Individuelles Projekt

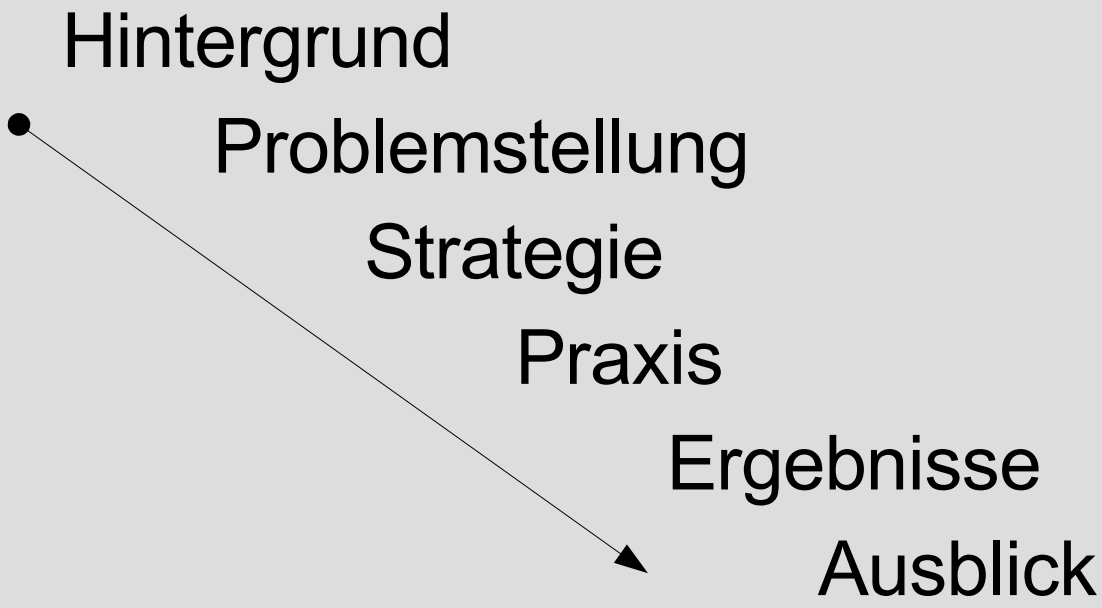
Verifikation und Validierung des Modellierungswerkzeuges 'Kiek/EcoScape'

Nina Marwede
9. Semester Diplom-Informatik

12.05.2005

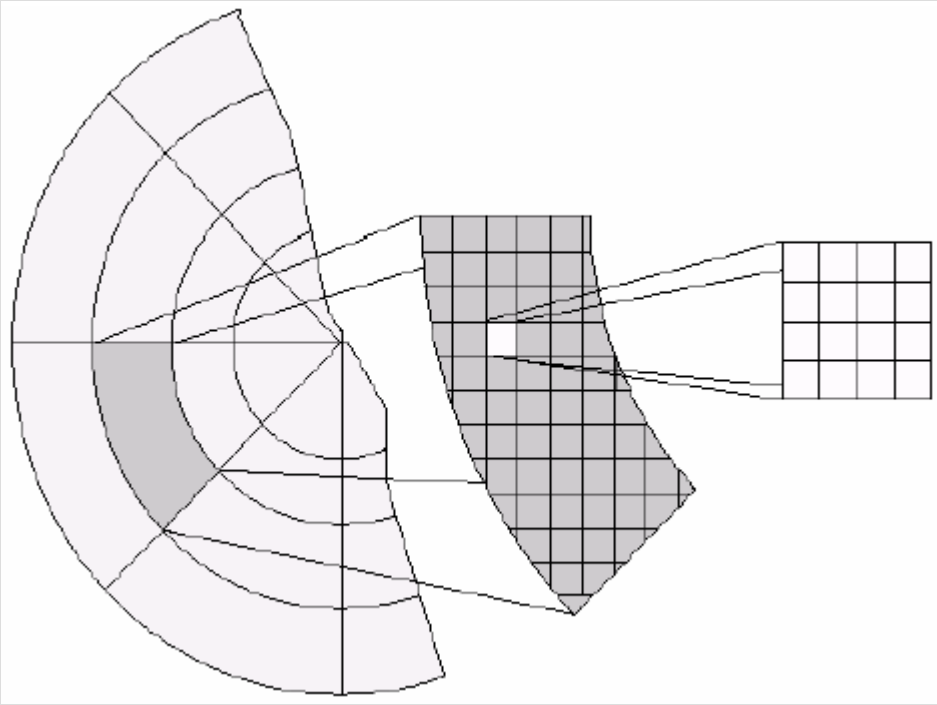
Betreuer: Prof. Dr. Michael Sonnenschein und Dr. Ingo Stierand

