HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

FHO Fachhochschule Ostschweiz

Evaluierung der Simulationssoftware AnyLogic, Arena und SIMIO

Bachelorarbeit

Abteilung Informatik
Hochschule für Technik Rapperswil

Frühjahrssemester 2013

Autor:	Urs Baumann
Betreuer:	Prof. Dr.-Ing Andreas Rinkel
Projektpartner:	IBM Zürich Research Laboratory, Rüschlikon, ZH
Experte:	Dr. Ulrich Schimpel, IBM Zürich Research
Gegenleser:	Prof. Stefan F. Keller

ERKLÄRUNG

Ich erkläre hiermit,

- dass ich die vorliegende Arbeit selber und ohne fremde Hilfe durchgeführt habe, ausser derjenigen, welche explizit in der Aufgabenstellung erwähnt ist oder mit dem Betreuer schriftlich vereinbart wurde,
- dass ich sämtliche verwendeten Quellen erwähnt und gemäss gängigen wissenschaftlichen Zitierregeln korrekt angegeben habe.
- das ich keine durch Copyright geschützten Materialien (z.B. Bilder) in dieser Arbeit in unerlaubter Weise genutzt habe.

Ort, Datum:

Name, Unterschrift:

AUFGABENSTELLUNG

Ausgangslage, Problemstellung

Im Modul System Modeling und Simulation wird momentan mit der Software Arena von Rockwell gearbeitet. Geplant ist, dass diese Software nächstes Jahr durch eine modernere Applikation, wie Simio oder AnyLogic abgelöst wird. Beide Programme verfolgen zum Teil unterschiedliche Modellierungsansätze und sind daher schwierig zu vergleichen.

Im Rahmen der Bachelorarbeit sind die Unterschiede der beiden Softwarelösungen anhand eines universellen Beispiels aufzeigen und zu bewerten. Anschliessend sind die Ergebnisse mit den Erfahrungen von Arena gegenüber zu stellen.

Ziele der Arbeit

- Inbetriebnahme der Software AnyLogic und Simio
- Einarbeitung in die beiden Programme und Aneignung der Simulationsgrundlagen.
- Entwicklung eines universellen Modelles,
- Entwicklung eines Vorgehensmodells zur Erstellung des Simulationsmodells in Simio und AnyLogic
- Optional sind die Schnittstellen der Simulationsapplikationen zu Fremdsystemen zu untersuchen

Zur Durchführung

Mit dem Betreuer findet in der Regel eine wöchentliche Besprechung statt. Das Sitzungsintervall ist flexibel und kann dem Bedarf angepasst werden. An den Besprechungen wird der aktuelle Stand des Projektes und das weiter Vorgehen besprochen. Das Projekt wird als Einzelarbeit durchgeführt.

Termine

Siehe Terminplan auf der HSR Seite: https://www.hsr.ch/Termine-Diplom-Bachelor-und.5142.0.html

Rapperswil, den

Verantwortlicher Dozent

ABSTRACT

Ausgangslage

Die Simulation als Hilfe zur Entscheidungsfindung spielt in Geschäfts- und Wirtschaftsbereichen eine immer grössere Rolle, da die zu optimierenden Prozesse oder Phänomene immer komplexer werden. Durch die enthaltene Dynamik kommt man bei diesen Problemen mit den analytischen Verfahren schnell an die Grenzen. An vielen Orten ist ein Ausprobieren oder Erstellen eines Prototyps nicht möglich, zu kostspielig oder zu gefährlich. Hier ist die Simulation das richtige Werkzeug, um die Prozesse zu simulieren und mit verschiedenen Parametern zu optimieren. Um erstellte Modelle zu simulieren, wird meist eine komplexe Simulationssoftware eingesetzt oder eine massgeschneiderte Applikation entwickelt. Im momentanen Markt gehören AnyLogic von XJ Technologies und SIMIO von SIMIO LLC zu den etabliertesten.

Vorgehen

Nach einer intensiven Einarbeitung in die Tools SIIO und AnyLogic und dem Besuch des Moduls «System Modeling and Simulation», das noch die Software Arena verwendet, werden die Grundlage der Simulation erarbeite. Um die Applikationen zu vergleichen, wird ein universelles Modell erarbeiten und mit den Tools simuliert. So lassen sich die Unterschiede der Tools gut herausarbeiten und studieren. Weiter wird untersucht, welche Schnittstellen zu externen Applikationen bestehen.

Ergebnis

Der Einstieg in AnyLogic und Arena ist einfacher und die Bedienung intuitiver als bei SIMIO. Hingegen wird der Modellierungsprozess in SIMIO deutlich einfacher wenn die grundlegenden Konzepte verstanden sind. AnyLogic zeichnet sich durch die Möglichkeit aus, verschiedene Modellierungstypen, wie zum Beispiel die agentenbasierte oder diskrete Modellierung zu unterstützen. Ferner erlaubt AnyLogic Javabefehlte zu hinterlegen. Damit können Prozesse sehr agil in AnyLogic realisiert werden. In SIMIO fehlt bis jetzt der Ansatz von Agentenmodellen. Dafür ist die Library für die diskreten Eventmodelle sehr mächtig und die Elemente können mit Add-on Prozessen, welche wiederum Eventbasiert sind, ergänzt werden. In Arena fehlen die modernen Modellierungsansätze, wie Objektorientierung und Datenkapselung. In SIMIO und in AnyLogic lassen sich schnell 2D/3D Animationen erstellen. Hier ist der Aufwand in Arena deutlich höher. In AnyLogic ist es möglich die Projekte in Java Applet zu exportieren. SIMIO und Arena sind im Bereich von Auswertungen und Generieren von Szenarien gegenüber AnyLogic klar im Vorteil.

INAHLTSVERZEICHNIS

Erklärung 2
Aufgabenstellung 3
Ausgangslage, Problemstellung 3
Ziele der Arbeit 3
Zur Durchführung 3
Termine 3
Abstract 4
Ausgangslage 4
Vorgehen 4
Ergebnis 4
Inahltsverzeichnis 5
A. Technischer Bericht 9
1 Einleitung 9
2 Umgebung 9
2.1 Wink 9
3 Modell 9
3.1 Entscheidung für ein Modell 10
3.2 Kurzbeschrieb 10
3.3 Ausblick 10
3.4 Modellierung 10
4 Umsetzung 11
5 Auswertung / Gegenüberstellung 11
5.1 Übersicht AnyLogic 11
5.2 Übersicht Arena 12
5.3 Übersicht Simio 12
5.4 Einfachheit 13
5.5 Animation 13
5.6 Szenarien 13
5.7 Auswertung 13
5.8 Externe Daten 13
5.9 Speed 13
5.10 Erweiterbarkeit 13
5.11 Debugging 14
5.12 Community 14
5.13 Zusammenarbeit 14

5.14 Tauglichkeit für Unterricht 14
B. Anleitung AnyLogic 15
1 Installation 15
2 Neues Projekt erstellen 16
2.1 Erste Simulation 16
2.2 Eigenes Element 17
3 Netzwerk 18
4 Netzwerkressourcen 23
5 Kochen 24
6 Personenbegrenzung 25
7 Prozess Verteilfunktionen 26
8 Zeitmessung 28
9 Diagramme 29
10 Parametrisierung 30
11 3D Animation 32
C. How To SIMIO 33
1 Installation 33
2 Neues Projekt erstellen 34
3 Ressourcen 35
4 Netzwerk 36
5 Neues Modell 37
6 Gruppengrösse 38
7 Ressource an das Modell Tisch übergeben 39
8 Aufteilung der Bestellung 40
9 Kochen 41
10 Symbole 42
11 Overflow 43
12 Prozess Verteilfunktionen 43
13 Debugging 44
14 Diagramme 44
15 Zeitmessung 45
16 Experimente 46
17 Kosten 47
18 Animation 47
19 OptQuest 48
D. Projektmanagement 49
1 Management Summary 49

1.1 Ausgangslage 49
1.2 Vorgehen 49
1.3 Ergebnisse 49
2 Projektplan 50
2.1 Projektorganisation 50
2.2 Zeitaufwand 50
2.3 Besprechungen 50
2.4 Infrastruktur 50
2.5 Risikomanagement 51
2.6 Arbeitspakete 51
2.7 Meilensteine 51
2.8 Zeitplan 52
2.9 Test 52
3 Selbstreflexion 53
4 Sitzungsprotokolle 54
4.1 Protokoll Kick Off Meeting 54
4.2 1. Sitzung 55
4.3 2. Sitzung 56
4.4 3. Sitzung 56
4.5 4. Sitzung 57
4.6 5. Sitzung 57
4.7 6. Sitzung 58
4.8 7. Sitzung 58
4.9 8. Sitzung 59
4.10 9. Sitzung 59
4.11 10. Sitzung 60
4.12 11. Sitzung 60
4.13 12. Sitzung 61
4.14 13. Sitzung 61
E. Anhang 63
1 Abbildungsverzeichnis 63
2 Modell 65
3 Zeitmanagement 69
4 Projektdokumentation Hardrock 71

A. TECHNISCHER BERICHT

1 Einleitung

Im Modul „System Modeling und Simulation“ wird momentan mit der Software Arena von Rockwell gearbeitet. Es ist geplant, dass diese Software nächstes Jahr durch eine modernere Applikation wie Simio oder AnyLogic abgelöst wird.

Die Simulatoren AnyLogic und Simio sind sehr komplex und verwenden zum Teil verschieden Ansätze. Damit die Applikationen besser verglichen werden können, wird ein Modell für eine Simulation erarbeitet und in beiden Programmen realisiert. Zudem wird jeden Simulator eine eigene Anleitung erstellt. Nach der Realisierung sind die Erfahrungen gegenüberzustellen und auszuwerten.

2 Umgebung

AnyLogic und Simio sind grosse und vielfältige Simulatoren. Um die Programme in der gleichen Umgebung zu testen, wurde je eine VMware-Umgebung mit Windows 7 x64 aufgesetzt. Damit das Verhalten mit verschiedenen Prozessoren ersichtlich war und zudem flüssig gearbeitet werden konnte, wurden der virtuellen Maschine vier Prozessoren und 4 GB RAM zugeteilt. Zu Beginn wurde nur mit einem Prozessor und 2 GB RAM gearbeitet. Beide Applikationen laufen problemlos mit den beschränkten Ressourcen.

2.1 Wink

Zur Erstellung der PrintScreens für die Anleitungen wurde das Freeware Tool Wink verwendet. Das Tool erstellt bei Maus- oder Tastaturevents PrintScreens. Zudem lassen sich Flash-Filme generieren, in welchen ersichtlich ist, wann welcher Event wo ausgelöst wurde.

Im Nachhinein lassen Sie die PrintsScreen bearbeiten und Texte einfügen. Im Rahmen dieser Arbeit wird das Projektfile für die verschiedenen Aufzeichnungen, sowie jeweils ein Flash-Film abgegeben. Die Filme erleichtert die Verständlichkeit der Anleitung. Mit den Projektfiles können die Filme weiter verfeinert oder weiter bearbeitet werden.

Die Applikation kann unter www.debugmode.com/wink/ herunter geladen werden.

