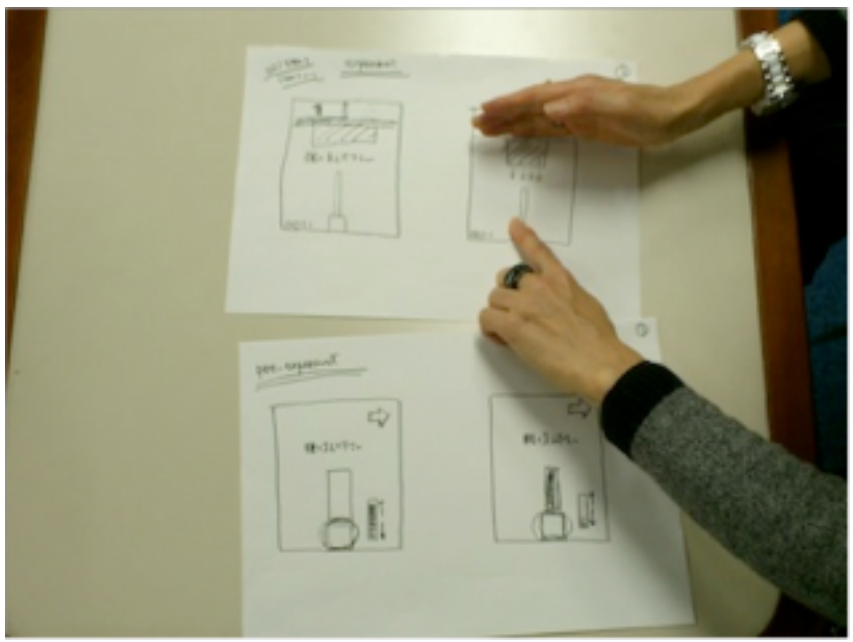
ライブ UI プロトタイピングに向けたマルチ言語環境 SOMETHINGit

小田朋宏 (株)SRA
中小路久美代 (株)SRA
山本恭裕 東京工業大学

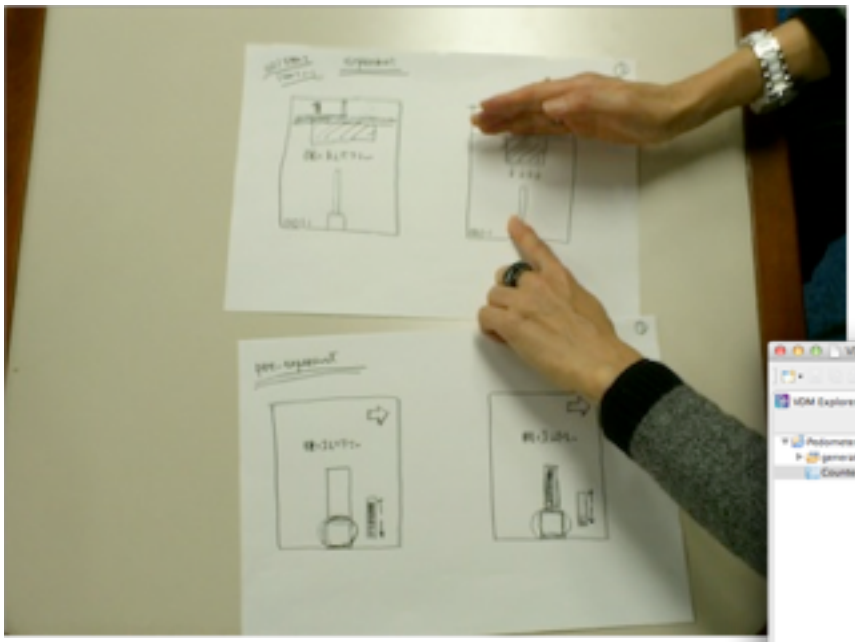
UI design
Lightweight Formal Method

動機

- 軽量形式手法をより多くの人が手軽に使える環境を作りたい
  - 例えば、UIデザイナ



軽量形式手法とUIデザイン
それぞれのプロトタイピングの特徴

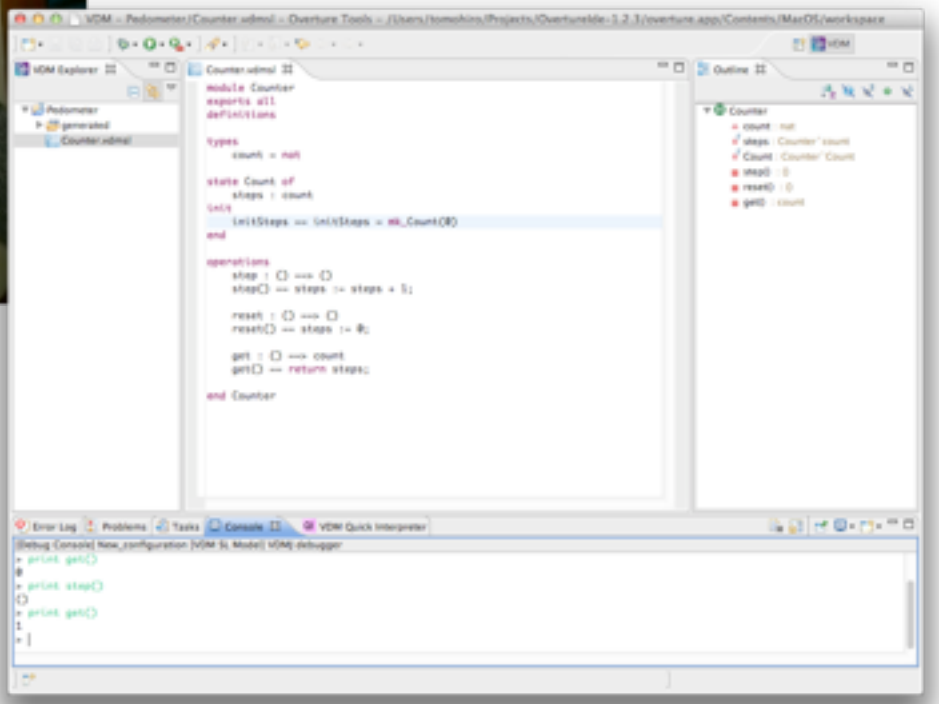


UIスケッチ

- 具体性
- ニュアンス

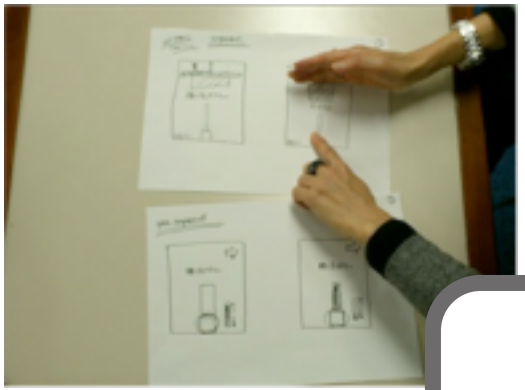
実行可能仕様

- 実行可能性
- 厳密性

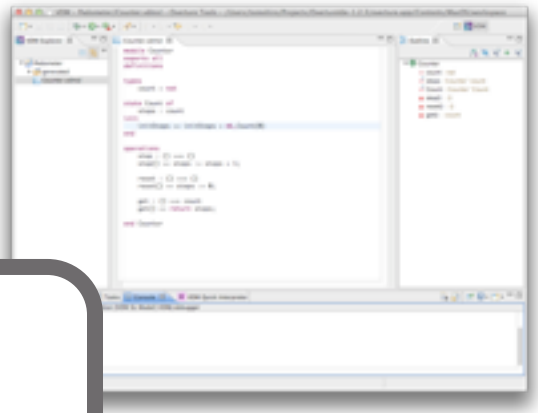


軽量形式手法＋UIプロトタイピング

UIデザイナ



形式手法技術者



実行可能な
UIプロトタイプ

デザインデシジョンの合意

めざすところ

- 「機能モデル的に実現可能なUIデザイン」

  かつ、

  「適切なUIを構築可能な機能モデル」

  ができるようなプロトタイプ環境を作りたい

本研究開発のアプローチ

UIをデザインすることと機能をモデリングすることを
共同デザインとして捉える

UIデザイナと形式手法技術者の間の「対話」のデザイン

**プロトタイピング環境構築に必要な
ライブラリの実装**

UIプロトタイピング環境の構築

LIVE

合意形成のためのプロトタイピング

- プロトタイプの定義とビュー双方について
  - 仮説的状況を実現して試行できること
  - 何が起こっているのか理解できること
  - 手に取って指し示して議論できること

そこで

  - プロトタイプを定義する記述環境
  - プロトタイプを使う実行環境

を同一にする　＝ ライブプロトタイピング環境

SOMETHINGit

SOMETHINGit

- Smalltalk (Squeak/Pharo) 上のライブラリ
  - 外部インタプリタを呼び出す
    - VDM-SL (VDMJ)
    - Haskell (GHCi)
- ライブプロトタイピングを指向
  - VDM-SLでの

