

Teil 2

SESAM: Grundidee und Überblick

Jochen Ludewig

1. Der Hintergrund

Als 1989 die Gruppe langsam auf ihre Sollgröße wuchs, stellte sich die Frage, an welchem Thema sie denn arbeiten sollte. Naheliegende, wenn auch keineswegs ganz bewußte Randbedingungen waren:

- Das Thema soll praxisnah sein, also nicht von einem unsinnig vereinfachten Bild der Praxis ausgehen, und Resultate liefern, die in der Praxis eingesetzt werden können.
- Es soll die spezifische Stärke der Hochschule zur Geltung bringen, nämlich die Freiheit von kurzfristigen Rentabilitätsüberlegungen und die Freiheit zur firmenübergreifenden Forschung.
- Es soll Raum für viele miteinander verbundene Arbeiten bieten und in absehbarer Zeit nicht zu erschöpfen sein.
- Es soll den Doktoranden eine auch im Hinblick auf ihre spätere Arbeit nützliche Erfahrung verschaffen.
- Und es soll allen Beteiligten Spaß machen.

So kam mir die Idee, den Prozeß der Software-Entwicklung in einem Computer-Spiel zu simulieren. „Auslöser“ war ein Zeitungsartikel über Ökopolopoly von F. Vester; viele Erinnerungen und Erfahrungen, beginnend mit meiner Diplomarbeit (ein Simulationssystem, 1973), auch frühe Spiele auf der VAX („Dungeon“), bildeten den Hintergrund.

2. Die Zielsetzung

Was soll SESAM? Das Projekt („Software-Engineering-Simulation durch animierte Modelle“) hat das Ziel, ein Software-System zu schaffen, das auf einer Workstation läuft und von einer Person, dem *Spieler*, bedient wird. Der Spieler wird durch die Mitteilungen des SESAM-Systems mit Informationen über ein Software-Projekt versehen. Er kann den Verlauf dieses Projekts in ähnlicher Weise beeinflussen wie bei realen Projekten der Projektleiter und es dadurch mehr oder minder gelingen oder scheitern lassen.

Bei der Ausbildung im Flugsimulator findet keine reale Flugbewegung statt, nur die Daten einer simulierten Bewegung werden erzeugt. Ebenso entsteht beim Spiel mit SESAM keine Software, nur Daten der Software, z.B. quantitative und qualitative Merkmale ihrer Komponenten, werden aus dem Spielverlauf berechnet.

SESAM kann aus verschiedenen Perspektiven beschrieben und präsentiert werden:

- A. Der Blickpunkt des Spielers ist klar und lädt zur Identifikation ein („das möchte ich auch spielen“).
- B. Aus der Sicht des forschenden Ingenieurs stellt SESAM, wenn es im Spiel ein plausibles Verhalten zeigt, eine kompakte Codifizierung wichtiger Gesetzmäßigkeiten im Software Engineering dar.

(A) SESAM als Spiel

Wenn uns ein solches System zur Verfügung steht, dann können wir auch Dinge unterrichten, die in der traditionellen Weise so gut wie gar nicht zu vermitteln sind. Denn die reale Welt der Software-Bearbeitung ist im Sinne der reinen Lehre überwiegend durch „Schmutzeffekte“ geprägt. Alle möglichen persönlichen Beweggründe, überraschende Ereignisse und zufällige, aber kaum zu ändernde Randbedingungen prägen die Resultate oft stärker als rationale Entscheidungen.

Solche Effekte können wir simulieren und damit erfahrbar machen. SESAM ist also ohne Frage attraktiv als Lehrmittel, und das nicht nur an der Hochschule, sondern überall, wo Software entwickelt wird.

(B) SESAM als Modell

Über Software Engineering gibt es seit einigen Jahren eine ganze Reihe teilweise dicker Bücher. Man sollte also meinen, daß eine ganze Menge Wissen verfügbar sein müßte. Das gilt aber nur mit erheblichen Einschränkungen. Wir wissen heute vor allem, wie man *nicht* vorgehen sollte, und wir können unsere Erfahrungen in Aussagen kleiden, die nach ihrer Präzision eher der Medizin des 19. Jahrhunderts denn der Physik des 20. ähneln.

In diesem Sinne ist SESAM das Modell, das die „Bauernregeln“ des Software Engineerings präzisiert und quantifiziert, so daß aus konkreten Daten erstmals konkrete Schlüsse gezogen werden können. Hier liegt der wissenschaftliche Reiz und die akademische Herausforderung des Projekts.

