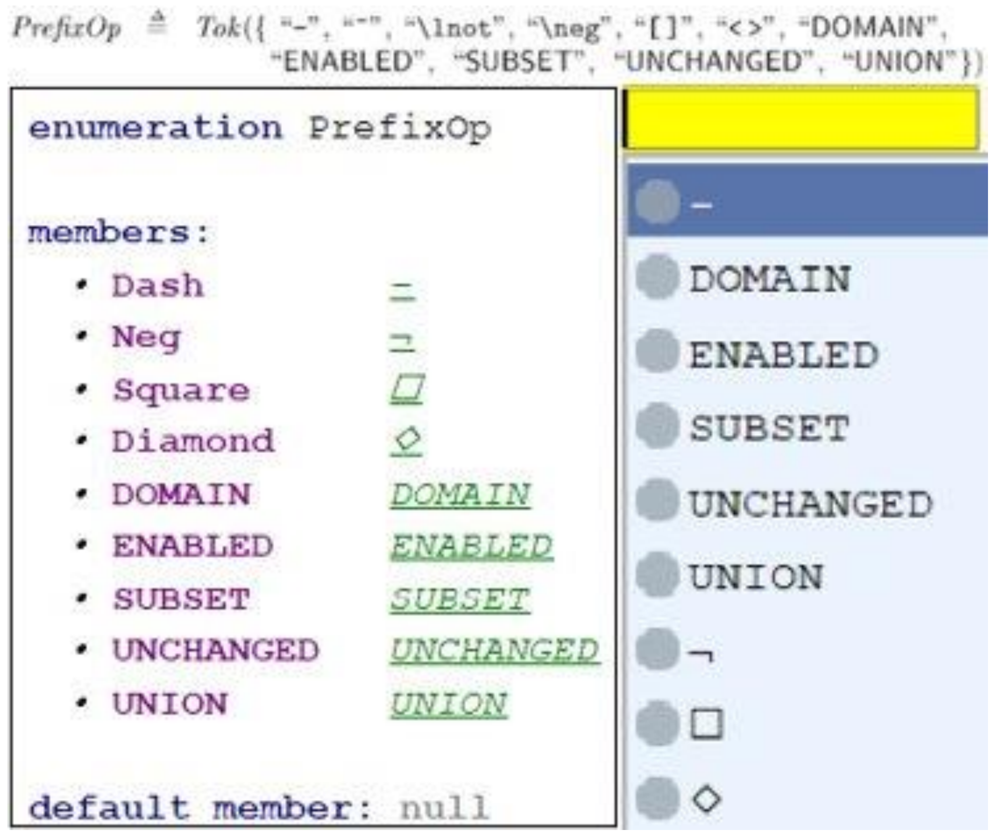
```
concept Module extends     Unit                   properties:
               implements <none>                  ModuleName : Name

  instance can be root: true                      children:
  alias: <no alias>                               SetOfUnits          : Unit[0..n]
  short description: <no short description>      SetOfModuleNames  : ModuleNameList[1]
```

MPS Editors

- ❑ View and Controller (in MPS model-view-controller pattern)
- ❑ Cells (which contain other cells or text)
- ❑ Style (indentation, color, etc.)