    Smalltalk流のライブな

    プロトタイピングを可能にする

SOMETHINGitによる Smalltalk流プログラミング環境の実現

- 情報隠蔽を破るバックドア
- 実行時改変と継続実行
- 自由度の高いFFI

Backdoors

情報隠蔽を破るバックドア

VDM-SLモジュールのstate内の変数はprivate

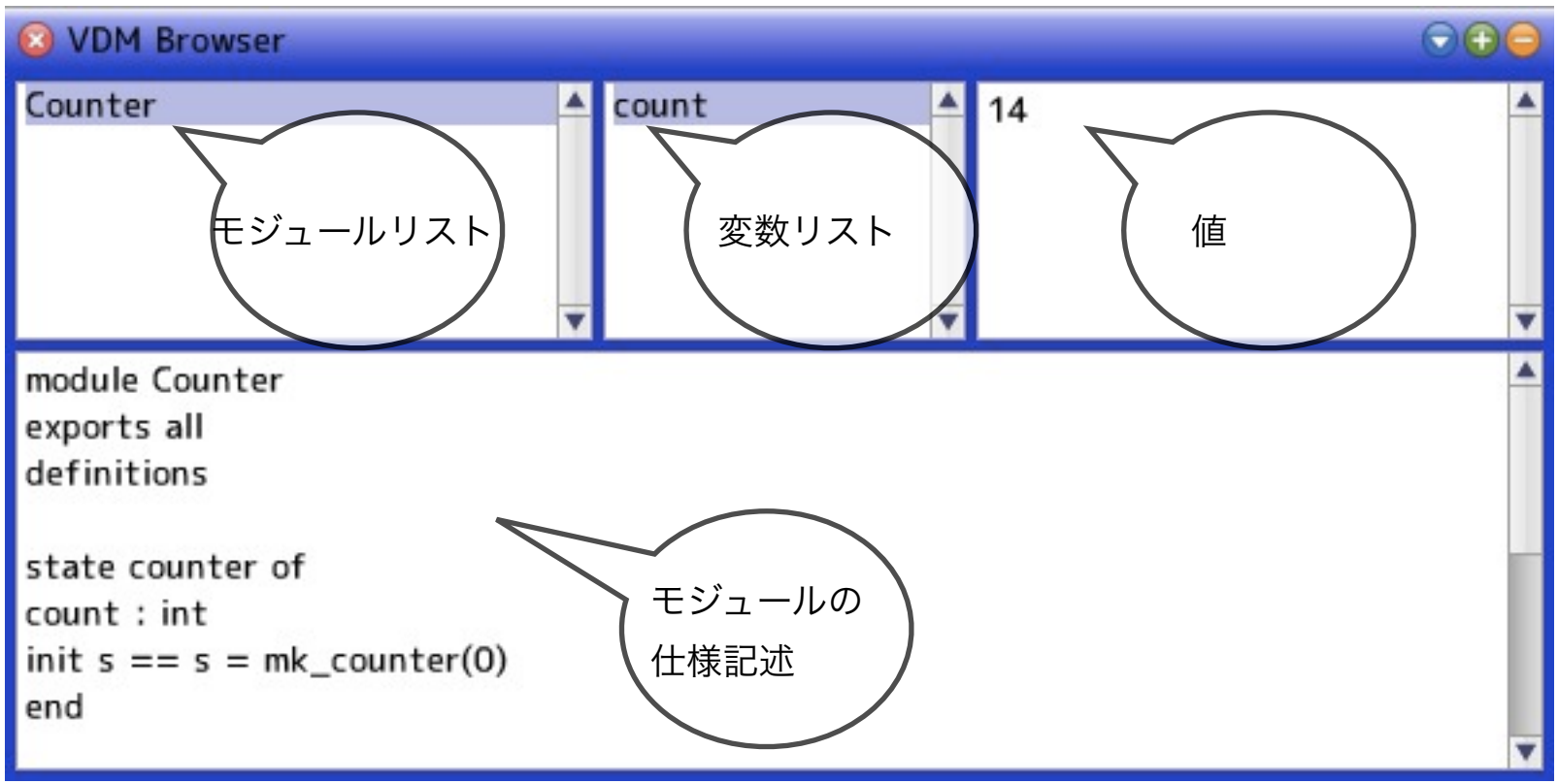
- 外部から値を読んだり書き込むためにはoperationを通す必要がある
- そこでstate内の全変数に一括して読み書きするpublicなアクセサを自動生成する
  - setter
  - getter

安全なバックドア

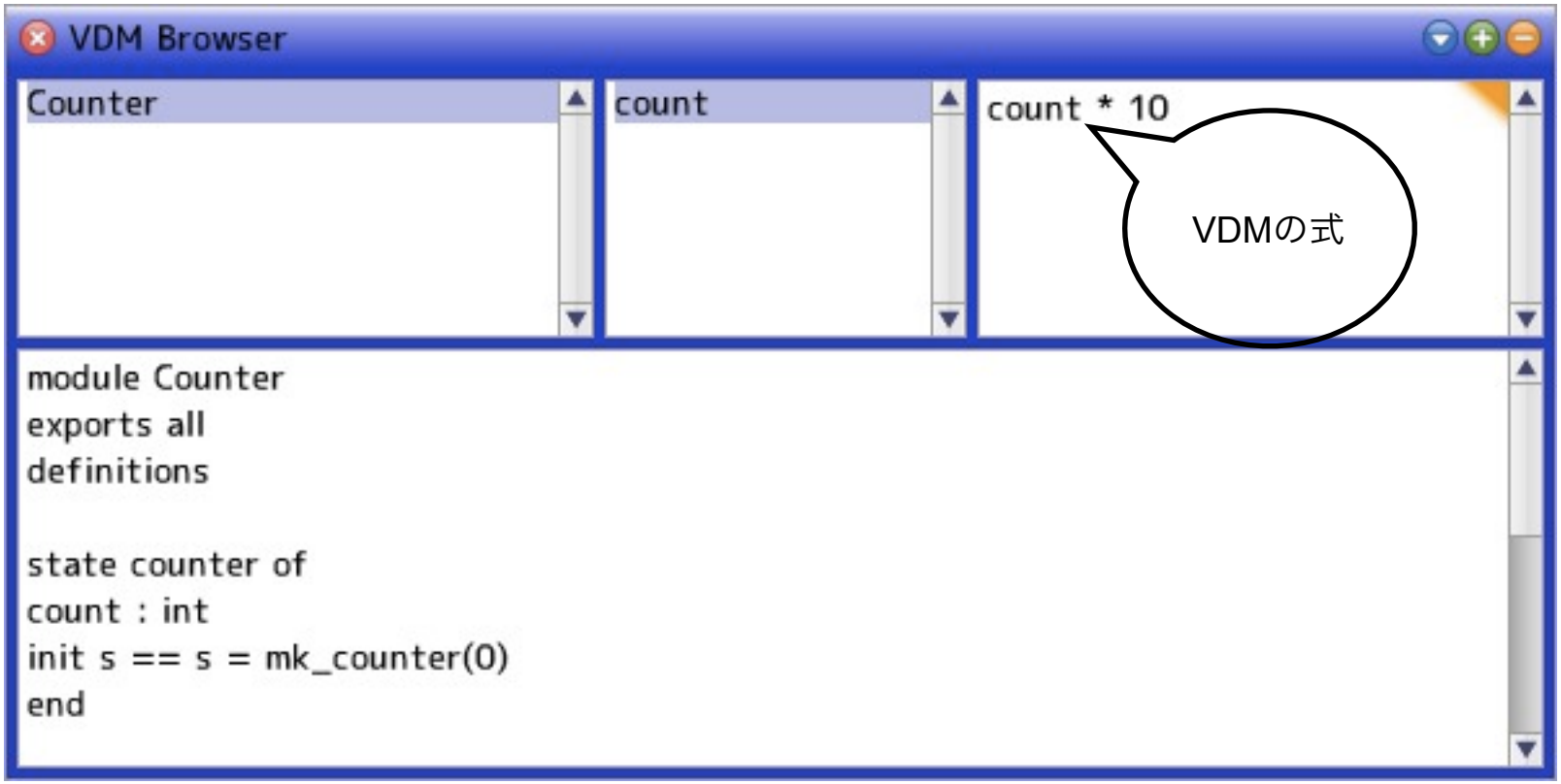
VDM-SL仕様中ではバックドアを利用できない

1. バックドア無しの仕様をVDMJにロードする
2. コンパイルエラーが発生しなければ
    a. バックドアを自動生成し
    b. バックドア付きの仕様を再ロードする