Unter Windows 7 x64 gab es leider Probleme im Zusammenhang mit Eclipse. Wink erkannte nicht, wann ein Maus- oder Keybord-Event geworfen wurde. Es wurde ein Workaround erstellt, bei dem Autohotkey.com verwendet wurde, um bei jedem Event die „Pause“ Taste zu drücken.

3 Modell

Es wurde ein universelles Modell entwickelt, welches in den Tools AnyLogic, Arena und Simio umgesetzt werden kann. Wichtig war, dass das Modell möglich viele verschiedene Aspekte hatte und somit gut geeignet ist, die Unterschiede der Simulatoren aufzuzeigen.

Parallel zur Bachelorarbeit wurde das Modul „System Modeling und Simulation“ besucht bei welchem es ein Projekt zu realisieren gab. Die Erstellung des Modells wurde zu gleichen Teilen von der Bachelorarbeit und von dem Projekt im Modul erstellt. Die Simulation und ausführliche Auswertung des Modells in Arena, welches durch das Projekt im Modul erarbeitet wurde, finden Sie im Anhang. Das Projektteam bestand aus Reto Schelbert, Simon Schreiber, Tobias Blaser und Urs Baumann. Die Erkenntnisse flossen in die Modellbildung mit ein. Für die Gegenüberstellung der Tools werden die angeeigneten Erfahrungen beigezogen.

3.1 Entscheidung für ein Modell

Der Entscheid fiel auf die Simulation eines Restaurants, da ein Restaurant viele Aspekte hat und dennoch sehr anschaulich und illustrativ ist. Man kann sich sehr gut vorstellen, wie die Prozesse in einem Restaurant ungefähr aussehen. Ausserdem sind viele Parteien vorhanden, welche miteinander interagieren.

Durch die Bindung an ein reales System, dem Hardrock Café in Atlantic, konnten gewisse Rahmenbedingungen festgelegt werden.

Das Modell eignet sich zudem gut, um zunehmend detailliertere und vielfältigere Aspekte zu simulieren. Das Modell lässt sich einfach anpassen.

3.2 Kurzbeschrieb

Gäste betreten das Restaurant in Gruppen von eins bis vier Personen und gehen zuerst an die Bar, um zu fragen ob ein Tisch respektive ein Platz frei ist. Ist dies der Fall, gehen die Gäste gleich weiter zum Tisch und warten auf eine Bedienung. Ist kein Platz verfügbar warten die Gäste an der Bar. Warten zu viele Gäste vor der Türe, überlegen es sichs die potentiellen Kunden anders und gehen weiter.

Die Gäste am Tisch warten bis eine Servicekraft die Bestellung aufnimmt. Diese wird automatisch in die Küche übertragen. In der Küche beginnen die Köche sofort mit der Zubereitung. Um das Modell einfach zu halten, geben die Gäste nur eine Bestellung pro Besuch ab. Sobald die die Köche die Bestellung zubereitet haben, wird das Essen von den Tellerbringern gebracht und die Gäste können mit dem Essen beginnen. Sobald gegessen wurde, verlassen die Gäste das Restaurant. Für das Zahlen wird nicht nochmal die Servicekraft beansprucht.

Genauere Informationen finden Sie in der Dokumentation des Projekts, welches im Modul System Modeling und Simulation erstellt wurde.

3.3 Ausblick

Bei einer Weiterentwicklung der Simulationen können zum Beispiel folgende Aspekte verbessert oder vertieft werden.

- Trinken an der Bar, vor und oder nach dem Essen
- Mehrfache Bestellungen
- Getränke an den Tisch bringen, Tische abräumen
- Servicekräfte bringen Gäste zum Tisch
- Gäste sitzen gruppiert an Tischen
- Mund-zu-Mund-Propaganda bei kurzen Wartezeiten

Dies ist nur eine kurze Liste als Inspiration. Dieses Modell ist gut für den Unterricht geeignet und kann beliebig erweitert oder angepasst werden. Bei mehreren Gruppenarbeiten könnten verschiedene Restaurants simuliert werden. In einem weiteren Schritt könnte der Markt mit einer Menge an potentiellen Kunden und den erstellten Restaurantssimulationen simuliert werden.

3.4 Modellierung

In Visio wurde mit selbst erweiterten Stencil das Modell vor Beginn der Umsetzung aufgezeichnet. Das Modell finden Sie ebenfalls im Anhang.

4 Umsetzung

Um ein möglichst schnellen Einstieg in die Applikationen und in die Grundlagen der Simulation zu erhalten, wurden verschiedene kleine Beispielprojekte simuliert. Da die Bücher „AnyLogic 6 in Three Days“ und „Simio & Simulation | Modeling, Analysis, Applications“ im Ausland bestellt werden mussten, tratgen Lieferverzögerungen auf. Bis zum Eintreffen der Bücher wurde mit Anleitungen und Onlinevideos gearbeitet.

Für AnyLogic und Simio wurden je drei Wochen aufgewendet um anhand der erwähnten Bücher den Einblick zu erhalten. In dieser Zeit wurden zahlreiche kleine Simulationen erstellt.

Danach wurden vier Wochen verwendet, um den Prototypen und danach die effektive Simulation in den jeweiligen Programmen zu realisieren. Der Prototyp muss zuerst erstellt werden, damit die Simulation danach ohne Zwischenfälle aufgebaut werden kann und im Hintergrund PrintScreens erstellt werden können.

Beim Erstellen einer Simulation ist es sehr empfehlenswert kleine Iterationen zu modellieren und das Ergebnis ständig zu überprüfen, da die Prozesse in höchstem Masse parallel laufen und ein Debugging in den Simulatoren ziemlich mühsam ist.

Die Anleitungen wurden so verfasst, dass Sie am Anfang sehr einfach und verständlich ist. Gegen Ende wird jedoch zum Teil voraussetzt, das die vorgegangenen Schritte durchgearbeitet und verstanden wurden. Es wurde darauf geachtet, dass die Anleitungen nicht zu lang wurden. Zusätzlich wurden Flash-Videos erstellt, welche bei Verständnisproblemen konsultiert werden können.

5 Auswertung / Gegenüberstellung

Durch die Einarbeitung und die Realisierung desselben Modells in beiden Simulatoren konnten ein guter Einblick in die Applikationen erarbeitet wurden. Die Erkenntnisse aus dieser Arbeit und die Erfahrungen welche im Projekt mit Arena gesammelt wurden fliessen nun in diese Gegenüberstellung ein.

5.1 Übersicht AnyLogic

Der Simulator AnyLogic ist eine Java Applikation und basiert auf Eclipse. Besitzt man bereits Erfahrung mit Eclipse ist der Einstieg sehr einfach. Sobald dann etwas kompliziertere Simulation modelliert werden sollen, sind Kenntnisse von Java oder anderen Programmiersprachen von Vorteil, da ab und zu kleine Einzeiler geschrieben werden müssen, um Events anzustossen oder Variablen zu setzen.

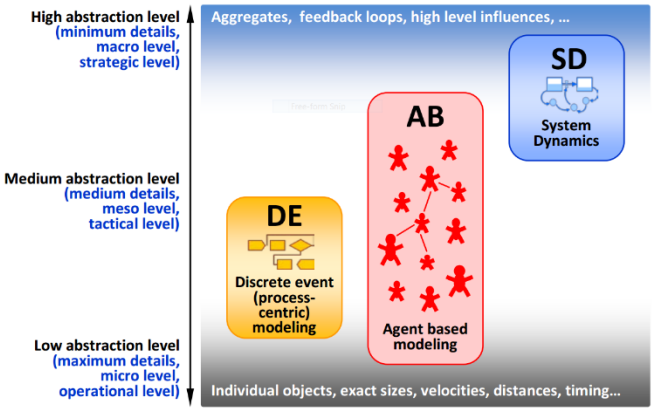


Abbildung 1 Modellierungsarten AnyLogic

AnyLogic unterstützt die Modellierungsarten[1] Discret Event Modeling, Agent based modeling und System Dynamics. Der Agent based Modellierungsansatz eignet sich gut um das Verhalten von Agenten zu simulieren. Beispiel: Marktverhalten zweier Produkten. Mit System Dynamic kann zum Beispiel die Verbreitung einer Krankheit simuliert werden. Personen sind gesund, einer steckt den Nächsten an und ist für eine gewisse Zeit ansteckend und ist danach immun gegen die Krankheit.

[1] http://www.anylogic.com/upload/Big Book of AnyLogic/Modeling_and_simulation_modeling.pdf

Die Erstellung der Simulationen ist sehr intuitiv. Mit wenig Aufwand können die Simulationen so gestalten werden, dass während dem die Simulation läuft, Werte geändert werden können und diese direkt in die Simulation einfliessen. Die Projekte in AnyLogic können schnell als Java Applet exportiert werden und so den Kunden ohne Lizenzkosten zur Verfügung gestellt werden. AnyLogic ist gut geeignet für die Erstellung eigener Simulatoren, welche Kunden selber parametrisieren können. Viele Beispiele finden Sie unter http://www.runthemodel.com/.

5.2 Übersicht Arena

Der Einstieg in Arena ist sehr einfach und die erste Simulation hat man schnell erstellt. Sobald jedoch komplexere Probleme simuliert werden sollen, wird es schnell sehr kompliziert und umständlich. Auch die Animationen von einfachen Abläufen brauchen ein wenig Übung. Man merkt, dass die Applikation keine neueren Programmieraspekte integriert hat.

Mit den Zusatzprogrammen ProcessAnalyser und OptQuest können Simulationen gut mit mehreren Szenarien durchlaufen werden. Bei OptQuest kann angegeben werden welche Parameter verändert werden sollen und welcher Wert zu optimieren ist.

5.3 Übersicht Simio

Simio ist eine auf C# basierende .Net Applikation und kommt im Ribbon Design daher, welches aus dem Microsoft Office bekannt wurde. Anfangs ist die Software sehr kompliziert und verwirrend. Viele Probleme lassen sich auf verschiedene Arten lösen und Optionen können an verschiedenen Orten eingestellt werden. Dies macht es anfangs schwierig den Überblick zu behalten. Andere Optionen lassen sich nur an bestimmten Orten einstellen und meist nicht dort, wo man als erstes sucht. Hat man sich erst an die Applikation gewöhnt, ist die Applikation sehr mächtig und komplizierte Prozesse können relativ einfach modelliert werden. Szenarien können sehr schnell erstellt werden und definierte Parameter werden gleich übernommen. Die Auswertungen sind sehr detailliert und lassen sich gut anpassen. Zudem generiert Simio gleich sehr gute und aussagekräftige Diagramme über einzelne Simulationen oder über auch über die Experimente.

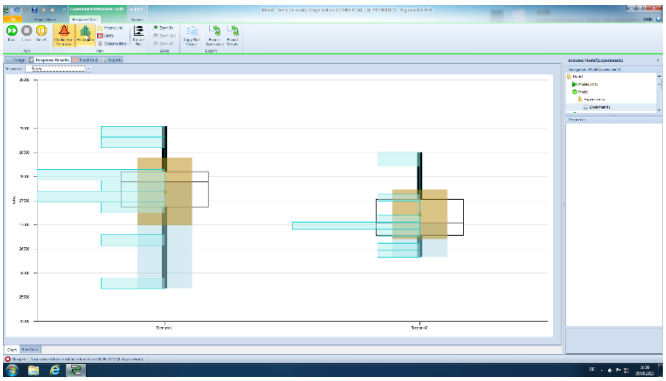


Abbildung 2 Diagramm in Simio

5.4 Einfachheit

Der Einstieg in AnyLogic und Arena ist relativ einfach. Bei Simio braucht es eine längere Einarbeitungszeit. Sobald die Projekte komplizierter werden, ist Simio jedoch gegenüber Arena klar im Verteil. Auch AnyLogic ist bei den komplizierten Projekten stark, da vielerorts mit Java Code unterstützt und eingegriffen werden kann.