3. Der Stand mit SESAM-1

Darum haben wir uns bislang kaum mit Fragen befaßt, die bei der Entwicklung eines Abenteuerspiels auf dem Rechner scheinbar vorrangig sind, beispielsweise mit der möglichst luxuriösen Benutzerschnittstelle. Die meiste Zeit ist damit vergangen, die

abstrakte Architektur des Systems zu schaffen. Damit ist die Beziehung zwischen Modell und Simulator gemeint.

Wir arbeiten an einem ausführbaren Modell, von dem wir wissen, daß es alles andere als perfekt ist. Wir brauchen darum einen Simulator, der durch das Modell quasi parametrisiert wird. Nur so ist eine schnelle Evolution der Modelle möglich. Aber das Prinzip der Parametrisierung schränkt die Freiheit des Modell-Schöpfers ein. Was *nicht* durch Parameter gesetzt werden kann, das ist durch den Simulator vorgegeben und nur mit sehr großem Aufwand änderbar. Das nach drei Prototypen jetzt fertiggestellte System SESAM-1 ist nach unserer Einschätzung für einige Jahre als Werkzeug tragfähig. Die Entwicklung der Modelle ist zurückgeblieben, denn bislang hatten wir keine Möglichkeit, neue Modelle in kurzer Zeit zu formalisieren und zu erproben.

Wir stehen also bei SESAM heute an einem Meilenstein: Wir zeigen das Werkzeug und seine Komponenten als wichtiges Halbprodukt; in den nächsten Monaten und Jahren werden wir vor allem die Evolution des Modells forcieren.

4. Die Modellierung des Unbekannten

Bei der Konzeption von SESAM standen wir immer wieder vor einem Dilemma: Simulieren kann man alles, was man gut verstanden hat. Der Prozeß der Software-Entwicklung ist aber keineswegs gut verstanden, und so ist er auch der Formalisierung, damit der Simulation entzogen. Andererseits ist diese Situation in der Wissenschaft nicht ungewöhnlich: Gerade durch eine – unzulängliche – Präzisierung des Problems wird sichtbar, wie eine bessere aussehen könnte.

Hier ist ein Zitat aufschlußreich: Hj. Siegenthaler, Prof. für Wirtschaftsgeschichte am Sozioökonomischen Seminar der Universität Zürich, schreibt in der NZZ vom 16.10.1993 in einem Artikel zur Vergabe des Nobelpreises für Wirtschaftswissenschaften

(„Neuer Blick in die Geschichte: Die innovativen Ansätze Robert Fogels und Douglass C. Norths“):

Fogel *quantifiziert* systematisch alle Feststellungen, die er zur Begründung seiner Vorstellungen trifft. Er tut dies auch dort, wo die Datenlage seinen Quantifizierungsversuchen nicht eben entgegenkommt. Dabei rückt er die Bedeutung statistischer Verfahren aus dem Zwielficht zweifelhafter Wahrheitsansprüche sehr entschieden heraus: Quantifizierung begründet keine Wahrheit, aber wer auf sie verzichtet, schreckt davor zurück, sich einer immerhin *kritisierbaren* «Wahrheit» überhaupt zu stellen.

Dies ist auch unser Ansatz: Wir wissen sehr wenig über die Zusammenhänge im Software Engineering. Darum simulieren wir sie.

5. Überblick zu den Beiträgen

Aus der Sicht des Spielers ist SESAM einfach ein großes, komplexes System. Aus der Sicht der Entwickler zerfällt es in viele Komponenten, deren Abgrenzung uns beträchtliche Mühe gemacht hat und die wir heute als ein wesentliches Resultat unserer Arbeit betrachten.

Die folgende schematische Darstellung zeigt die Gliederung und die Themen der Beiträge.

Kurt Schneider befaßt sich mit der Gliederung selbst, dann vor allem mit dem Zusammenspiel von Schema, Regeln und Situationsmodell.

Jinhua Li geht auf die Realisierung des Simulationssystems ein.

Anke Drappa diskutiert die verschiedenen Möglichkeiten, Informationen für den Modellbauer zu beschaffen, also die Quellen für ein Modell.

Marcus Deininger betrachtet SESAM aus der Sicht des Lehrers, also den Einsatz von SESAM.

Jürgen Schwille präsentiert einen wichtigen Implementierungsaspekt, den generischen Editor vis-A-vis, der den drei Editoren für den Modell-Bauer zugrundeliegt.