Gliederung


Hintergrund
Problemstellung
Strategie
Praxis
Ergebnisse
Ausblick

Hintergrund

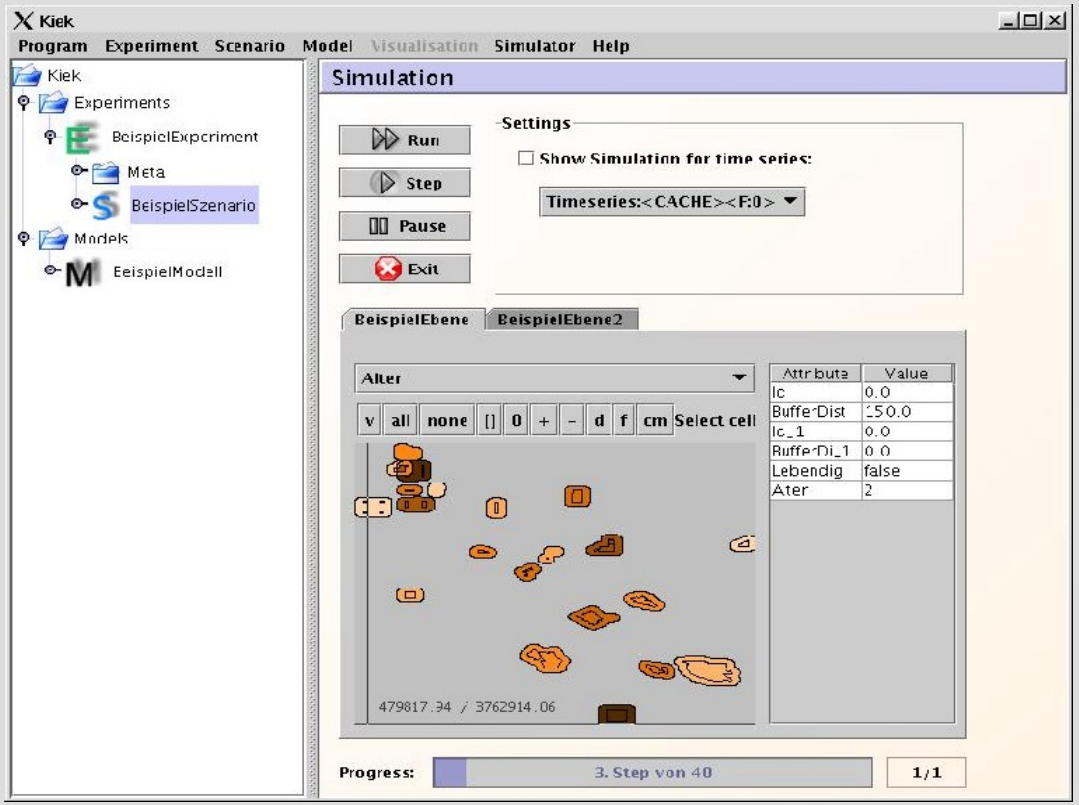
- Hierarchische Asymmetrische Zellulare Automaten
  - Zellen sind endliche Automaten
  - Zustand bestimmt durch Attribute
  - Zustandsübergänge abhängig von Nachbarschaft und Hierarchie
- Diplomarbeit: EcoScape
- Projektgruppe: Kiek



aus der PG-Vorstellung, M. Sonnenschein 2003

Problemstellung

- Ergebnis eines früheren IP (2004) : Das Modellierungswerkzeug "Kiek" funktioniert nicht.
  - Mangelhafte Benutzerführung
  - Fehlerhafte Simulation: falsche Ergebnisse bzw. Nullwerte
  - Abstürze unter Windows