5.5 Animation

Eine 3D/2D Animation ist in AnyLogic und Simio sehr schnell erstellt. Bei Simio sind die Elemente bereits in 3D und Bestandteil der Animation. Bei AnyLogic ist die Modellierung separat zur Animation. Die 2D oder 3D Animation ist jedoch sehr schnell und einfach hinzuzufügen. Bei Arena ist dies nicht ganz so komfortabel. Für eine einfache Animation müssen zusätzliche Elemente zum Modell hinzugefügt werden.

5.6 Szenarien

In Arena und Simio lassen sich einfach Tabellen mit Szenarien erstellen, bei welchen verschiedenen Szenarien definiert werden können. Simio erleichtert einem hier die Arbeit, da es automatisch die parametrisierten Eigenschaften integriert. Bei der Auswertung ist Simio klar stärker als Arena. Simio bietet detaillierte Auswertungen und generiert Grafiken über die definierten Ausgabewerte. Beide Tools verfügen auch über eine Zusatzapplikation um die Simulation zu optimieren. AnyLogic verfolgt hier ein anderen Ansatz. In AnyLogic lassen sich Simulationen erstellen, welche vom Benutzer direkt konfiguriert werden können.

5.7 Auswertung

Arena generiert nach der Simulation eine relativ detaillierte Auswertung. Simio geht hier noch etwas weiter: die Auswertung in der Pivot Tabelle kann angepasst und umsortiert werden. In AnyLogic muss die Auswertung selbst mit Diagrammen oder mit Datensets realisiert werden.

5.8 Externe Daten

In AnyLogic wie auch in Simio können externe Daten gelesen und geschrieben werden. AnyLogic hat eine eigene Palette mit Elementen um auf Datenbanken oder Dateien zuzugreifen Zusätzlich könnte mit Java Code gearbeitet werden. In Simio können die Tabellen an eine Datenbank oder eine Excel Datei gebunden werden.

5.9 Speed

Arena arbeitet nur mit einem CPU ist aber trotzdem relativ schnell. Simio nutzt alle verfügbaren CPUs und ist bei der Berechnung der Szenarien sehr schnell. Bei AnyLogic steht die Animation immer im Vordergrund und die CPU wird deshalb nie ausgenutzt.

5.10 Erweiterbarkeit

AnyLogic ist mit Java erweiterbar. Es können eigene Klassen geschrieben und integriert werden. In Simio ist es möglich mit C# zu arbeiten. Somit sind beide Tools beliebig erweiterbar.

5.11 Debugging

In AnyLogic und Simio können Breakpoints gesetzt werden. Mit dem „Watch“ Fenster können in Simio die Inhalte der Variablen betrachtet werden, was sehr nützlich sein kann. In Arena ist die Fehlersuche sehr mühsam. Es wird in allen Simulatoren empfohlen das Projekt regelmässig zu testen und zu überprüfen ob das Ergebnis auch dem erwarteten Verhalten entspricht.

5.12 Community

Für AnyLogic finden sich im Internet viele Hilfen und Anleitungen. Für Simio ist die Online-Community nicht ganz so gross.

5.13 Zusammenarbeit

AnyLogic verwendet ein XML Format für die Projektdaten. Bei Simio und Arena handelt es sich um binäre Dateien. Grundsätzlich wäre die Teamarbeit mit Verwendung eines Repository Server mit AnyLogic möglich. Sobald jedoch an der gleichen Stelle gearbeitet wird, wird ein Zusammenfügen sehr umständlich und mühsam.

5.14 Tauglichkeit für Unterricht

AnyLogic und Simio sind für den Unterricht geeignet. Der Vorteil bei Simio ist sicher die Auswertung und die Szenarien. Leider braucht es jedoch ein wenig Einarbeitungszeit. Bei AnyLogic kann schnell mit einem Projekt begonnen werden jedoch ist dort die Gefahr sehr gross, dass sich Studenten mit den Java Kenntnisse durchmogeln und somit die Ansätze der Simulation nicht richtig lernen und anwenden. Dafür hat AnyLogic bis jetzt als einziges Tool mehrere Modellierungsarten.

B. ANLEITUNG ANYLOGIC

In AnyLogic erstellen wir ein Modell, um ein Restaurant zu simulieren. Als Vorlage wird das Hardrock Cafe Atlantic City verwendet.. Die Bar wird als Queue für die Tische verwendet und zur Vereinfachung wird nur das Essen simuliert. Diese Anleitung hat keinen Anspruch auf die perfekte Lösung und darf nach eigenem Ermessen angepasst werden. Sie soll den Einstieg in die Simulation mit AnyLogic erleichtern und zeigt ein paar Aspekte.

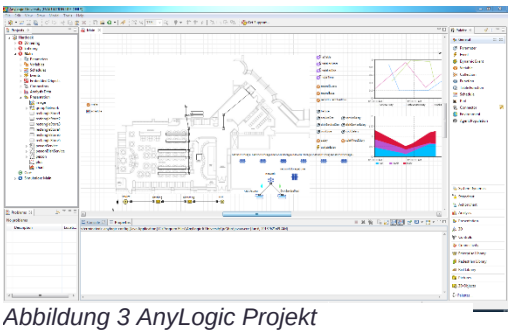


Abbildung 3 AnyLogic Projekt

1 Installation

Für dieses Tutorial wurde AnyLogic 6.9.0 auf einem Windows 7 x64 Betriebssystem verwendet. Als Lizenz wurde ein time-Limited Evaluation Key verwendet. Somit standen nicht alle Funktionen zur Verfügung.

Es wird empfohlen die Installation mit Administrationsrechten auszuführen. Nach dem die License Agreement akzeptiert wurde können Sie die gewünschte Sprache wählen. Bei dieser Installation wurde Englisch verwendet, da die Suche nach Problemen in Englisch zielführender ist als in anderen Sprachen. Der Installationsassistent kann mit den vorgegebenen Optionen installiert werden.

Beim ersten Start wird man gleich nach der Lizensierungsart gefragt. Hier kann ein Lizenzserver oder zum Beispiel ein USB Dongle angegeben werden. Es kann auch gleich ein Evaluation Key angefordert werden, welcher danach per E-Mail empfangen wird und im nächsten Dialogfeld eingegeben werden muss. Der ganze Key kann einfach hinein kopiert werden.

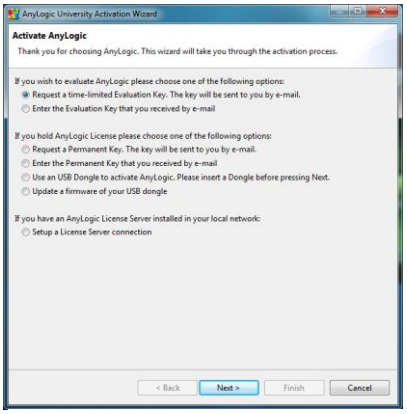


Abbildung 5 Anforderung Key

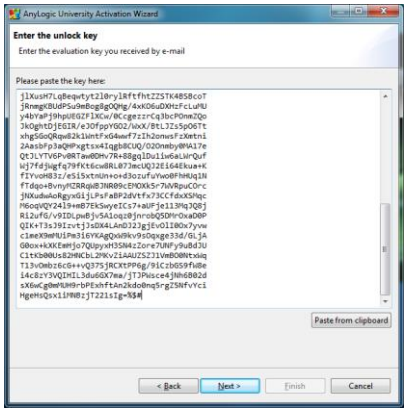


Abbildung 4 Eingabe Evaluation Key

Projektexplorer: Mit einem Doppelklick können die Elemente in der Arbeitsfläche geöffnet werden.

Arbeitsfläche: Hier können die Objekte modelliert werden.

Palette: Hier finden Sie die Elemente, welche auf die Arbeitsfläche gezogen werden können.

Eigenschaften: Konfiguration der Elemente.

Fehler: Liste der Fehler.

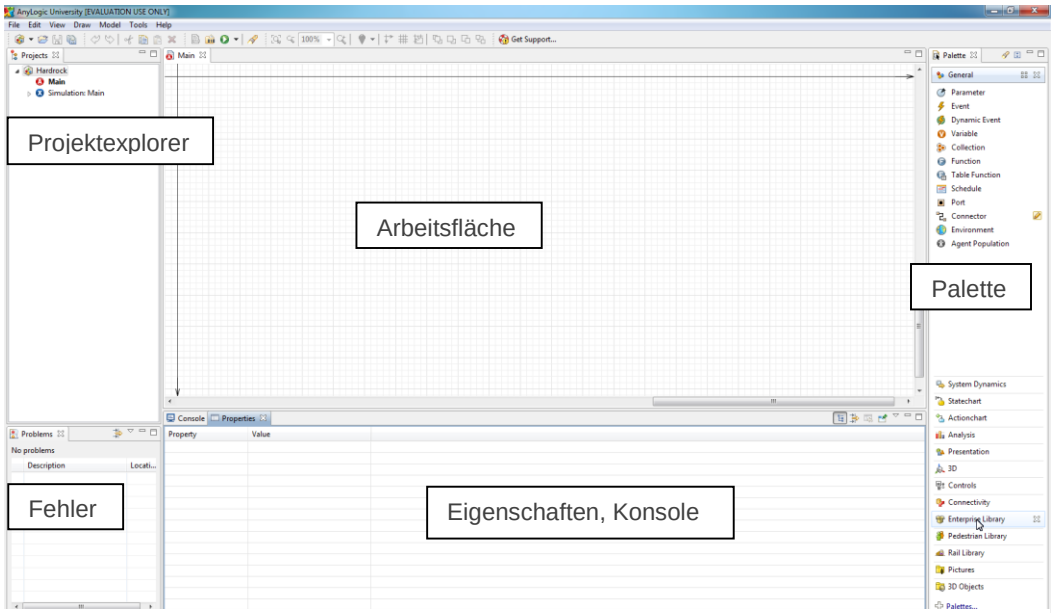


Abbildung 6 AnyLogic Hauptfenster

2 Neues Projekt erstellen

2.1 Erste Simulation

1. Nachdem die Willkommensseite geschlossen wurde kann über `File > New > Model` ein neues Projekt erstellt werden.
2. Geben Sie nun den Namen des Modells ein und wählen Sie den Ordner in welchem das Modell abgelegt werden soll. Im selektierten Ordner wird automatisch ein Ordner mit dem Modellnamen erstellt.
3. Um AnyLogic besser kennen zu lernen wird kein Template verwendet. Mit `Finish` wird das Projekt erstellt.
4. Ziehen Sie von der `Enterprise Libary` die Elemente `Source`, `Queue`, `Delay` und `Sink` auf die Arbeitsfläche von `Main`.
5. Nun müssen die Elemente miteinander verbunden werden. Dies geschieht durch einen Doppelklick in das Kästchen von welchem die Verbindung startet, gefolgt von einem Klick in das Zielkästchen.
6. Vor dem ersten Ausführen muss mit dem kleinen Pfeil neben dem „Run“ Symbol das entsprechende Projekt ausgewählt werden.
7. Nun kann die erste Simulation bereits gestartet werden.

