教育環境におけるテストとデバッグフレームワーク

- Skeleton – etoys の為の動的スプレッドシート -

1. 背景

Skeleton を一言で簡単に説明すると、少し変なスプレッドシートという事が言えると思います。はじめテストやデバッグのツールとして開発を始めたのになぜスプレッドシートのような物が出来てしまったのかを背景として簡単にご説明します。

生徒達、初学者、そしてプログラミングを趣味とする人々、これらをオムニユーザと呼びましょう。オムニユーザの為のプログラミング言語としては、最初期の LOGO を筆頭に数々の試みがあります。また、オフィスソフト等に付属するマクロ言語も、職業プログラマでは無い人々によって使われる事を前提としている点で、同じ括りに入れる事が出来るでしょう。これらのプログラミング環境には、効率や堅牢性を求められるプロ用のツールとは別の性質が求められます。それは理解のしやすさであり、プログラミングが本来持つ楽しさです。

プログラミング環境において、言語の文法やライブラリの仕様は大変重要です。しかし別の視点でプログラミングを捉えた場合、むしろ開発に使われるツールの使い勝手にも大切な要素は多く含まれています。周辺ツールは言語の文法に比べてよりユーザのコンピュータ生活に密着した物であり、言語を使うときの『手触り感』に大きく影響します。その意味で、当初言語仕様などに比べて注目される事の少ない周辺ツールであるテストツールやデバッグツールに注目しました。

Alan Kayの近年の仕事である子供向けプログラミング環境 Squeak etoys を開発対象として作業を進める過程で、なぜデバッガやテストツールが必要になるのかという問題に突き当たりました。その一つの回答として二つのキーワードが浮かび上がります。一つは『宣言的』。もう一つは『具体的』です。デバッガによって動的な変数の値を具体的に捉え、テストによって処理の流れに宣言的な制約を付ける事が、これらツールの大きな役割になっています。

一方で、これらの問題に正面から取り組んだ試みとして、1980年代のAlan BorningによるThingLabを筆頭とする制約によるユーザインタフェースの構築という仕事がありました。オブジェクト間に制約と呼ばれる宣言的なルールを設定し、プロトタイプと呼ばれる具体的なオブジェクトの編集によってプログラミングを行うというのがThingLabのスタイルです。このアイデアはオムニユーザ･プログラミング環境にとって非常に魅力的な物と言えます。

Skeletonは、この制約というアイデアと、日ごろ私たちが日常的に使用するスプレッドシートの見た目をミックスさせた物です。実はスプレッドシートには、制約的な性質と具体的な性質のどちらもが備わっています。ルールは数式として関数的に記述され、セルには常に具体的な値が表示されます。これに、より動的な性質を付け加える事によって楽しいプログラミング環境を構築する事が出来ます。

2. 目的

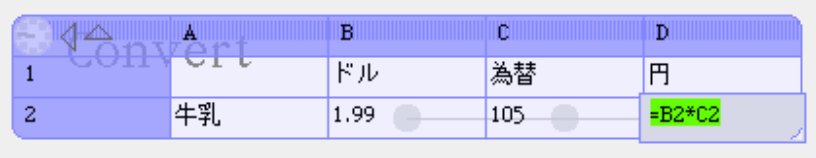
Skeletonはプログラム言語ではなく、言語と共に利用する小さなツールです。これで本格的なプログラムを作るわけには行きませんが、ちょっとした実験や日常の為の簡単なツールを直感的に作る事が出来ます。

見た目は貧弱なスプレッドシートですが、普通のスプレッドシートには出来ないことが二つあります。一つは双方向のルールを書くことが出来る事。スプレッドシートでは、通常一つのセルには値か式のどちらかしか書くことが出来ません。そこからの帰結として、問題のセルと答えのセルとの役割は固定されています。一方で、日常的なさまざまな現象では、しばしば問題と答えが入れ替わるでしょう。単価から全体の金額を求めたり、逆に全体から部分を求めたりします。Skeletonは双方向のルールによって、こういった問題を自然な形で記述する事が出来ます。ただし、逆方向のルールについては明示的に記述する必要があり、自動的に推論を行う事はしません。推論には困難な問題が伴うため、現在の実装ではそれを避けています。