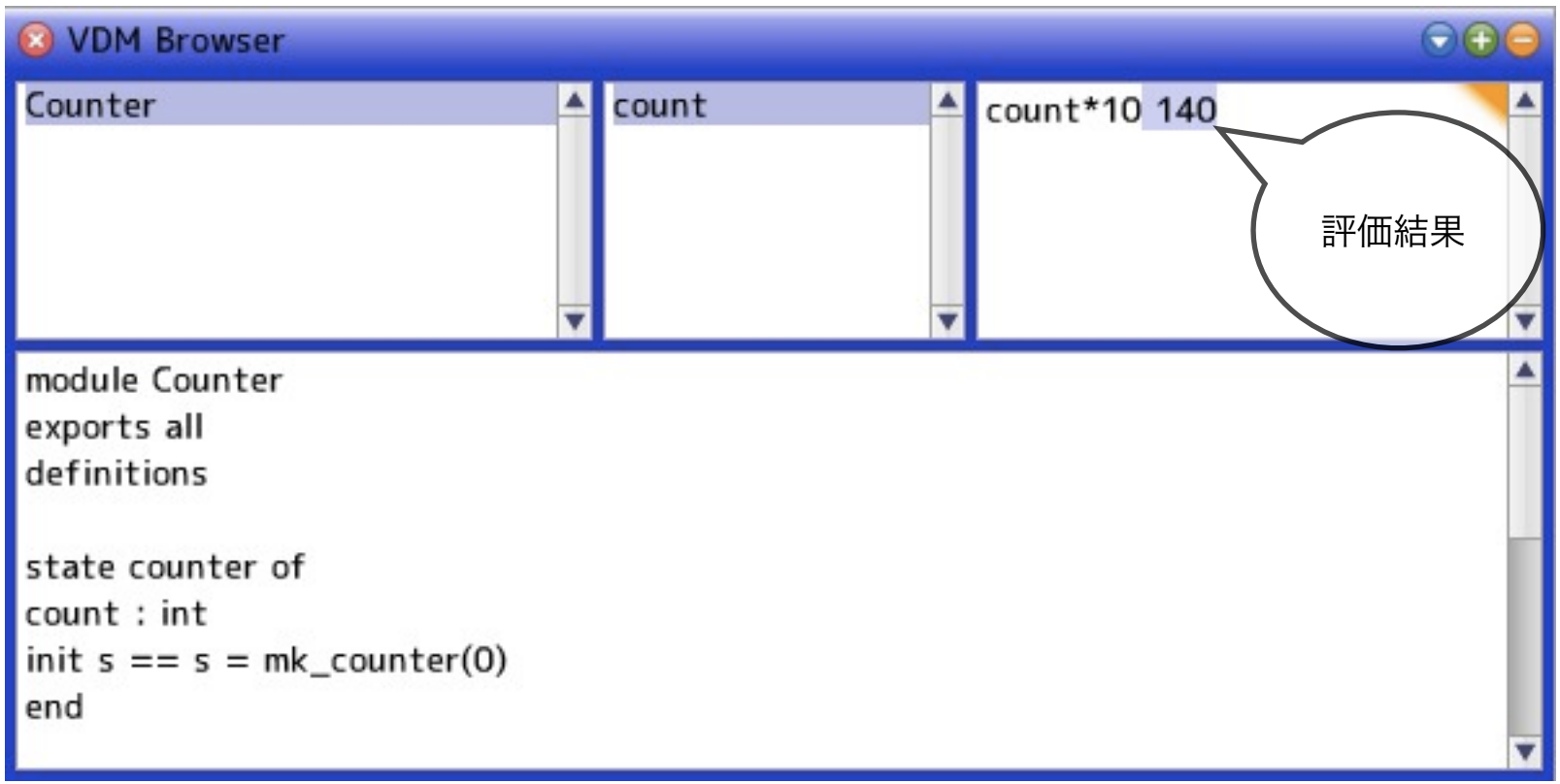
例：state変数の値を直接操作する



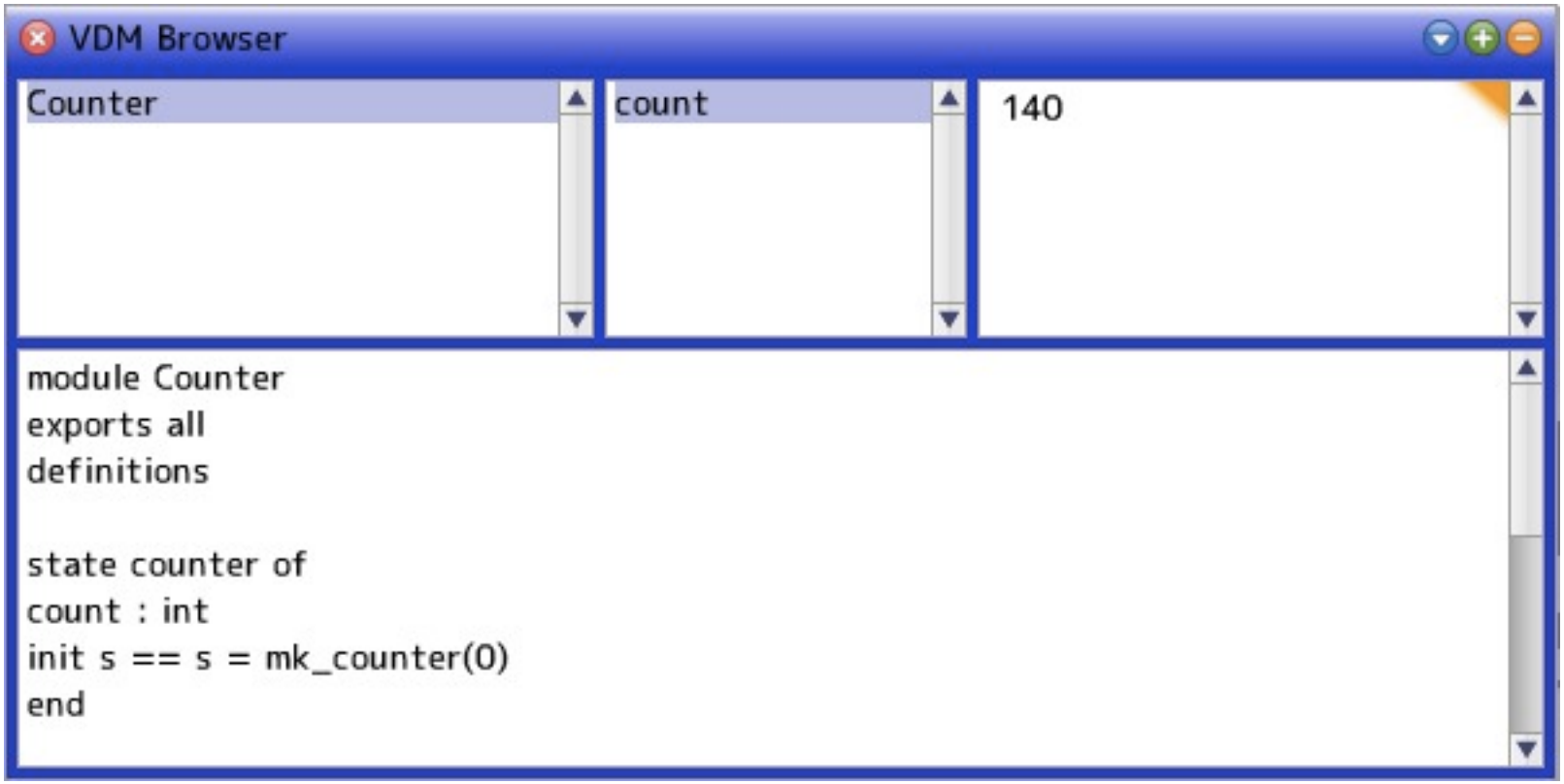
count * 10を計算する



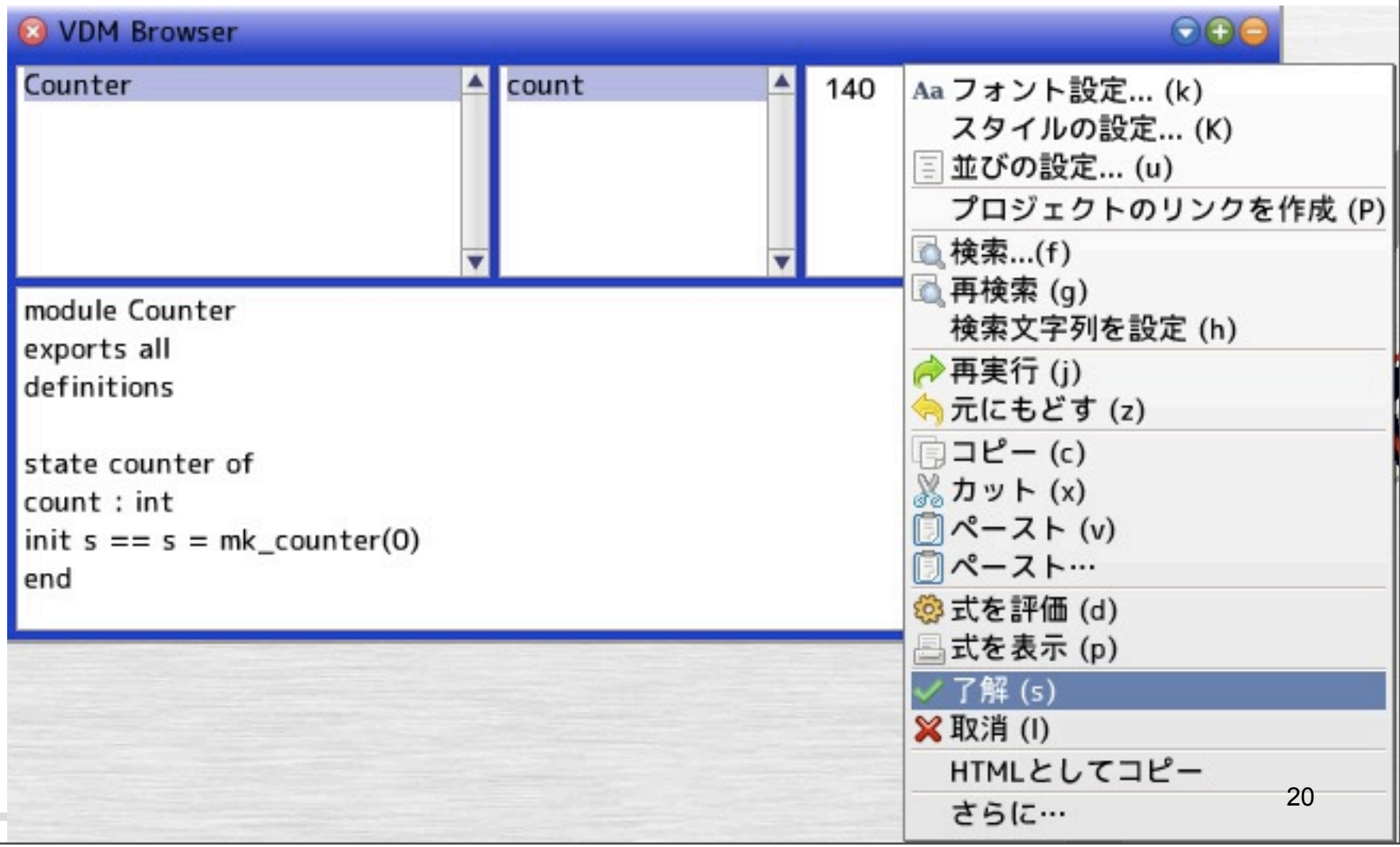
count * 10 の答えは140