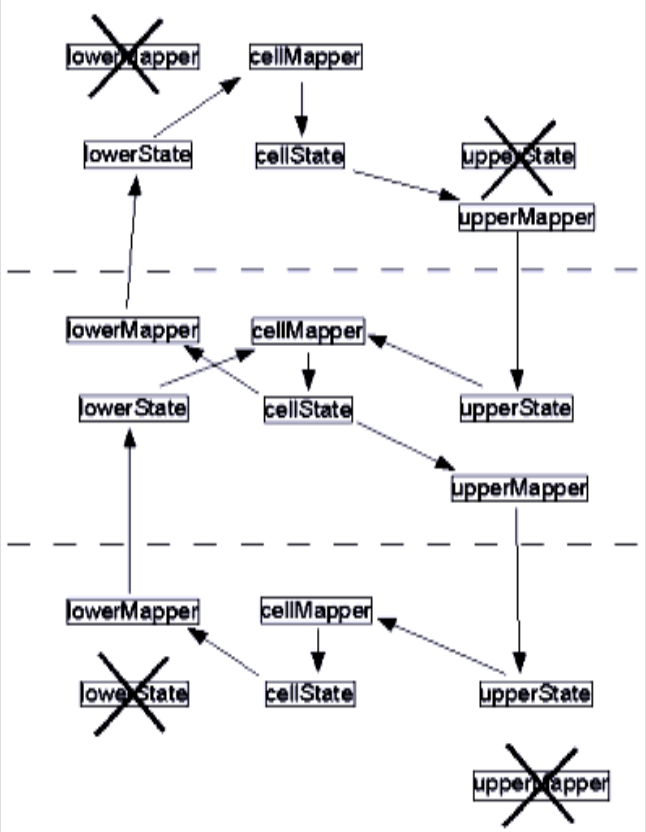
Kiek-Screenshot, PG Kiek 2004

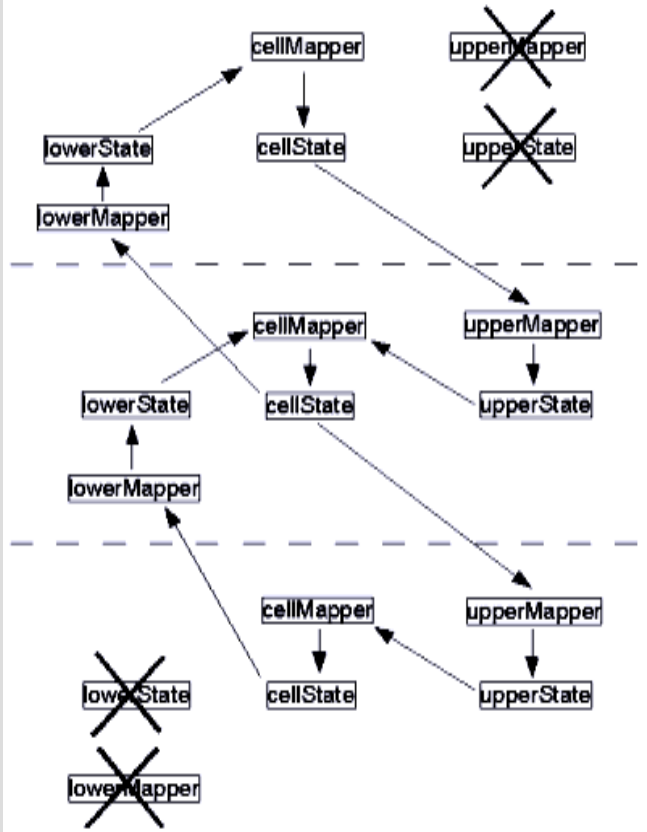
Problemstellung

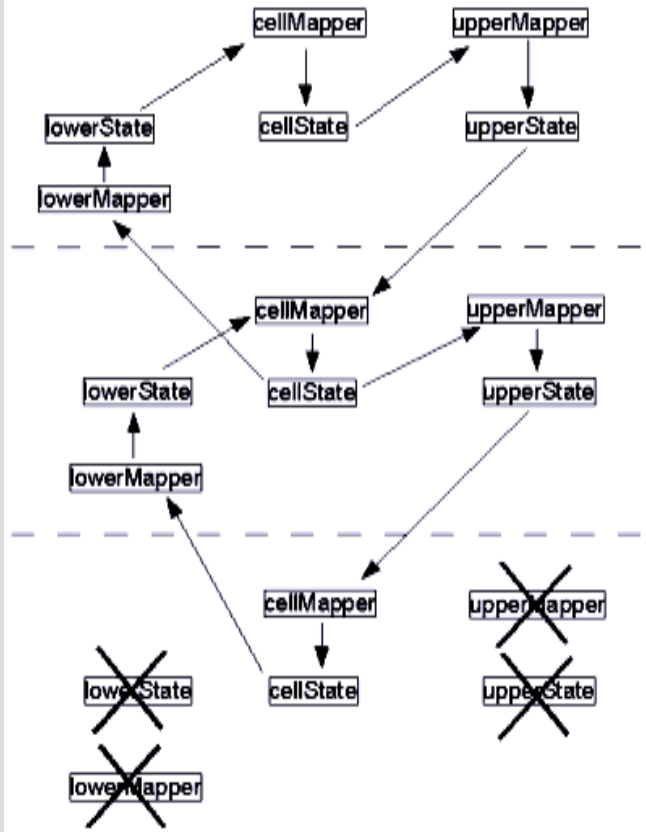
- Wie funktioniert die Hierarchie?
- Sieben verschiedene Varianten:
  - EcoScape-Diplomarbeit
  - EcoScape-Tutorial "EcoLife"
  - EcoScape-Check-Code (scpModel::check())
  - EcoLife-Original-Code
  - PG-Seminar
  - PG-Programmierung
  - Evaluationsmodell "Springbrunnen" (voriges IP)
- Davon drei im Prinzip funktionstüchtige...

Problemstellung

- Vergleich: EcoScape-Doku, PG und Springbrunnen







Strategie

- Bohm (1979):
  - "**Validierung**: Erstellen wir das richtige Produkt?"
    - Bedürfnisse der (zahlenden) Kundschaft
  - "**Verifikation**: Erstellen wir das Produkt richtig?"
    - Spezifikation
- Myers (1991):
  - "**Testen** ist der Prozess, ein Programm mit der Absicht auszuführen, Fehler zu finden."
- **Debugging**
  - Nicht Lokalisierung, sondern Behebung von Fehlern!

