Développement à l'Aide de Langages Dynamiques : Smalltalk

Nada Ayad, Damien Cassou et Annabelle Souc

module IT308: Méthodologies et outils logiciels

Présentation des Langages Dynamiques : Typage Dynamique

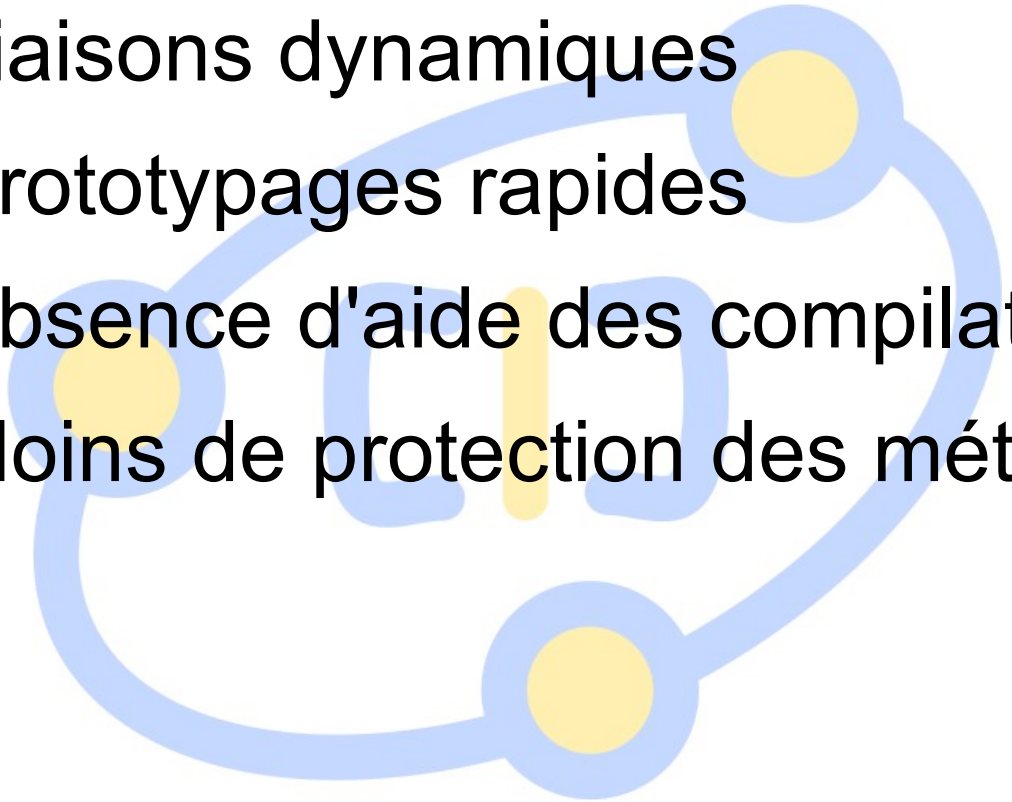
- Variables non typées
- Valeurs typées
- Pas de vérification de type par les compilateurs
- Ruby, Perl, Python...

Ruby
A Programmer's Best Friend

python™

Spécificités des Langages Dynamiques

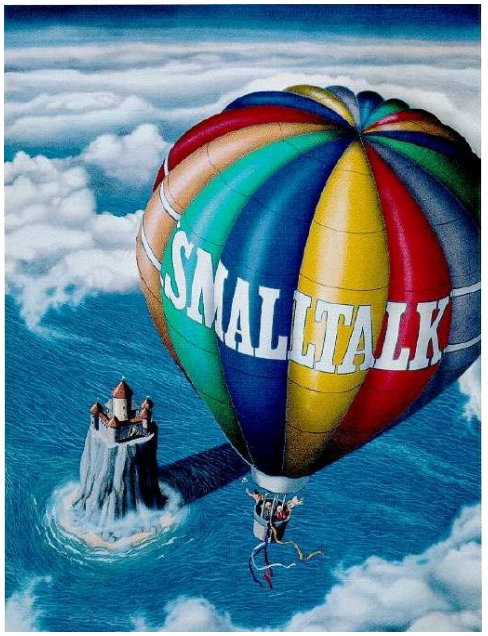
- Liaisons dynamiques
- Prototypages rapides
- Absence d'aide des compilateurs
- Moins de protection des méthodes