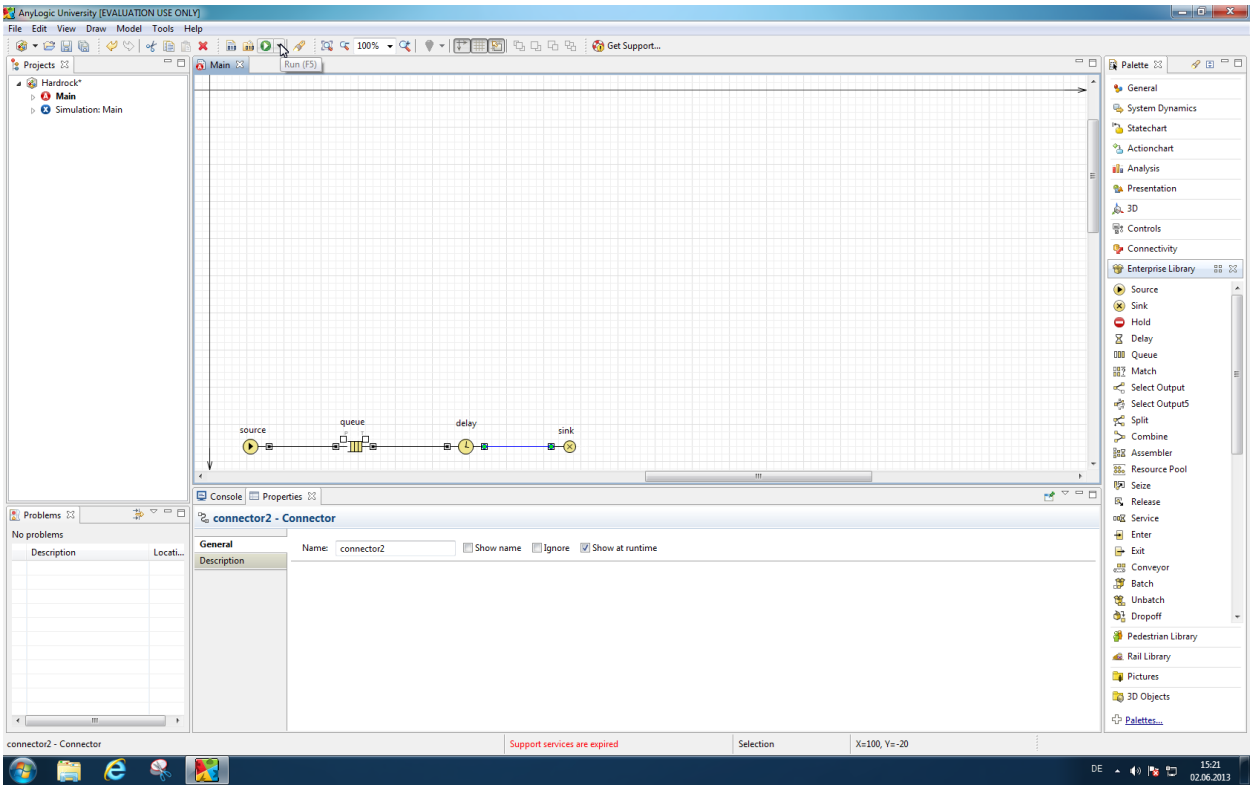


Abbildung 7 Erste Simulation

> *Die grünen Punkte der Verbindung zeigen an, dass sie mit dem Port des Elements verbunden sind.*
> *Sind die Punkte einer selektierten Verbindung nicht grün, so sind die zwei Elemente nicht verbunden und es können keine Entitäten übergeben werden.*

2.2 Eigenes Element

Da bei der Evaluation Version maximal 20 aktive Elemente auf einer Arbeitsfläche erlaubt sind, erstellen wir von Anfang an zwei Unterelemente `Entering` und `Dinnering`

1. Durch einen Rechtsklick auf das Projekt im Projektexplorer und danach auf `New > Active Object Class` kann ein neues Element erstellt werden. Nennen Sie das erstellte Element `Entering`. Die Arbeitsfläche öffnet sich automatisch.
2. Damit das Element von der Aussenwelt Entitäten erhalten kann und diese wieder weitergeben kann, braucht es sogenannte Ports. Diese findet man in der Palette `General`. Ziehen Sie zwei auf die Arbeitsfläche und beschriften Sie sie mit `in` und `out`. Es bildet sich ein Rahmen mit der Beschriftung `Icon` um die zwei Ports. Alles in diesem Rahmen wird als Element Icon verwendet. Holen Sie aus der Palette `Pictures` nun das Bild `Message` und platzieren Sie es zwischen den Ports. Nun ist das Icon erstellt.

> *Ist das Element markiert, kann in den Eigenschaften mit `Show name` angegeben werden, ob der Name angezeigt werden soll oder nicht. Dies kann die Unterscheidung der Elemente stark vereinfachen.*

3. Fügen Sie wie in 2.1 eine `Queue` und einen `Delay` ein und verbinden Sie den `in` Port mit der Queue, die Queue mit dem Delay und diesen mit dem `out` Port.
4. Mit Rechtsklick auf die `Entering` Klasse im Projektexplorer und danach auf `Copy` kann die Klasse kopiert werden. Nun kann mit Rechtsklick und danach `Paste` eine Kopie erstellt werden.
5. Benennen Sie `Entering1` in `Dinnering` um indem Sie die Klasse selektieren und im Eigenschaftenfenster (`Properties`) das Feld `Name` bearbeiten.
6. Wechseln Sie zurück zur Arbeitsfläche `Main`, löschen Sie die Elemente `Queue` und `Delay` und ersetzen Sie diese durch die Elemente `Entering` und `Dinnering`. Eigene Elemente können hinzugefügt werden, indem sie per Drag and Drop vom Projektexplorer in die Arbeitsfläche gezogen werden. Nun müssen die Verbindungen neu angeheftet werden.

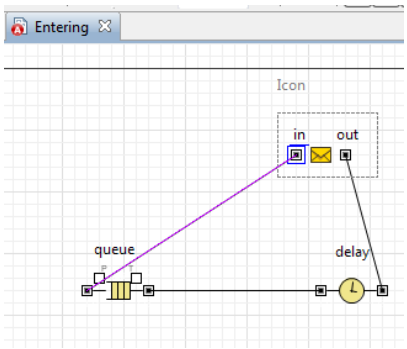


Abbildung 8 Submodel

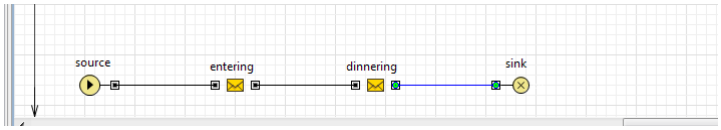


Abbildung 9 Modell mit Submodel

> *Es ist empfehlenswert, immer kleine Veränderungen vorzunehmen und die Simulation einer Funktionskontrolle zu unterziehen. Die Fehlersuche kann sich als sehr schwierig und verzwickt herausstellen, wenn viele Schritte miteinander gemacht worden sind. Fehler werden meistens erst beim Kompilieren des Projekts angezeigt. Beim Ausführen der Simulation wird die Kompilierung automatisch durchgeführt.*
>
> *Es ist ausserdem empfehlenswert, das Projekt in regelmässigen Abständen zu speichern.*

7. In der Simulation kann (per Navigationsdropdown Menü) zu den eingebetteten Elementen gewechselt werden. In der Symbolleiste kann ausserdem die Simulationsgeschwindigkeit angepasst werden. Bei den Elementen ist anhand der Zahlen ersichtlich, wie viele Entitäten in Bearbeitung sind, das Element betreten und verlassen haben.

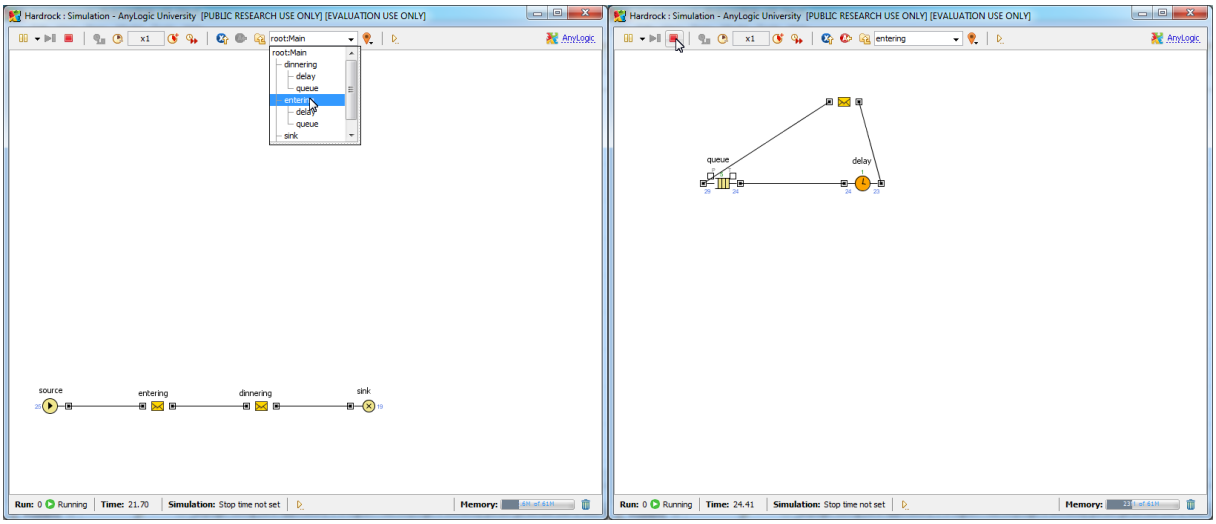


Abbildung 10 Laufende Simulation

> *Die Simulation eignet sich gut um zu prüfen, ob das Verhalten richtig ist. Es ist wichtig, sich stets präsent zu halten, was man mit den letzten Veränderungen beabsichtigte zu erreichen.*

3 Netzwerk

Für die Animation und die Platzierung wird der Ansatz mit einem Netzwerk und Netzwerkspeicher wie in einem Grosslager gewählt. Netzwerkressourcen und die Entitäten bewegen Sich anhand der Pfade durch das Netzwerk.

1. Um die Netzwerkpfade einzuzeichnen, fügen Sie zuerst ein Bild mit dem Gebäudeplan ein. Die Grafik hardrock.jpg ist 864 Pixel breit. In der Realität ist der Raum ca. 57.6 Meter breit. Somit ergibt sich das Verhältnis: 15px = 1m.
2. Fügen Sie von der `Presentation` Palette ein `Image` Element zur Arbeitsfläche hinzu und fügen Sie in den Einstellungen mit `Add image` das Bild hardrock.jpg hinzu. Zudem müssen `Original size` und `Show in 3D scene` aktiviert werden.
3. Das Netzwerk wird mit `Polyline` und `Rectangle` gezeichnet. Beides findet man in der Palette `Presentation`. Der gelbe Stift beschreibt, dass mit einem Doppelklick auf das Element die Figur gezeichnet werden kann. Wie die anderen Elemente können diese auf die Arbeitsfläche gezogen werden. Die Rechtecke dienen als Warteräume, Eintrittspunkte oder als Kreuzungen. Die `Polyline` verbinden diese Knoten. Wichtig ist, dass diese in jedem `Rectangle` einen Punkt haben. So muss beim Zeichnen in jedes Rechteck geklickt werden. Auch normale Linien können für Netzwerkverbindungen nicht verwendet werden. Zeichnen Sie das Netzwerk wie auf folgender Grafik dargestellt.

