

SESAM – Motivation und Ziele

Problemstellung

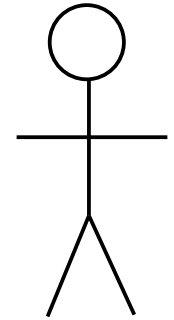
- ❑ Ausbildung von Projektleitern ist oft unzureichend
- ❑ Vermittlung von Projekt-Management-Fähigkeiten ist schwierig
- ❑ aktuelle Ansätze:
 - ❑ Ausbildung an Hand von Büchern (☞ unzureichend)
 - ❑ Erfahrung durch Leitung realer Projekte (☞ riskant/ teuer)

SESAM – Motivation und Ziele

Zielsetzung

- ❑ neuer Ansatz zur Ausbildung (angehender) Projektleiter
 - ❑ Software Engineering Simulation durch Animierte Modelle
 - ❑ Ausbildung durch interaktive Simulation von Software-Entwicklungsprojekten

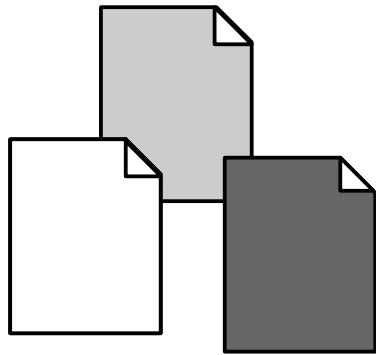
SESAM-Grundidee



