Mac開発者のための Smalltalk紹介

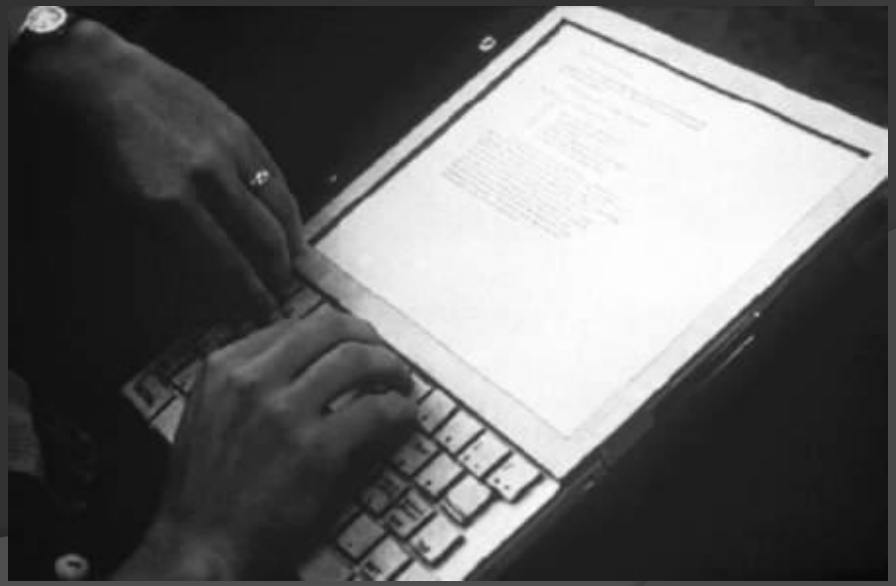
Masashi Umezawa

iPadとDynabook

- 似ている!

2010 iPad

1972 Dynabook



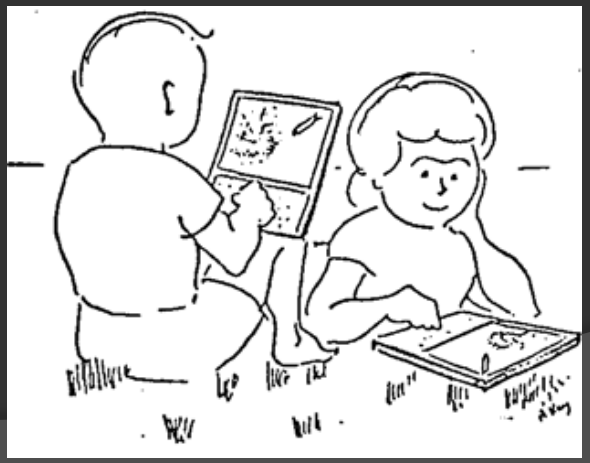
http://www.tomshardware.com/news/alan-kay-steve-jobs-ipad-iphone,10209.html

iPad・iPhoneがもたらしたもの

- コンピュータに対する敷居を劇的に下げた
  - 誰もがどこでも使える
  - 世界中の情報に直ちにアクセスできる
- 革命的なデバイス
  - iPad・iPhoneについてアラン・ケイのコメント
    - “Make the screen five inches by eight inches, and you’ll rule the world.” – Alan Kay
- 真のパーソナルコンピューティングに近づいた
  - 「使う」という点においては

Dynabookの目指したもの

- 子供でもプログラミングできるコンピュータ
  - “Personal Dynamic Media”
    - http://www.newmediareader.com/book_samples/nmr-26-kay.pdf
  - 実際に子供たちが「作って」いる!



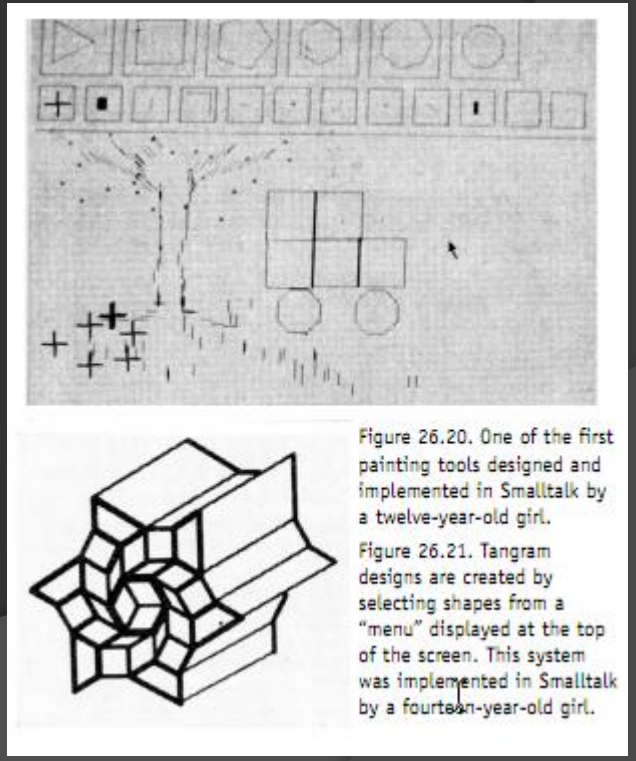


Figure 26.20. One of the first painting tools designed and implemented in Smalltalk by a twelve-year-old girl.