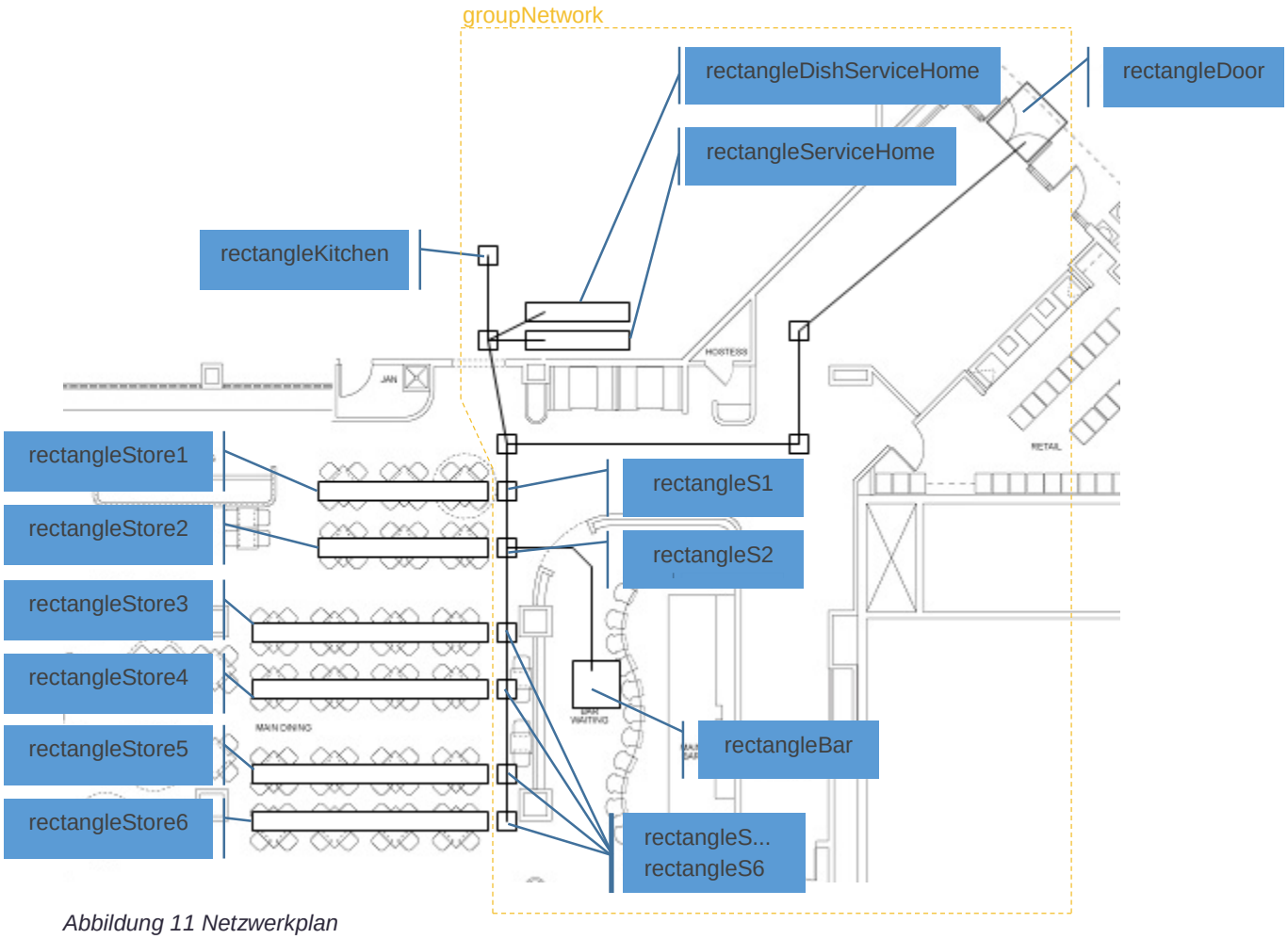


Abbildung 11 Netzwerkplan

4. Ändern Sie die Namen der Formen ebenfalls entsprechend der Grafik. Auf diese Namen wird im Weiteren der Anleitung Bezug genommen.
5. Markieren Sie alle Elemente innerhalb des orangen Rahmens, gruppieren Sie diese per Rechtsklick `Grouping > Create a Group` und nennen Sie die neue Gruppe `groupNetwork`.

> *Elemente können auch über den Projektexplorer markiert werden. Erweitern Sie hierzu den Baum/Ast mit dem [+] Symbol und markieren Sie die gewünschten Elemente. Tipp: Auswahl mit CTRL oder SHIFT ist möglich.*

6. Um das Netzwerk zu definieren, ziehen Sie das `Netwok` Element aus der `Enterprise Library` auf die Arbeitsfläche und geben unter `Group of netwok shapes` die in Punkt 5 erstellte Gruppe `groupNetwork` an.
7. Zur Simulation der Tische werden Netzwerkspeicher verwendet. Ziehen Sie sechs `Network Storage` Elemente ins Modell.
   Bei allen Tischen wird als `Network` das erstellte Netzwerk `network` angegeben. Als `Shape of the aisle` dienen die Formen `rectangleStore1` bis `rectangleStore6` und als `Front end entry node` und `Back end entry node` dienen `rectangleS1` bis `rectangleS6` Als Einstiegs- wie auch als Ausstiegspunkt wird jeweils der gleichen Knoten verwendet. Um auf 250 Plätze zu kommen wurde `Positions (per row)` folgendermassen definiert.

Storage Name	Postitions (per row)	Shape of the aisle	Front / Back end entry node
networkStorage	14	rectangleStore1	rectangleS1
networkStorage1	15	rectangleStore2	rectangleS2
networkStorage2	24	rectangleStore3	rectangleS3
networkStorage3	24	rectangleStore4	rectangleS4
networkStorage4	24	rectangleStore5	rectangleS5
NetworkStorage5	24	rectangleStore6	rectangleS6

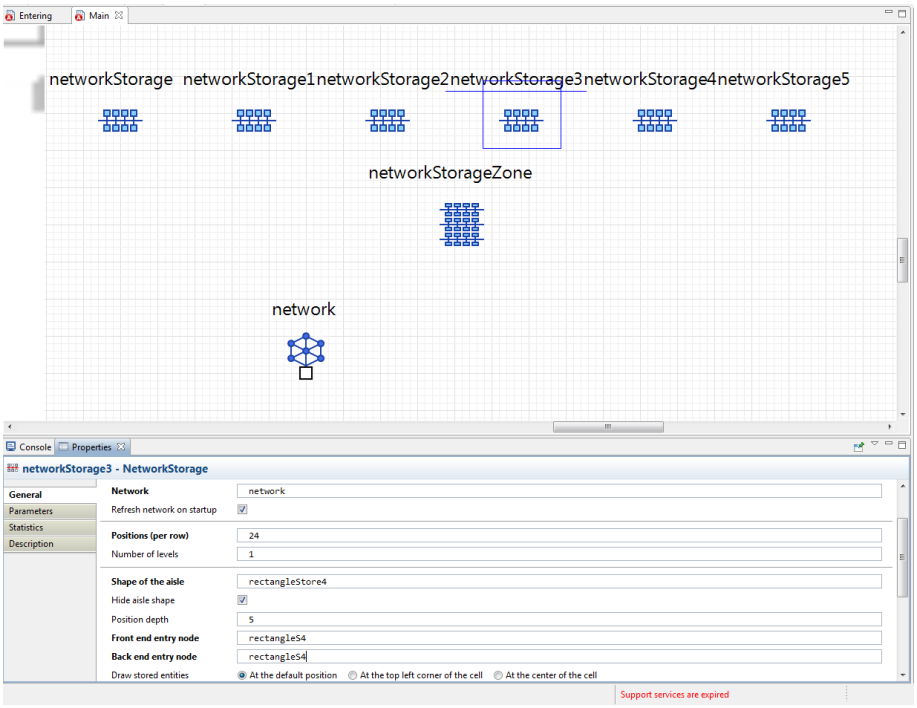


Abbildung 12 Konfiguration Netwok Storage

> *Mit [Ctrl] + Leertaste kann die Autovervollständigung aufgerufen werden. So muss nicht immer alles von Hand ausgeschrieben werden. Wie in Eclipse üblich, sieht man die zur Verfügungen stehenden Objekte oder Variablen.*

8. Um die Speicher nicht alle einzeln anzusprechen kann ein `Network Storage Zone` Element verwendet werden. Fügen Sie ein solches Element dem Modell hinzu und geben Sie unter `Storage` die Speicher wie folgt als Menge an: `{ networkStorage, networkStorage1, networkStorage2, networkStorage3, networkStorage4, networkStorage5 }`. Wichtig sind die geschwungenen Klammern.
9. Bevor es zu den Submodellen geht, macht es Sinn eine Variable für die Grösse eines Meters zu definieren. Hier geben Sie an wie viele Pixel auf dem Plan einen Meter lang sind. Platzieren Sie das Element `Variable` von der Palette auf der Arbeitsfläche. Nennen Sie die Variable `meter` und setzen Sie den Typ auf `int` und `Initial value` auf `15`.

> *In der Simulation ist nur das 4. Quartal des X Y Systems sichtbar.*

10. Wechseln Sie nun zu `Entering`.
11. Löschen Sie den Delay und fügen Sie folgende Elemente aus der `Enterprise Library` hinzu: `Restricted Area Start`, `Network Enter`, `Network Move To` und `Network Storage Put`.

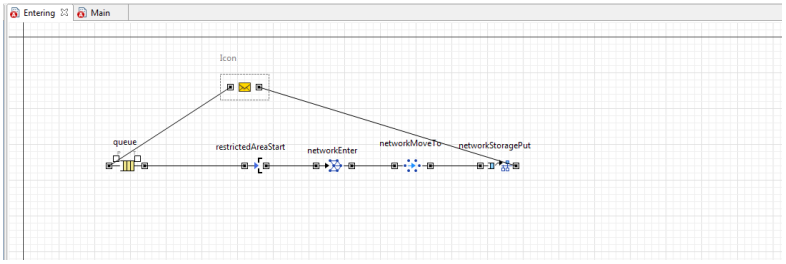


Abbildung 13 Eintritt Netzwerk

12. Mit der Restricted Area wird eine Zone erstellt, welche nur eine gewisse Anzahl an Entitäten enthält. Dies ist nötig, da `Network Storage Put` keine Queue hat und einen Fehler auslösen würde, sobald der Speicher voll ist und die nächste Entität abgelegt werden müsste. Die `Capacity` setzen wir vorerst mal auf 250.
13. Mit dem Element `Network Enter` sind die Entitäten in diesem Netzwerk. Als Netzwerk muss der Name des erstellten Netzwerks angegeben werden. In diesem Fall ist dies `network`. Als Einstiegspunkt wird das Rechteck bei der Türe angegeben. Da dieses jedoch eine Ebene oberhalb definiert ist, muss zuerst auf diese zugegriffen werden. `Entry node` wird deshalb als `get_Main().rectangleDoor` angegeben. Als Geschwindigkeit (`Speed`) geben wir einen Meter pro Simulationseinheit an. Diesen „Meter“ haben wir zuvor im `Main` Modell definiert. Mit `get_Main().meter` wird die Geschwindigkeit der Entitäten auf einen Meter pro Zeiteinheit gesetzt.

