PTC

Common Lisp in 3D CAD: CoCreate Modeling

European Lisp Symposium 2009

Who's this guy?

Software architect at Parametric Technology

Working in the CoCreate Modeling team since 1991

Belated Lisp convert `(it (took (me (ten years!))))`

`http://www.clausbrod.de/Blog`

Who's this guy?

Software architect at Parametric Technology

Working in the CoCreate Modeling team since 1991

Belated Lisp convert `(it (took (me (ten years!))))`

`http://www.clausbrod.de/Blog`

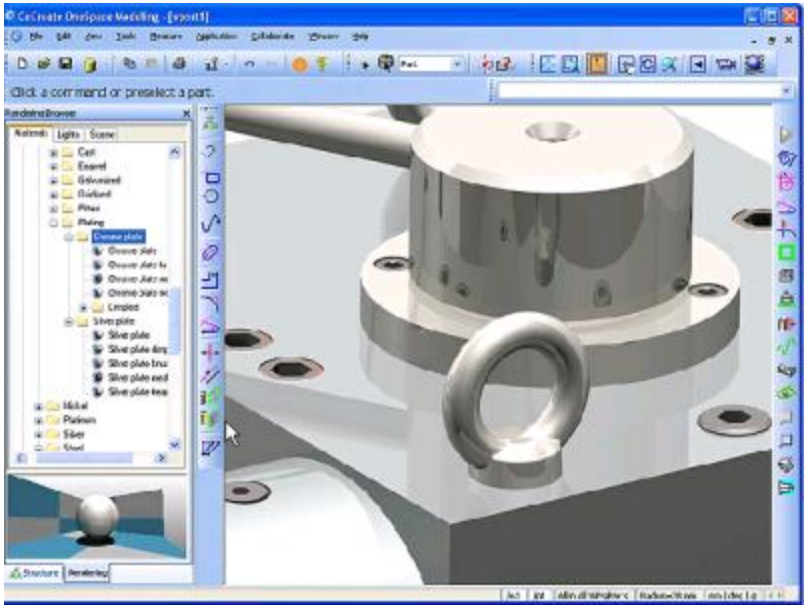
CoCreate Modeling

One of the leading 3D CAD modelers in the mechanical engineering and electronics markets

***The* leader in „explicit modeling“**

Initial prototyping started 1989, first release in 1992

Main implementation languages: C++, Common Lisp



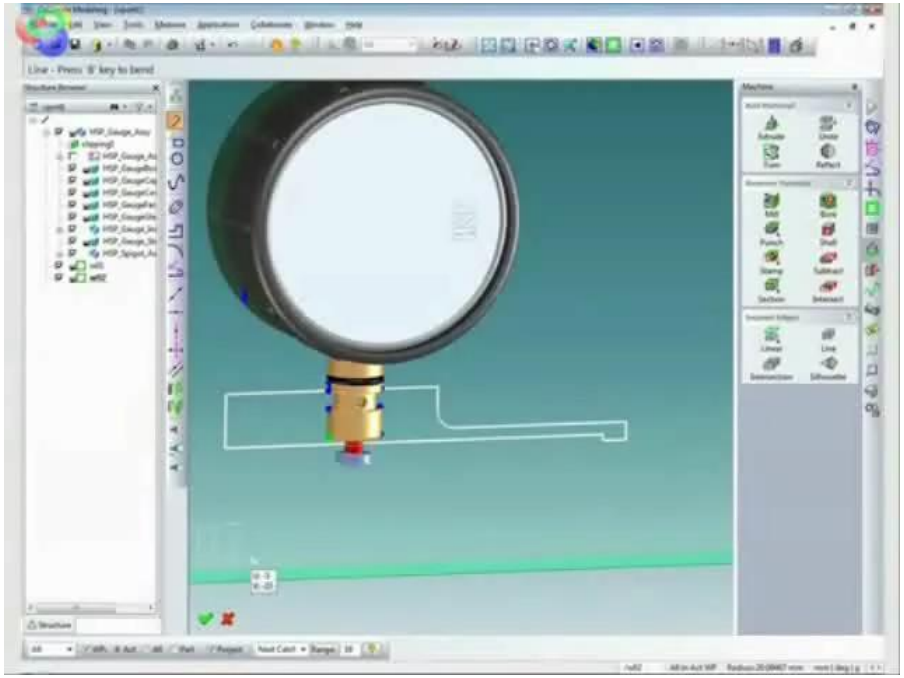
Common Lisp in CoCreate Modeling

Lisp implementation („HCL“) derived from KCL and AKCL

Lisp compiler generates C/C++ code

Close to CLtL1 (pre-standard):