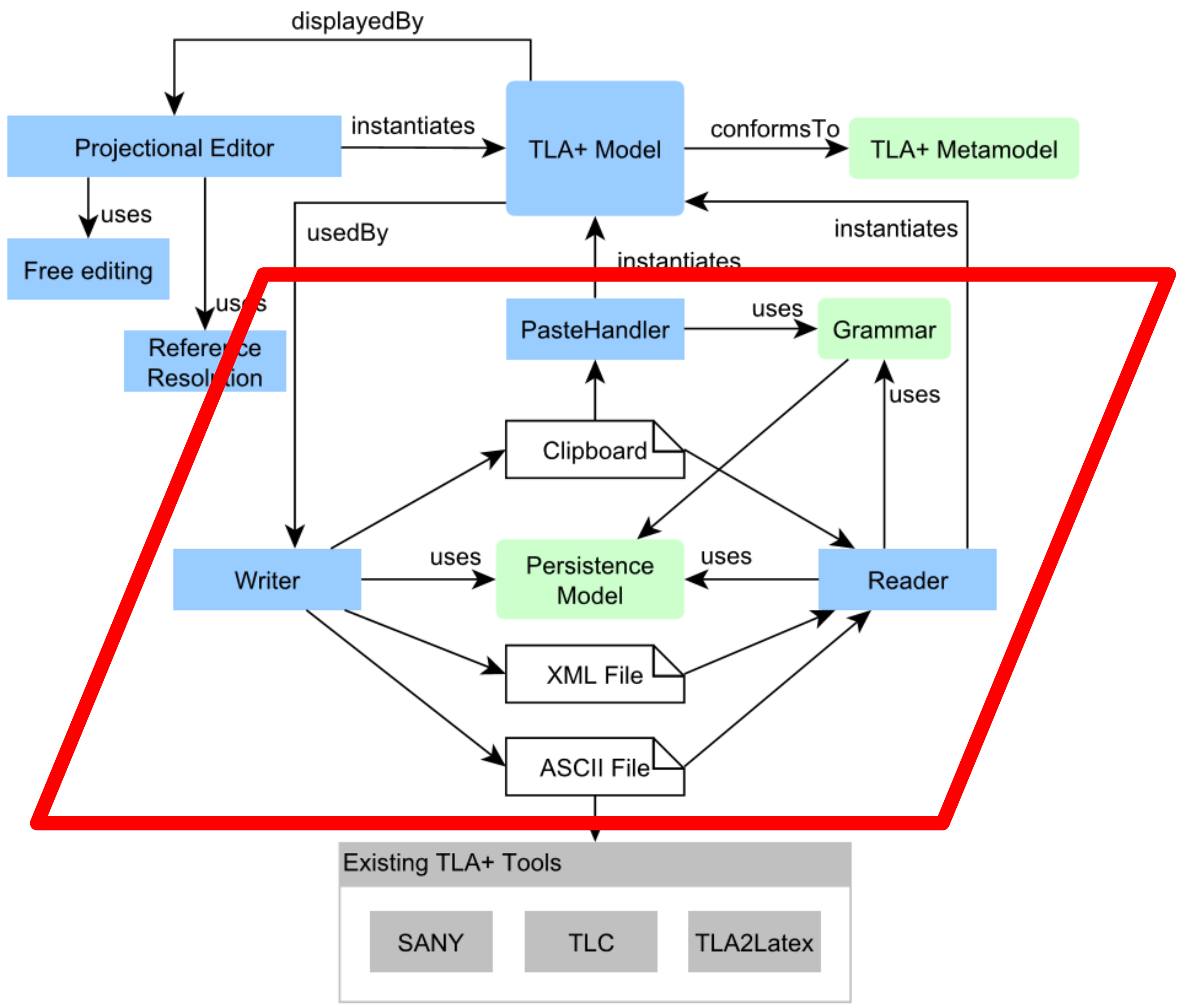


```
<default> editor for concept CaseArm
  node cell layout:
    [- % Expr1 % → % Expr2 % -]
```

