

Entwurf von Optimierungen für Squeak

Studienarbeit
Universität Karlsruhe
IPD GOOS

Betreuender Mitarbeiter:
Dipl.-Inform. Dirk Heuzeroth

- Squeak ist ein Programmiersystem für Kinder
- Besondere Anforderungen an die Implementierung

Aufgabe:

- Analyse der bestehenden Ansätze
- verbesserter Entwurf

- Objektorientierte Programmiersprache
- Smalltalk Dialekt
- Einfache graphische Syntax für Kinder
- Beispiel: Auto fahren



Auto Demo

! Auto script1 paused

Auto drehe Dich um 3 ▶

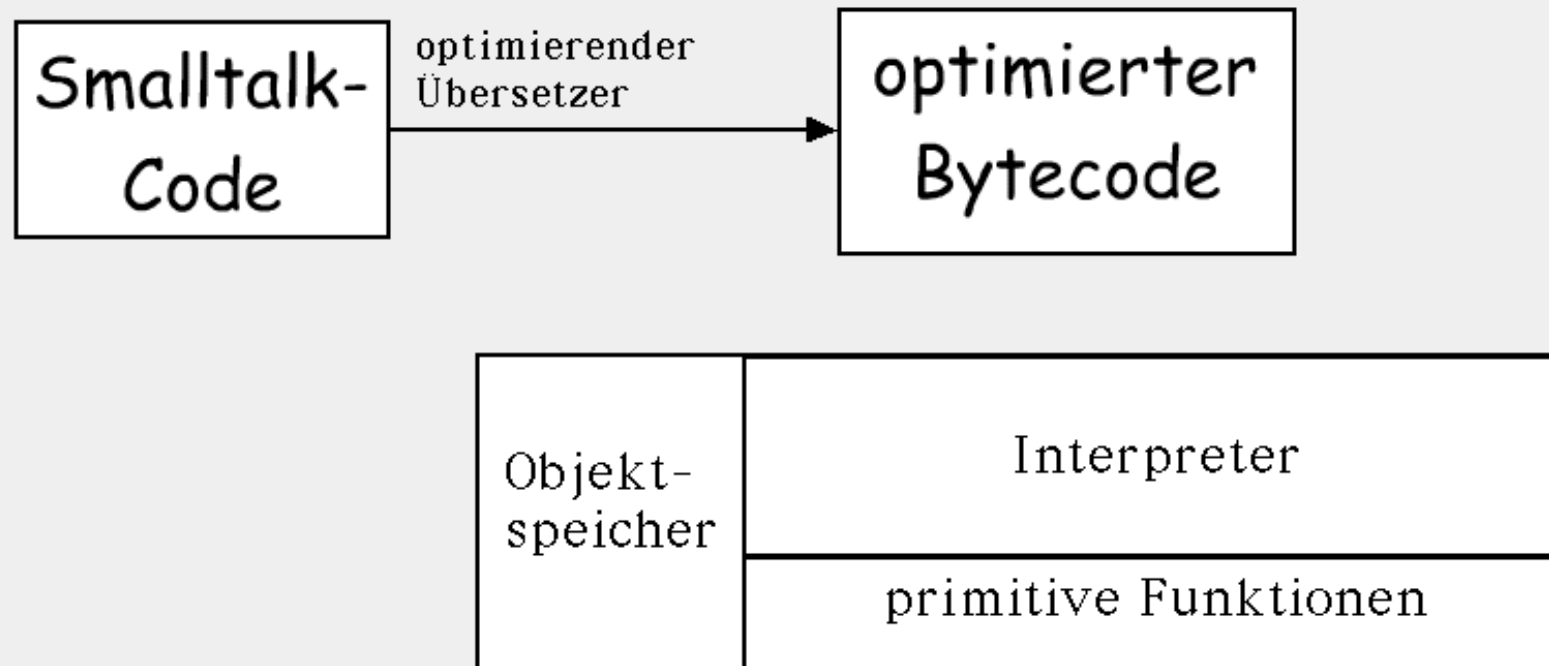
Auto gehe vorwärts um 5 ▶



1. Dynamische Metaprogrammierung
2. Portabilität der Laufzeitumgebung
3. Erwartetes Verhalten:
Konflikt zwischen Spezifikation
und Implementierung
4. Performanz

Existierende Ansätze

Aktuelles System: - Virtuelle Maschine
- Bytecode Interpreter



Probleme des Interpreters

Positiv:

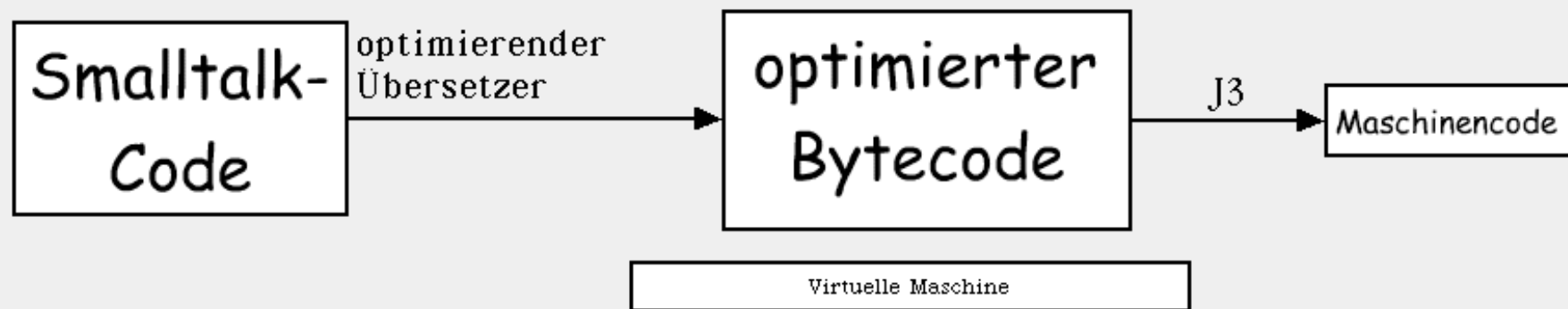
- Dynamische Metaprogrammierung
- Portabel

Probleme:

- Performanz: Teile in C implementiert (primitive Funktionen)
- Erwartetes Verhalten: nicht garantiert wegen nicht-transparenter Optimierungen

J3 Laufzeitübersetzer

- Klassischer Laufzeitübersetzer
- Ziel: Beschleunigung der Bytecodeausführung



- Objektspeicher und primitive Funktionen wiederwendet

Positiv: Schneller

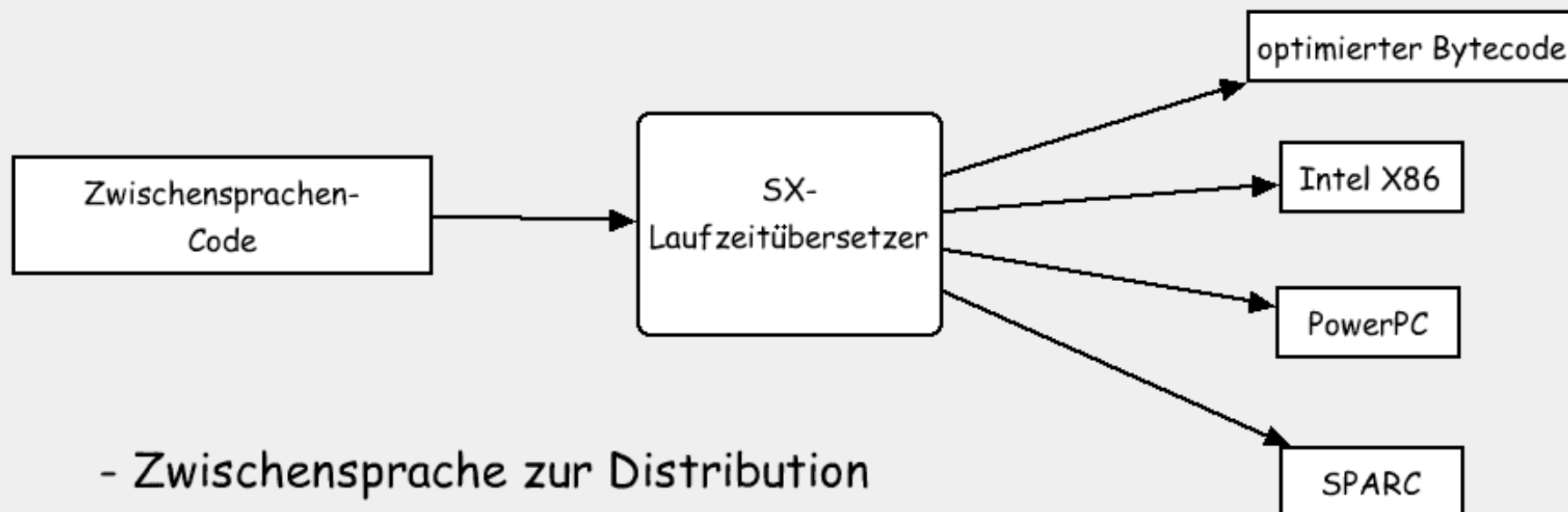
- Beschleunigung:
- Faktor 3 Bytecodeausführung
 - Faktor 7 Methodenaufrufe

Probleme:

- Implementiert in C++:
 - Experimente schwierig
- Ignoriert Probleme der Optimierung des Bytecodes

Neuer Entwurf: SX

- SX "Squeak eXtreme": neue Architektur
- Laufzeitübersetzer als zentrale Komponente



