

BALI — Programmieren im *Internet*

Bernd Krieg-Brückner – Berthold Hoffmann – Zhenyu Qian – Hui Shi • Universität Bremen

EINLEITUNG

Das *Internet* ist ein weltweites Netz zum Austausch von Informationen wie Nachrichten und Dateien. Das *world wide web* benutzt dieses Netz zur Realisierung eines weltweiten multimedialen Informationssystems mit Hypertexten, Bildern, Grafik, Filmen, Ton.

In diesem Projekt soll eine *Sprache* für das Programmieren in *Internet* und *world wide web* entworfen und in einem *System* implementiert werden. In Anlehnung an die heiß diskutierte Sprache *Java* könnte diese Sprache *BALI* heißen.¹

In der Sprache *BALI* können aktive grafische Objekte fürs *world wide web* entwickelt werden. Die Arbeitsweise des *BALI*-Systems ist in Abb. 1 skizziert und kann wie folgt beschrieben werden:

- *BALI*-Moduln werden übers Netz geladen.
- Eingaben werden über aktive Objekte wie Menüs und Formulare im *world wide web* eingelesen.
- Das Programm wird lokal ausgeführt (im wesentlichen).
- Seine Ergebnisse werden im *world wide web* angezeigt.

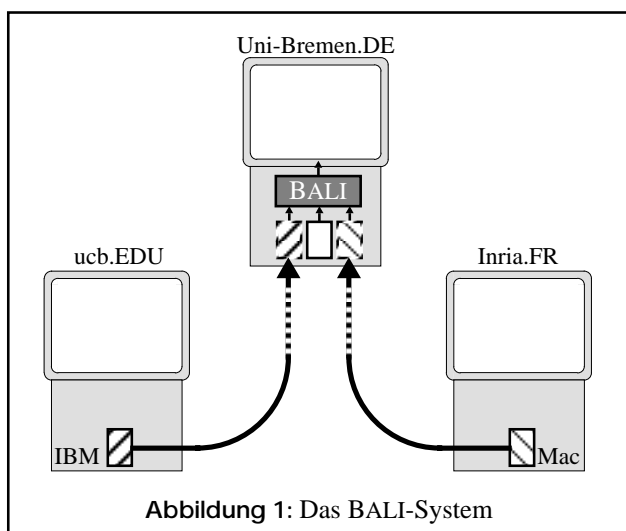


Abbildung 1: Das BALI-System

1. Allerdings versteht man in den USA unter "Java" eher einen "Pott Kaffee" als die Insel westlich von Bali.

THEMA UND ZIEL

Das Thema dieses Projektes berührt mehrere Fachgebiete:

- Programmieren von grafischen Benutzeroberflächen
- verteiltes Programmieren
- Entwurf von Programmiersprachen
- Visuelle Programmiersprachen
- Übersetzerbau
- Netze

Zunächst werden wir ganz allgemein beschreiben, welche Eigenschaften die Sprache und das System *im Idealfall* haben könnten, und dann auf mögliche Schwerpunkte eingehen, die innerhalb des studentischen Projektes bewältigt werden können.

Die Traumsprache

Die Anforderungen an eine Sprache fürs Programmieren im Netz erfordern einige wünschenswerte Eigenschaften:

- Das Programmieren von grafischen Benutzungsumgebungen erfordert eine Sprache, die *modular* ist, deren Moduln leicht *erweiterbar* sind, und die *Nebenläufigkeit* mindestens soweit unterstützt, daß gleichzeitige Benutzeraktionen (Mausklick, Tastatureingabe) bearbeitet werden können. Kurz gesagt, die Sprache sollte *objektorientierte Konzepte* unterstützen.
- Programmieren im Internet geht nur mit einer *Plattform-unabhängigen* Sprache, da Moduln auf Netzknoten mit verschiedener Hardware und/oder verschiedener Betriebssoftware laufen sollen. Die Moduln müssen *robust* auf Fehler reagieren und gegen unberechtigten Zugriff *geschützt* sein. *Verteilte* Ausführung von Programmen ist nicht unverzichtbar, aber eine naheliegende Ergänzung.