Smalltalk

- Typage dynamique
- Tout est objet : même les entiers et les classes
- Tout est modifiable : classes ouvertes etc.
- Tout est cohérent : pas d'exception à la règle
- Multi plates-formes (machine virtuelle)
- Garbage collector
- Closures (blocs)

Smalltalk



- Héritage simple
- Classes publiques
- Méthodes publiques
- Variables d'instances protégées
- Pas d'interface mais classes abstraites
- Valeur de retour par défaut : l'objet courant

Smalltalk : Historique

- 1972 : Première version inspirée par Simula
- 1976 : Nouvelle conception :
  - Hiérarchie des classes avec racine unique
  - Syntaxe du byte-code fixée
  - Sémaphores
- 1978 : Expérimentation avec les 8086
- 1980 : Norme Smalltalk-80

Smalltalk : Syntaxe

- Nombre limité de mots réservés :
  - **`self, super, true, false, nil, thisContext`**
  - **`52 en Java, 38 en Ruby, 76 en C#`**
- Nommage des méthodes :

```
envoie: unMessage a: unePersonne
```

- Appel de méthodes :

```
adam := Personne new.
eve := Personne new.
adam envoie: 'Fallait pas' a: eve
```

Smalltalk : Syntaxe

- Pas de constructeur :
  - seulement des méthodes qui retournent de nouvelles instances

```
Personne class>>nom: uneChaine
   |personne|
   personne := self new.
   personne nom: uneChaine.
   ^ personne


eve := Personne nom: 'Eve'.
```

Smalltalk : Syntaxe

- Utilisation de fonctions anonymes :
```
salaries select: [:salarie | salarie estFemme]
```
- Pas d'instruction de contrôle (if, while, etc.) :
```
(eve messageRecu) ifTrue: [eve senVeut]
```
- Création d'une classe :
```
Object subclass: #Personne
   instanceVariableNames: 'prenom nom genre'
   classVariableNames: ''
   poolDictionaries: ''
   category: 'Annuaire-modele'
```

Squeak

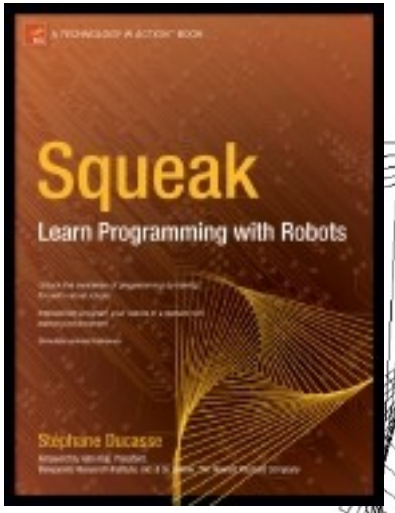
- Implémentation de Smalltalk
- Environnement de développement
- Pas de fichiers sources
- Machine virtuelle implémentée en Smalltalk
- Dernière version stable : décembre 2006

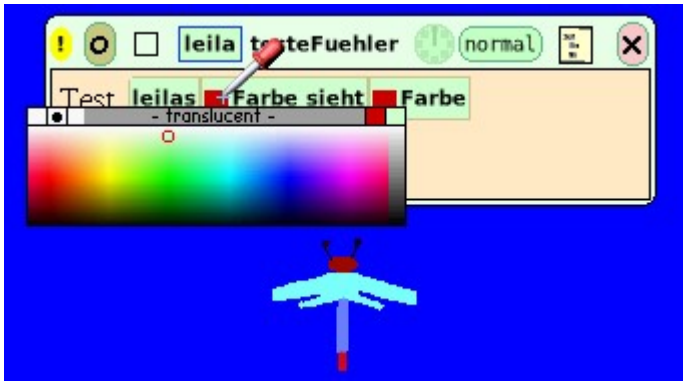
Squeak : Installation

- Téléchargement de la machine virtuelle : 1 exécutable (www.squeak.org)
- Téléchargement d'une image : 2 fichiers
- Drag&Drop de l'image sur la machine virtuelle
- C'est installé !