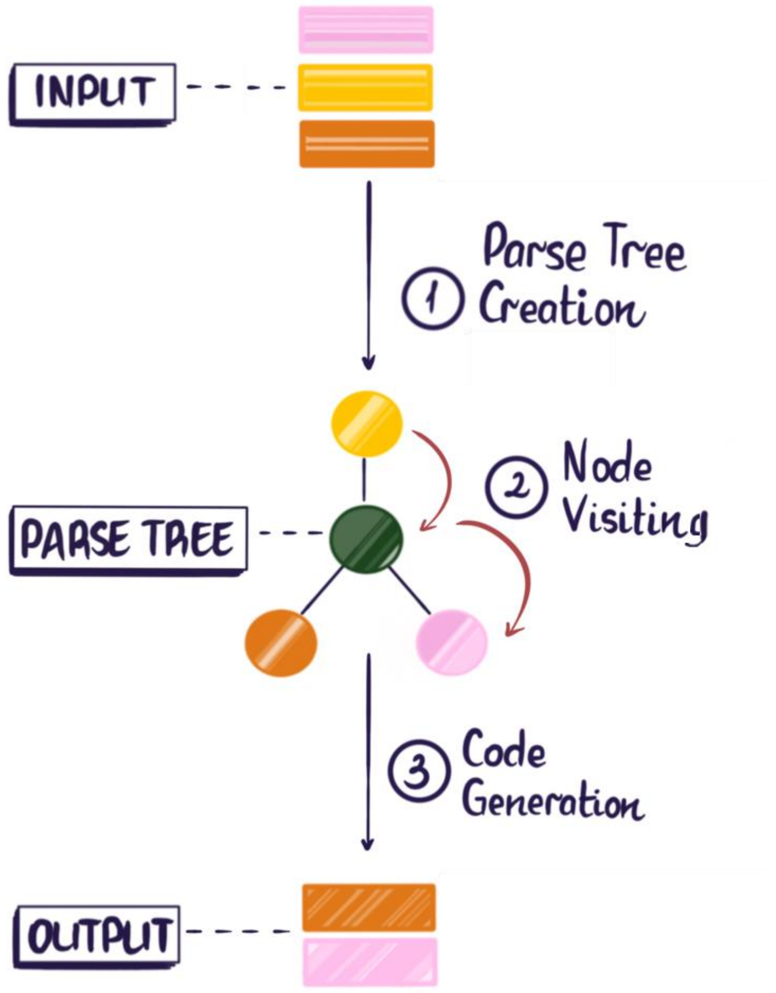


$PrefixOp \triangleq Tok(\{$ "-", "¬", "\lnot", "\neg", "[]", "<>", "DOMAIN", "ENABLED", "SUBSET", "UNCHANGED", "UNION" $\})$

```
enumeration PrefixOp

members:
  • Dash        −
  • Neg         ¬
  • Square      □
  • Diamond     ◇
  • DOMAIN      DOMAIN
  • ENABLED     ENABLED
  • SUBSET      SUBSET
  • UNCHANGED   UNCHANGED
  • UNION       UNION

default member: null
```