Allgemein sollte die Sprache so *einfach* wie möglich sein (aber nicht einfacher, sagt Niklaus Wirth mit Einstein), *abstrakt*, *mächtig* und natürlich *effizient*.

Ähnlichkeiten mit existierenden Sprachen sind *nicht* zufällig, denn auch SUN (1995) beschreibt die für Internet-Programmierung entwickelte Sprache *JAVA* bescheiden als: einfach, objektorientiert, verteilt, interpretiert, robust, sicher, Architektur-neutral, portabel, effizient, nebenläufig und dynamisch.

Mit diesen Eigenschaften können wir eine vorläufige Wunschliste von Konzepten aufstellen:

- Klassen
- Vererbung
- dynamisches Binden
- *persistente Objekte*
- strikte Typisierung
- Ausnahmebehandlung
- automatische Speicherbereinigung
- keine Zeiger
- *Mehrfachvererbung*
- *generische Klassen*

Alle bis auf die kursiv gesetzten Konzepte sind in JAVA auch vorhanden. JAVA könnte denn auch ein Ausgangspunkt für die Entwicklung von BALI sein. (Aber nicht unbedingt das Vorbild!)

Die grafischen Gesichtspunkte könnten noch weiter getrieben werden, indem man BALI selbst als eine *visuelle Sprache* zum "Programmieren mit Bildern" konzipiert.

Das Traumsystem

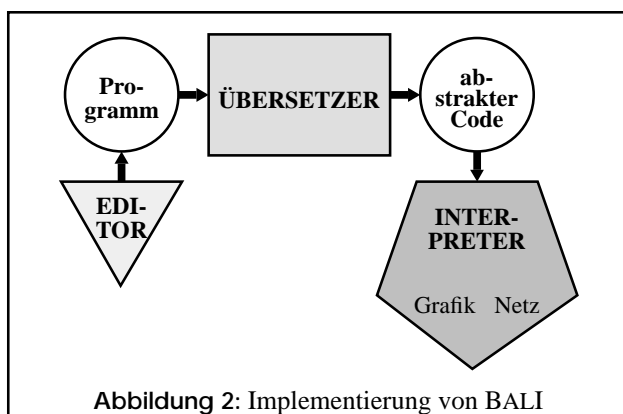
Das ideale BALI-System besteht aus einem Editor, einem Übersetzer und einem Interpreter (siehe Abb. 2).

Der *Editor* ist sprachorientiert: Er "kennt" die Regeln der Sprache und macht den Programmierer schon beim Erstellen des Programms auf Fehler aufmerksam. Mit ihm wird das *Programm* konstruiert (als Syntaxbaum).

Der *Übersetzer* transformiert Programme in *abstrakten Code*, der maschinennah, aber portabel ist und auf herkömmlichen Plattformen effizient ausgeführt werden kann.

Der *Interpreter* führt abstrakten Code aus. Er muß Moduln übers *Netz* laden, die dynamische Bindung von Methoden an Objekte vollziehen. Zu ihm gehört auch eine Bibliothek für das Programmieren von *Grafik* und grafischen Benutzungsoberflächen.

Um den Zusatzaufwand für die Interpretation des abstrakten Codes (teilweise) zu vermeiden, könnte der abstrakte Code auch in den lokalen Maschinencode *assembliert* werden. Beim Importieren von Moduln müßte das wegen der Heterogenität im Netz im allgemeinen lokal und zur Laufzeit geschehen (dynamische Übersetzung).



Das System soll nach der Methode des objektorientierten Entwurfs entwickelt werden. Dabei werden auch typische Übersetzerwerkzeuge wie Parsergeneratoren benutzt, und auf existierende Bibliotheken, z. B. für Grafik und Verteiltheit zurückgegriffen.

Mögliche realistische Projektziele

Um in der gegebenen Zeit überhaupt etwas fertigzustellen, müssen einige Abstriche an Traumsprache und Traumsystem gemacht werden. Im Projekt nach den Interessen der Teilnehmenden (Studierende *und* Veranstaltende) ein Schwerpunkt für die Arbeit gesetzt werden, z. B.:

- mächtige und abstrakte Sprachkonzepte *oder*
- visuelles Programmieren *oder*
- Effizienz des Systems *oder*
- Verteiltes Programmieren

Für die als weniger interessant erachteten Aspekte des Projektes müßte (und könnte) auf existierende Lösungen zurückgegriffen werden. Schließlich könnten unerfüllte Träume dann immer noch in Diplomarbeiten realisiert werden.