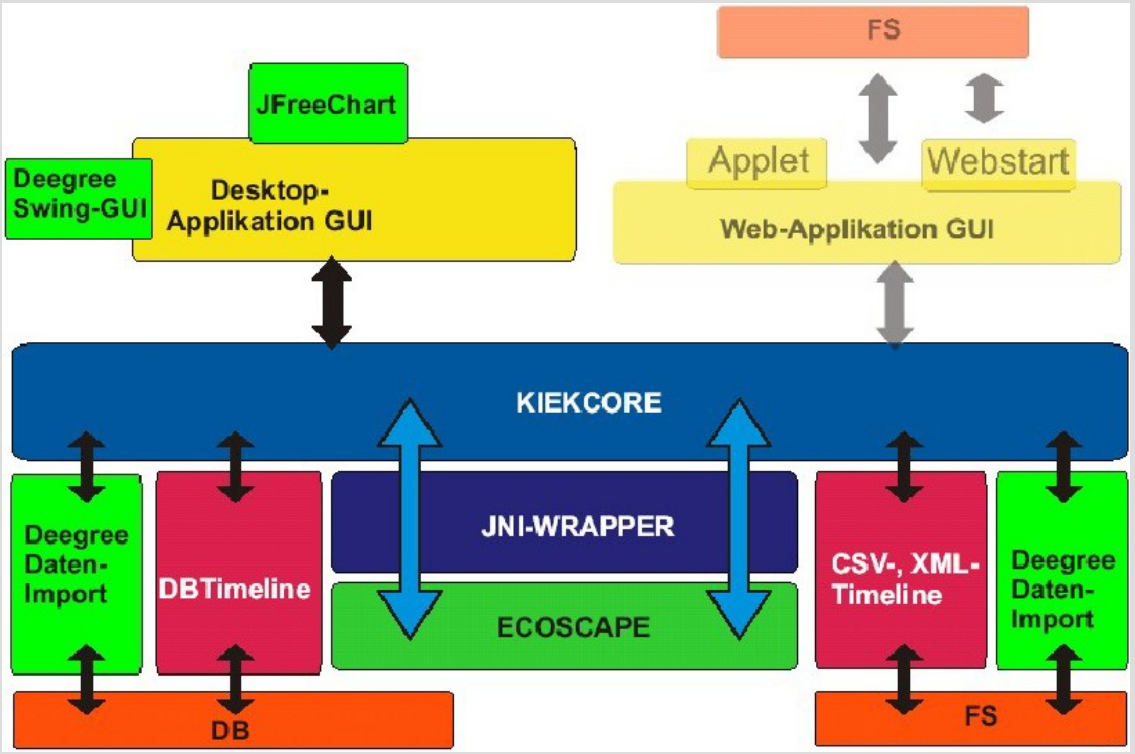
Strategie

- Validierung erstmal beiseite
- Priorität hat die Fehlersuche

- Inspektionen
  - Analyse von Dokumenten
- Testen
  - Ausführung einer Implementierung mit Testfällen
  - Fehlertests
  - Regressionstests
  - Back-to-back-Tests

Strategie

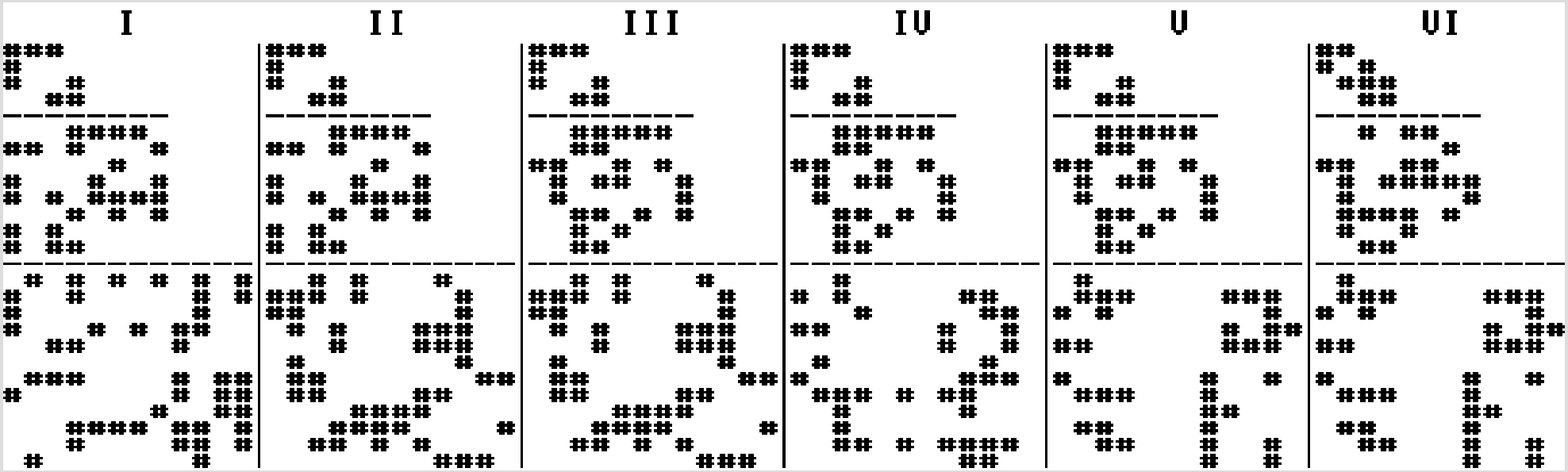
- Top-down: grobe Lokalisierung der Fehler
- Bottom-up: Verifikation der verdächtigen Komponenten



Kiek-Architektur, PG Kiek 2004

Praxis

- Validierungstest: EcoLife
  - Fehler im Tutorial
  - scheint aber zu funktionieren (besser als Kiek)
  - "echte" Verifikation schwierig



Praxis

- Fehlertest: Springbrunnen
  - In Kiek deutliche Fehler sichtbar
  - In EcoScape kein Fehler erkennbar!
  - Ist die Springbrunnen-Hierarchie also die richtige?

