Software-Entwicklung mit Prolog

Übersicht

- Entwickeln am Toplevel
- Software-Produkte
  - Bibliothek
  - Applikation
- Applikationsarten
  - Befehlszeilenapplikation
  - GUI-Applikation
- Dokumentation

Entwickeln am Toplevel

Klassischer Prolog-Entwicklungszyklus

- Schreiben > Konsultieren > Anfragen > Debuggen > Schreiben…

Dateiorganisation

- Alle Quelldateien in einem Verzeichnis.
- Bei mehr als 7 Dateien allenfalls sinnvolle Unterverzeichnisse
- Zentrale "Lade-Datei": `loadall.pl`
  - Das Rekonsultieren von loadall.pl lädt alle **geänderten** Dateien neu.
  - Achtung Falle: Im Text vollständig gelöschte Prädikate bleiben trotzdem in der Wissensbasis erhalten!

```
:- ensure_loaded([
  datei1,
  datei2,
  datei3,
  datei4
   ]).
```

Skelett einer Ladedatei

Typische Software-Produkte

Bibliotheken

- Sammlung von Prädikaten für (andere) Programmierende
- Schnittstelle: Möglichst wenige, aber mächtige "öffentliche" Prädikate
  - Benutzung des Modulsystems von Prolog für das Verstecken von Hilfsprädikaten
    - Leider noch nicht so richtig Standard, obwohls dazu einen ISO-Standard gibt!

Applikation

- Ablauffähiges Programm für Endbenutzer
- Schnittstelle
  - Befehlszeilenbasiert
  - Interaktiv; heute fast immer mit graphischer Benutzeroberfläche (GUI) und Mausinteraktion auf verschiedenste graphische Elemente: GUI-Applikation

Applikationsarten

Befehlszeilenapplikation

- Ablaufmodell
  - Einlesen der Optionen und Befehlszeilenargumente, verarbeiten, beenden
    - Einfach zu machen! Gut zum Kombinieren von automatischen Filtern!

GUI-Applikation (**g**raphical **u**ser **i**nterface)

- Eventbasiertes Ablaufmodell
  - Aufstarten und Voreinstellungen lesen
  - Benutzer erzeugt irgendein Ereignis, welches vom Programm entsprechend verarbeitet wird
  - Programm verarbeitet Ereignis
    - Aufwändiger und meist plattformabhängiger!
      - Mit Java und TclTk kann +/– plattformunabhängig gearbeitet werden!

Befehlszeilenapplikation mit SICStus

Saved States

- Binäre, kompilierte Repräsentation
- `save_program(+Dateiname,+BeweiszielBeiAufruf)`
- Ablauffähig, falls SICStus Prolog in der **gleichen** Version installiert ist auf der Maschine

  `$ main.sav`
  - Keine Unabhängigkeit von SICStus Entwicklungssystem!
  - Funktioniert ähnlich bei SWI-Prolog!

Application Builder

- Das SICStus-Prolog-Tool spld erlaubt das Erzeugen von Prolog-Applikationen, welche auf Maschinen **ohne SICStus Entwicklungssystem** laufen!

```
:- ensure_loaded(loadall).
main :-
   process,
   halt(0).

save :-
   save_program('main.sav', main).
```

Skelett von main.pl

Applikationen mit SICStus Prolog

spld – Das Befehlszeilentool mit 1001 Optionen

- Unabhängig von der Entwicklungsumgebung und einer Prolog-Lizenz des Endbenutzers
- Erzeugt ca. 6MB grosse Applikationen
  - Funktioniert unter UNIX, Linux, MacOS X;
  - Win benötigt Microsoft Visual C++ 6.0

```
$ spld --static --main=restore --resources-from-sav \
       --resources=main.sav=/mystuff/main.sav -o main.exe
$ main.exe
```


Dokumentation

Unterschiedliches Zielpublikum

- End-Benutzer
  - Aus der Gebrauchsperspektive schreiben: Beispiele
  - Keine irrelevanten Details aus der Implementation(-sgeschichte)!
- Entwickler
  - Konzeptuelle Dokumentation zum implementierten Ansatz
  - Sorgfältige Aufrufsspezifkation für öffentliche Prädikate bei Bibliotheken!
    - Implementationsspezifische Details im Quellkode selbst!

Verknüpfung von Kode und Dokumentation

- Separat unterhaltene Dokumente, oder
- "Literate Programming": Eine Datei, aus dem Programm und Dokumentation erzeugt werden. (lpdoc für Prolog; angelehnt an javadoc für Java)
  - http://clip.dia.fi.upm.es/~logalg/slides/G_lpdoc_cl2000/G_lpdoc_cl2000.html

Integration von Prolog

Prolog-Beweiser als C-Bibliothek

- Verstecken des Beweisers in interne Funktion eines C-Programmes

GUI

- TclTK-Bibliothek
  - Bidirektionales Interface, .d.h. Prolog kann Hauptapplikation sein oder TclTK kann Hauptapplikation sein

Datenbank

- Für grosse Datenmengen
  - Persistente Speicherung von Prolog-Termen mit variabler und mehrfacher Indizierung (nicht bloss First-Argument-Indexing)

und vieles mehr...