PLANUNG

Wir gehen auf Erwartungen an die TeilnehmerInnen des Projektes und den geplanten Ablauf ein und geben einige Hintergrundinformation über die VeranstalterInnen des Projektes und ihre Arbeitsgruppe.

Erwartungen an die TeilnehmerInnen

Interesse für praktische Programmierung, Programmiersprachen und Übersetzer ist sicher eine Grundvoraussetzung dafür, sich aus diese Projekt einzulassen. Jede ProjektteilnehmerIn wird Gelegenheit bekommen, Fragestellungen zu finden und in kleinen Gruppen selbstständig zu bearbeiten.

Daneben müssen alle TeilnehmerInnen Verantwortung für das Projekt als Ganzes übernehmen, indem sie sich für die Organisation engagieren. Die VeranstalterInnen verstehen sich als Berater, nicht als Organisatoren der Projektarbeit.

Zum Ablauf

Das Projekt besteht aus einem Arbeitsvorhaben und begleitenden Lehrveranstaltungen. Kursiv gesetzte Veranstaltungen sind für das Projekt unbedingt notwendig. (Die Studienordnung nennt sie "Pflicht".)

Winter 96

Der Schwerpunkt des Projektes wird gemeinsam festgelegt. Sprachkonzepte werden daraufhin untersucht, ob sie für BALI *wichtig* und *notwendig* sind. Parallel dazu werden Er-

fahrungen mit dem Programmieren im Internet gesammelt (anhand von Java). Ein ausführlicher *Anforderungskatalog* (für Sprache und System) und ein Arbeitsplan für die Folgesemester wird erstellt.

Lehrveranstaltungen

- *BALI – Analyse von Konzepten für die Internet-Programmierung* (2, Krieg-Brückner, Hoffmann, Qian, Shi)
- *Programmiersprachen* (3, Hoffmann)
- *Programmieren in Java* (2, Qian, Shi)
- Fortgeschrittene funktionale Programmierung (2, Krieg-Brückner)

Sommer 97

[Die Planung dieses und der folgenden Semester können sich je nach gewähltem Schwerpunkt ändern.]

Die Sprache BALI wird entworfen. Syntax, Kontextbedingungen und Semantik werden präzise beschrieben. Ein Prototyp der *Syntaxanalyse* wird mit Werkzeugen implementiert.

Lehrveranstaltungen

- *BALI – Entwurf, Implementierung* (4, Hoffmann, Qian, Shi)
- *Übersetzer* (4, Hoffmann)
- Objektorientierte Analyse und Entwurf (2, Qian)
- Verteilte Programmierung (2, Shi)

Winter 97

Editor, Übersetzer und Interpreter des Systems werden implementiert, bzw. bestehende Implementierungen werden angepaßt. Dabei werden zunächst nur wenige Funktionen realisiert, und später Erweiterungen vorgenommen. Ergebnis ist ein arbeitende Version für eine Teilmenge der geplanten Konzepte.

Lehrveranstaltungen

- *BALI – Implementierung* (4, Hoffmann, Qian, Shi)
- *Übersetzerwerkzeuge* (2, Hoffmann)
- [Objektorientierte Sprachen (2, Qian)]

Sommer 98

Die Implementierung wird vervollständigt. Fallstudien dienen der Überprüfung der gesetzten Anforderungen. Der Projektbericht wird geschrieben.

Lehrveranstaltung

- *BALI – Implementierung, Fallstudien* (6, Hoffmann, Qian, Shi)

Weitere Ergänzungen zum Thema Netze, Sicherheit in Netzen und gesellschaftlichen Auswirkungen der Vernetzung sind sicherlich sinnvoll.

Projektmanagement

Zusammen mit anderen Projekten soll im Winter 96 eine kurze Einführung in das Projektmanagement gegeben werden.

Für die Softwareentwicklung soll die Methode des objektorientierten Entwurfs verwendet werden.