II - Plaintext support

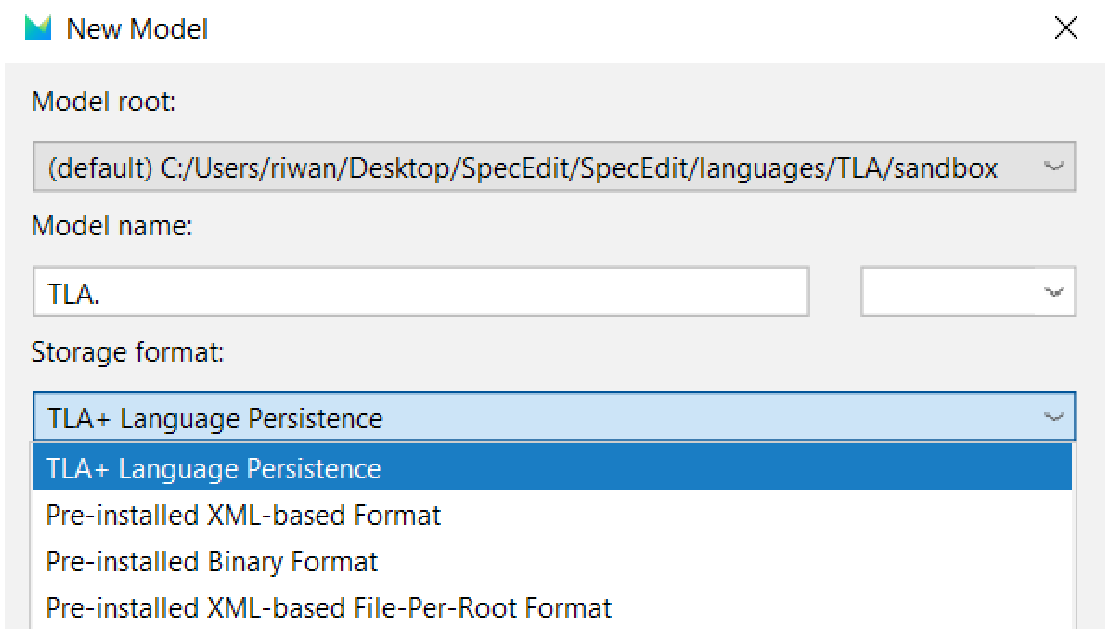


Custom paste handler

- ❑ In charge of managing paste events
- ❑ Integration of ANTLR modules in MPS
    - ❑ Lexing, parsing, visiting
    - ❑ Overriding of the methods of the visitor
    - ❑ Transpiling (ASCII TLA+/MPS Language TLA+)
- ❑ Plugin inserting an entry in the context menu

Custom persistence model (+TextGens)

- ❑ Models are saved in an XML-based format
- ❑ Similar approach (different source)
- ❑ Rewriting loading/saving strategy
- ❑ Plugin (set as a ModelFactoryProvider)