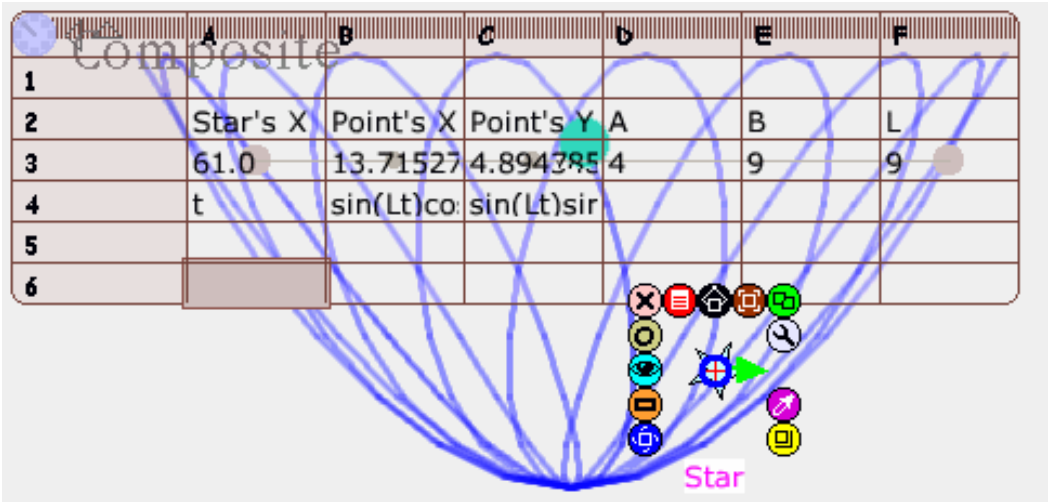
もう一つの特徴としては、このスプレッドシートが数や文字列に限らない、あらゆるオブジェクトを扱えるという事です。あるセルと図形オブジェクトの座標を関連付け、図形を数式に従って動かしたり、二つの図形の間にルールを設定し、幾何学的な証明をデモンストレーションするコンテンツを作成したりする事が可能です。セルは単に値を保持する入れ物として働くのではなく、getter, setter を伴うプロパティを通してシステムのあらゆるオブジェクトを安全に操作する事が出来ます。

Skeleton の名前の由来は、コンピュータを生物とみなした時の制約の役割から来ています。いわば、生物が骨と肉で形作られるようにプログラムを書くという例えです。肉は自由度があり力を生みますが、重力の元で形を維持する事は出来ません。骨はそれ自体動かない物ですが、生物にとって有用な形を作り、梃子の原理で肉の動きを助けます。さらに骨は化石となった後でも、生物学者はその動作を力学的に予想する事も出来ます。興味深い事に、動きを記述するのは実際に動作する肉ではなく、それを補完する骨の方なのです。

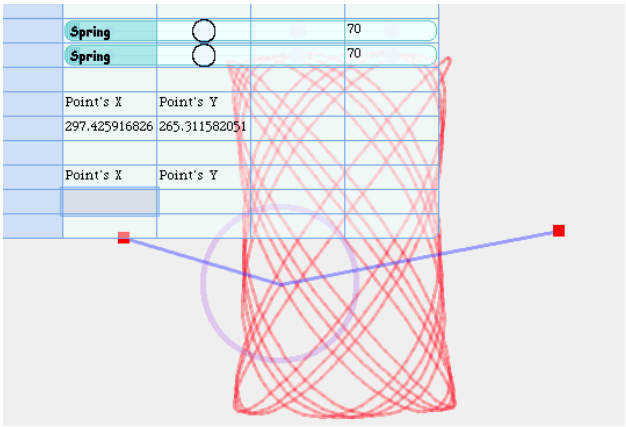
3. 開発の内容



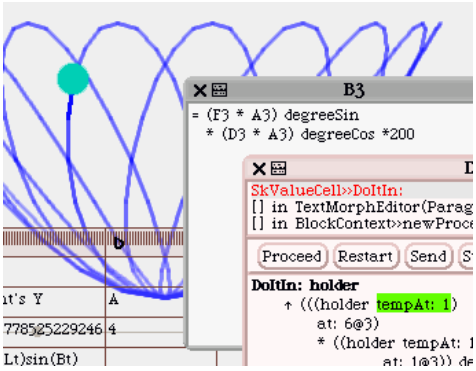
　見た目はスプレッドシートのような画面です。図のように、通常のスプレッドシートと同じようにセル参照や数式を使います。



　数式と図形を関連させると、このような幾何学的な図形を描く事が出来ます。



　組み込みのばねの動作をするモジュールを使って軌跡を描いています。値が変化しながら図形が描かれてゆく様をゆっくり見る事が出来ます。



Smalltalkの標準デバッガで数式をデバッグする事も出来ます。

4. 従来の技術（または機能）との相違

制約をとっても、スプレッドシートにしても、この分野には多くの独創的な研究があり、Skeletonによって初めて成し遂げられた成果はありません。先行研究に対するSkeletonの意味とは、1)制約とスプレッドシートを適度なバランスでまとめた事。2)Squeakという開かれたプラットフォームで他のアプリケーションと関連付けながら使える事。があるでしょう。特殊な環境を要求する他の多くの研究成果に対して、Skeletonは比較的シームレスに一般的な環境の中で利用する事が出来ます。

5. 期待される効果

プログラミングの面白さの一端がより身近になる事を期待します。

6. 普及（または活用）の見通し

まだリリース前の段階ですが、現在スペインの教育プロジェクトLinExで使いたいという申し出を受けています。

7. 開発者名（所属、e－mailアドレス）

山宮隆　propella@yuri.sakura.ne.jp

8. URL

http://languagegame.org:8080/propella/57