

# Le langage de script Tcl/Tk

Matthieu Herrb

The logo for LAAS-CNRS, featuring the text "LAAS-CNRS" in a blue, sans-serif font. Above the text is a thick red horizontal line, and below it is a thick yellow horizontal line.

LAAS-CNRS

Compil, 25 novembre 2010

<http://homepages.laas.fr/matthieu/talks/compil10-tcltk.pdf>

# Agenda

- 1 Introduction
- 2 Tcl
- 3 Tk
- 4 Exemple d'utilisation au LAAS
- 5 Conclusion

- Initialement développé pour UNIX à l'Université de Berkeley par le Dr. John Ousterhout (1988),
- Le langage est maintenant soutenu par la société ActiveState,
- License libre, open-source : Tcl peut être librement modifié et intégré dans des applications commerciales,
- Dernière version : 8.5.9.

**Prononciation** : [tisil], on entend aussi « tickle ».

# Agenda

- 1 Introduction
- 2 Tcl**
- 3 Tk
- 4 Exemple d'utilisation au LAAS
- 5 Conclusion

# Caractéristiques

- Langage interprété/compilé à la volée
- multi-plateformes : Unix/Linux, Windows, MacOS, MacOS X
- Syntaxe très simple (mais elle peut dérouter)
- Variables typées dynamiquement, conversions automatiques
  - Chaînes UTF-8
  - Entiers précision infinie
  - Flottants (double)
  - Données binaires (images,...)
  - Listes, tableaux associatifs
- Visibilité locale des variables par défaut
- Peut s'interfacer simplement avec C/C++/...
- Nombreuses extensions

# Syntaxe

- `$foo`  
remplacé par la valeur de la variable correspondante.
- `[clock seconds]`  
commande exécutée et remplacée par le résultat de l'exécution.
- `"quelque chose"`  
rassemble plusieurs mots en un seul. Les dollars et les crochets sont interprétés à l'intérieur.
- `{quelque chose}`  
rassemble plusieurs mots en un seul. Aucune interprétation des caractères spéciaux.
- `\`  
introduit des caractères spéciaux (ex. `\n`) et supprime l'interprétation du dollar et du retour à la ligne

- `cmd arg arg...`

une commande Tcl est composée de mots séparés par des espaces. Le premier mot est le nom de la commande et les autres sont les arguments.

- **Pas d'autre grammaire.**

# Groupement et substitution

L'analyseur syntaxique de TCL procède en 3 étapes :

- 1 **groupement des arguments** sur la base des espaces entre les mots et des groupements à l'aide de quotes et d'accolades. Une commande est terminée par un retour à la ligne ou un point-virgule.
- 2 **substitutions** des valeurs des variables et des résultats de commandes entre crochets
- 3 **appel des commandes** en utilisant le premier mot de chaque commande comme index dans la table des procédures connues (intégrées ou définies par l'utilisateur en C ou en Tcl).



# Expressions : `expr`

- À la mode C (int et doubles)
- Substitution de commandes et de variables à l'intérieur des expressions
- Évaluées par `expr`, `if`, `while`, `for`.

```
set b 5
```

```
→ 5
```

```
expr ($b*4) - 3
```

```
→ 17
```

```
expr $b <= 2
```

```
→ 0
```

```
expr 6 * cos(2*$b)
```

```
→ -5.03443
```

## Exemple: Fibonacci

```
set fibvalues(0) 1
set fibvalues(1) 1

proc fib {n} {
    global fibvalues

    if [info exists fibvalues($n)] {
        return $fibvalues($n)
    }
    return [set fibvalues($n) \
        [expr [fib [expr $n - 1]] + [fib [expr $n - 2]]]]
}
```

# Agenda

- 1 Introduction
- 2 Tcl
- 3 Tk**
- 4 Exemple d'utilisation au LAAS
- 5 Conclusion

# Un toolkit multi-plateformes

- X11, Windows, Carbon, Cocoa
- Look natif
- basé sur Tcl
- nombreux widgets standards: menus, boutons, sliders,...
- quelques widgets puissants: canvas, text,...
- extensible et nombreuses extensions (BWwidgets,...)

# Hello Tk

```
package require Tk

button .hello -text "Hello, world" -command {
    puts "Hello, world"
    destroy .
}

pack .hello
```

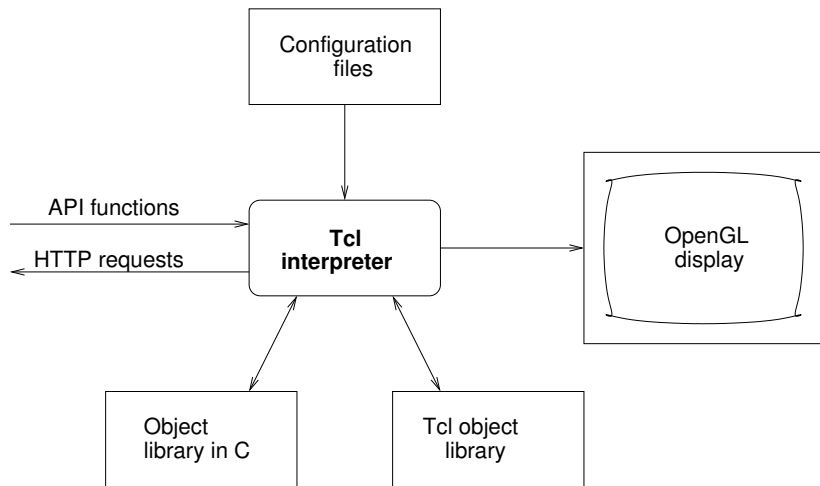
# Agenda

- 1 Introduction
- 2 Tcl
- 3 Tk
- 4 Exemple d'utilisation au LAAS**
- 5 Conclusion