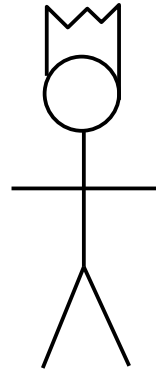
Projektleiter

SESAM-Grundidee

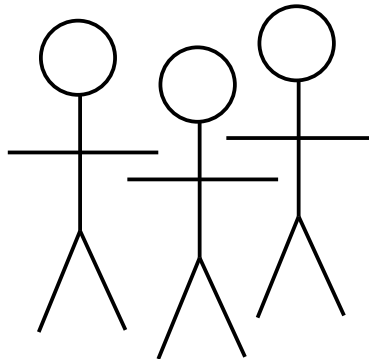
SESAM



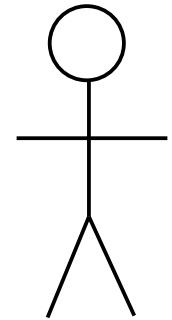
Dokumente



Kunde

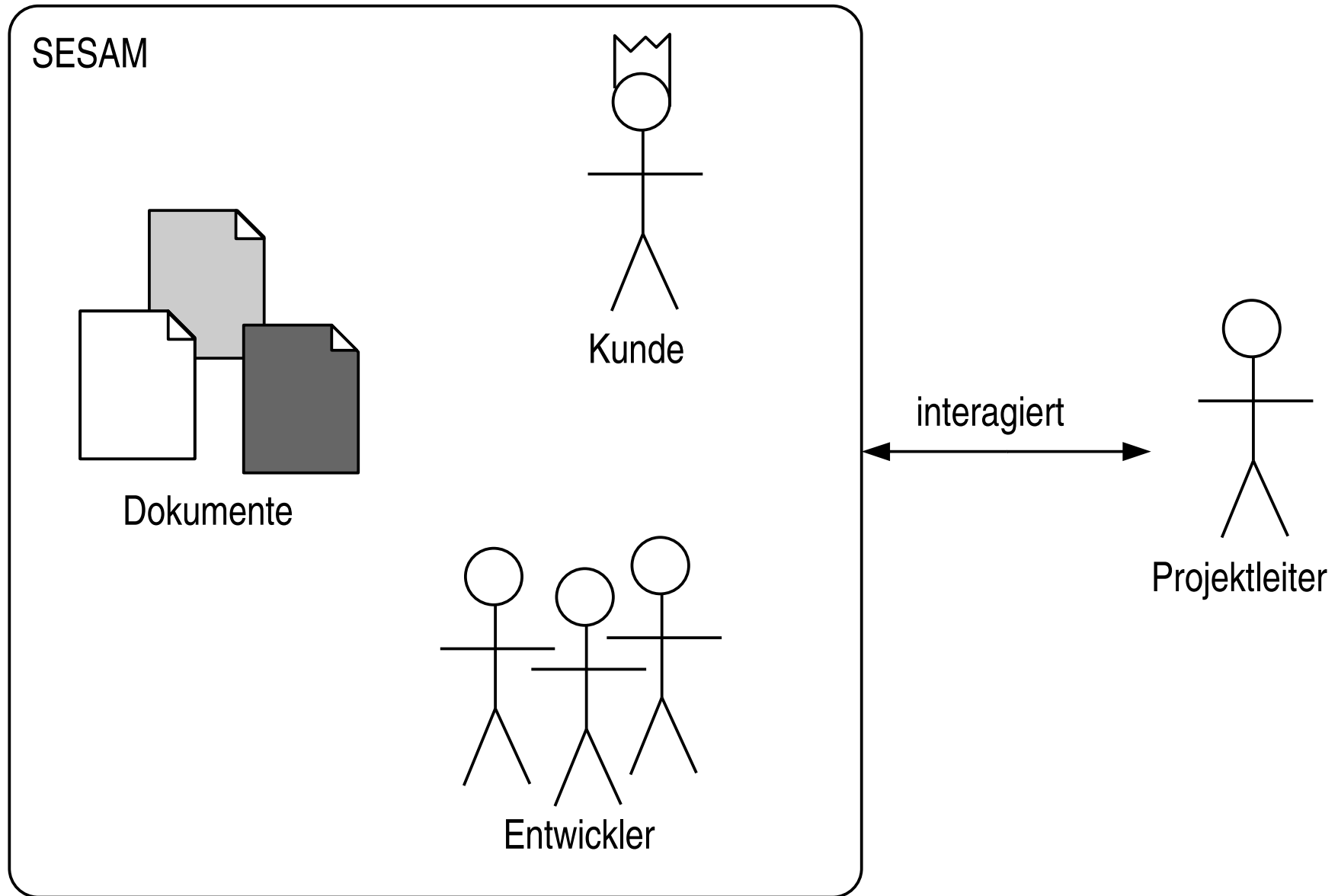


Entwickler

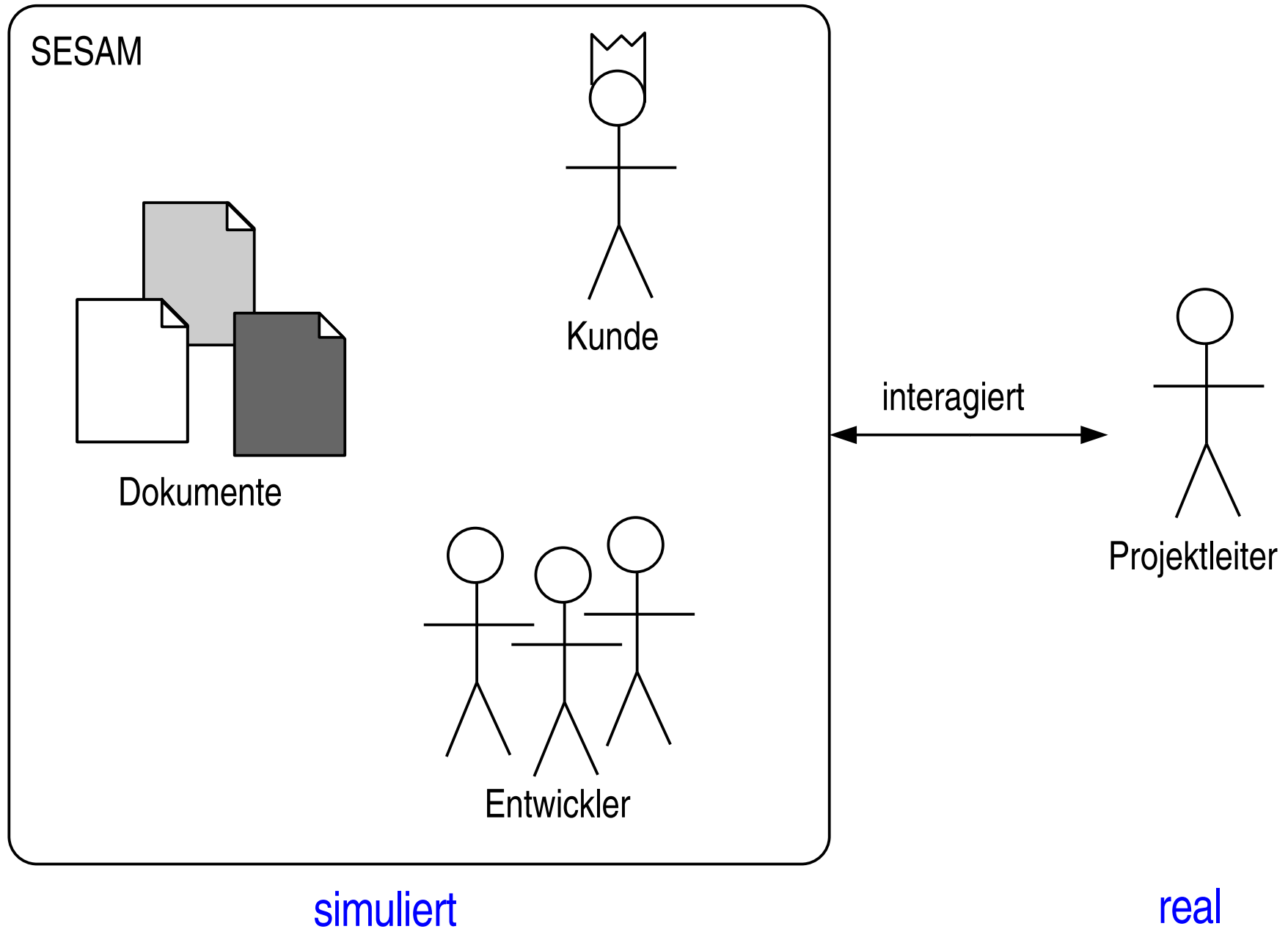


Projektleiter

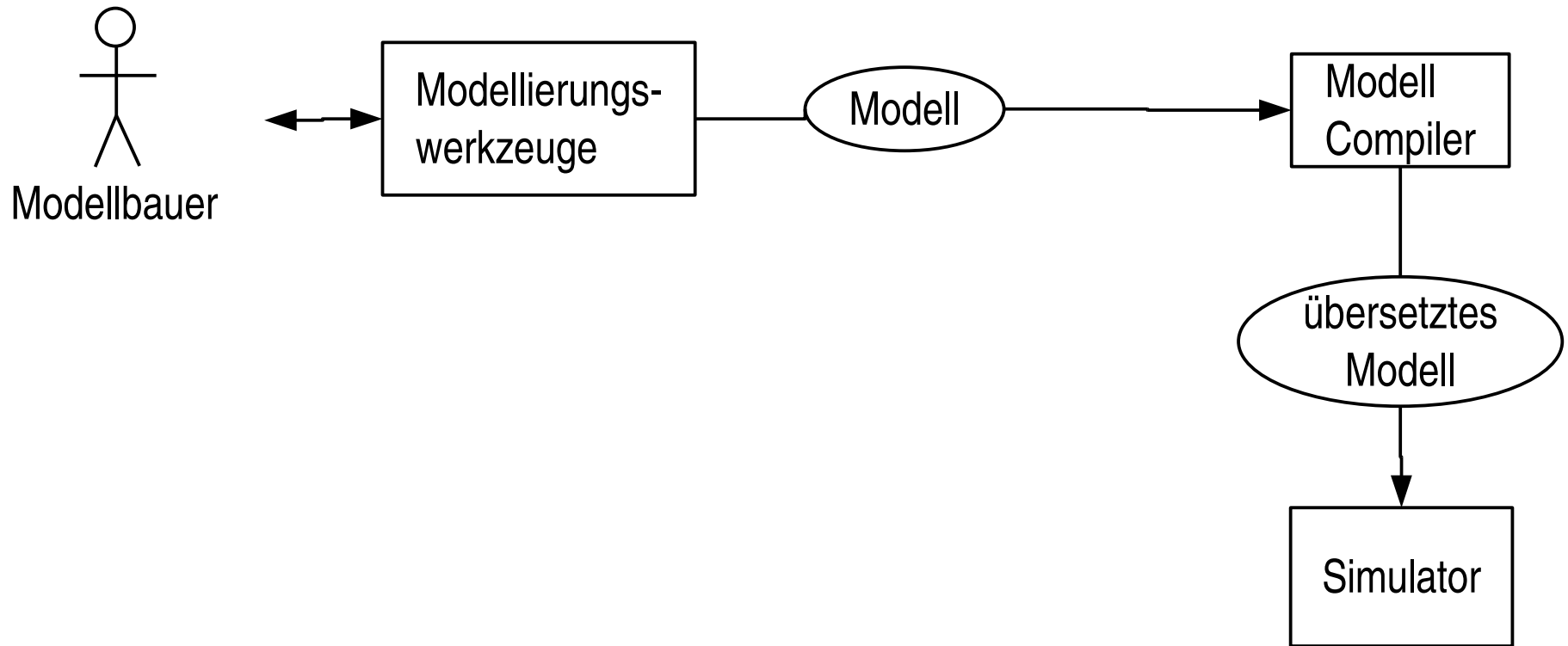
SESAM-Grundidee