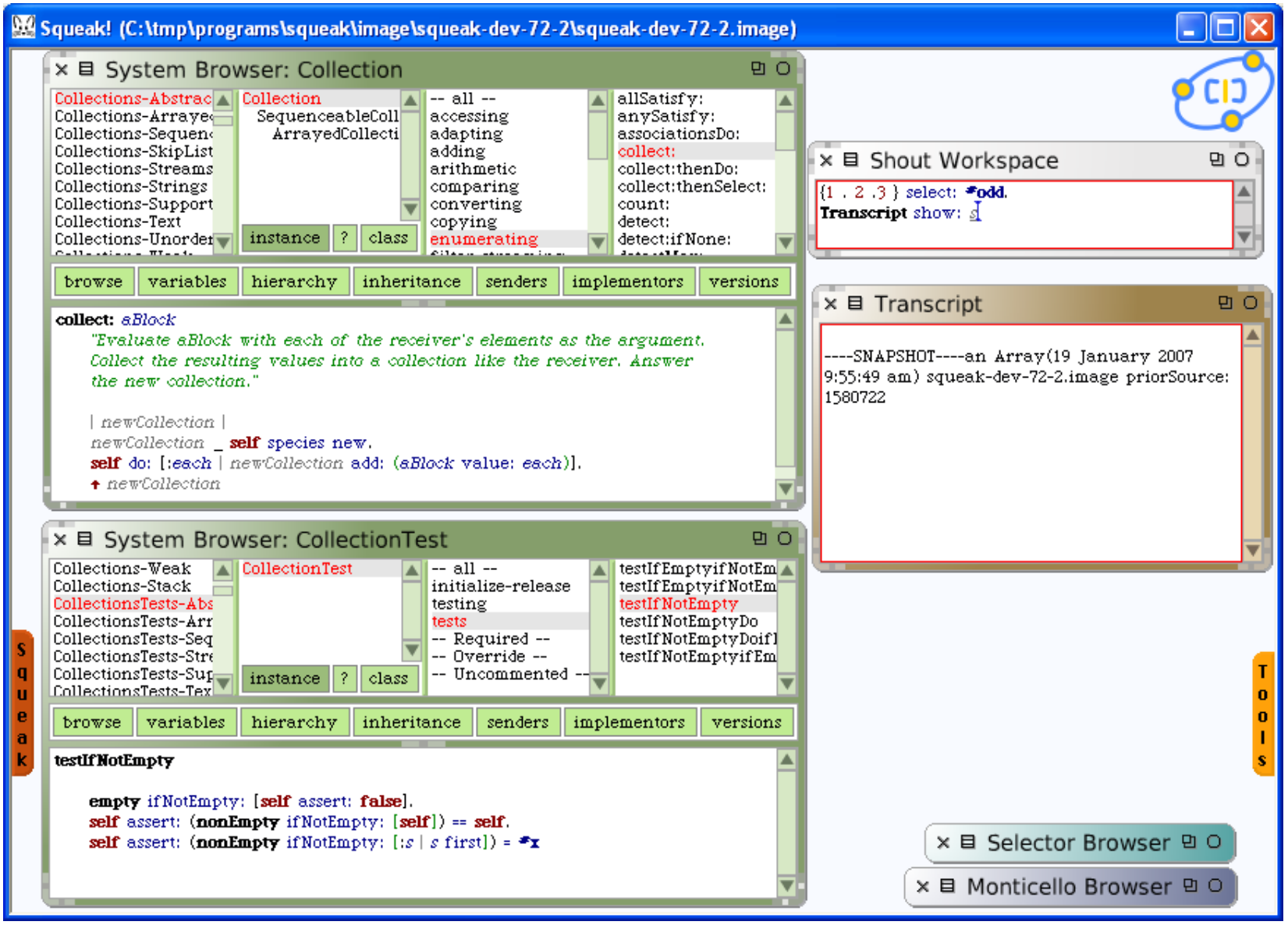
Utilisation 1: Apprentissage de la Programmation