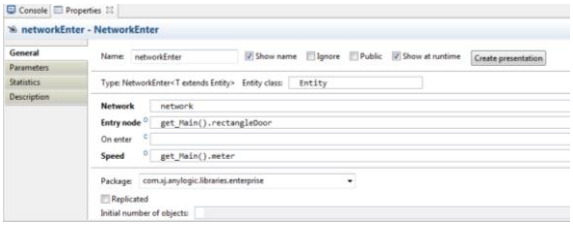


Abbildung 14 Konfiguration Network Enter

> *Die Geschwindigkeit bezieht sich auf die zurückgelegte Strecke pro definierte Zeiteinheit (Default: Strecke/min). Zu einem späteren Zeitpunkt dieser Anleitung wird die Zeiteinheit von Minute auf Sekunde geändert.*
> *Deshalb wird der Wert auf `get_Main().meter / second() * minute()` geändert*

14. Wenn der Gast das Restaurant betritt, begibt er sich zuerst zur Bar um dort auf einen Platz an einem Tisch zu warten. Mit `Network Move To` können die Entitäten über das Netzwerk zur Bar gesendet werden. Als Zielort (`Node`) wird `get_Main().rectangleBar` angegeben.
15. Um die Entitäten im Speicher abzulegen wird das Element `Network Storage Put` verwendet. In der Konfiguration wird die erstellte Storage Zone angegeben. Da die Gäste selber zum Tisch gehen können braucht es keine Ressource um sie zu bewegen oder begleiten.
16. Um die Simulation zu testen, müssen die Entitäten das Restaurant auch wieder verlassen können.

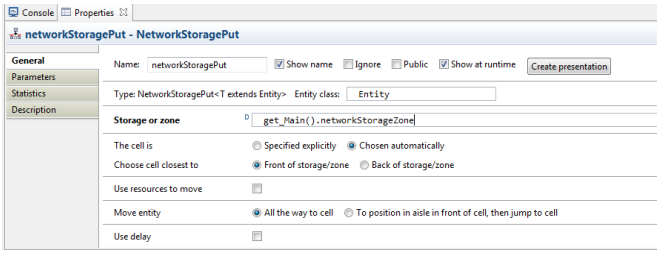


Abbildung 15 Storage Put

17. Öffnen Sie `Dinnering` im Projektexplorer mittels Doppelklick.
18. Ergänzen Sie das Modell mit `Network Storage Pick`, `Network Exit` und `Restricted Area End` aus der `Enterprise Library`.
19. Gleich wie bei `Storage Put` muss beim `Storage Pick` der Speicher angegeben werden. Als Zieladresse (`Destination node`) wird die Türe angegeben: (`get_Main().rectangleDoor`).

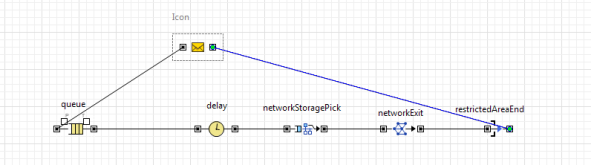


Abbildung 16 Austritt Netzwerk

20. Beim Verlassen des Netzwerks muss nichts konfiguriert werden.
21. Damit die `Restricted Area` weiss, dass die Gäste das Restaurant verlassen haben, muss beim `restricedAreaEnd` das Objekt angegeben werden, welches den Prozess gestartet hat. Dieses Element können wir über `get_Main().entering.ristricedAreaStart` referenzieren.

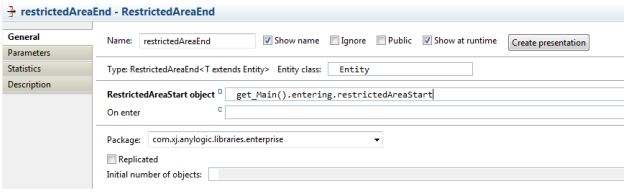


Abbildung 17 Restriced Area End

22. Nachdem alle Verbindungen gezeichnet sind, kann die Simulation getestet werden.
23. Wenn die Kompilierung fehlschlägt, wird eine Warnung angezeigt und man kann die Ausführung abbrechen. Ein Doppelklick auf die Fehlermeldung zeigt direkt die fehlerhafte Stelle.
    In diesem Fall wird `network` nicht erkannt, da `get_Main().` nicht angeführt wurde.

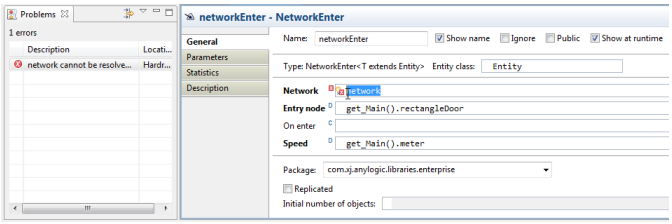


Abbildung 18 Fehler network Auflösung

24. Damit in der Animation die Bewegung der Gäste sichtbar ist, benötigen die generierten Entitäten ein Symbol. Gehen Sie zur `Main` Ansicht und ziehen Sie von den `Pictures` das Bild `Person` neben das `source` Objekt. In den Eigenschaften von `source` muss nun bei `Entity animation shape` das erstellte Symbol `person` angegeben werden.
25. Wenn das Netzwerk nicht angezeigt wird, muss überprüft werden ob `Show at runtime` richtig gesetzt ist.

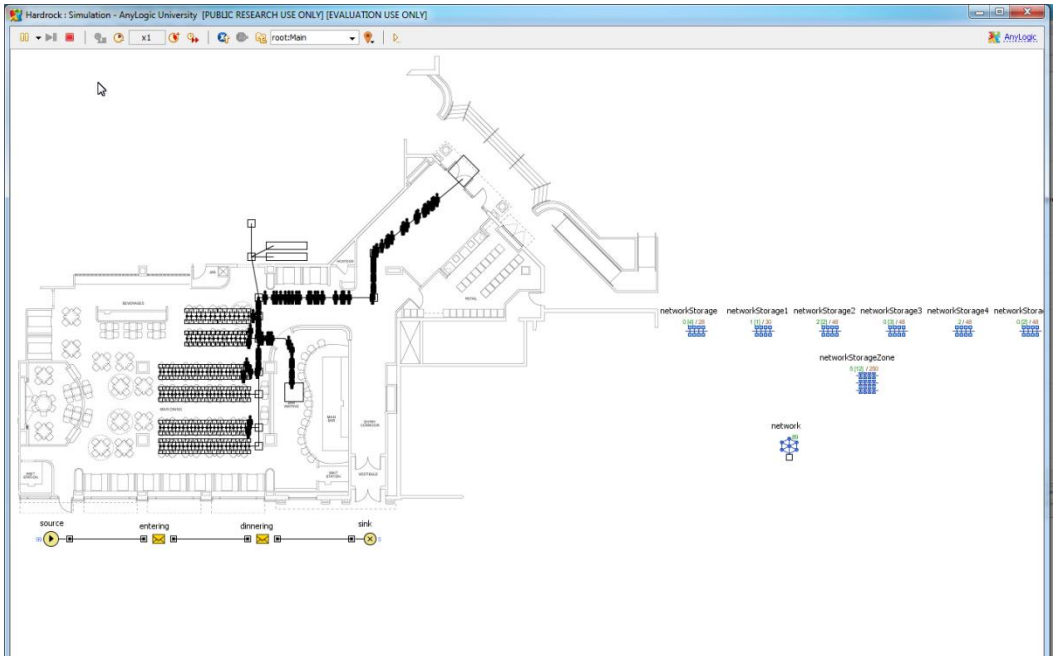


Abbildung 19 Simulation Netzwerk

4 Netzwerkressourcen

Als nächstes werden die Netzwerkressourcen hinzugefügt. Dies sind die Servicekräfte und die Tellerbringer.

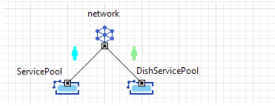


Abbildung 20 Netzwerk Ressourcen

1. Fügen Sie zwei Objekte vom Typ `Network Resource Pool` aus der `Enterprise Library` zum `Main` Modell hinzu und verbinden Sie diese mit dem Netzwerk. Nennen Sie die beiden Pools `ServicePool` und `DishServicePool`.
2. Fügen Sie, wie bereits bei der Source, pro Pool das Bild `Person` von der Palette `Pictures` ein. Benennen Sie die Bilder `personService` und `personDishService`. Um die Farbe der Shape zu ändern kann im Projektexplorer bei `Main`, `Presentation` die entsprechende `Shape`Gruppe aufgeklappt werden. Markieren Sie das Element `shapeBody` und ändern Sie die Farbe (`Fill color`).
3. Die erstellten Symbole werden nun in der Konfiguration der jeweiligen Pools als `Idle unit animation shape` und `Busy unit animation shape` definiert. Als `Home node` werden die vorbereiteten Rechtecke `rectangleServiceHome` und `rectangleDishHome` definiert. Bei `Capacity` wird die Anzahl Ressourcen definiert. Zum Starten definieren Sie 10 Servicekräfte und 4 Tellerbringer.

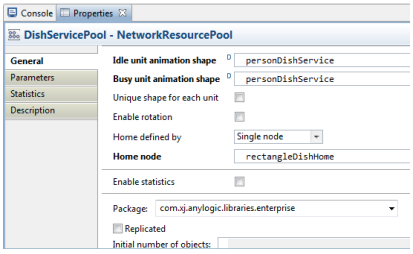


Abbildung 21 Network Resource Pool

4. Wechseln Sie zur Arbeitsfläche von `Dinnering`. Nun werden zuerst die Servicekräfte und danach die Tellerbringer zu den Gästen gesendet.

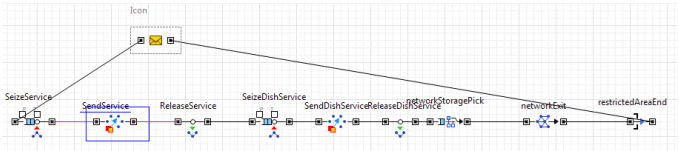


Abbildung 22 Sende Ressourcen zum Tisch

5. Ersetzen Sie sowohl `Queue`, als auch `Delay` durch folgende Elemente aus der `Enterprise Library`: `Network Seize`, `Network Send To`, und `Network Release`. Benennen und platzieren Sie die sechs neuen Elemente wie in der Abbildung 22 dargestellt.
6. Bevor eine Ressource zu einer Entität gesendet werden kann, muss diese mit einem Seize reserviert werden. Bei `SeizeService` muss die Ressource als `{get_Main().ServicePool}` bei `List of resources` angegeben werden. Zusätzlich ist die interne Queue auf das Maximum zu setzen, da die Anzahl der Personen im Restaurant über die `Restricted Area` geregelt wird.
7. Die Servicekraft wird mit SendService zur Entität am Tisch gesendet. Dafür muss in den Eigenschaften die `Destination` auf `Entity` gesetzt werden. Als `Resources to send` wird `{get_Main().ServicePool}` angegeben. Dies ist nötig, da an die Entität mehrere Ressourcen angebunden sein können.
8. Beim Freigeben der Ressource `ReleaseService` wird die gleiche Liste (`{get_Main().ServicePool}`) wie beim Seize angegeben.
9. Wiederholen Sie die Schritte 6 bis 8 entsprechend angepasst für die anderen drei Elemente.
10. Nun kann die Simulation getestet werden. Gerne werden bei den Ressourcen die geschwungenen Klammern vergessen. In diesem Falle erscheint eine `Type mismatch` Fehlermeldung.