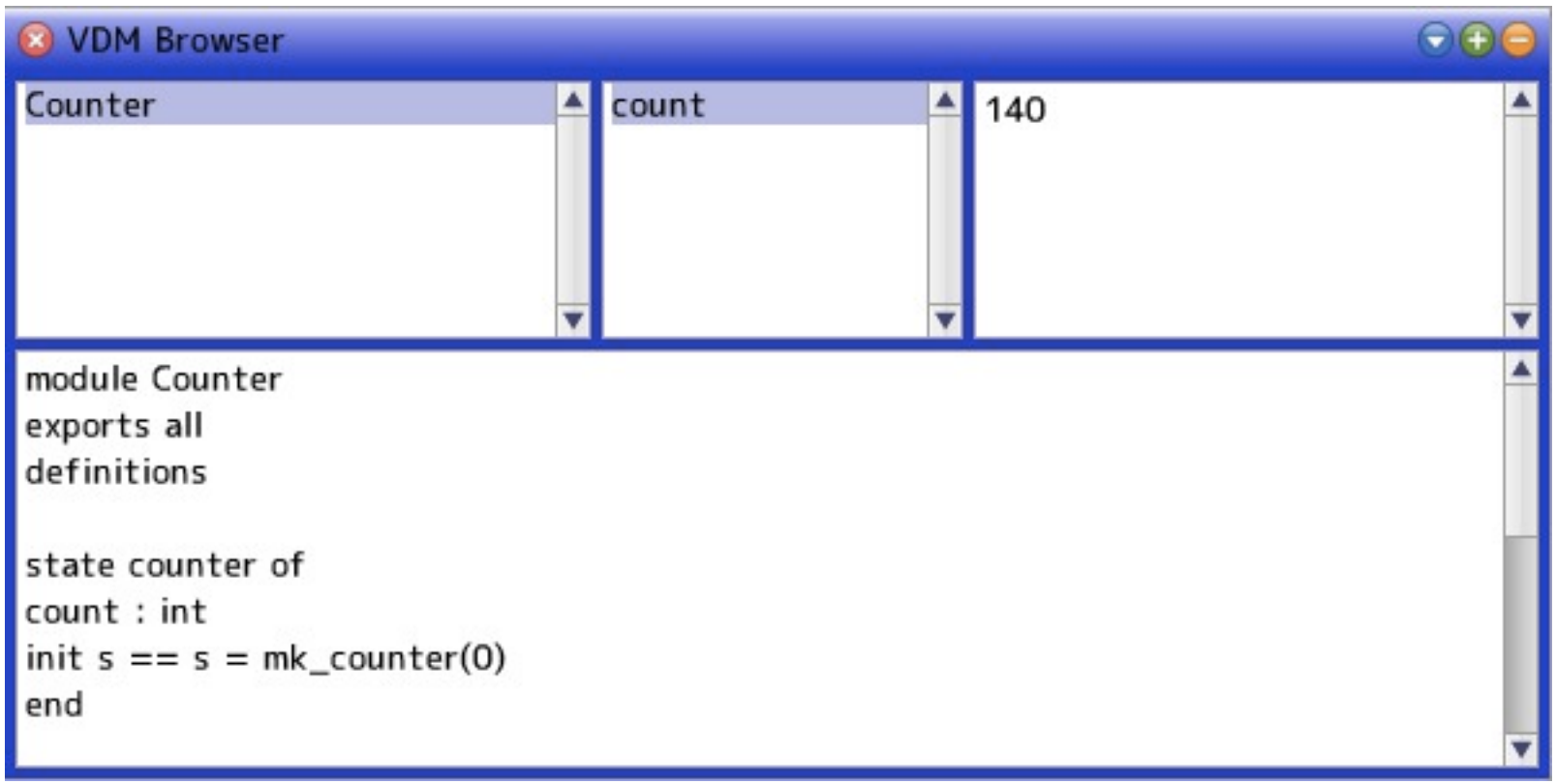
140を


VDM Browser
Counter
count
140
module Counter
exports all
definitions
state counter of
count : int
init s == s = mk_counter(0)
end