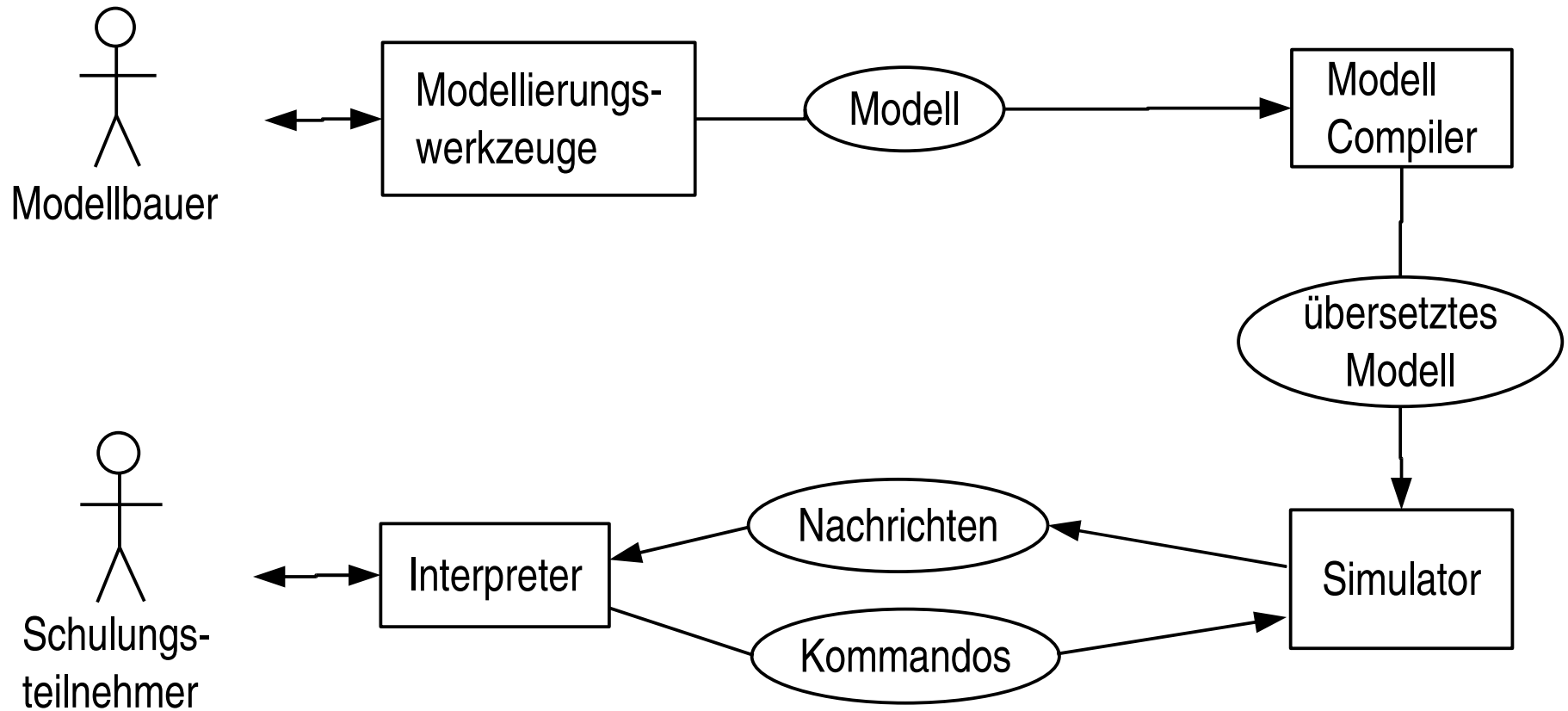
SESAM-Grundidee



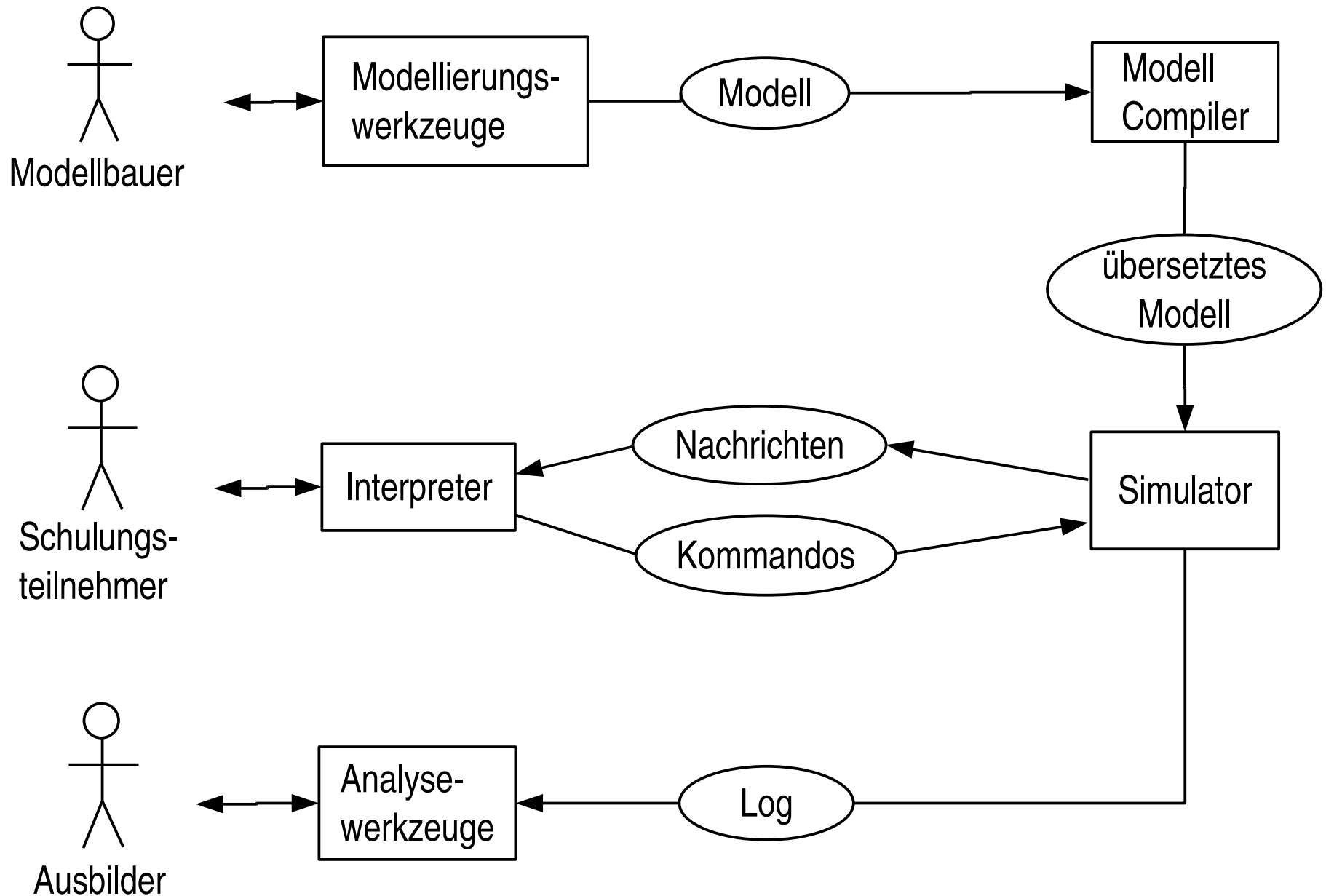
Das SESAM-System – Ein Überblick



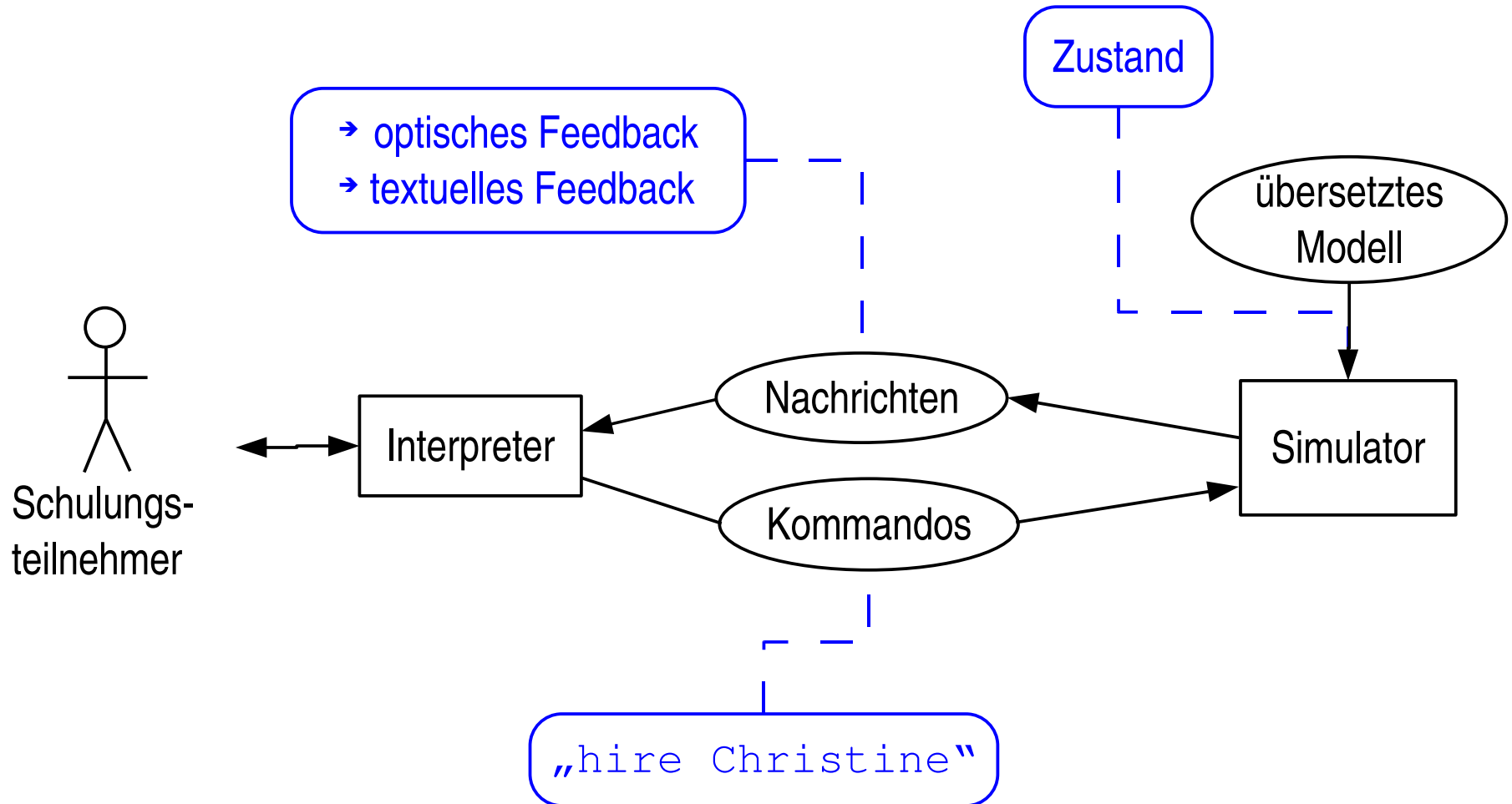
Das SESAM-System – Ein Überblick



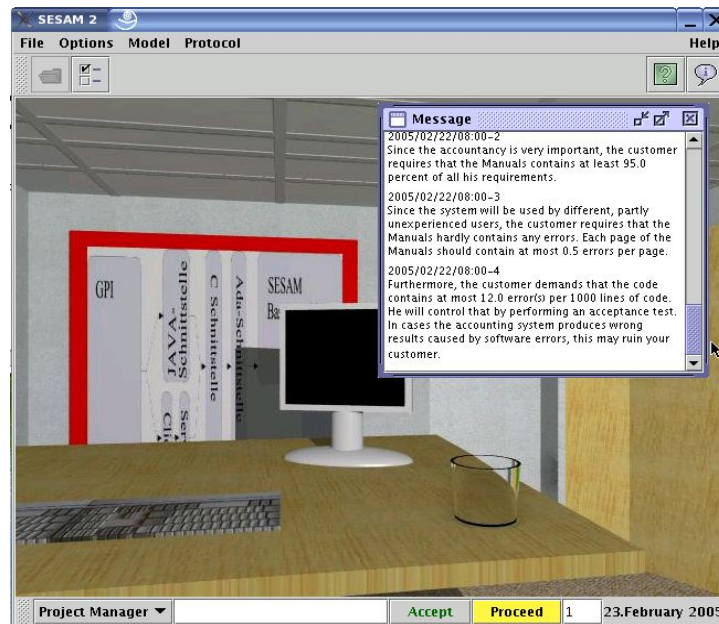
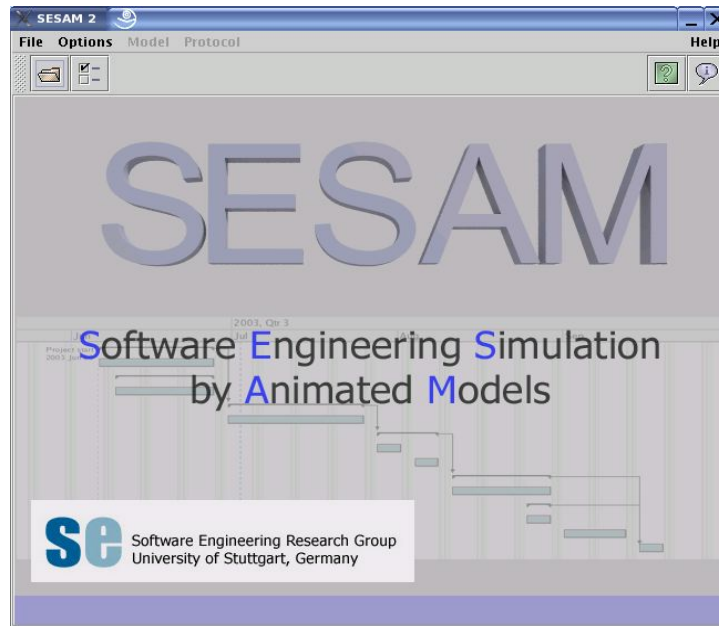
Das SESAM-System – Ein Überblick