- No conditions
- CLOS in embryonic state
- Parts of CL „library“ missing (example: logical pathnames)



How we use Lisp in CoCreate Modeling

Embedded *configuration language* for CAD admins and end users

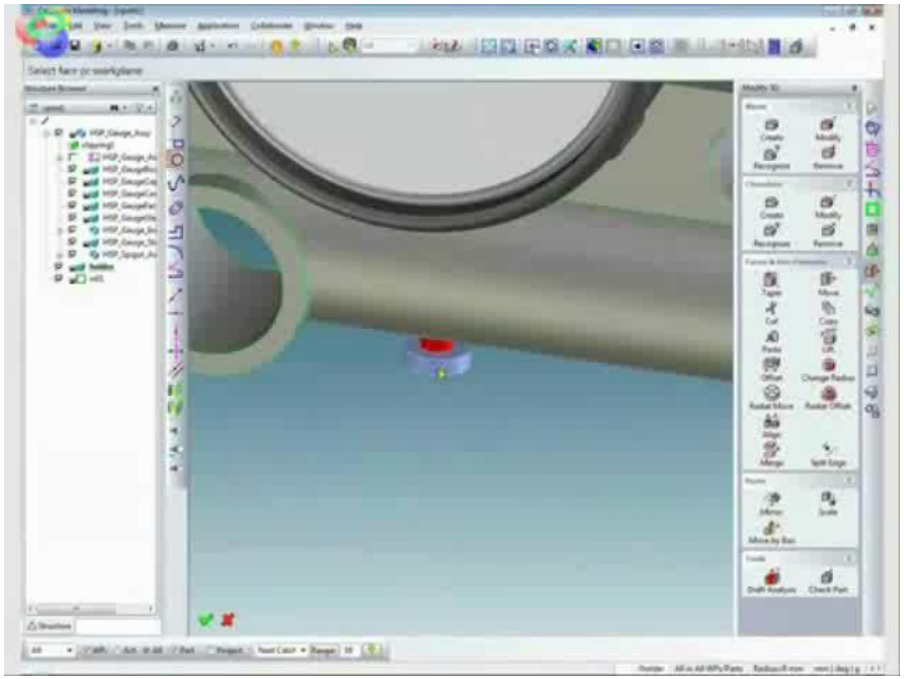
***Scripting language* for automation and writing tests**

Foundation of action routine and dialog *DSLs*

***Implementation language* for internal and external developers**

Our main external API („Integration Kit“) is a Lisp library

Bridge to COM/ActiveX/.NET



So does Lisp scale? (1)

This is a massive project

Several million lines of code in Lisp or Lisp-based DSLs (2/3 of our code overall)