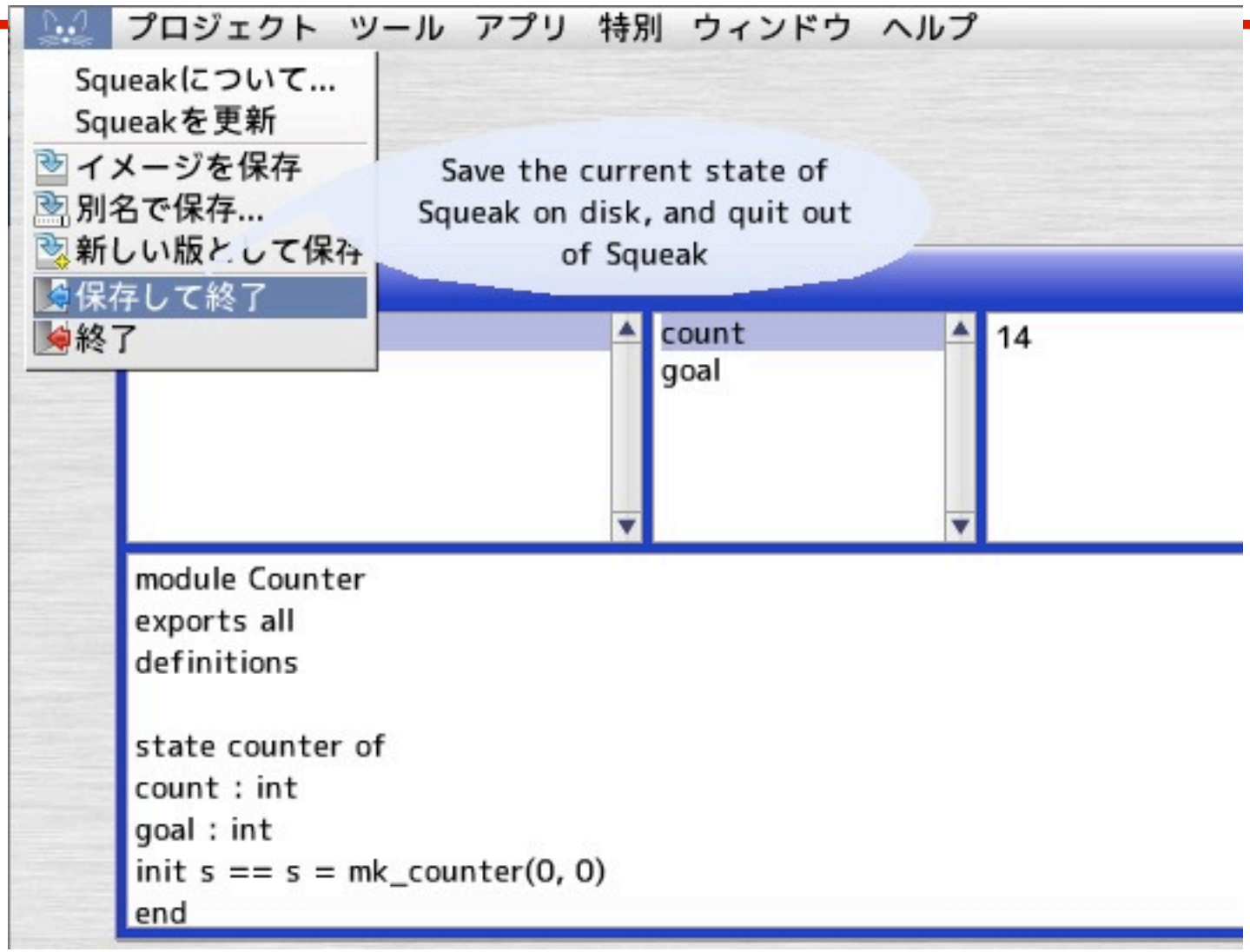
140を変数countの値に設定する



変数countの値が140になる

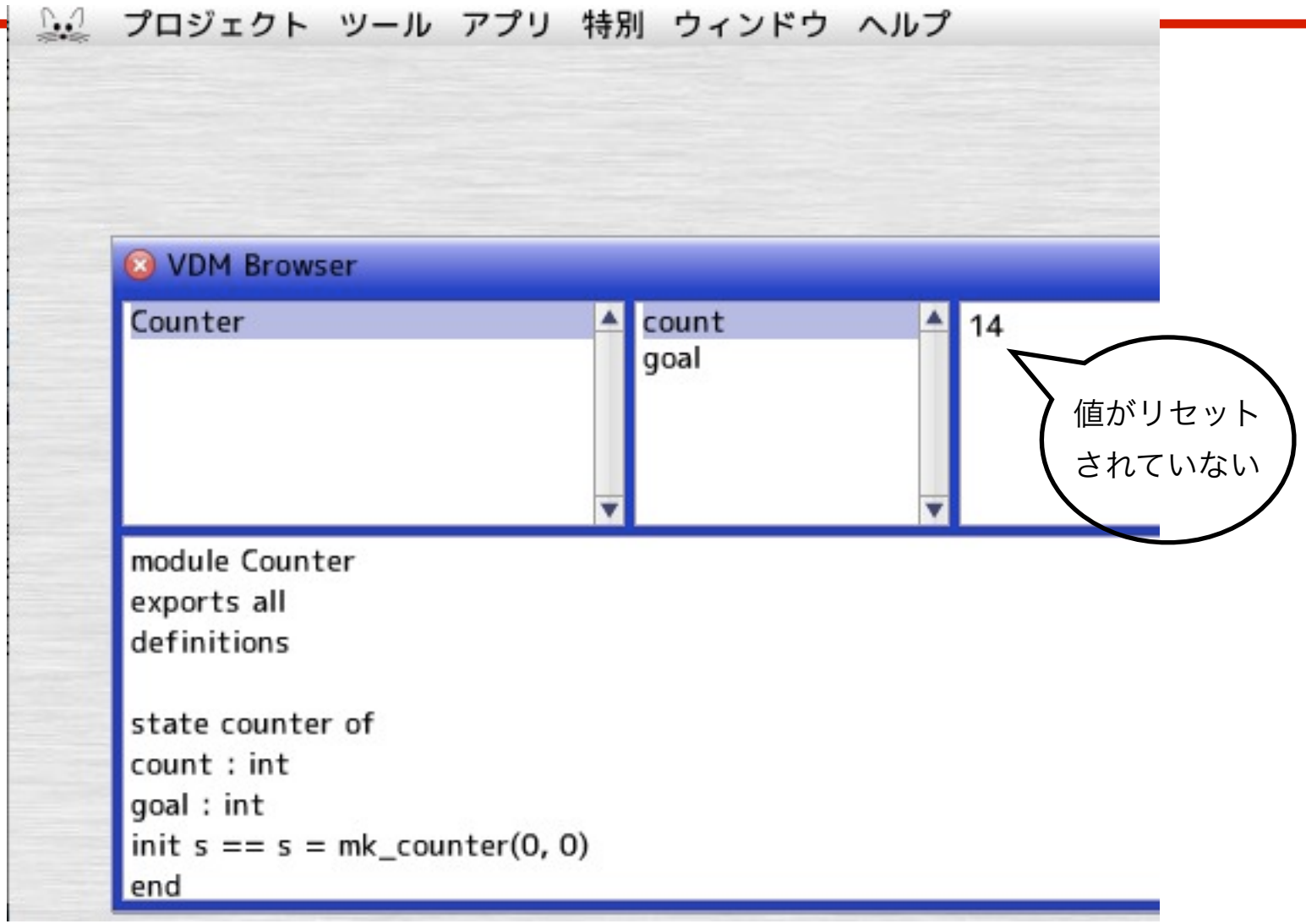


開発環境を「保存して終了」する



一旦、開発環境を終了した後で

もう一度開発環境を開くと
終了前の状態が回復する

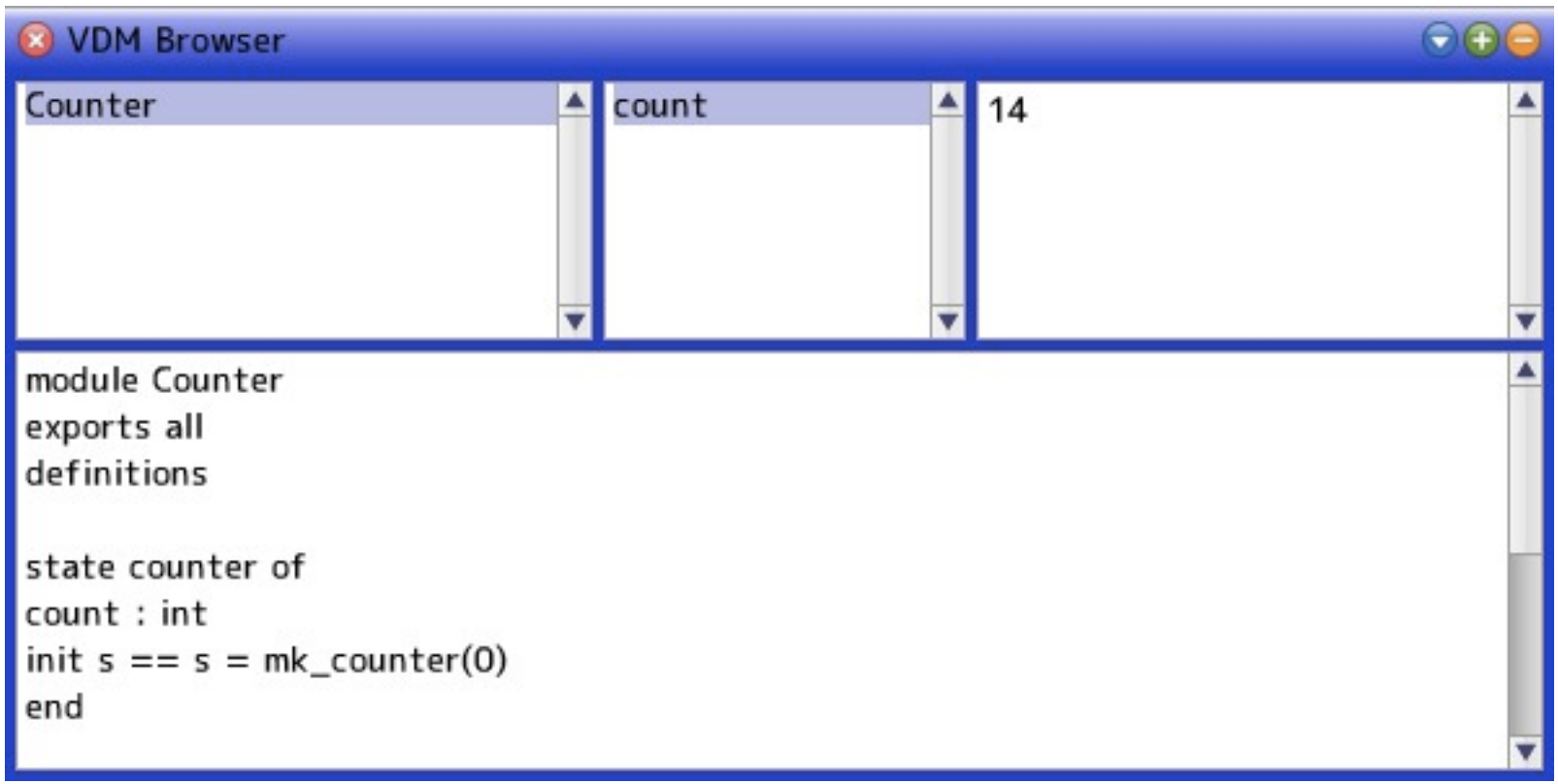


Liveness

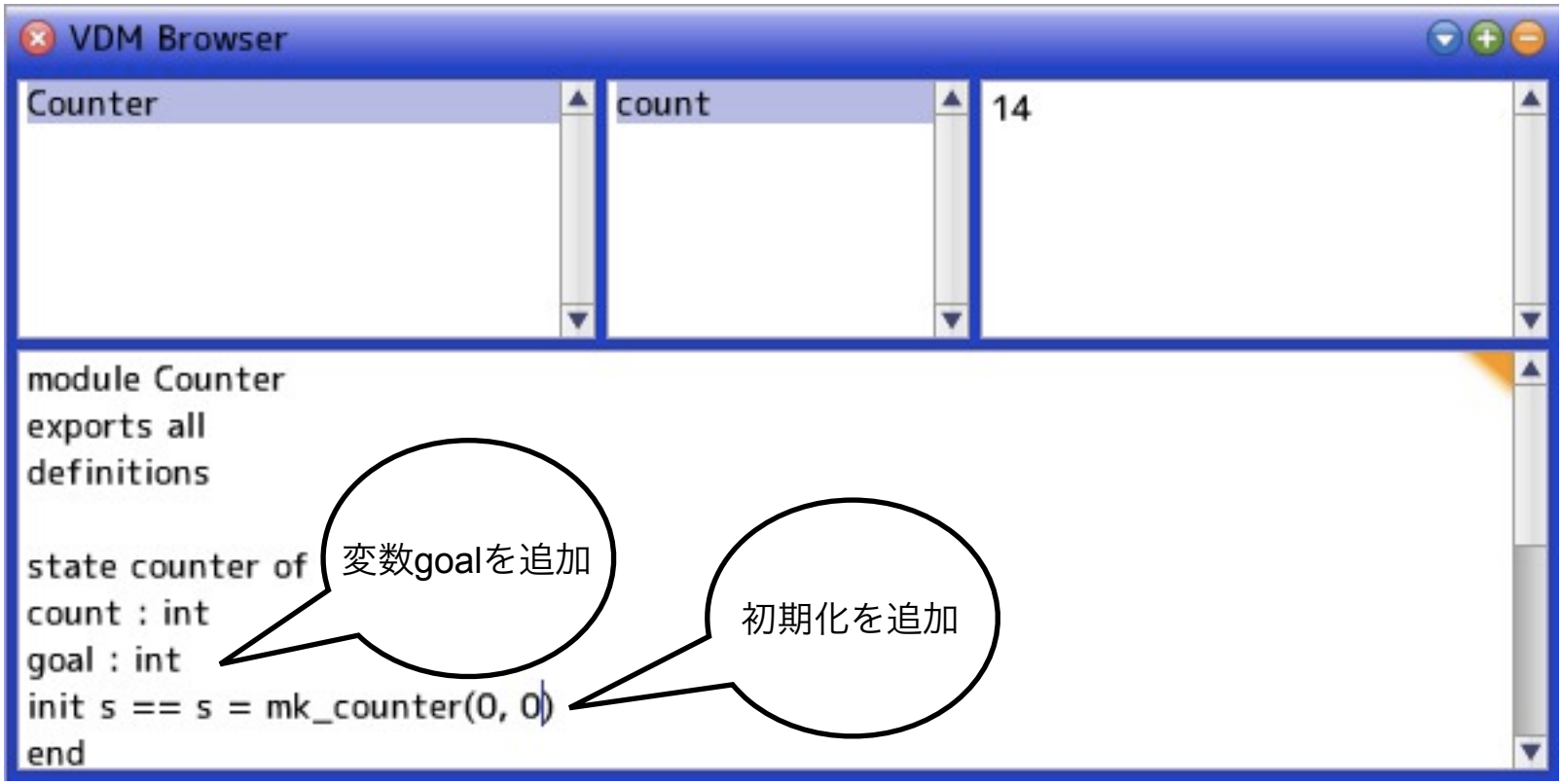
ライブプログラミング

実行中のプログラムのソースコードを
変更したら即座にその実行に反映される
プログラミング