Das SESAM-System – Ein Überblick

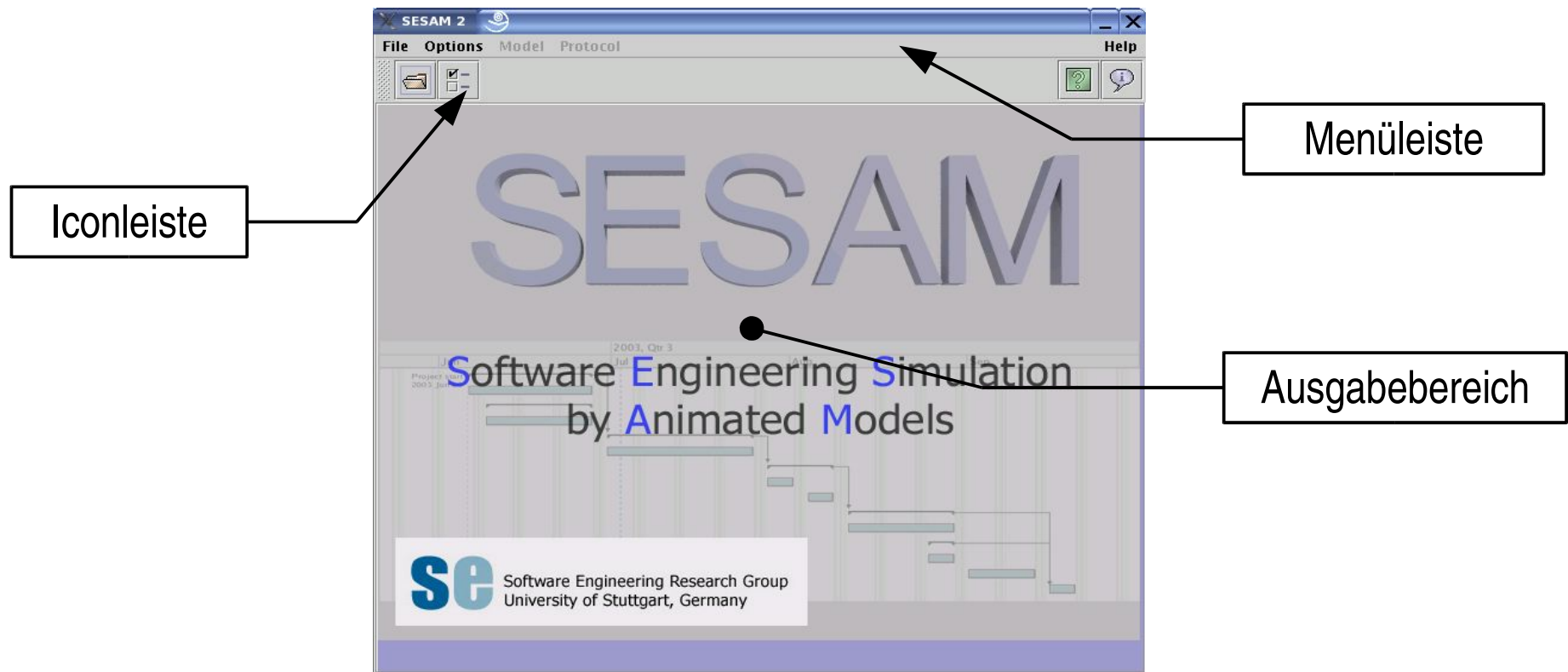


Spielerschnittstelle



Bedienung des Simulators

- ❑ Starten des Simulators
 - ❑ Eingabe des Befehls `sesam`
- ❑ Die graphische Benutzungsoberfläche





❑ Menüleiste

❑ Kommandos im Menü File

- ❑ Load Model
- ❑ Exit

❑ Kommandos im Menü Options

- ❑ Game Options
- ❑ Show Tips on Startup

❑ Kommandos im Menü Model

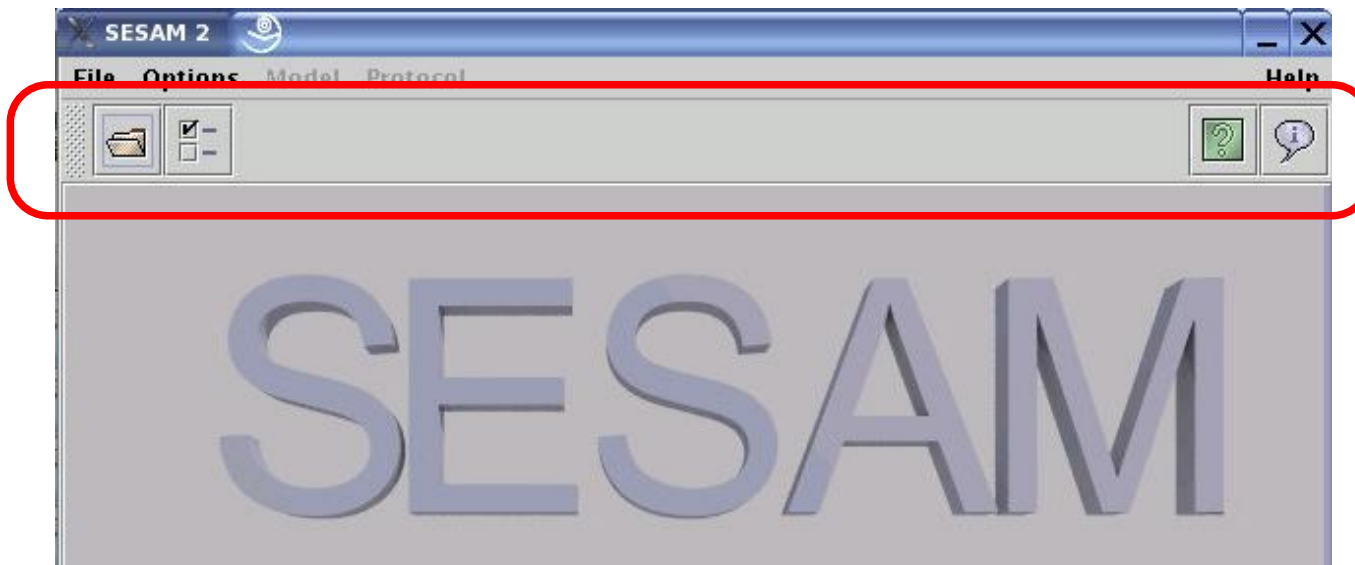
- ❑ Load State
- ❑ Save State

❑ Kommandos im Menü Help

- ❑ Help
- ❑ About

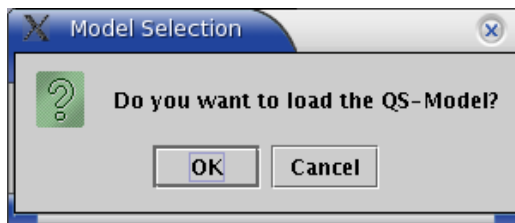
❑ Kommandos im Menü Protocol

- ❑ Open Protocol

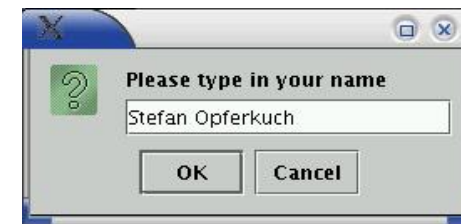


- ❑ Iconleiste

- ❑ Load a Base2 Model



- ❑ Set Game Options



- ❑ Show Online Help
 - ❑ Show About

❑ Räume in SESAM – Cafeteria





- ❑ Eingabe von Kommandos
 - ❑ Eingabe mit <RETURN> bestätigen oder
 - ❑ Accept drücken
 - ❑ **Wichtig:** jedes ausgeführte Kommando benötigt Zeit!