Über die VeranstalterInnen

Bernd Krieg-Brückner wird im ersten Semester seine Erfahrung bei der Durchführung von Projekten weitergeben; außerdem ist er Experte für Sprachentwurf.

Berthold Hoffmann beschäftigt sich mit Programmiersprachen, Übersetzer und vor allem auch mit Grafik und Visualisierung (einem möglichen Schwerpunkt des Projektes).

Zhenyu Qian beschäftigt sich mit objektorientierter Softwareentwicklung, Konzepten von Programmiersprachen und insbesondere mit Typisierung.

Hui Shi beschäftigt sich mit paralleler und verteilter Programmierung.

Alle begreifen das Projekt als Gelegenheit, ihre Kenntnisse weiterzugeben, auszutauschen und zu vertiefen.

Über die Arbeitsgruppe Krieg-Brückner

Die Arbeitsgruppe von Bernd Krieg-Brückner vertritt seit 1982 das Fach *Programmiersprachen und Übersetzer* im Studiengang Informatik. Sie beteiligt sich an der Durchführung der LV Praktische Informatik und bietet im Hauptstudium regelmäßig die Kurse "Prinzipien von Programmiersprachen", "Übersetzer" und "Übersetzerwerkzeuge" an.

Im Laufe der Jahre hat sich das Forschungsinteresse ausgeweitet auf *Methoden und Werkzeuge zur Entwicklung sicherer Systeme* und *kognitive Robotik* (mit neuronalen Netzen). Ein Teil der Arbeitsgruppe beschäftigt sich mit der *Visualisierung und regelbasierten Ersetzung von Graphen*.

In der folgenden Aufstellung werden die Projekte in den verschiedenen Forschungsgebieten der Gruppe genannt; studentische Projekte erscheinen kursiv.

- Programmiersprachen und Übersetzer
 - *Überquelle*, *PORQUE*, *FUSEL*, AnnA, EXSPEC, TOFL.
- Methoden und Werkzeuge für sicherer Systeme
 - *SIESTE*, PROSPECTRA, CORSO, COMPASS, Uni-ForM, CoFi.
- neuronale Netze und kognitive Robotik
 - *FORANN*, *SAUS*, KOGNITAS, Raumkognition
- Graphvisualisierung und Graphersetzung
 - *FORAUS*, daVinci, CompuGraph, GRACE.

Das vorgeschlagene Projekt hängt nicht direkt mit einem der laufenden Forschungsprojekte zusammen, obwohl es

einige Berührungspunkte mit den Projekten TOFL (Krieg-Brückner und Qian) und GRACE (Hoffmann) gibt.

Mehr Informationen ...

... über die Arbeitsgruppe, die genannten Projekte und die VeranstalterInnen findet sich im www unter der Adresse

<http://Informatik.Uni-Bremen.DE/~agbkb>

Unter Staff→Hoffmann→Lehre findet man auch schon etwas über Bali.

ZUSAMMENFASSUNG

Im Projekt BALI werden Kenntnisse und praktische Fähigkeiten auf Gebieten der Informatik vermittelt, die für die spätere Praxis wichtig sind:

- Objektorientierter Softwareentwurf
- Grafik-Programmierung
- Moderne Konzepte von Programmiersprachen
- Visuelle Programmiersprachen
- Moderne Methoden des Übersetzerbaus

Literatur

Flanagan (1996), Lemay und Perkins (1996) und SUN (1995) beschreiben (Programmieren mit) JAVA.

Glinert (1990) ist eine Artikelsammlung zu Sprachen und Systemen für visuelles Programmieren.

David FLANAGAN (1996). *Java in a Nutshell*. Sebastopol, California: O'Reilly & Associates. [30 DM]

Ephraim P. GLINERT (1990). *Visual Programming Environments*. Los Alamitos, CA: IEEE Comp. Soc. Press Tutorial.

Laura LEMAY, Charles L. PERKINS (1996). *Teach Yourself Java in 21 Days*. Indianapolis, IN: Sams.net Publishing [85 DM]

<http://www.mcp.com/samsnet/books/yjava>

SUN (1995): *The Java Language Specification, Version 1.0 Beta*. Mountainview, 30.10.1995.

<ftp://ftp.javasoft.com/docs/javaspec.ps.zip>