# Gdhe - Graphic Display for Hilare Experiments

- Gdhe est un élément de l'architecture LAAS pour la robotique (<http://www.openrobots.org/>)
- Outil de visualisation 3D
- Basé sur OpenGL et Tcl/Tk
- Bibliothèque d'objets (robots) pré-définis
- Extensible en C/C++ ou Tcl/Tk
- Interface avec les modules Genom
- Open source, Licence BSD :  
<http://gdhe.openrobots.org/>

# Architecture



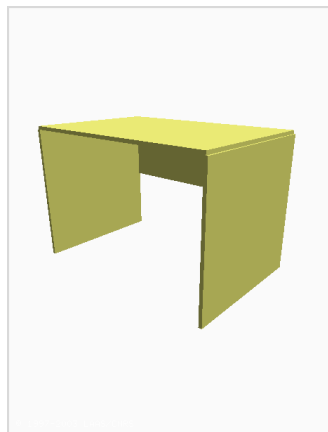


# Exemple de modèle d'objet

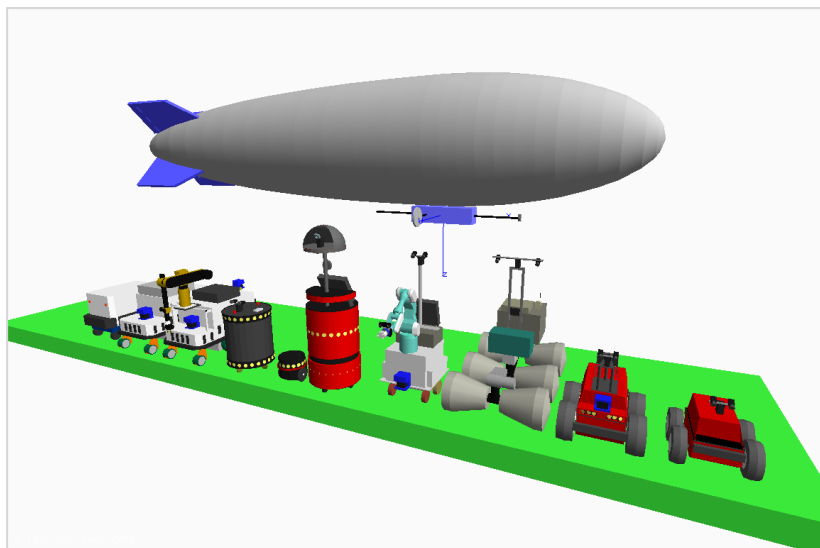
Un objet est défini par une procédure qui le dessine.

Exemple:

```
proc desk {} {  
  pushMatrix  
  translate 0.6 0.4 0  
  color 200 200 100  
  # Upper plane  
  box 0 0 0.8 1.20 0.8 0.02  
  # sides  
  box -0.6 0 0 0.02 0.8 0.8  
  box 0.6 0 0 0.02 0.8 0.8  
  # bottom  
  box 0 0.4 0.4 1.20 0.02 0.4  
  popMatrix  
}
```



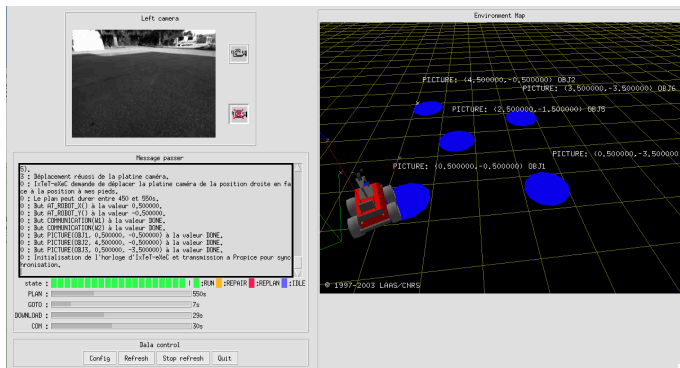
# Galerie d'objets



# Graphical User Interfaces

L'interface utilisateur est programmable (Tk).

→ embarquement de la fenêtre Gdhe dans une interface plus générale



# Récupération de données XML par HTTP

**picoweb**: petit serveur HTTP dédié aux modules Genom

- écoute sur le port 8080
- scripts CGI standard générés par Genom
- lit les posters et produit des structures XML
- plugins pour les posters dynamiques ou des structures spécifiques (images)

**http** : package standard Tcl client HTTP

**tdom** : package standard Tcl parser XML/DOM

→ Scripts Tcl pour mettre à jour la scène à partir des données des posters

# Agenda

- 1 Introduction
- 2 Tcl
- 3 Tk
- 4 Exemple d'utilisation au LAAS
- 5 Conclusion**

# Conclusion

- Tcl est un langage riche et puissant tout en restant peu gourmand en ressources.
- Largement répandu
- Permet une utilisation interactive facile
- Résiste bien face à la concurrence (python, ruby, lua,...)

Questions ?