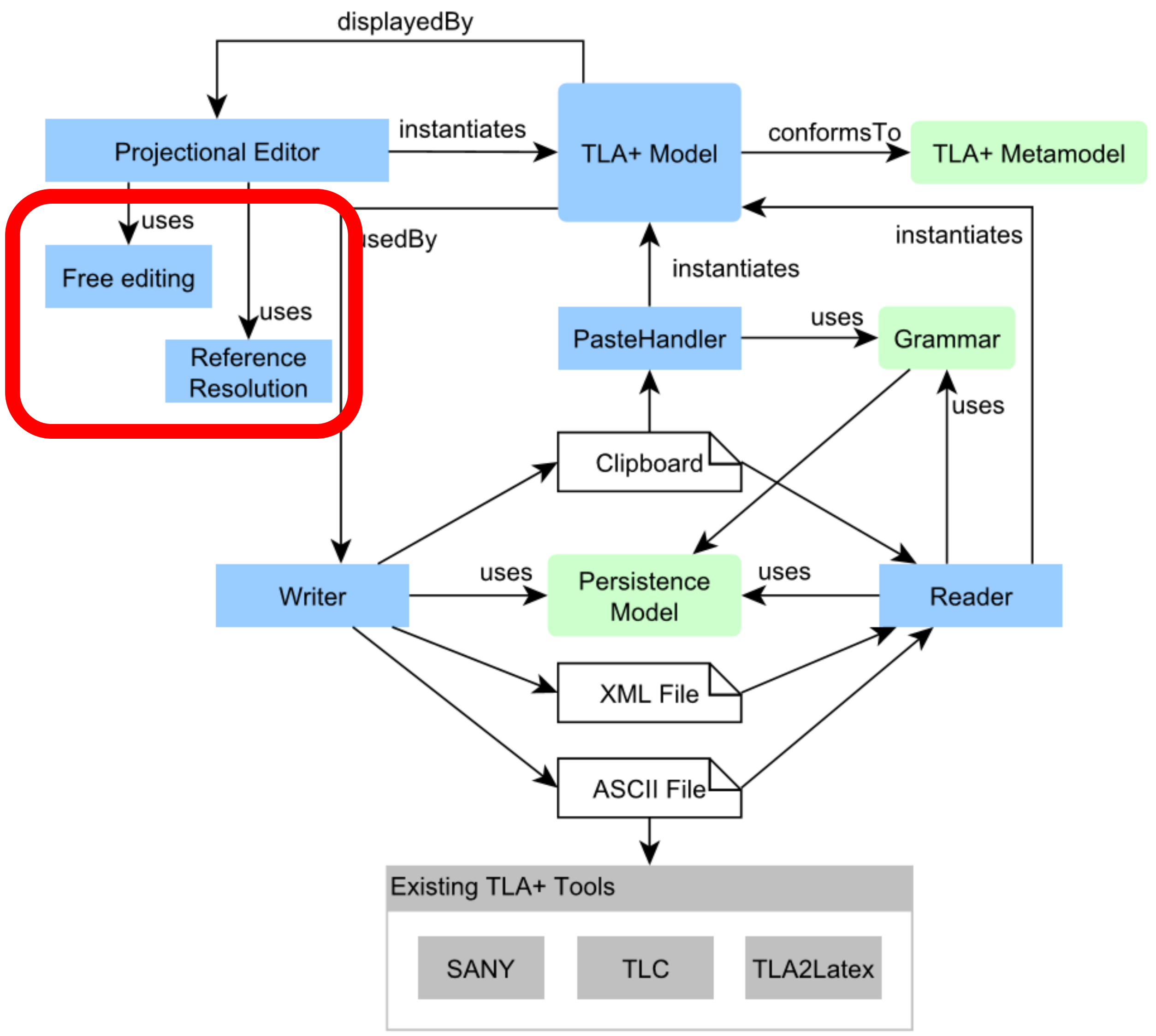

New Model

Model root:
(default) C:/Users/riwan/Desktop/SpecEdit/SpecEdit/languages/TLA/sandbox

Model name:
TLA.

Storage format:
TLA+ Language Persistence

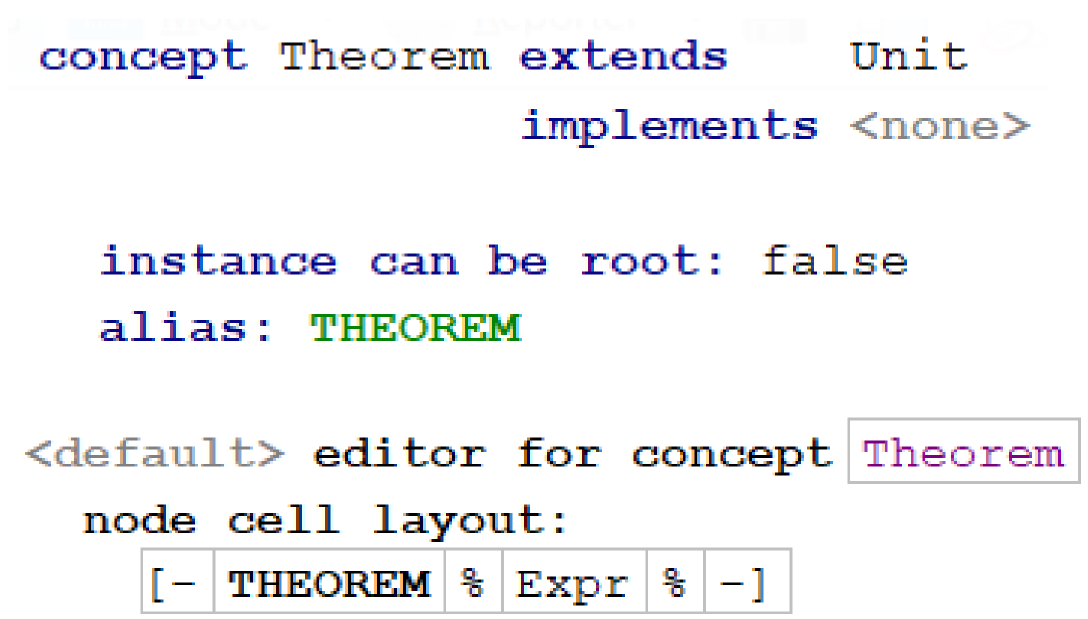
TLA+ Language Persistence
Pre-installed XML-based Format
Pre-installed Binary Format
Pre-installed XML-based File-Per-Root Format



III - Customization of user experience (UX)

