Vienna Talk

A Formal Method Environment on Pharo

Tomohiro Oda
Software Research Associates, Inc.

Tomo openInEsug

- Smalltalking since 1994
    - Smalltalk/V, VisualSmalltalk, VisualWorks, Squeak and then Pharo
    - 3D graphics (Jun on VW and later on Pharo), research prototypes
    - a big fan of ESUG since 2016
- Working for Software Research Associates, Inc.
    - SRA is a software vendor since 1967
    - one of the early adapters of OO in Japan
    - co-organized Smalltalk Day Japan 2019 with Smalltalk Study Group Tokyo.
- Collaborating with
    - smalltalkers: YOU
    - formal methods: VDM communities in Europe and Japan
    - scientists: chemists, biologists, economists and so on.

ViennaTalk is LIVE and FORMAL
(formal tools with the spirit of Smalltalk)

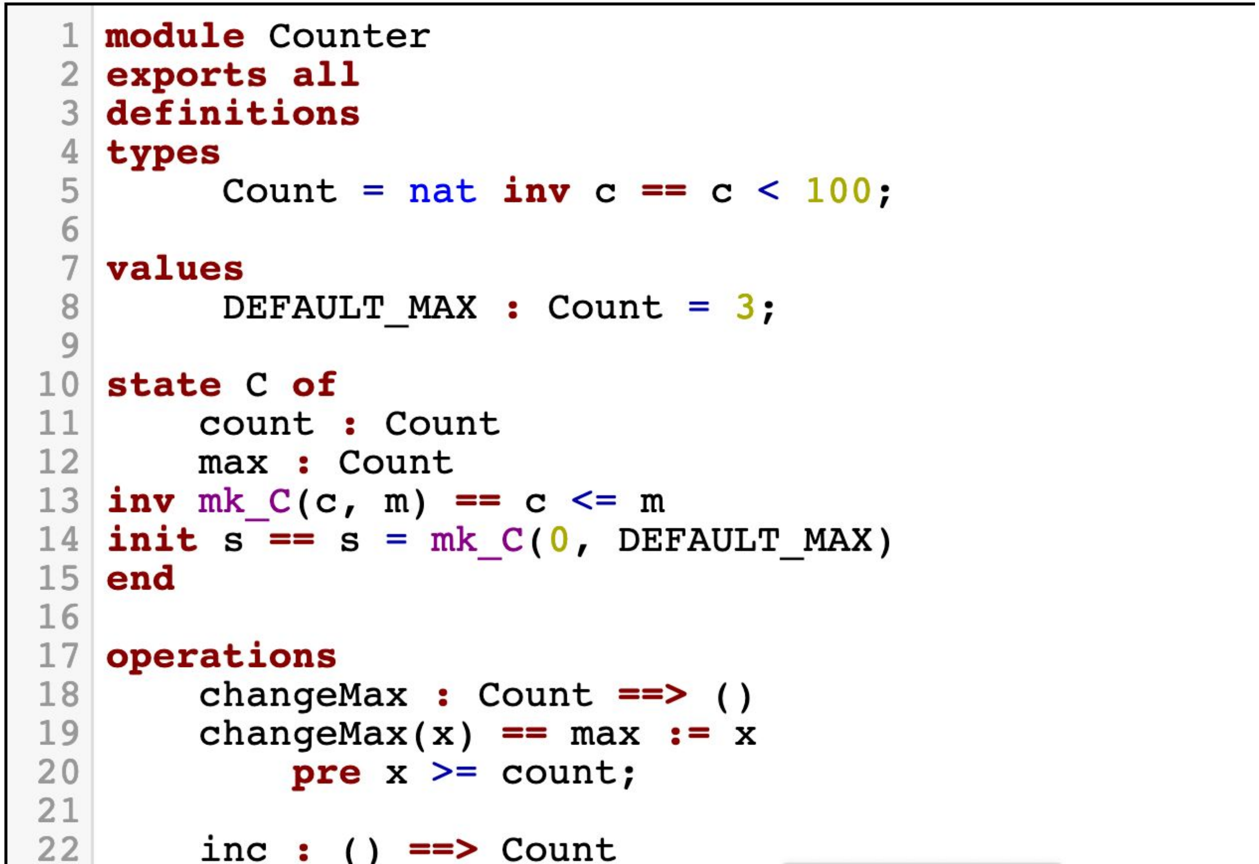
IDE for VDM-SL (Vienna Development Method - Specification Language)

- **Execution engines** to **simulate** the specified system
    - external interpreter (with VDMJ)
    - cloud interpreter (by ViennaTalk)
    - Smalltalk code generators
- **VDMPad**: Web IDE to **scratch idea**
- **VDMBrowser**: Smalltalk-like Browser to **explore solutions**
    - pretty printer / syntax highlighter / Workspace / Inspector / DoIt-PrintIt
- **Lively Walk-Through**: UI prototyping tool to **communicate** with UI designers
- **Webly Walk-Through**: HTTP server to **publish** a spec as an Web API
- **ViennaDoc**: JavaScript library to **document** a specification with eval and test.