5 Kochen

Nachdem eine Servicekraft die Bestellung aufgenommen hat, soll die Bestellung sofort in die Küche gehen. Da keine Gruppen sondern nur einzelne Gäste simuliert werden, kann davon ausgegangen werden, dass eine Bestellung nur ein Gericht enthält und die Bestellung somit nicht aufgeteilt werden muss. Dies wäre in AnyLogic mit den Elementen `Split`, `Match` und `Combine` gut möglich. Möchte man die Bestellungen und das Kochen animieren, könnte man einfach die Entität als Bestellung kopieren und so darstellen.

1. Nachdem der Tellerbringer das Essen gebracht hat, ist der Gast eine bestimmte Zeit mit essen beschäftigt. Dies wird mit einem `Delay` realisiert.
2. Fügen Sie zwischen `ReleaseDishService` und `networkStoragePick` einen `Delay` ein und nennen Sie ihn `eating`. Da alle Gäste gleichzeitig essen können, muss in den Eigenschaften des `Delays` die Checkbox `Maximal capacity` aktiviert werden.
3. Nun fügen Sie das Element `Service` aus der `Enterprise Library` Palette dem Modell hinzu. Platzieren Sie den neuen Service zwischen `ReleaseService` (Servicekraft hat Bestellung abgeschlossen) und `SizeDishService` (Tellerbringer wird angefordert). Nennen Sie den neuen Service `cooking`. Unterhalb des Service soll ein `Resource Pool` mit dem Namen `resourcePoolCook` (ebenfalls aus der `Enterprise Library` Palette) angehängt werden.
4. Bei der Servicestation kann nun als `ResourcePool object` der soeben erstellte Pool `resourcePoolCook` angegeben werden. Die interne Queue wird auf das Maximum eingestellt, da über die Poolgrösse gesteuert wird, wie viele Köche gleichzeitig kochen können.

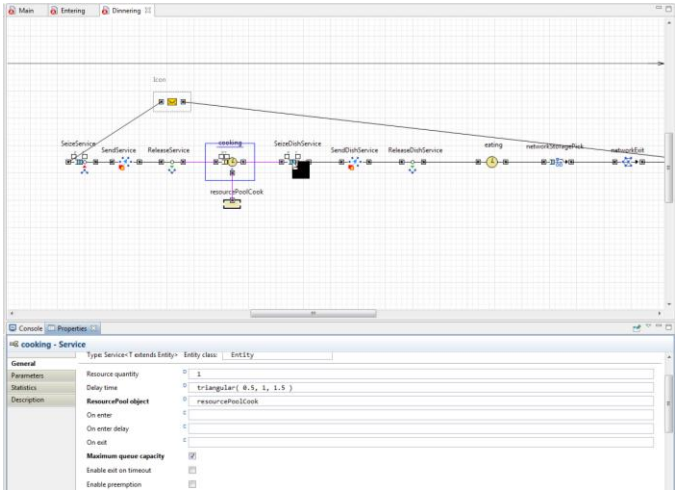


Abbildung 23 Service Station Kochen

5. Die `Capacity` bei `resourcePoolCook` wird auf 10 eingestellt. Somit hat es 10 Köche.
6. Da der Tellerbringer zuerst in die Küche gehen muss um die Teller zu holen, aktivieren Sie bei `SeizeDishService` die Option `Send seized resources`. Der Tellerbringer soll zuerst zum `Node` `get_Main().rectangleKitchen` gesendet werden. Mit `SendDishService` kommt der Tellerbringer danach zum Tisch.
7. Nun kann die Simulation getestet werden. Die Netzwerkressourcen sollten sich zu den Tischen bewegen.

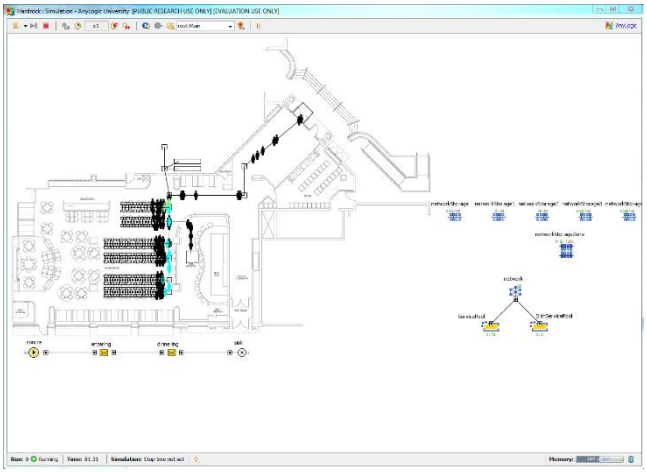


Abbildung 24 Simulation Tellerbringer

6 Personenbegrenzung

Im Restaurant hat nur eine gewisse Anzahl an Gästen Platz. Es werden nur so viele Gäste eingelassen, wie es Plätze hat. Ist die Warteschlange vor der Tür zu lange, gehen die Kunden ohne das Restaurant zu betreten.

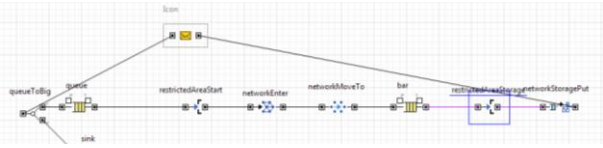


Abbildung 25 Entering Restricted Area

1. Öffnen Sie das Modell `Entering`.
2. Fügen Sie vor der Queue einen `Split` ein, bei welcher der mit `F` gekennzeichnete Port auf einen `Sink` führt, welcher ebenfalls hinzugefügt werden muss. Beide Elemente finden Sie in der `Enterprise Library` Palette. Hier wird nun eingestellt, dass die Gäste nicht anstehen wenn bereits zu viele Gäste in der Warteschlange vor der Türe stehen.
3. In der Konfiguration des Splitters muss eingestellt werden, dass der Output auf Grund einer `If condition` ausgewählt wird. Als `Condition` wird `queue.size() < 20` angegeben. Somit stellen sich nur neue Gäste in der Warteschlange an, wenn sich weniger als 20 Personen in dieser befinden.
4. Damit die wartenden Gäste angezeigt werden, muss beim Element `queue` das Attribut `Animation guide shape` auf `get_Main().rectangleDoor` gestellt werden. Der `Animation type` muss auf `bag` geändert werden. Ansonsten erscheint beim Kompilieren die Fehlermeldung `root.entering.queue: animationGuide for PATH type must be a ShapePolyLine`.

> *Taucht neben einem Inputfeld eine kleine Glühbirne auf, können weitere Informationen angezeigt werden. Bewegen Sie dazu die Maus über die Glühbirne.*
> *Meistens erfährt man so, wie auf Entitäten oder andere spezielle Objekte zugegriffen werden kann.*

5. Nun muss zwischen `networkMoveTo` und `networkStoragePut` eine Queue namens `bar` und ein `Restricted Area Start` mit dem Namen `restrictedAreaStorage` eingefügt werden.
6. Bei der Queue `bar` wird `Maximum capacity` aktiviert, da nur so viele Elemente wie über die `Restricted Area` definiert in der Bar sein können.
7. Bei `Restricted Area` Elementen kann beim Feld `Capacity` nicht auf die Grösse der definierten `StorageZone` zugegriffen werden, da diese beim Initialisieren noch Null und das Feld statisch ist.

> *Ein blaues C neben dem Eingabefeld signalisiert, dass Code in diesem Feld erwartet wird und ein Semikolon nach dem Befehl folgen muss. Ein blaues D bedeutet, dass dieses Feld dynamisch ist und wenn nichts angegeben wird, das Feld von statischer Natur ist, jedoch über einen Code geändert werden kann.*

8. Erstellen Sie für die Bargrösse auf dem `Main` Modell eine Variable `barSize` mit dem `Integer` Wert 40.
9. Damit die `Capacity` der `Restricted Areas` über die Variable und die `Capacity` der `StorageZone` definiert werden kann, wird der `Startup code` des `Entering` Objekts genutzt. Um zu den Eigenschaften zu gelangen klicken Sie im Modell `Entering` einfach auf eine freie Arbeitsfläche
   `Startup code:`
   ```
   restrictedAreaStorage.capacity = get_Main().networkStorageZone.capacity();
   restrictedAreaStart.capacity = get_Main().networkStorageZone.capacity() +
   get_Main().barSize;
   ```
   Durch diesen Code kommen in die Zone `restrictedAreaStorage` nur so viele Gäste, wie es Tischplätze hat und in die Zone `restrictedAreaStart` noch jene mehr, welche in der Bar Platz finden.

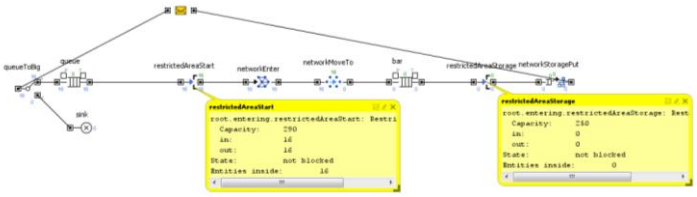


Abbildung 26 Restrict Area Simulation

7 Prozess Verteilfunktionen

Beim Erarbeiten des Modells wurden die Zeitfunktionen geschätzt, da die Zeit für Messungen und lange Analysen nicht gegeben war. Somit handelt es sich um fiktive Daten.

Es wurde definiert, wie viele Gäste pro Stunde das Restaurant besuchen wollen. Die Öffnungszeiten sind von 10:00 Uhr bis Mitternacht.

1. Fügen Sie einen `Scheduler` aus der Palette `General` zum `Main` Modell hinzu und konfigurieren Sie ihn wie auf dem nachfolgendem Bild.