Free editing support

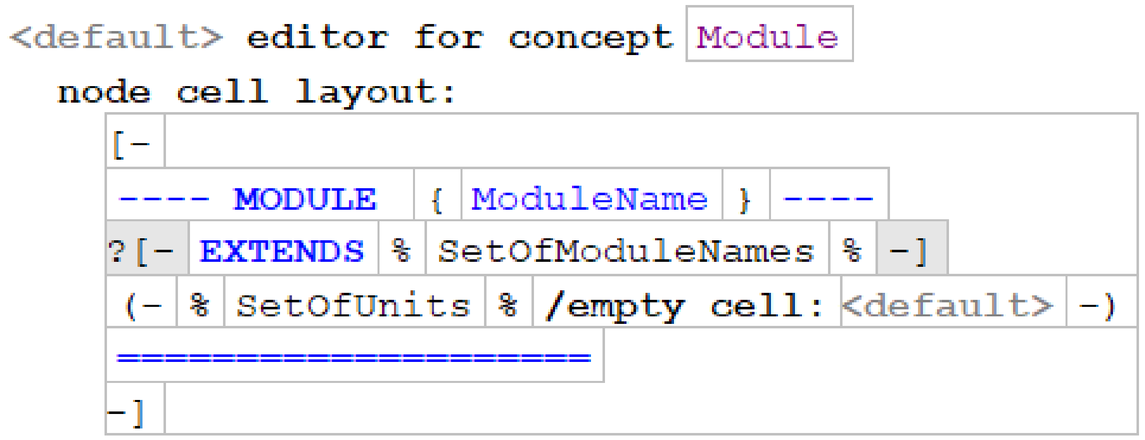
- Prohibited by default (Context menu for completion)
- “Editable” property: Not enough (syntactic error)
- Node instantiation triggered for given string via transformation menus and aliases



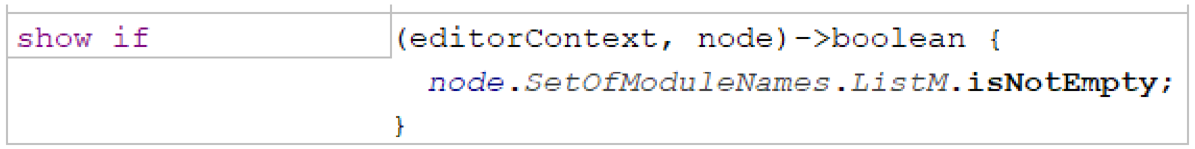
```
concept Theorem extends     Unit
                implements <none>

  instance can be root: false
  alias: THEOREM

<default> editor for concept Theorem
  node cell layout:
    [- THEOREM % Expr % -]
```

Optional field management

- Side transformations (available when users type from the left or right part of a cell)
- Combination with hidden fields
- Definition of actions to be executed on a given written string to unhide fields



```
<default> editor for concept Module
  node cell layout:
    [-
    ---- MODULE { ModuleName } ----
    ?[- EXTENDS % SetOfModuleNames % -]
     (- % SetOfUnits % /empty cell: <default> -)
    ======================
    -]
```

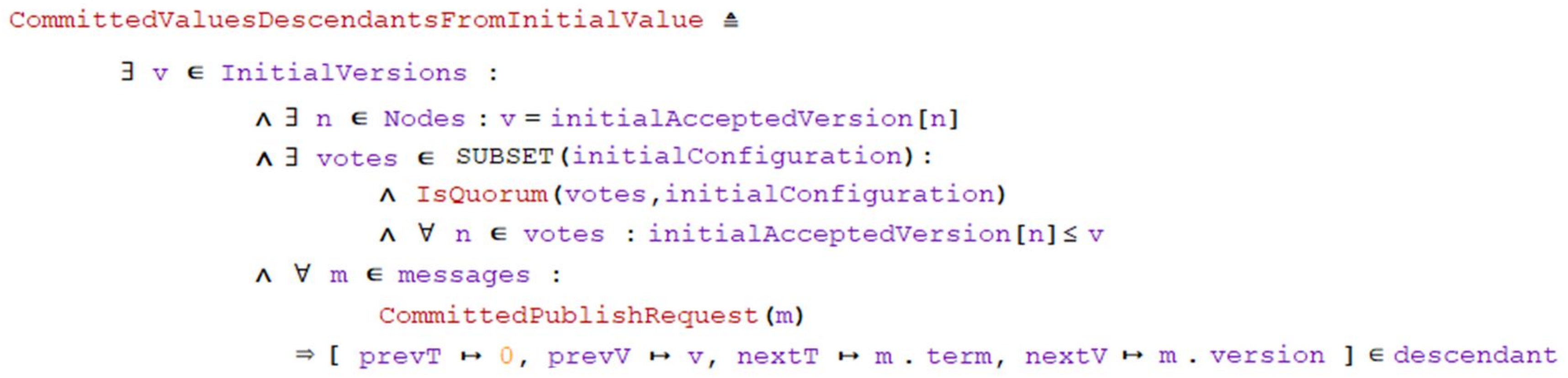


```
show if    (editorContext, node)->boolean {
             node.SetOfModuleNames.ListM.isNotEmpty;
           }
```