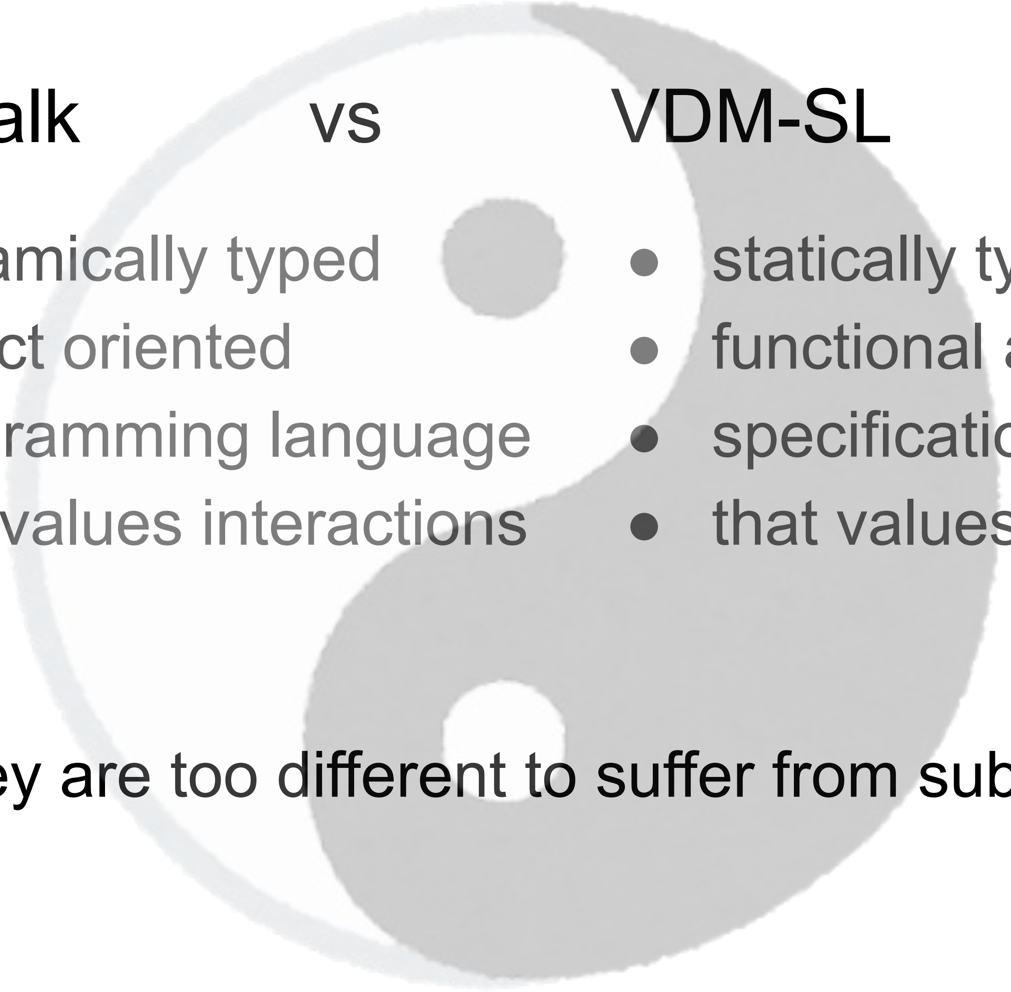
Smalltalk is not alone.