= 脱 Edit-Compile-Run サイクル

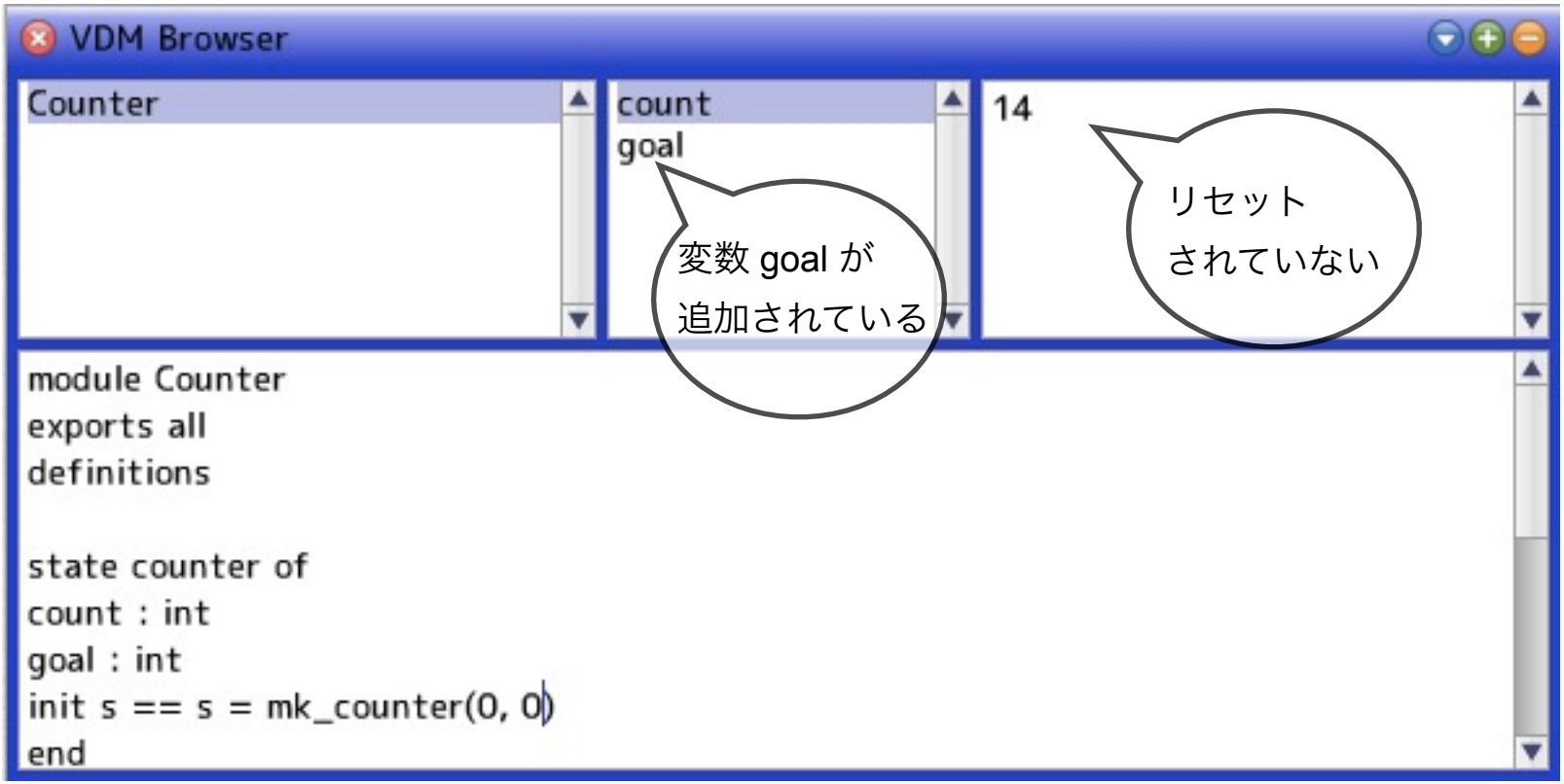
例：プロトタイプ実行中の仕様変更



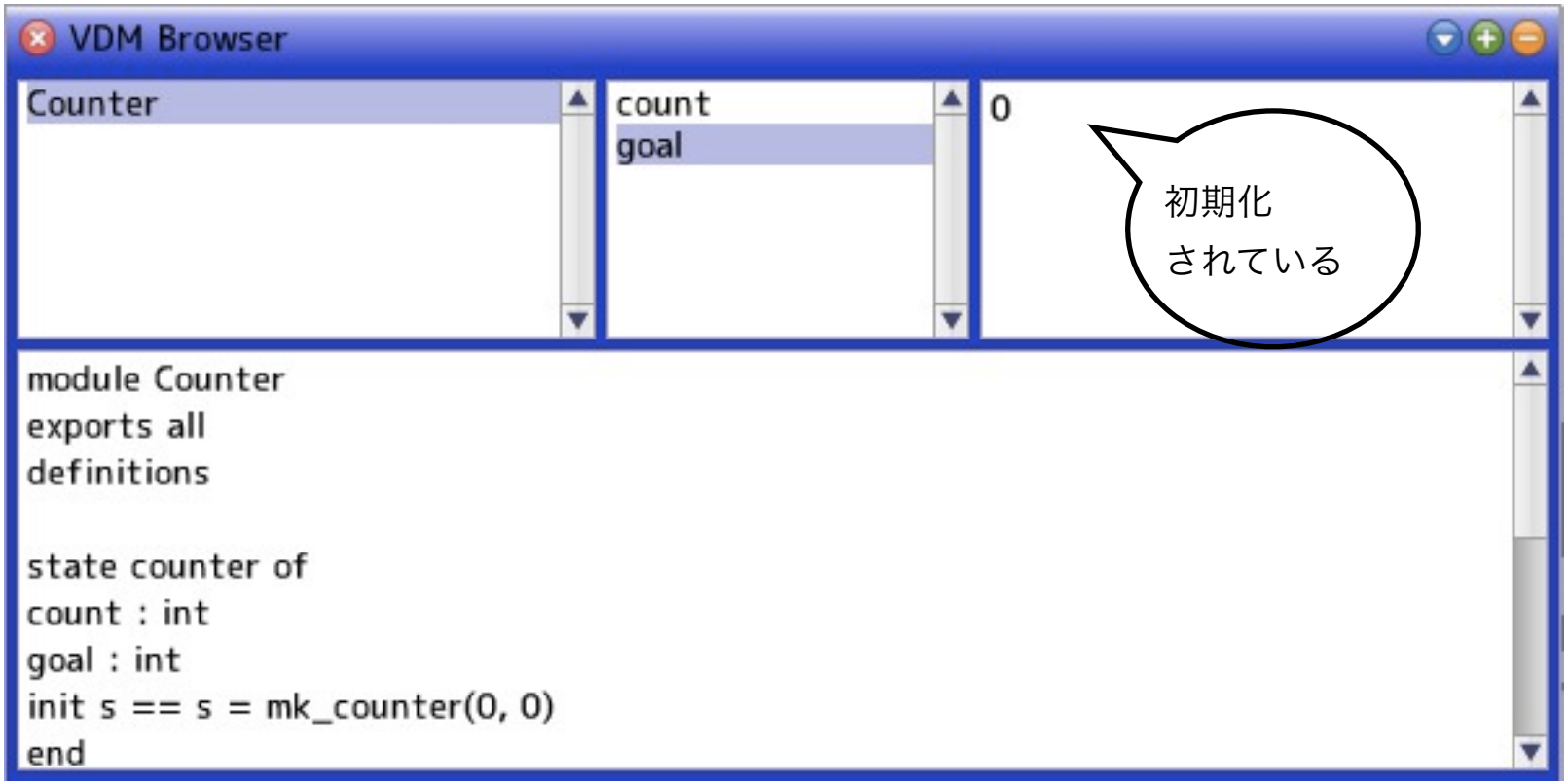
state変数としてgoal : intを追加



変数goalを追加しても
元の変数の値は維持されている

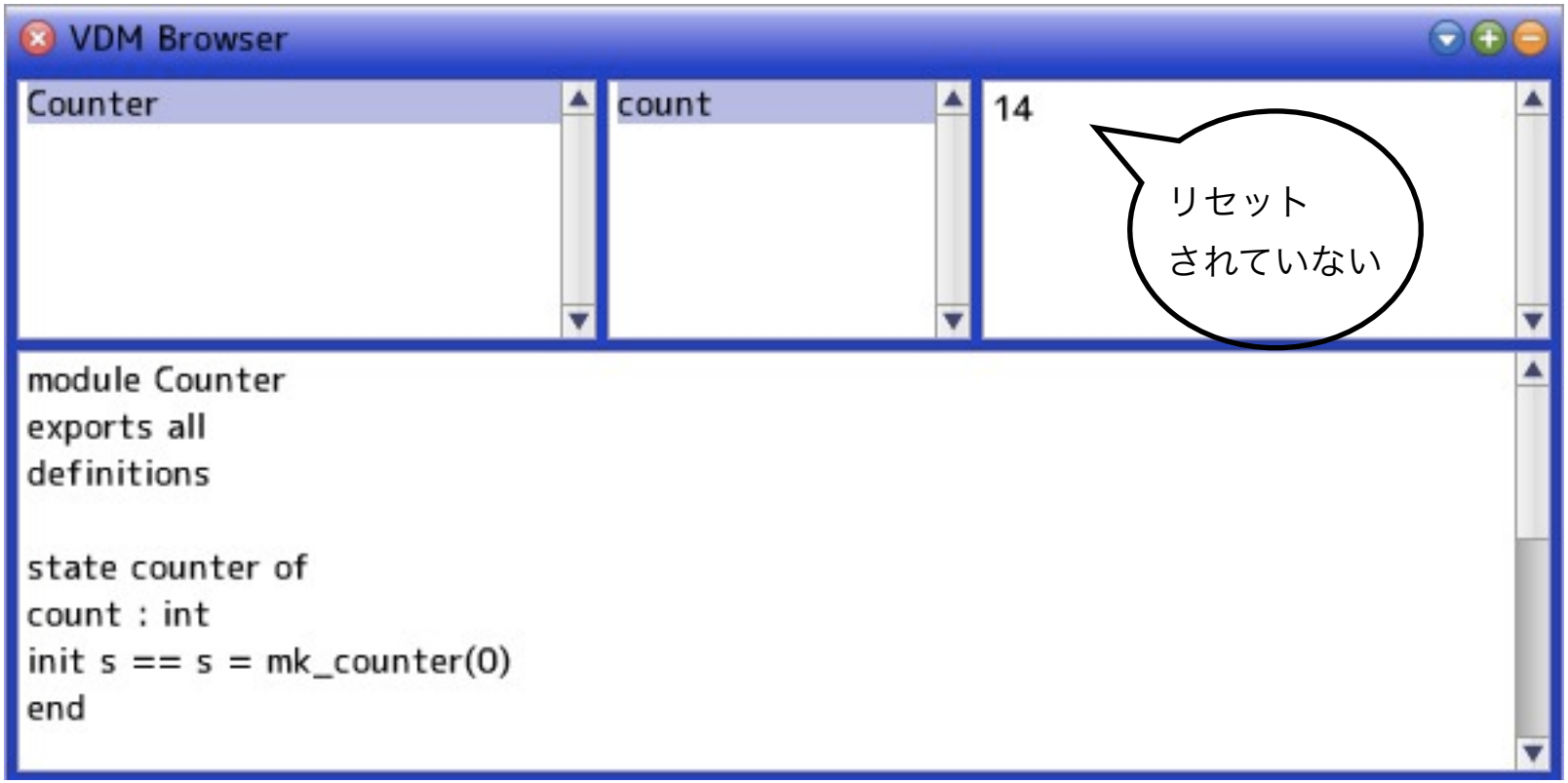


変数goalの値はinitの定義通り0



変数goalを削除しても

変数countの値は維持される



実行時改変と継続実行のしくみ

1. バックドアを使って全モジュールのstateを読み出す