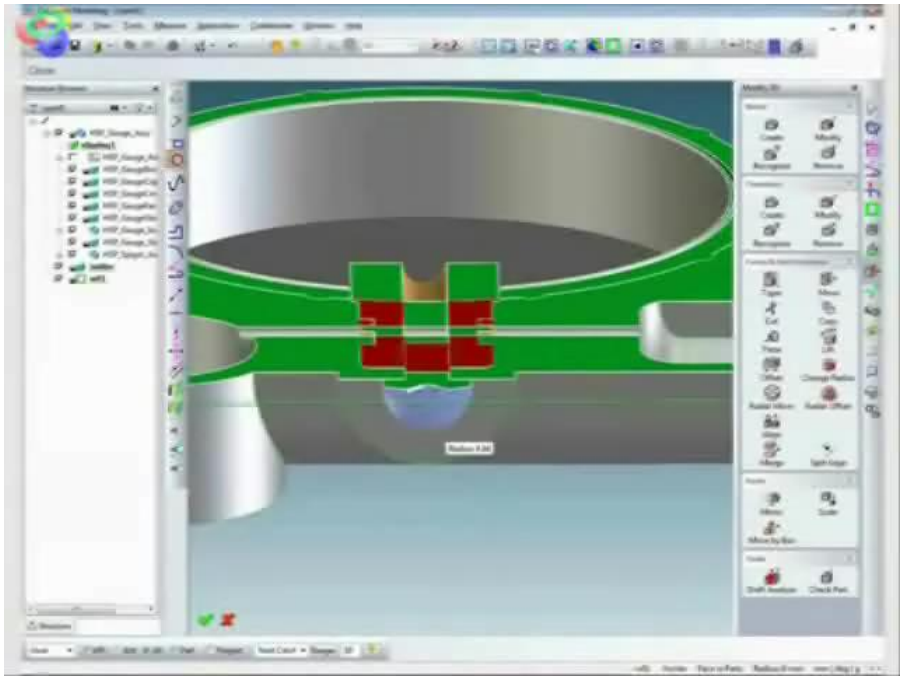
Full Lisp compilation can take a day

Homegrown tools to detect dependencies

Significant distributed development team

~30 customer releases (plus patch releases)

60000 active 2D+3D users, plus 140000 registered users of free version (CoCreate Modeling PE)



So does Lisp scale? (2)

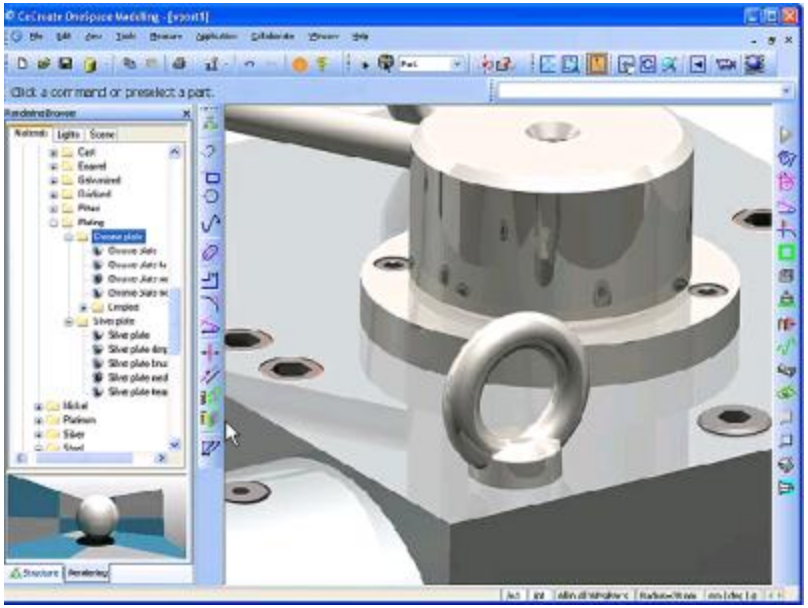
Lisp performance and memory consumption usually a minor issue

Lisp mostly used in front-end, i.e. UI and application logic

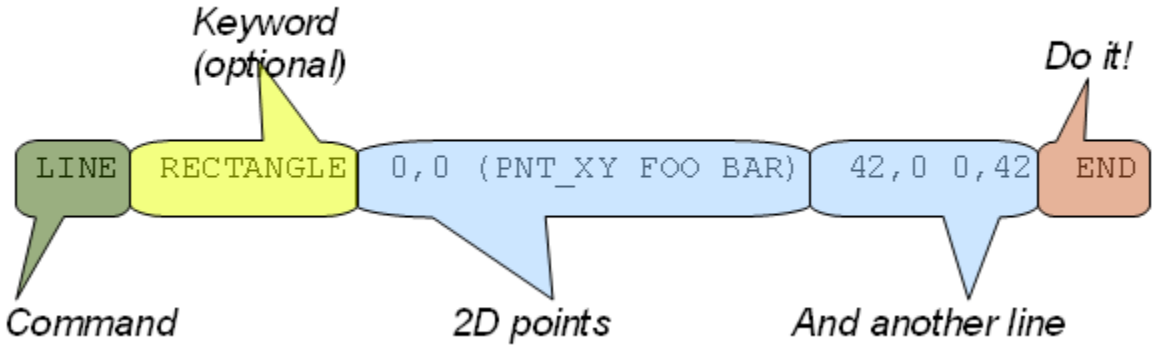
Performance-critical code (modeling kernel!) lives in C++