I-III	IV-VI	VII-IX
cell1 = 0 cell2a = 0 - 0 cell2b = 0 - 0 cell3 = 1 - 1 - 1 - 1	cell1 = 0 cell2a = 0 - 0 cell2b = 4 - 4 cell3 = 0 - 0 - 0 - 0	cell1 = 16 cell2a = 0 - 0 cell2b = 0 - 0 cell3 = 0 - 0 - 0 - 0
cell1 = 0 cell2a = 2 - 2 cell2b = 0 - 0 cell3 = 0 - 0 - 0 - 0	cell1 = 0 cell2a = 0 - 0 cell2b = 0 - 0 cell3 = 4 - 4 - 4 - 4	cell1 = 0 cell2a = 0 - 0 cell2b = 16 - 16 cell3 = 0 - 0 - 0 - 0
cell1 = 4 cell2a = 0 - 0 cell2b = 0 - 0 cell3 = 0 - 0 - 0 - 0	cell1 = 0 cell2a = 8 - 8 cell2b = 0 - 0 cell3 = 0 - 0 - 0 - 0	cell1 = 0 cell2a = 0 - 0 cell2b = 0 - 0 cell3 = 16 - 16 - 16 - 16

Praxis

- Mailkontakt mit dem Autor der von EcoScape
  - Das Hierarchie-Konzept des Springbrunnen ist tatsächlich mit hoher Wahrscheinlichkeit korrekt bzw. entspricht zumindest der Realisierung in EcoScape.
  - Mindestens zwei entscheidende Abbildungen in der Diplomarbeit sind falsch.
  - Somit *kann* Kiek natürlich nicht funktionieren; die Projektgruppe ist von falschen Voraussetzungen ausgegangen.

Praxis

- Klassisches Debugging: Kiek-Abstürze
  - Debug-Bibliotheken machen Probleme
    - Zuweisung von `string`-Objekten
    - Ursache unbekannt
  - Fehlerhafte Erzeugung von Zeitpunkten im Simulator-Konstruktor
    - Zeiger ungültig
    - Speicherfreigabe schlägt fehl
    - Lösung: Template-ähnliche Konstruktion
  - Ergebnis: keine Abstürze mehr, aber die Visualisierung in Kiek scheitert mit Java-Ausnahmebehandlung

Praxis

- Bottom-up: ein Schritt nach oben
- Invokation der Java Virtual Machine misslingt
- Taktik: JNI-Wrapper ohne JNI
  - C++-Tests sollen jene JNI-Methoden benutzen, die eigentlich von Java aus aufgerufen werden
- "Kastrierung" des JNI-Wrappers
  - Signaturen
  - Zeichenkettenkonvertierung
  - Felder