2. 変更後の仕様をロードする
3. 変更後の仕様のバックドアを生成する
4. バックドア付きの仕様をロードする
5. バックドアを使って、全モジュールのstateに書き込む

Expression-based
Foreign Function Interface

自由度の高い言語間インターフェイス

通常の言語間インターフェイス(FFI)

- 関数名、引数の数と型、返り値の型

を前もって公開する

SOMETHINGitの表現式ベースのFFI

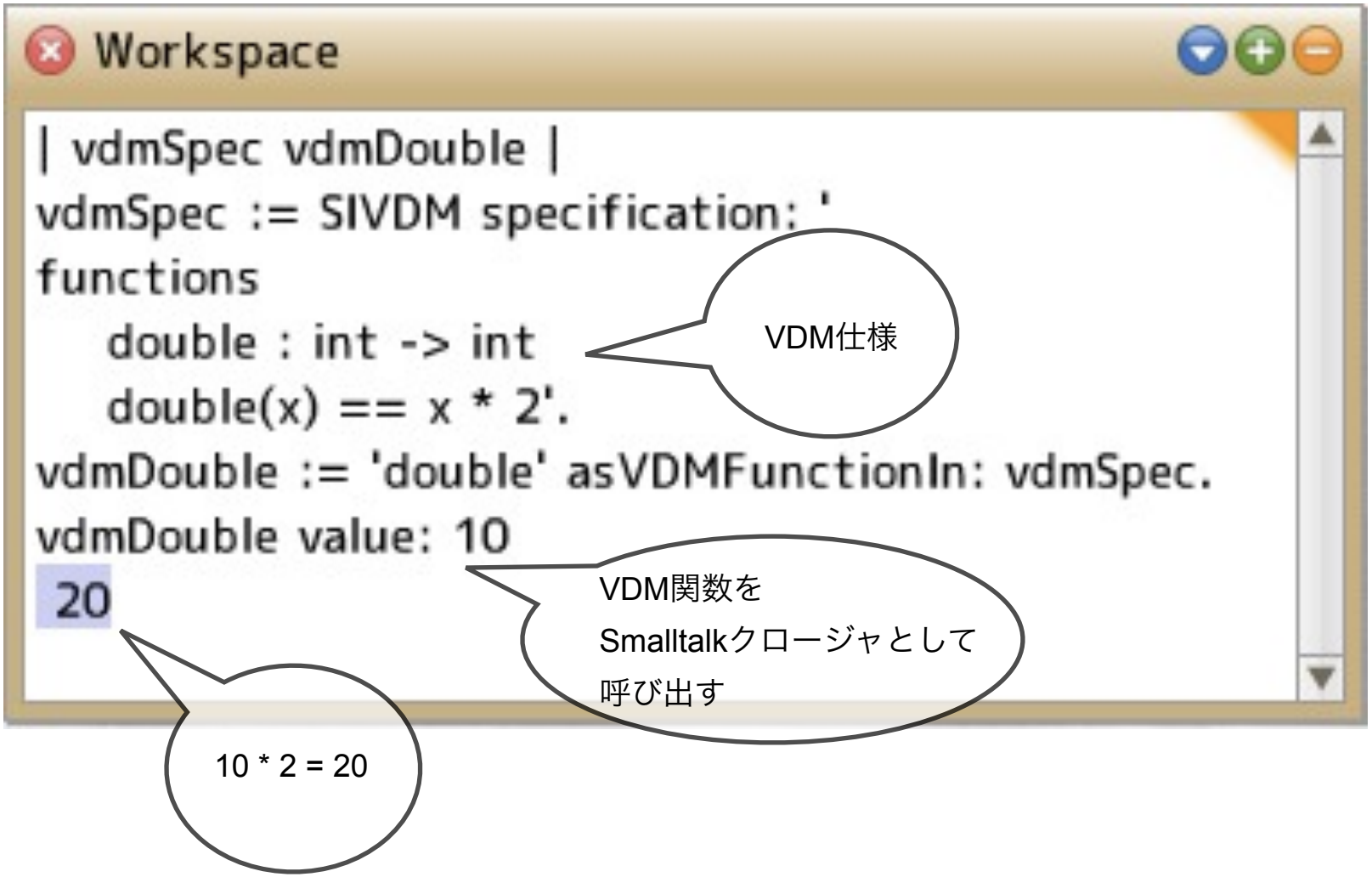
- 外部DSLとして任意の表現式を埋め込むことが可能
- ホスト言語側でゲスト言語の表現式を合成することで、単なる関数呼び出し以上の表現が可能