- ❑ Eingabe von Kommandos
 - ❑ Eingabe mit <RETURN> bestätigen oder
 - ❑ Accept drücken
 - ❑ **Wichtig:** jedes ausgeführte Kommando benötigt Zeit!
- ❑ Zeitfortschaltung (Proceed)
 - ❑ Proceed drücken → Zeitfortschaltung um die Anzahl der rechts daneben stehenden Schritte



- ❑ Eingabe von Kommandos
 - ❑ Eingabe mit <RETURN> bestätigen oder
 - ❑ Accept drücken
 - ❑ **Wichtig:** jedes ausgeführte Kommando benötigt Zeit!
- ❑ Zeitfortschaltung (Proceed)
 - ❑ Proceed drücken → Zeitfortschaltung um die Anzahl der rechts daneben stehenden Schritte
- ❑ Raumwechsel
 - ❑ Cafeteria
 - ❑ Project Manager
 - ❑ <Mitarbeiter>'s Office
 - ❑ Kommandos sind raumgebunden

❑ Räume in SESAM – Mitarbeiterzimmer

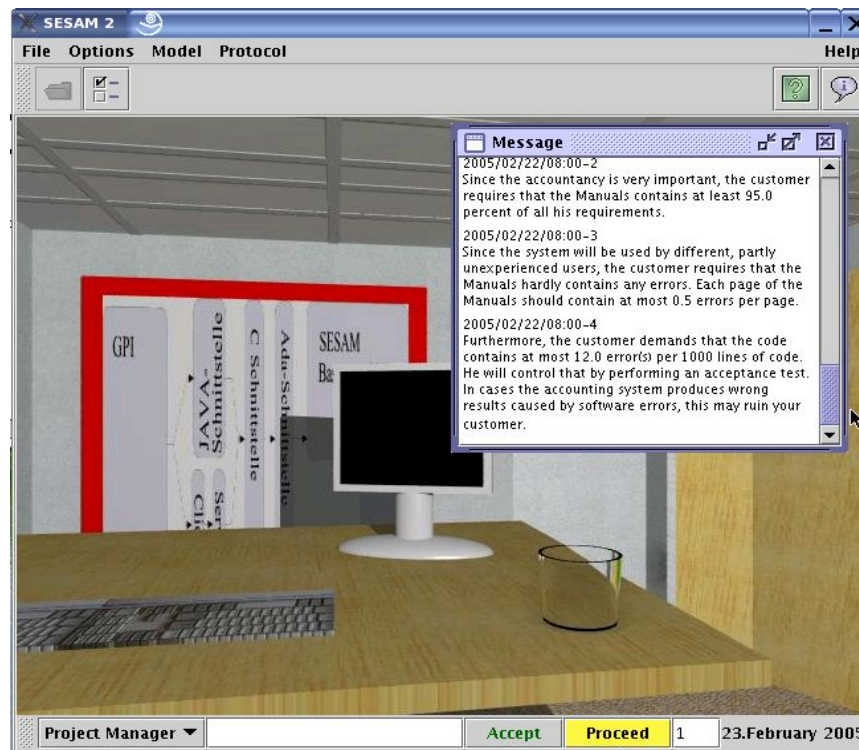


Nachricht von SESAM

Nachrichten von SESAM

- ❑ Nachrichten von Mitarbeitern
- ❑ Nachricht über Beginn und Ende von Tätigkeiten
- ❑ Informationen über Entwickler

❑ Räume in SESAM – Das Zimmer des Projektleiters



Nachrichten von SESAM – nur im Projektleiterzimmer

- ❑ Projektstart und -ende
- ❑ Fortschritt von Reviewsitzungen
- ❑ Zustand von Dokumenten
- ❑ Ende von Tätigkeiten der Entwickler, falls nicht im Mitarbeiterzimmer

Simulation eines Softwareprojekts

Laden eines Modells (Load Model)

- ❑ Voraussetzung: Simulator ist gestartet
- ❑ Dialog Model Selection aufrufen
 - ❑ Menüpunkt File → Load Model anwählen oder
 - ❑ Tastenkombination <Ctrl>-l drücken oder
 - ❑ Icon Load a Base2 Model anklicken
- ❑ Frage Do you want to load the QS-Model? mit OK bestätigen
- ❑ Ladevorgang ist beendet, wenn Cafeteria und Mitarbeiter erscheinen

Spielen mit dem Modell

- ❑ Beginnen eines *neuen Spiels* nach dem Laden eines neuen Modells
- ❑ Spiel durchführen
 - ❑ beliebige Folge von Kommandos und `Proceeds` eingeben
 - ❑ Spieler erhält auf einige Kommandos sofort Rückmeldung (Bsp.: `determine spent resources`)
 - ❑ bei anderen Kommandos erfolgt die Meldung erst nach einem `Proceed` (Bsp. `create specification`)
 - ❑ **Achtung:** `Proceed` nicht zu exzessiv verwenden
 - ❑ **besser:** sinnvolle Aktionen durchführen:
 - ❑ Jede Aktion des Projektleiters verbraucht Zeit
 - ❑ Die angefallene Projektlaufzeit führt dazu, dass ein Zeitschritt fortgefahren wird und die Ausgaben erscheinen

Laden und Speichern von Spielständen

(Load State und Save State)

- ❑ Voraussetzung: Modell geladen
- ❑ Laden und Speichern eines Spielstands ist zu jedem beliebigen Zeitpunkt möglich
 - ❑ Menüpunkt Model → Load State bzw. Save State auswählen
 - ❑ Speichern: Dateiname für den zu speichernden Spielstand auswählen oder neu eingeben
 - ❑ Laden: Dateiname des zu ladenden Spielstands (Endung .sit) auswählen
 - ❑ Ladevorgang ist beendet, wenn rechts unten das neue Datum erscheint

Laden und Speichern von Spielständen

(Load State und Save State)

- ❑ Wichtig: In regelmäßigen Abständen den Spielstand speichern, um im Notfall auf den zuvor gesicherten Spielstand wieder aufsetzen zu können

Das QS-Modell

Modellkonzepte und Metriken

- ❑ Dokumente
 - ❑ Inhalt: # Adjusted Function Points (AFP)
 - ❑ Qualität: # Fehler
 - ❑ Größe: # Seiten, # Lines of Code
- ❑ Entwickler
 - ❑ Kosten: Kosten pro Stunde (in EUR)
 - ❑ Erfahrung: keine/ niedrig/ mittel/ hoch
(Aktivitäten)
- ❑ Aktivitäten
 - ❑ Produktivität: # AFP/ Stunde
 - ❑ Fehlerrate: # Fehler pro AFP
 - ❑ Verlustquote: % AFP

Kommandoübersicht

Kommandos zur Produktion von Software

- ❑ ask <Name> to write the code
- ❑ do the system design
- ❑ ...