Praxis

- Back-to-back-Test: JNI-Springbrunnen
  - Reihenfolge muss geändert werden
  - Zellenerzeugung muss geändert werden
  - andere Definition der Zustandsübergänge
  - parallele Modifikationen am Original-Springbrunnen
  - Regressionstests
  - Zusicherungen (assert)
- JNI-GameOfLife
  - gemeinsames Modell macht nicht viel Sinn

```
//layer1->setCellState( cellState1 );
Java_kiekcore_jniEcoScapeImpl_KiekModelLayer_jniSetCellState( layer1, cellState1 );

...

//cell = new scpModelCell();
//ring = new scpGeoLineRing();
//ring->insertPointAtEnd( new scpGeoPoint(  100,  100 ) );
//ring->insertPointAtEnd( new scpGeoPoint(  600,  100 ) );
//ring->insertPointAtEnd( new scpGeoPoint(  600, 1100 ) );
//ring->insertPointAtEnd( new scpGeoPoint(  100, 1100 ) );
//cell->setGeometry( new scpGeoPolygon( ring ) );
//cellSpace->addCell( cell );
double pts02[] = { 100,  100,
                   600,  100,
                   600, 1100,
                   100, 1100 };
res = Java_kiekcore_jniEcoScapeImpl_KiekScenarioLayer_jniAddModelCell( layer2, pts02, 8 );
assert( res == 32 );

...

//layer1->setLowerStateMapper( new SprbrnLowerMapper1() );  // lower1 = sum(cell2a)
prg = Java_kiekcore_jniEcoScapeImpl_KiekScenarioLayer_jniSetLowerStateMapper( layer1,
  "LSA(lower1) = 0; \
   FOREACH EMBEDDED { \
      LSA(lower1) = LSA(lower1) + CSA(cell2a); \
   }", 0 );
assert( prg );
assert( prg[0] == 0x0000001a );    // 1a PUSHCONSTI
assert( prg[1] == 0x00000000 );    //     <const>
assert( prg[2] == 0x00004022 );    // 22 POPI      40 lower  00 index    lower1
assert( prg[3] == 0x00100131 );    // 31 BEGINLOOP 01 ctr    10 cell     FOREACH {
assert( prg[4] == 0x00004012 );    // 12 PUSHI     40 lower  00 index    lower1
assert( prg[5] == 0x00031012 );    // 12 PUSHI     10 cell   03 index    cell2a
assert( prg[6] == 0x00000091 );    // 91 ADD                             +
assert( prg[7] == 0x00004022 );    // 22 POPI      40 lower  00 index    lower1
assert( prg[8] == 0xfffc0132 );    // 32 ENDLOOP   01 ctr    fffc adr    }
assert( prg[9] == 0x000000f0 );    // f0 RETURN
delete[] prg;
test = Java_kiekcore_jniEcoScapeImpl_KiekModelState_jniGetParserErrorString();
assert( strlen(test) == 0 );
```

Ergebnisse

- Fehler in der EcoScape-Diplomarbeit
  - Hierarchie
- Fehler im EcoScape-Tutorial "EcoLife"
  - Hierarchie
- Fehler in EcoScape-Model
  - eine potenzielle Endlosschleife, sonst Kleinigkeiten
- Fehler in EcoScape-Simulator
  - fehlerhafte Zeitpunkt-Konstruktion
    - Lösung gefunden
  - defekter Nachbarschafts-Iterator
    - Ursache unklar

Ergebnisse

- Fehler im Parser und EcoShell
  - FOREACH EMBEDDED ist unvollständig implementiert
    - Token ist deklariert, aber Bytecode-Erzeugung fehlte
  - Potenzielle Fehler im Flex-Code
    - behoben
  - EcoShell-Logik-Operatoren arbeiten nicht wie erwartet
- Fehler im JNI-Wrapper
  - ungünstige Rückgabewerte
    - können zu Endlosschleife führen
  - Copy-Paste-Fehler
  - fehlende/falsche Initialisierungen
- Fehler in Kiek

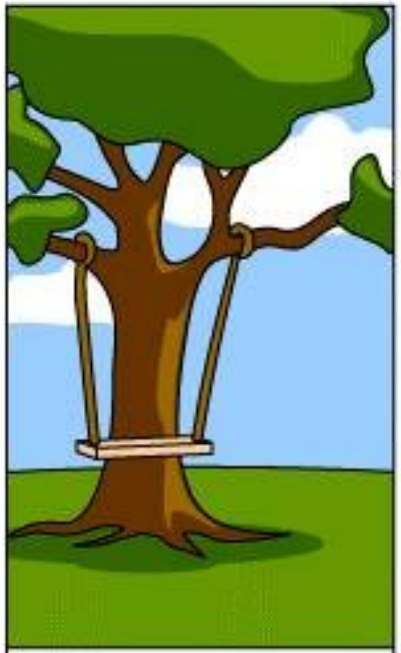
Ergebnisse

- JNI-Springbrunnen funktioniert
  - testet aber nur Hierarchie, keine Nachbarschaften
- JNI-GameOfLife funktioniert nicht
  - Neighborhood-Iterator im Simulator ist irgendwie "kaputt"
  - aber nur in Verbindung mit der JNI-Komponente!
- Gegenüberstellung deutet auf die Konstruktion der Zustandsübergänge und die Zuweisung der Übergangsfunktion
- leider keine Zeit für weitere Untersuchungen