Figure 26.21. Tangram designs are created by selecting shapes from a "menu" displayed at the top of the screen. This system was implemented in Smalltalk by a fourteen-year-old girl.

Smalltalk

- Dynabookのためのプログラミング言語
  - すぐに習得できる簡潔な文法
    - 予約語は五個のみ
      - nil, true, false, self, super
  - 後から自由にシステムを拡張できる柔軟性
    - メッセージングモデルに基づく「動的」な
      オブジェクト指向言語

Objective-C (1)

- C とSmalltalkのハイブリッド言語
  - Objective-C = C + Smalltalk
    - 作者のBrad CoxはSmalltalkの解説書を見て開発
    - Cをオブジェクト指向拡張したのではなく、別の言語(Smalltalk)をCの中に同居できるようにした
      - インラインでSmalltalkの書けるC
    - オブジェクトにメッセージを送る箇所を[] でくくって書く

Objective-C (2)

- Objective-Cのメッセージ送信の例

```
Book *book = [Book new];
[book setPrice: 3000];
int price = [book price];
```

  - Smalltalkでは…

```
book := Book new.
book setPrice: 3000.
price := book price.
```

Objective-C (3)

- 文法だけでなく思想もSmalltalkに近い
  - 強制よりも自由
- C++やJavaなどに比べ、非常に動的な性質を持つ
  - セレクタによる間接的なメッセージ送信
    - performSelector:
      - Smalltalkではperform:
  - 動的なクラスの作成、メソッドの追加・すげ替えなどが可能
- 柔軟なMac OSプラットフォームの基礎となっている
  - 参考: “Dynamic Objective-C”
    - http://www.amazon.co.jp/dp/4861006414

Smalltalkはなぜ置き去られたか?

- 当時のハードウェア能力の限界
  - Smalltalkを動作させるには高価なワークステーション級のコンピュータが必要だった
    - Dynabookも模型にすぎない
- ハードウェアの制約が厳しい状況ではCが選択されるのも必然
  - Objective-Cもそうした制約下で誕生
- 今は??
  - iPad2の性能は1994年のスーパーコンピュータ並
  - http://www.tuaw.com/2011/05/09/ipad-2-would-have-bested-1990s-era-supercomputers/

今、Smalltalkは十分速い

- 例: 形態素解析のベンチマーク結果
  - IgoのJava, VisualWorks Smalltalk, Ruby版で比較
    - http://d.hatena.ne.jp/kaminami/20110509/p1

	Java	Smalltalk	Ruby
100行	4	10	2970
300行	14	35	19442
1000行	53	170	251818

※msec

- 洗練された高速なVM
  - 動的な言語の中ではもっとも高速な部類

今、Smalltalkはオープン

- フリーの処理系が多く存在する
  - Squeak
  - Pharo
  - VA Smalltalk
  - GNU Smalltalk
  - Dolphin Smalltalk
  - Smalltalk/X
  - VisualWorks NC版
- 多くの処理系はマルチプラットフォームで動作する
  - Mac OS X, Windows, Linux

強力なライブラリ群

- MVC
  - 1980年代からSmalltalkのUIに導入
- デザインパターンの元ネタ
  - Adapter, Template Method, Factory Method, Singleton, Flyweight…
- VisualWorksで2000を超える標準クラスライブラリ
  - Collection, Stream, File, Socket, Web, Database, etc.
- All-in-oneで必要なものが一通りそろっている
  - 言語・開発環境をあれこれ物色する必要がない
  - Smalltalk自身で自由に拡張できる

シンプルで習得しやすい文法

- Objective-C = C + Smalltalk ということは…
  - Smalltalk = Objective-C - C !
- Smalltalkを覚えるにはC部分を忘れればよい
  - 決まり事は最小限
    - すべてはオブジェクト(基本データ型を含め)
    - オブジェクトにメッセージを送る