Vorgehen

- ❑ Entwickler schnüren sich bestimmte Anzahl Funktionen zu einem Arbeitspaket
- ❑ nur ein Entwickler ist Autor einer bestimmten Funktion in einem speziellen Dokument
- ❑ Abarbeiten eines Arbeitspakets dauert einige Tage
- ❑ Entwickler nehmen sich solange neue Arbeitspakete, bis es nichts mehr zu tun gibt, der Projektleiter ihnen die Tätigkeit entzieht oder eine andere Aufgabe zuteilt

Kommandos zum Prüfen von Software

- ❑ ask <Name> to test the modules
- ❑ ask <Name1> and <Name2> to review <Dokumentname>
- ❑ ...

Vorgehen

- ❑ Gutachter prüfen bei einem Review je Sitzung eine bestimmte Anzahl Funktionen bzw. entsprechende Anzahl Seiten
- ❑ alle 3 Tage findet eine Review-Sitzung statt, solange, bis entweder der Prüfling vollständig geprüft ist oder aber der Projektleiter den Abbruch des Reviews befiehlt
- ❑ bei einem Review werden nur diejenigen Funktionen geprüft, von denen *keiner* der Gutachter Autor ist (d.h., dass Funktionen, deren Autor einer der Gutachter ist, nicht als Arbeitspaket für das Review zusammengeschürt werden)

Kommandos zur Korrektur von Software

- ❑ `ask <Name> to correct the module design`
- ❑ `correct specification`
- ❑ `...`

❑ Vorgehen

- ❑ nach erfolgter Prüfung können die Entwickler Dokumente korrigieren
- ❑ der Projektleiter hat die Wahl, entweder *alle* Dokumente oder *nur den* Prüfling korrigieren zu lassen
- ❑ Entwickler korrigieren nur solange noch etwas zu tun ist → bei paralleler Prüfung und Korrektur muss man ggf. den Auftrag zur Korrektur mehrfach erteilen

Inspektoren

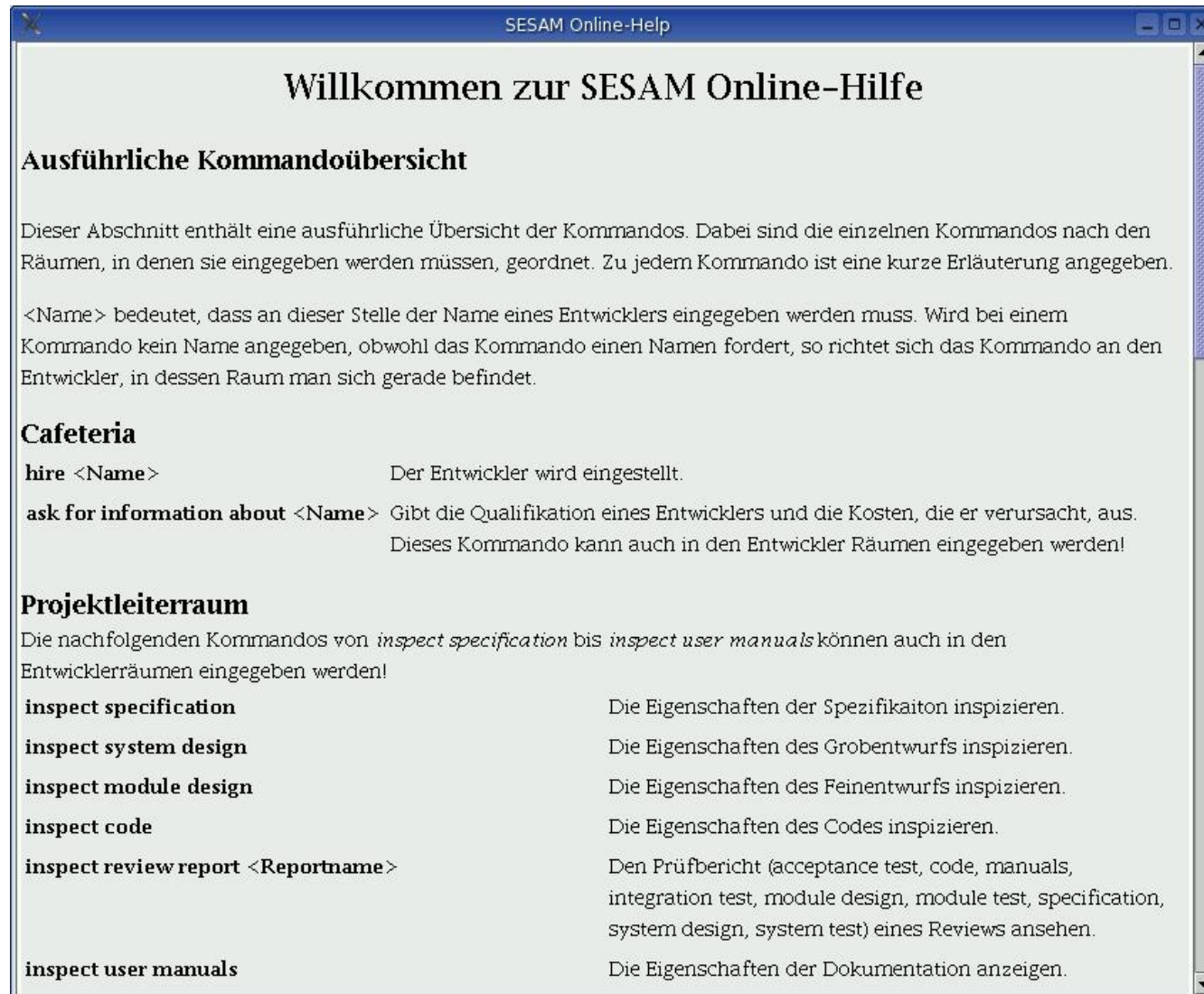
- ❑ Stand des Projekts durch verschiedene Kommandos grob überprüfen
 - ❑ `determine spent resources`
 - ❑ `inspect specification`
 - ❑ ...
- ❑ Achtung: Informationen z.T. bewusst ungenau
 - ☞ selbst den Überblick bewahren!
 - ☞ eigene Planung außerhalb des Simulators machen!

Sonstige Kommandos

- ❑ `finish task`
- ❑ `deliver system`
- ❑ ...