GBC:

- Can be a problem when loading large uncompiled multilevel macros (DSL code)
- Ameliorated by idle-time GBC



Lisp-based DSL: Action routines



Customers loved extension language in precursor product ME10 (2D CAD)

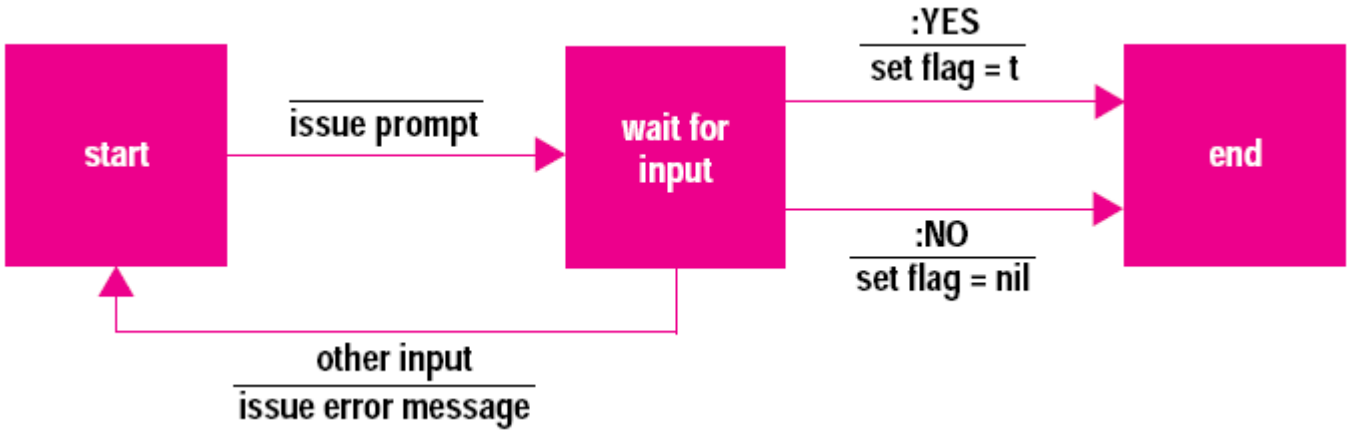
Each command is basically a mini-language, implementable as a state machine

Action routines in CoCreate Modeling:

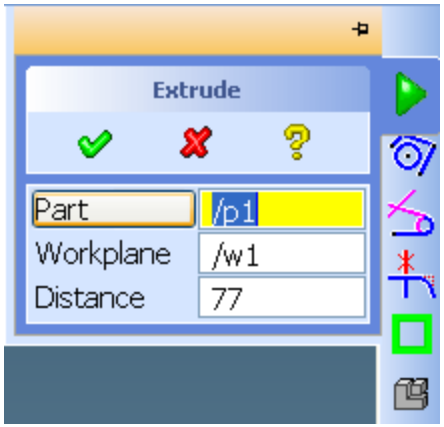
- **Define commands which accept ME10-like syntax**
- **Basically a state machine DSL**

Action routines: simple_exit command

```
(defaction simple_exit
  (flag) ; local variable

  (; state descriptions
    (start nil "Terminate CoCreate Modeling?" nil
      (:yes (setq flag t) answer-yes end)
      (:no (setq flag nil) answer-no end)
      (otherwise (display_error "Enter either :YES or :NO.") nil start))
    (end (do-it) nil nil))

   (; local functions
     (do-it () (when flag (quit)))))
```