例：関数double

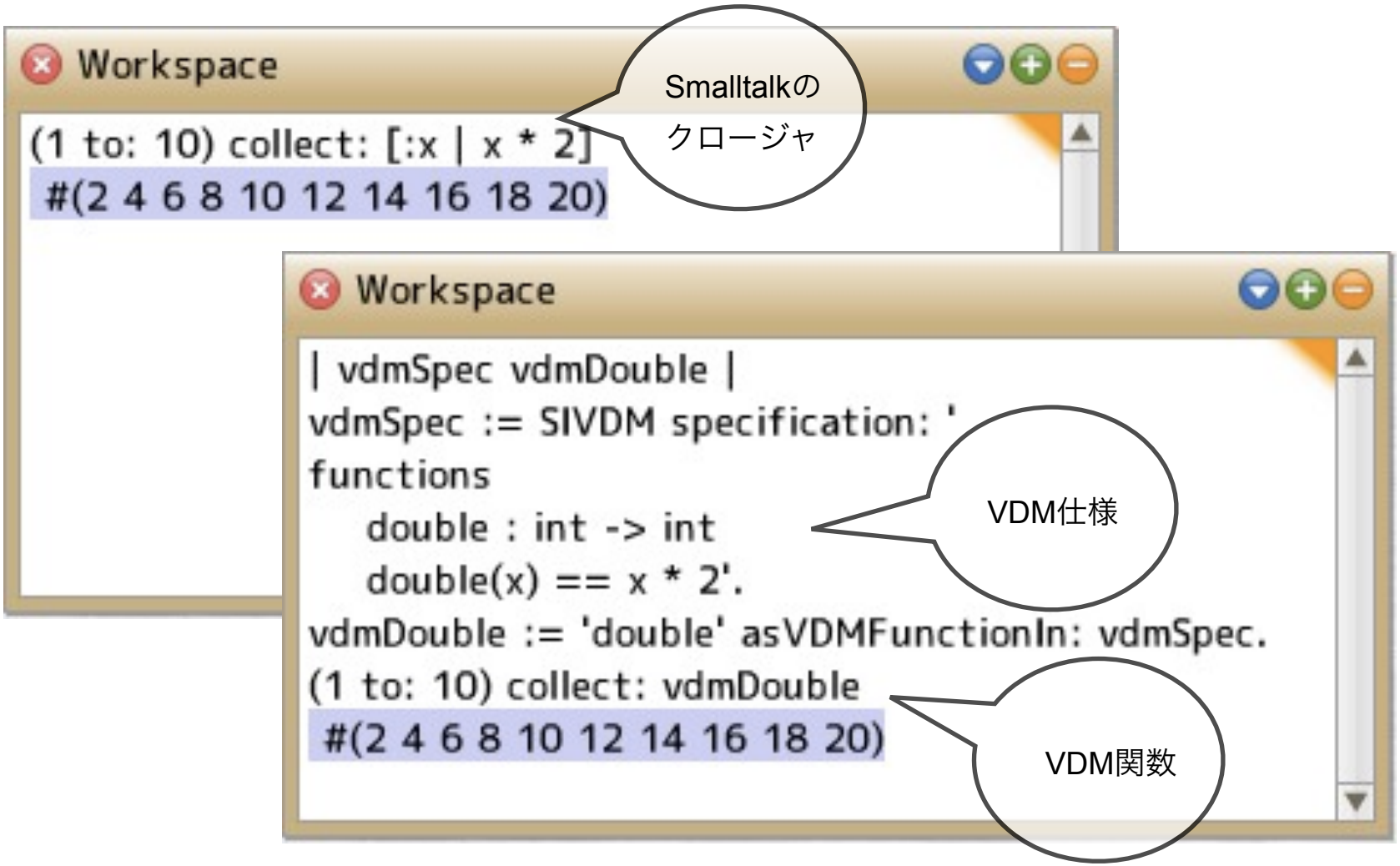
functions
    double : int -> int
    double(x) == x * 2

引数の２倍を返す関数

例：double(10)をSmalltalkで評価する

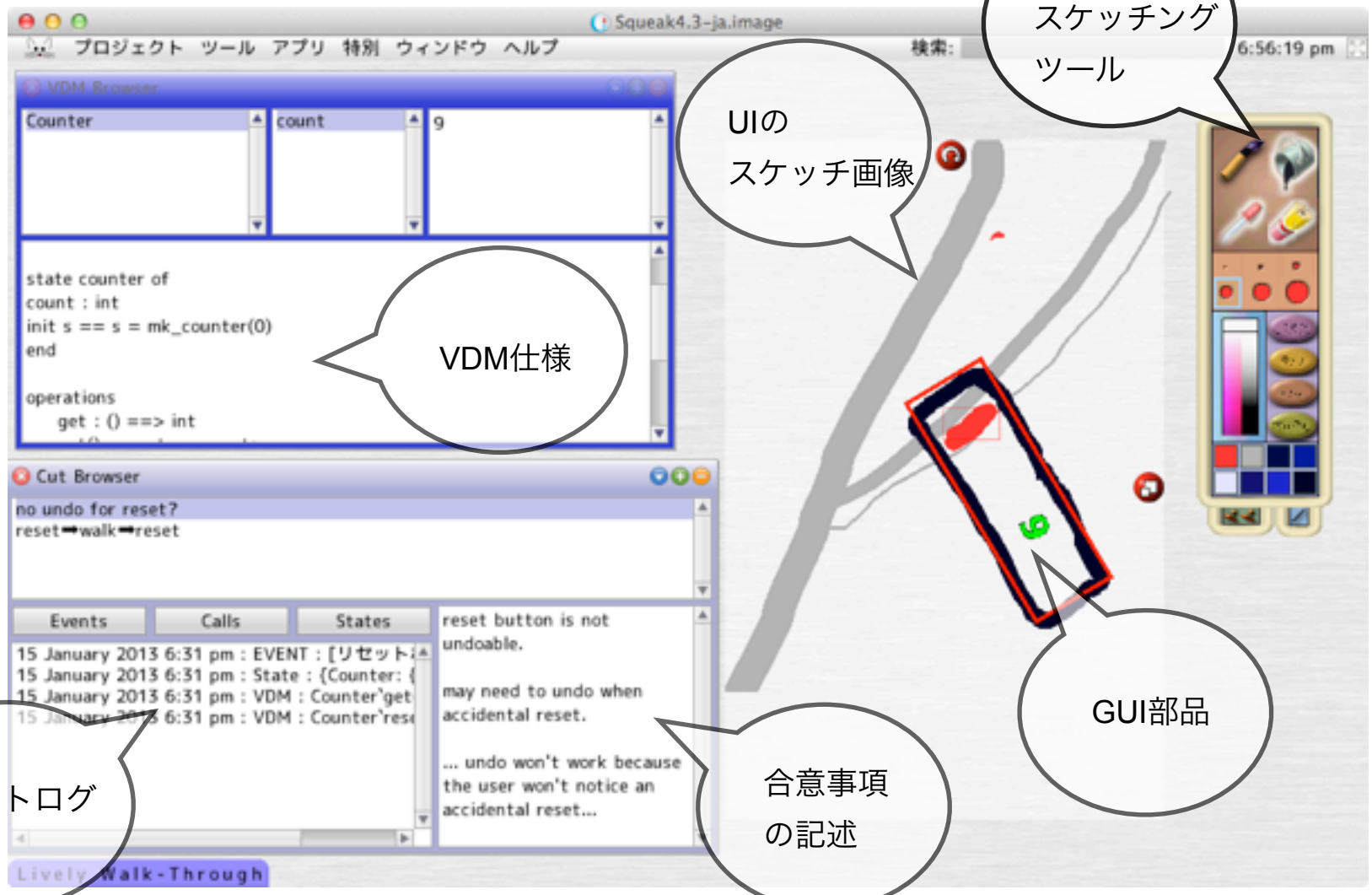


例：Smalltalkのクロージャとしての VDM double関数



応用：ライブUIプロトタイピング環境

Lively Walk-Through



Conclusion

まとめ

- 軽量形式手法とUIデザイン
- UIデザイナと形式仕様技術者の対話としてのライブUIプロトタイピング
- VDM-SLでライブプログラミング
  - 情報隠蔽を破るバックドア
  - 実行時改変と継続実行
  - 自由度の高いFFI

今後の課題

- さらに
  - 生き生きとした
  - 技術者やデザイナの間で会話が進む
  - 多様なUI形態のデザインに使える
    （タッチデバイス、ウェブUI、Kinect等）
  - 使っていて楽しい

  環境をめざします

Thank you.