配列の例 (Objective-C)

```
NSAutoreleasePool * pool = [[NSAutoreleasePool alloc] init];

NSArray *items = [NSArray arrayWithObjects:
@"orange",@"apple",@"pineapple",nil];
NSString *itemStr = [items objectAtIndex: 0];

NSMutableArray *fruits = [items mutableCopy];
[fruits addObject: @"kiwi"];

[fruits sortUsingComparator: ^(id flu1, id flu2){
  return [[NSNumber numberWithInt: [flu1 length]]
   compare: [NSNumber numberWithInt: [flu2 length]]];
}];

[pool drain];
```

auto release準備

配列作成

要素にアクセス

変更可能配列に

要素の追加

文字数でソート

オブジェクトの解放

配列の例 (Smalltalk)

```
NSAutoreleasePool * pool = [[NSAutoreleasePool alloc] init];

items := Array withAll: #('orange' 'apple' 'pineapple').
itemStr := items at: 1.

fruits := items asOrderedCollection.
fruits add: 'kiwi'.

fruits sort: [:flu1 :flu2 | flu1 size <= flu2 size].

[pool drain];
```

配列作成

要素にアクセス

追加可能コレクションに

要素の追加

文字数でソート

動的なメッセージングの例 (Objective-C)

```
Class targetClass = NSClassFromString(@"Target");

Target *target = [[targetClass alloc] init];

SEL sele = NSSelectorFromString(@"doSomething:");

if ([target respondsToSelector: sele]){

        [target performSelector: sele withObject: @"hello"];

}
```

文字列から
クラスを取得

文字列から
セレクタを取得

メッセージに答えられるか?

セレクタによる
間接的なメッセージ送信

動的なメッセージングの例 (Smalltalk)

```
targetClass := Smalltalk at: #Target.

target := targetClass new.

sele := #doSomething:.

(target respondsTo: sele) ifTrue: [
        target perform: sele with: 'hello'.
]
```

文字列から
クラスを取得

セレクタを用意

メッセージに答えられるか?

セレクタによる
間接的なメッセージ送信

ビジネスでもSmalltalk

- 実は事例も存在する
  - ビジネス用に特化したVisualWorksで特に多い
  - JP Morgan, AMD, FedEx, Texas Instrumentsなど
  - http://www.cincomsmalltalk.com/userblogs/cincom/blogView?content=successes
- 日本でも
  - 東洋ビジネスエンジニアリング
    - MC Frame (生産管理システム)
      - http://www.mcframe.com/

Macの開発者にとってのメリット

- Objective-Cを既に覚えている場合
  - Cとオブジェクト指向を行き来しないで良くなる
  - 動的なオブジェクト指向はそのままで、サーバアプリ開発やマルチプラットフォーム対応が行える
- WindowsからMac系のプラットフォームへと展開したいがObjective-Cに抵抗を感じている場合
  - 実はSmalltalkが楽!

VisualWorksを試してみよう(1)

- インストール
  - 基本インストーラを起動するのみ
- 起動
  - LaunchPadからプロジェクトを指定して起動
  - イメージファイルをVMにDrag & Dropでも良い

VisualWorksを試してみよう(2)

- ユーザインターフェース
  - 簡単なUIを付属のUI Painterで作ってみる
  - TO DO リスト
    - ApplicationModelを継承してMVCで作成
      - ツールによるスケルトンの自動生成
      - コールバックの実装で完成

VisualWorksを試してみよう(3)

- Webアプリケーション
- Seaside
  - 話題のWebアプリケーションフレームワーク
    - WebObjectsの再来
      - コンポーネント指向
  - 最近のSmalltalkの多くの事例もSeasideから
    - プランニング
      - http://yesplan.be/
    - CMS
      - http://www.cmsbox.com/en/cms

Seasideの例

- TO DO リスト
  - WAComponentを継承して作成 (VC)
    - レンダリングとコールバックを定義

```
MyToDoWeb >> renderContentOn: html
 html form: [
      html multiSelect list: self items.
      html textInput callback: [:v | self addItem: v].
      html submitButton with: '追加'.
   ]
```

```
MyToDoWeb >> addItem: aString
  self items add: aString
```

Seasideの例(2)

- 数当てゲームを10行程度で
  - Webらしからぬ「自然な」コード
  - ページ遷移を意識しない

```
MyNumberGuessGame >> go
   | counter random answer |
   counter := 0.
   random := (1 to: 10) atRandom.
   [answer := (self request: 'Guess number?') asNumber.
   answer < random ifTrue: [self inform: 'Smaller'].
   answer > random ifTrue: [self inform: 'Bigger'].
   counter := counter + 1.
   answer = random] whileFalse.
   self inform: (‘Got answer in <1p> times' expandMacrosWith: counter)
```

コミュニティの紹介

- Smalltalk-users.jp
  - http://smalltalk-users.jp
  - 勉強会を毎月開催
- Smalltalkers'メーリングリスト
  - http://www.smalltalk.jp/SML/index.html
  - Smalltalk全般に関する老舗のML
- Squeak-jaメーリングリスト
  - http://www.smalltalk.jp/mailman/listinfo/squeak-ja
  - Squeakに特化したML

まとめ

- Mac OS Xネイティブの力を存分に発揮したいのであればObjective-C
- 開発効率、マルチプラットフォーム展開、サーバアプリ開発等でSmalltalkも選択肢となる