Lisp-based DSL: Dialog generator



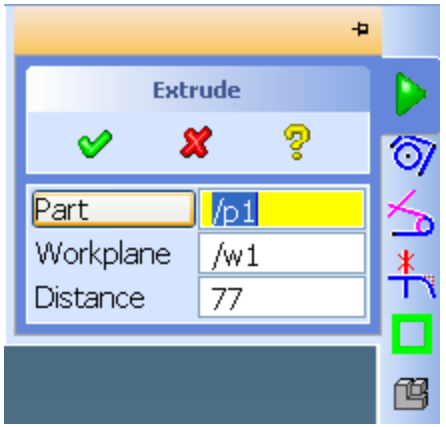

```
(sd-defdialog 'my_extrude
  :dialog-title "Extrude"

  :variables
  '((A_PART :value-type :part-incl-new
            :modifies :contents
            :title "Part"
            :prompt-text "Identify part to extrude.")
    (A_WP :value-type :wp-with-profile
          :title "Workplane"
          :prompt-text "Identify workplane to extrude.")
    (DISTANCE :value-type :distance
          :prompt-text "Specify distance to extrude."
          :initial-value 42))

  :ok-action
  '(sd-call-cmds (extrude :part a_part
                          :wp a_wp
                          :distance distance)))
```

Dialog generator:

- **Automatic command generation**
- ***Plus* automatic UI**
- **Specification of input variables and workflow without state machines**

Services provided for free by the dialog generator (sd-defdialog)

- Workflow
- Automatic UI generation
- Consistent look and feel
- Online help
- UNDO support
- Unit conversion
- User prompting
- Feedback (highlighting, 3D helper objects...)
- Input validation
- Command recording
- ...



Interesting Lisp tidbits in CoCreate Modeling

Unicode support

Lisp/COM interoperability

Integration with IE - Lisp calls JavaScript/IE, IE/JS calls Lisp

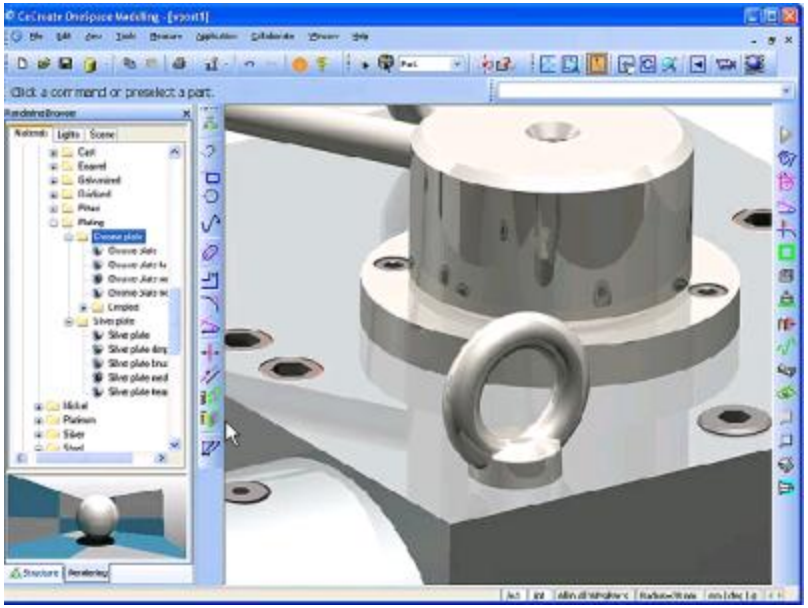
Infix input for arithmetic expressions

Vector notation:

- 0,0 is read as a 2D vector
- 42,42,42 is read as a 3D vector

Socket support

...



Links and references

Article in „HP Journal“ on action routines:

http://www.hpl.hp.com/hpjournal/95oct/oct95a7.htm

Free download (CoCreate Modeling PE):

http://www.cocreate.com/free

My ramblings:

http://www.clausbrod.de/Blog

Official company propaganda:

http://www.ptc.com/products/cocreate/modeling