1. We build systems using many languages.
2. Building dev envs for guest languages is fun.

VDM-SL at a glance



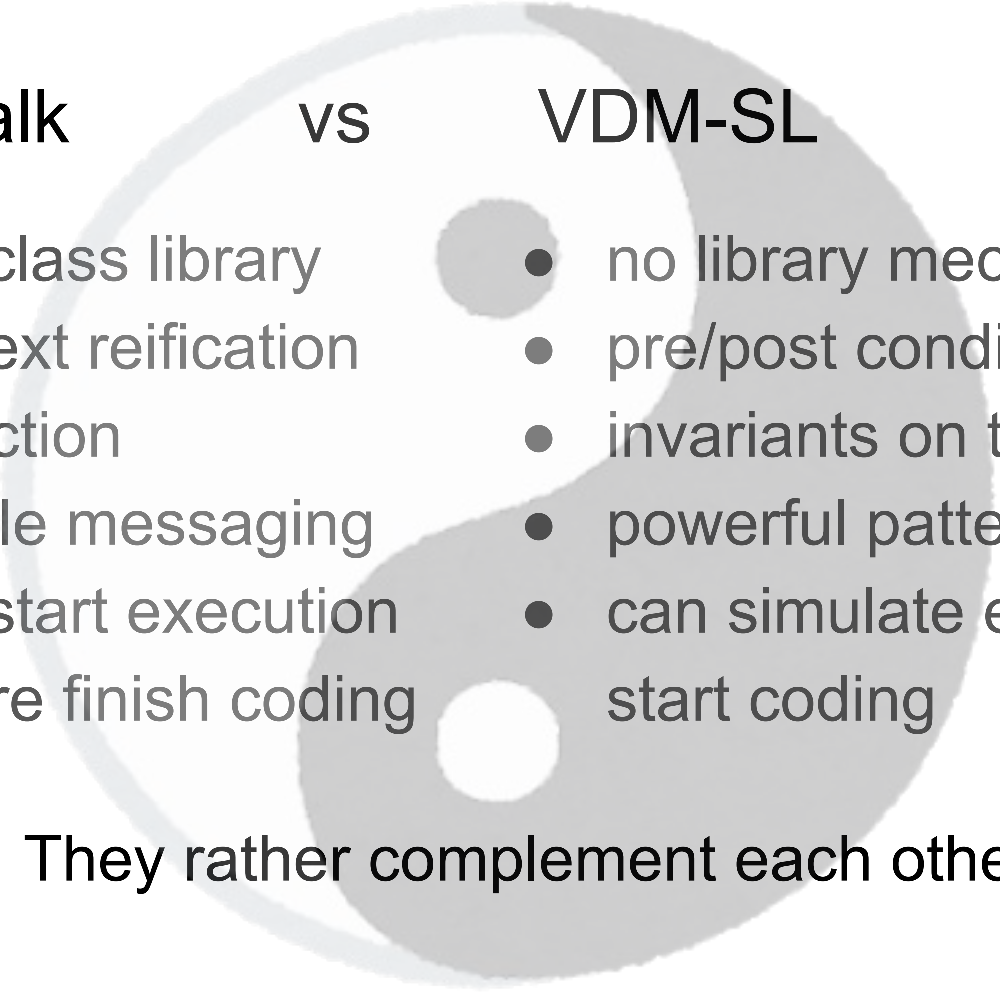
```
module Counter
exports all
definitions
types
     Count = nat inv c == c < 100;

values
     DEFAULT_MAX : Count = 3;

state C of
    count : Count
    max : Count
inv mk_C(c, m) == c <= m
init s == s = mk_C(0, DEFAULT_MAX)
end

operations
    changeMax : Count ==> ()
    changeMax(x) == max := x
        pre x >= count;

    inc : () ==> Count
```



Smalltalk	vs	VDM-SL
● dynamically typed		● statically typed
● object oriented		● functional and procedural
● programming language		● specification language
● that values interactions		● that values computations

They are too different to suffer from subtle conflicts.

Smalltalk vs VDM-SL



Smalltalk

- rich class library
- context reification
- reflection
- simple messaging
- can start execution before finish coding

VDM-SL

- no library mechanism
- pre/post conditions
- invariants on types and variables
- powerful pattern matching
- can simulate execution before start coding

They rather complement each other.

Why exploratory specification?

Because there's no perfect specification

- The world is always changing.
- Everyone has only limited knowledge.
- We learn from our products in action.