- Créé à l'origine pour les enfants
- Nombreux projets
- Projet « One Laptop per Child »

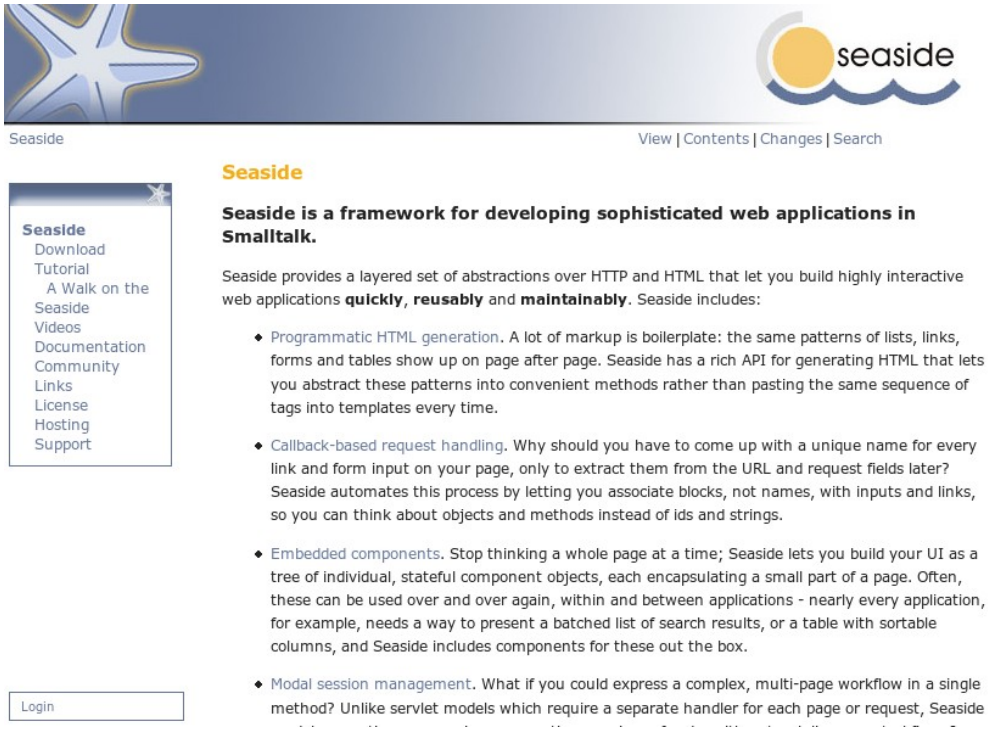




Utilisation 2: Environnement de Développement



Utilisation 3: Développement web



- Seaside
  - Orientation composant
  - Abstractions (http, html, javascript...)
- Magritte
  - Génération de formulaires
  - Méta modélisation

Fonctionnalités de l'Environnement de Développement

- Interface sous forme d'un bureau
- Navigateur de code
  - Découpage en paquetages et catégories
  - Code source ouvert et disponible
  - Compilation incrémentale
- Outils
  - Outils de recherche de méthodes
  - Exécution du code et inspections rapides
  - Navigateur pour les tests...

Et Maintenant

- Une démonstration !

Conclusion

- Conclusion
  - Langages à typage dynamique
  - Smalltalk
  - Squeak
- Points de vue personnels