- Zwischensprache zur Distribution (vgl. ANDF, Slim Binaries)
- Bytecode ist Zielsprache
- Übersetzer in Smalltalk implementiert
- Laufzeitübersetzer nicht "Just-In-Time"

- Zielbytecode kann optimiert werden:
Semantikerhaltende Optimierungen statt fehlerhafter
- Zielbytecode und Zwischensprachencode verschieden
- Zwischensprachencode einfacher:
Keine Optimierungen

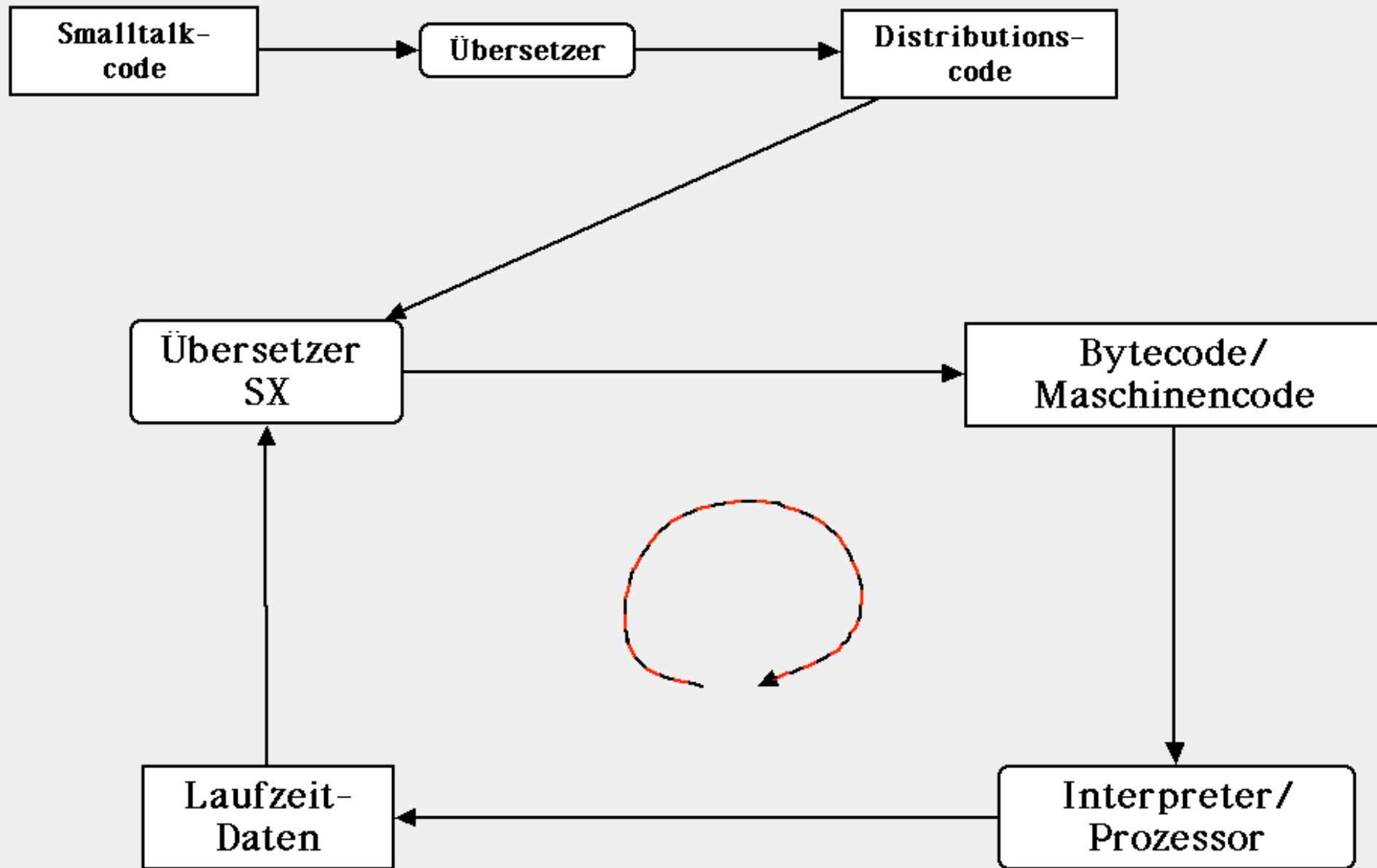
Implementiert in Smalltalk

- Fazit J3-Projekt: Implementierung in C++
problematisch
- Squeak Interpreter in "Slang": wenig mächtig

Daher: SX in Smalltalk implementiert

- > Entwicklungsumgebung von Squeak kann
genutzt werden

Laufzeitübersetzer nicht Just-In-Time



Weitere Möglichkeiten:

- > Statischer Smalltalk Übersetzer
- > Anbindung von Werkzeugen
(z.B. Kinderskriptsprache)

Realisierung:

- > Prototyp und Experimente als Teil
der Projekte AOSTA und Squeak VI4