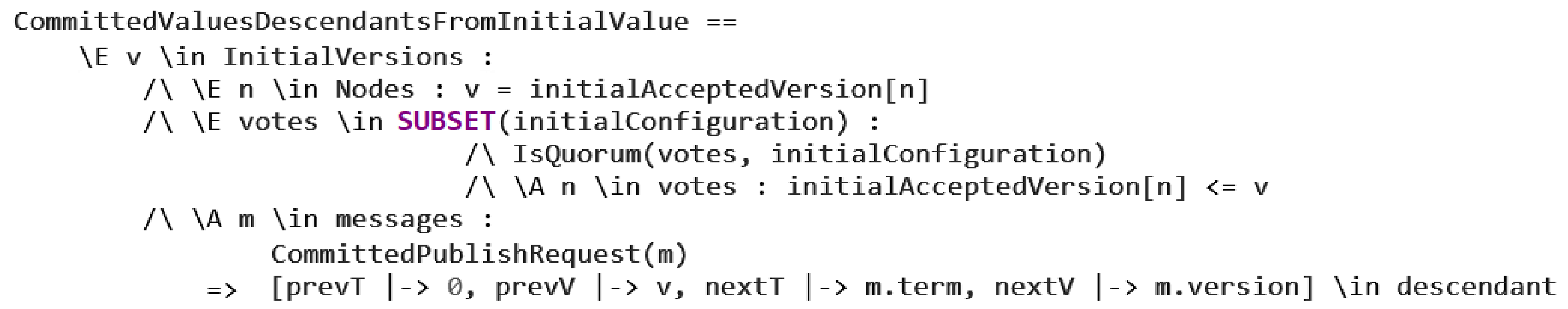
IV - SpecEdit in practice

Concrete example

- Elasticsearch
- Comparison of the rendering between TLA+ Toolbox and SpecEdit



```
CommittedValuesDescendantsFromInitialValue ≜

      ∃ v ∈ InitialVersions :

           ∧ ∃ n ∈ Nodes : v = initialAcceptedVersion[n]
           ∧ ∃ votes ∈ SUBSET(initialConfiguration):
                ∧ IsQuorum(votes,initialConfiguration)
                ∧ ∀ n ∈ votes : initialAcceptedVersion[n] ≤ v
           ∧ ∀ m ∈ messages :
                CommittedPublishRequest(m)
             ⇒ [ prevT ↦ 0, prevV ↦ v, nextT ↦ m . term, nextV ↦ m . version ] ∈ descendant
```



```
CommittedValuesDescendantsFromInitialValue ==
    \E v \in InitialVersions :
        /\ \E n \in Nodes : v = initialAcceptedVersion[n]
        /\ \E votes \in SUBSET(initialConfiguration) :
                            /\ IsQuorum(votes, initialConfiguration)
                            /\ \A n \in votes : initialAcceptedVersion[n] <= v
        /\ \A m \in messages :
                CommittedPublishRequest(m)
            =>  [prevT |-> 0, prevV |-> v, nextT |-> m.term, nextV |-> m.version] \in descendant
```

Conclusion and perspectives

- Projectional approach
- Merging of the existing syntaxes
- Need to formalize a new language model
- Not yet a full-fledged IDE
- Meant at epitomizing what can be achieved through projectional editing
- Further research directions: Model federation and tabular/graphical projections

Project repository: github.com/RiwanC/SpecEdit
Demo video: youtu.be/8JGlZt_DNt8