Ausblick

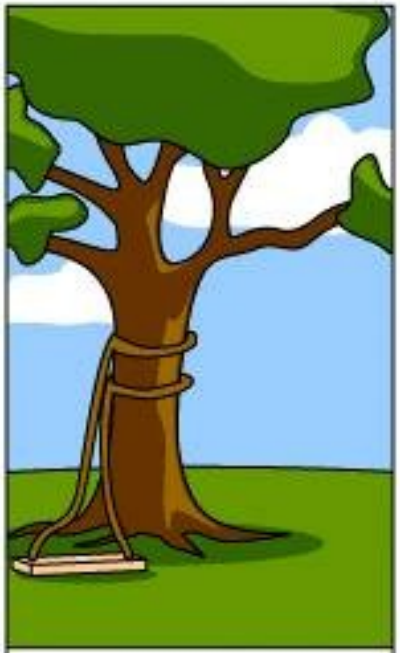
- Weitere Annäherung der funktionierenden und nicht-funktionierenden Varianten
  - mit und ohne JNI
- Mini-Kiek
  - Modellierungsprozess auf nächst-höherer Ebene
- Bottom-up
  - bis zur GUI

How the customer explained it



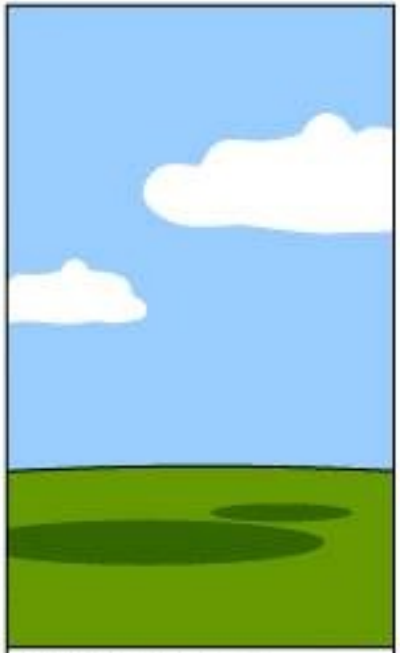
How the Project Leader understood it

How the Analyst designed it

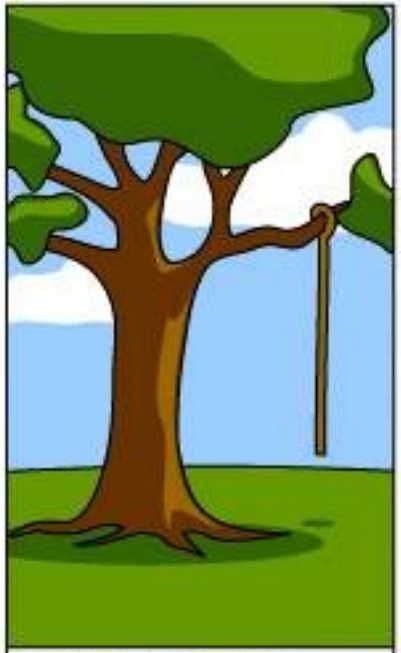


How the Programmer wrote it

How the Business Consultant described it

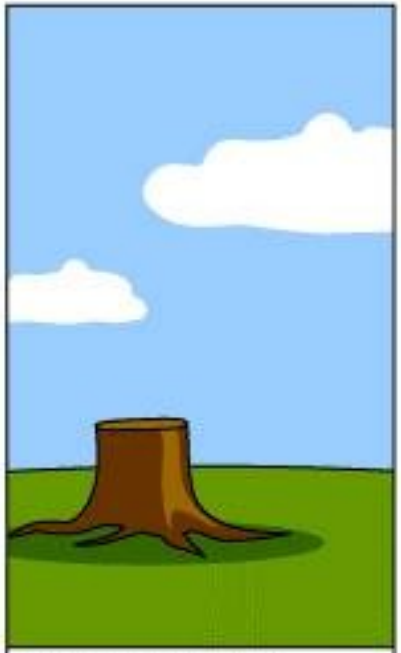


How the project was documented



What operations installed

How the customer was billed



How it was supported

What the customer really needed