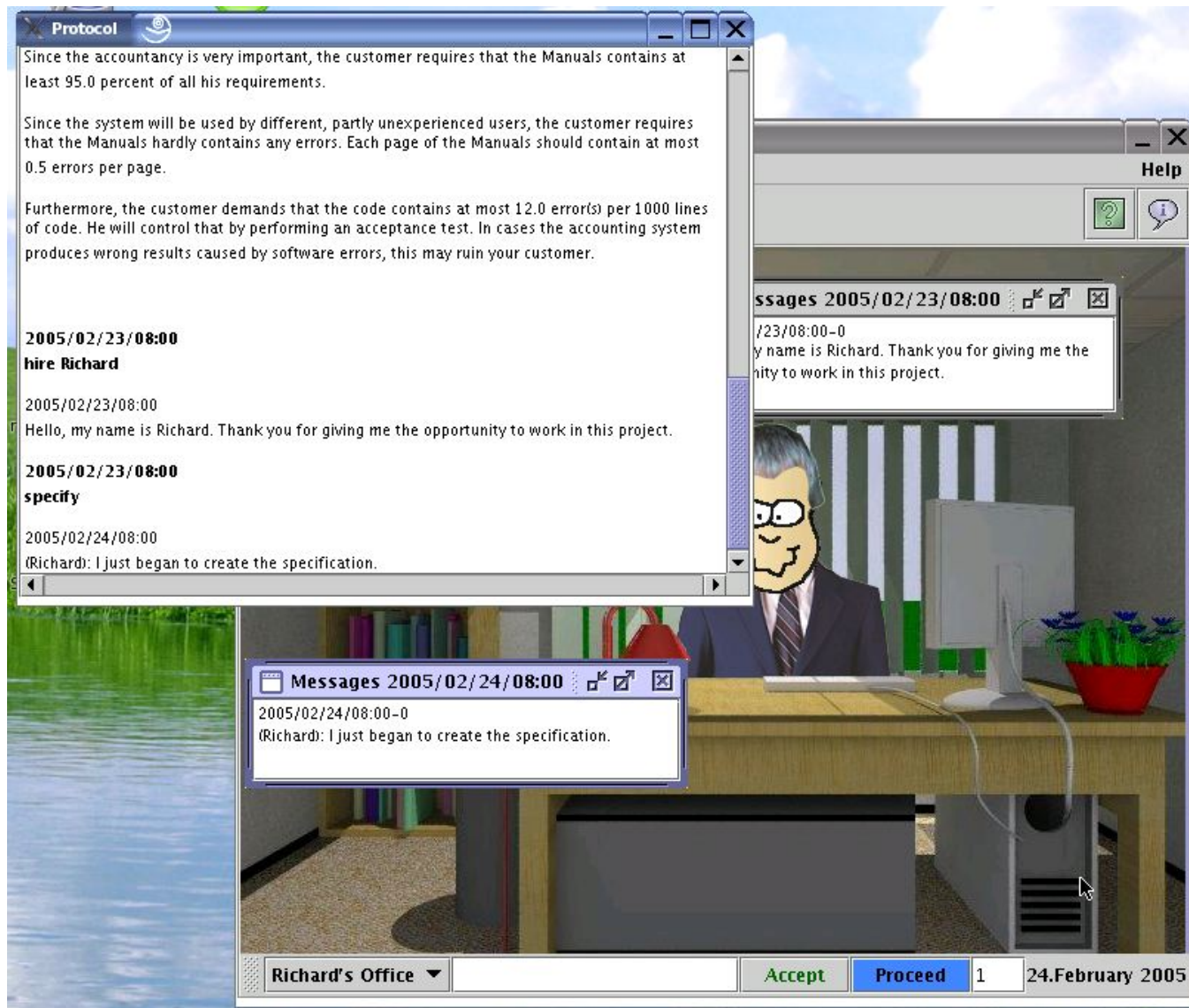
Die Online-Hilfe

Menüpunkt Help



Das Protokoll-Fenster

Menüpunkt Protocol



Besonderheiten des Modells

- ❑ Ein Entwickler kann zum gleichen Zeitpunkt nur eine Aufgabe ausführen (Ausnahme: Reviews)
- ❑ Wird einem Entwickler, der noch nicht eingestellt ist, eine Tätigkeit zugewiesen, führt er diese Tätigkeit *nicht* aus. Nach seiner Einstellung muss die Tätigkeit erneut vergeben werden.

Besonderheiten des Modells

- ❑ Werden abgesetzte Kommandos nicht sofort oder nach spätestens einem `Proceed` mit einer Nachricht bestätigt, gilt, dass das Kommando *nicht* ausgeführt wurde.
- ❑ Ausnahmen:
 - ❑ `finish task` und `quit current task`
 - ❑ `cancel project` und `deliver system`

- ❑ Beispiele für Situationen, in denen das Kommando ohne Wirkung bleibt:
 - ❑ einem Entwickler wird eine Tätigkeit erneut zugeteilt, an der er aktuell noch arbeitet
 - ❑ `fire`: ein Entwickler kann nur aus dem Projekt entlassen werden, wenn er aktuell keine Tätigkeit ausübt
 - ❑ `hire` und `fire`: mit seiner Entlassung wird der Entwickler für eine zufällige Zeitspanne zwischen 21 und 60 Tagen gesperrt. `hire` bleibt solange wirkungslos, bis der Entwickler wieder verfügbar ist.
Beachte: `hire` und `fire` sind hier nur projektbezogen!

- ❑ Beispiele für Situationen, in denen Rückmeldungen auch noch nach mehr als einem Zeitschritt erfolgen können:
 - ❑ `finish task`
 - ❑ `deliver system`
 - ❑ ein Entwickler, der aktuell noch mit einer Aufgabe beschäftigt ist, erhält einen neuen Auftrag

Hinweise und Tipps

- ❑ RTFM
- ❑ Papier und Bleistift evtl. Projektplanungstool bereit halten
- ❑ Parallelisieren
- ❑ Überlappen
- ❑ Prüfung impliziert nicht Korrektur
- ❑ Autoren sind schlechte Wahl als Gutachter
- ❑ Gutachter arbeiten max. 2h am Tag am Review; Reviews nur alle drei Tage
- ❑ Dokumente müssen nicht vollständig sein fürs Review
- ❑ Kunden einbinden

Hinweise und Tipps

- ❑ Neue Mitarbeiter einstellen
 - ❑ brauchen Einarbeitungszeit (je später sie dazu kommen, desto mehr Zeit brauchen sie)
 - ❑ verringern auch Produktivität anderer Mitarbeiter
- ❑ „Hire and Fire“ wohl dosieren:
 - ❑ Untätige Mitarbeiter kosten Geld
 - ❑ Produktivität von Mitarbeitern variiert mit Fähigkeiten im jeweiligen Einsatzgebiet
 - ❑ Mitarbeiter sind nach dem Feuern für gewisse Zeit gesperrt

Organisatorisches

- ❑ **Einführung am Rechner**
- ❑ **1. SESAM-Spiel:** bis ...
- ❑ **Feedback zum Spiel:**
- ❑ **2. SESAM-Spiel:** von ... bis ...

- ❑ zur Verfügung stehende Rechner
 - ❑ Linux-Kisten x^* in Ebene 0
- ❑ Nach Spiel: *Spiel_final* im Homeverzeichnis abspeichern
- ❑ Bitte Dateien *Spiel_final.sit*, *Spiel_final.sit.prot* und *Spiel_final.sit.gui* per Email an Pierre Frenzel.

Demonstration von SESAM