Smalltalk's exploratory programming style to formal specification!

demo

ViennaTalk is LIVE and FORMAL
(formal tools with the spirit of Smalltalk)

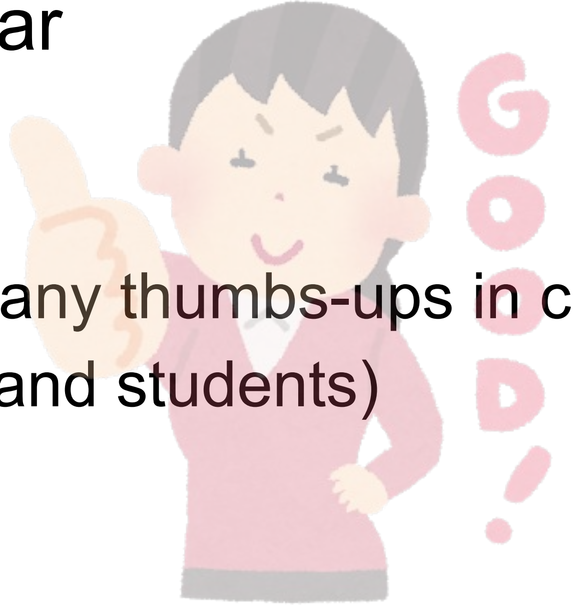
IDE for VDM-SL (Vienna Development Method - Specification Language)

- **Execution engines** to **simulate** the specified system
  - external interpreter (with VDMJ)
  - cloud interpreter (by ViennaTalk)
  - Smalltalk code generators
- **VDMPad**: Web IDE to **scratch idea**
- **VDMBrowser**: Smalltalk-like Browser to **explore solutions**
  - pretty printer / syntax highlighter / Workspace / Inspector / DoIt-PrintIt
- **Lively Walk-Through**: UI prototyping tool to **communicate** with UI designers
- **Webly Walk-Through**: HTTP server to **publish** a spec as an Web API
- **ViennaDoc**: JavaScript library to **document** a specification with eval and test.

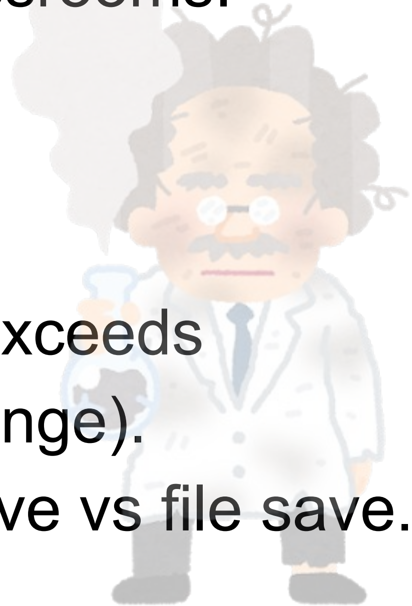
Evaluations so far

Good

- VDMPad got many thumbs-ups in classrooms.
  (by both tutors and students)



Bad



- Auto-generated methods sometimes exceeds
  the limitation of bytecode (e.g. jump range).
- Users are often confused by image save vs file save.

a pile of todos

- git support
- spec2 migration
- type checker / type inference
- code completion
- ViennaVM (VM for VDM)
- code generator for ViennaVM

Packages in ViennaTalk

- ViennaTalk-Animation-Core
- ViennaTalk-Animation-Transpiler
- ViennaTalk-Browser-Core
- ViennaTalk-Engine-Core
- ViennaTalk-Html-Core
- ViennaTalk-Launcher-Core
- ViennaTalk-LivelyWalkThrough-Core
- ViennaTalk-LivelyWalkThrough-Widgets
- ViennaTalk-Parser-Core
- ViennaTalk-Parser-Formatters
- ViennaTalk-Parser-Highlighters
- ViennaTalk-Parser-UI
- ViennaTalk-Transpiler-Core
- ViennaTalk-Transpiler-Debugger
- ViennaTalk-Transpiler-Test
- ViennaTalk-Type-Core
- ViennaTalk-VDMPad-Core
- ViennaTalk-Value-Core
- ViennaTalk-WeblyWalkThrough-Server
- ViennaTalk-WeblyWalkThrough-Translators

- ViennaTalk-Animation-Tests
- ViennaTalk-Engine-Tests
- ViennaTalk-Parser-Tests
- ViennaTalk-Type-Tests
- ViennaTalk-Value-Tests

ViennaTalk uses many Smalltalk/Pharo technologies

- Pharo (OpalCompiler, Slot)
- CogVM
- Zinc/NeoJSON
- PetitParser2
- OSSubprocess/ProcessWrapper
- MoldableDebugger

visit http://viennatalk.org/ for details.

supported by Grant-in-Aid for Scientific Research (S) 24220001,
Grant-in-Aid for Scientific Research (C) 26330099, and
JSPS KAKENHI Grant Number JP 18K18033.