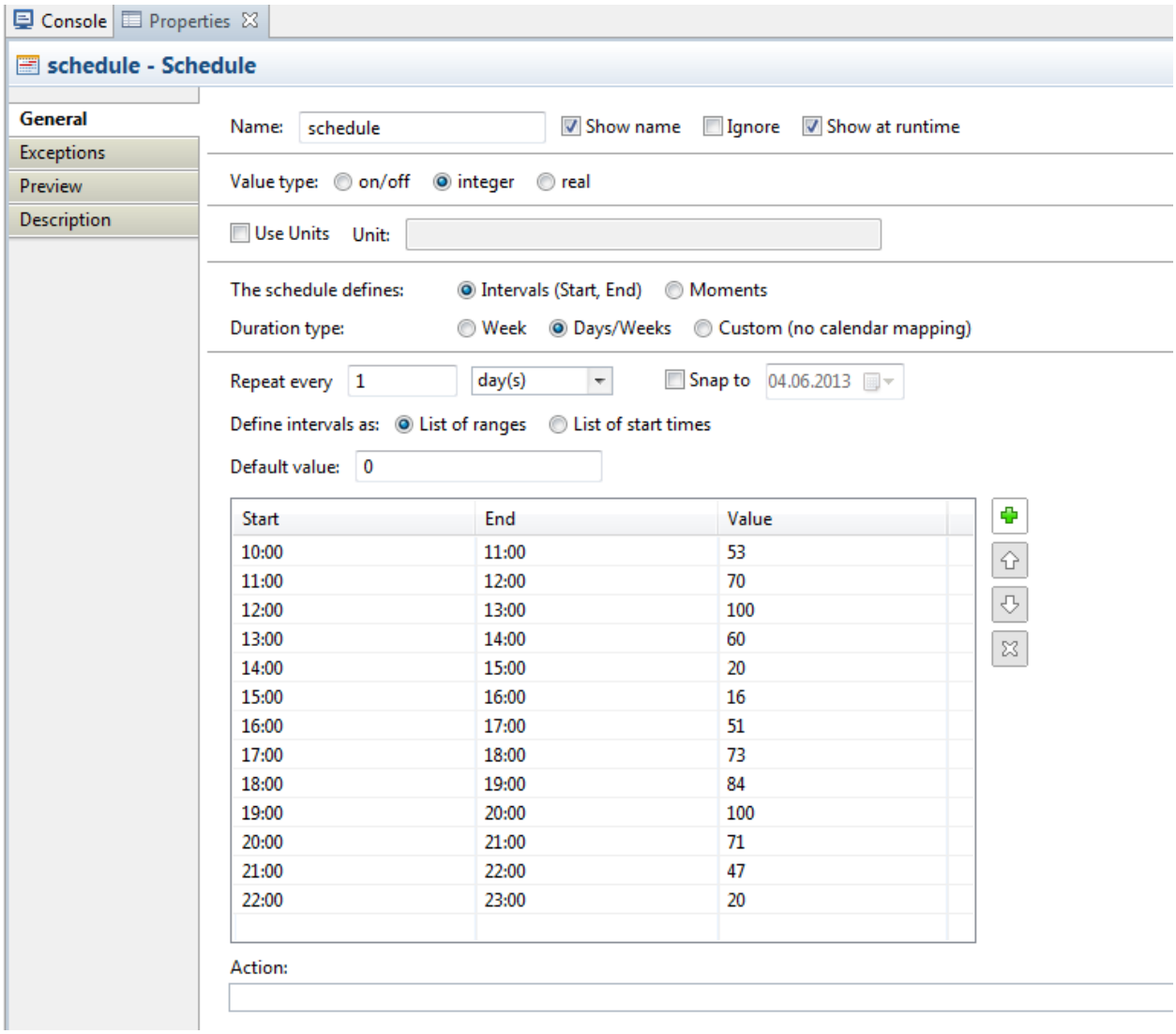


Abbildung 27 Konfiguration Scheduler

2. Nun muss dieser Scheduler beim Element `source` angegeben werden. Nach dem die `Arrivals defined by` auf `Rate schedule` eingestellt wird, kann im Feld `Rate schedule` der erstellte Schedule angegeben werden. Da die Werte pro Stunde definiert sind, muss `Modify rate` aktiviert werden um die Rate bei `Rate expression` durch eine Stunde teilen zu können. `baseRate/hour()`

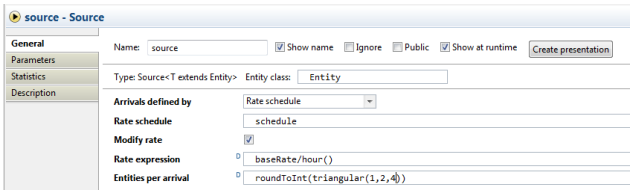


Abbildung 28 Konfiguration Source

3. Da Personen gerne in Gruppen in ein Restaurant gehen, wird nun bei `Entities per arrival` die Verteilfunktion `roundToInt(triangular(1,2,4))` angewandt, welche die Gruppengrösse abstrahiert. So kommen Gäste in Gruppen von 1 bis 4 Personen an, wobei die meisten in Zweiergruppen unterwegs sind.
4. Fügen Sie im Modell `Entering` einen Delay namens `ordering` zwischen dem `SendService` und `ReleaseService` ein.
5. Als `Delay time` wurde `triangular( 4, 7, 9 )` definiert.
6. Vor ReleaseDishService ist ebenfalls ein Delay mit dem Namen `dish` einzufügen.
7. Passen Sie die restlichen Verzögerungen wie in der folgenden Tabelle angegeben an:

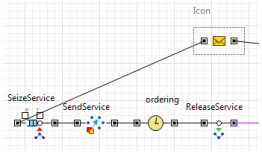


Abbildung 29 Ordering

Element	Feld	Wert
cooking	Delay time	triangular(1, 5, 10)
dish	Delay time	0.5
eating	Delay time	triangular(15, 40, 60)
networkEnter	Speed	get_Main().meter() / second() * minute()
ServicePool	Speed	meter / second() * minute()
DishServicePool	Speed	0.5 * meter / second() * minute()

Alle Zeiten stehen in Relation zu der Basissimulationszeiteinheit. Die `Time unit` ist beim Simulation Experiment in den Optionen einstellbar.

8. Öffnen Sie die Simulation über den Projektexplorer. Damit ein Tag simuliert wird, werden die `Model Time` Einstellungen wie im Bild dargestellt konfiguriert.

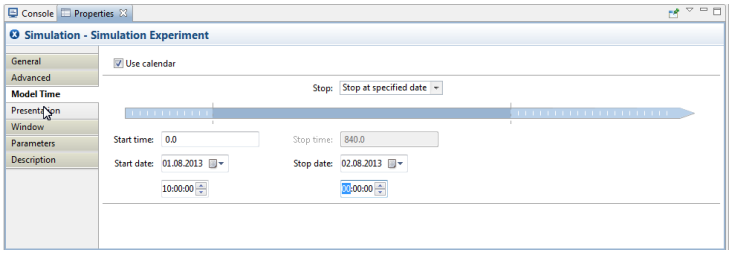


Abbildung 30 Simulationszeit Einstellungen

8 Zeitmessung

Damit mit den Entitäten Zeitmessungen gemacht werden können, müssen an bestimmten Stellen die Simulationszeit der Entität angehängt werden. Damit dies möglich ist, wird eine eigene Entitätsklasse erstellt.

1. Mit Rechtsklick in den Projektexplorer und danach auf `New > Java Class` kann eine neue Klasse erstellt werden. Geben Sie als Name `Cust` an und als `Superclass` muss `Entity` ausgewählt werden, damit diese Klasse als Entität verwendet werden kann. Damit kommen Sie zur Konfiguration der Klassenvariablen.
2. Erstellen Sie die Variablen `timeCreate`, `timeTable` und `timeOrder`. Die automatische Generierung der Konstruktoren kann weggelassen werden, da der Konstruktor von Hand angepasst wird. Mit `Finish` wird die Klasse generiert.

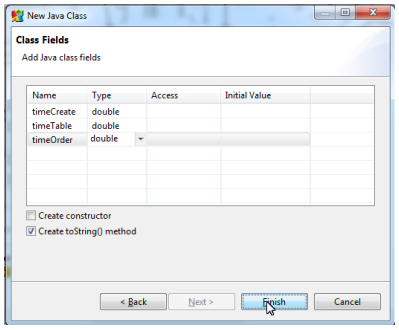


Abbildung 31 Eigene Klasse

3. Fügen Sie nun folgenden Konstruktor hinzu:
```
public Cust(double create) {
    this.timeCreate = create;
}
```
4. Damit die selber erstellte Klasse verwendet wird, muss bei dem Element `source` die `Entity class` auf `Cust` geändert werden. Zudem muss natürlich das Feld `New entity` angepasst werden. Über den Konstruktor kann die Zeit der Erstellung bereits mitgegeben werden. Somit lautet der Befehl `new Cust(time())`.

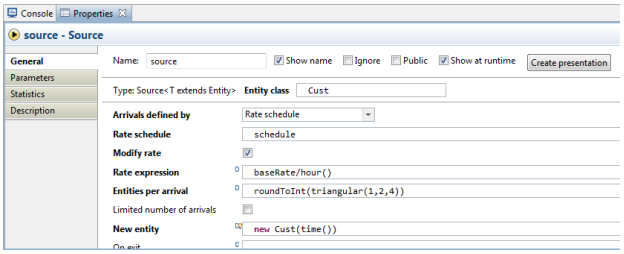


Abbildung 32 Generierung Cust Entitäten

5. Lassen Sie die Simulation laufen, um zu sehen, ob noch alles funktioniert.
6. Fügen Sie vier `Statistics` Elemente (`atTable`. `waitForOrder`, `waitForDish` und `totalTime`) aus der `Analysis` Palette und drei `Variablen` (`recordGuests`, `recordSales` und `recordGuestOverflow`) aus der `General` Palette zur `Main` Arbeitsfläche hinzu. Diese werden für die Datensammlung verwendet.

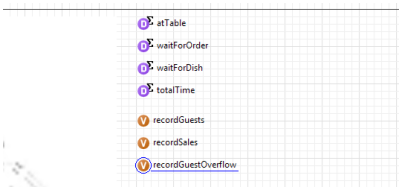


Abbildung 33 Statistikelemente

7. Beim Element `SeizeService` muss die `Entity class` ebenfalls auf `Cust` geändert werden. Danach kann nun bei `On exit` mit `entity.timeTable = time();` die Zeit gespeichert werden, bei welcher der Gast den Tisch erreicht. Zudem kann mit der weiteren Zeile `get_Main().atTable.add(entity.timeTable – entity.timeCreate);` das Statistikelement gefüllt werden.

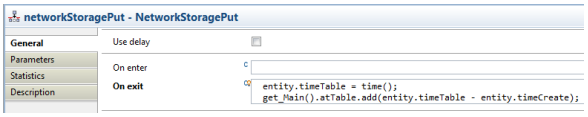


Abbildung 34 Zeitmessung

8. Sobald der Delay `ordering` beginnt, soll der Wert `timeOrder` gesetzt werden und die Wartezeit (Kunde am Tisch → Aufgabe der Bestellung) in das Statistikelement `waitForOrder` geschrieben werden. Dies wird dadurch erreicht, dass bei `ordering` die `Entity class` wiederum auf `Cust` geändert wird und bei `On enter` folgender Code ausgeführt wird:
```
entity.timeOrder = time();
get_Main().waitForOrder.add(entity.timeOrder – entity.timeTable);
```

9. Bei folgenden Elementen muss die `Entity class` auf `Cust` umgestellt werden und im entsprechenden Feld der Code angegeben werden.

Element	Feld	Wert
dish	On enter	`get_Main().waitForDish.add(time() - entity.timeOrder);`

sink (Main Modell)	On enter	`totalTime.add(time() - entity.timeCreate);`
		`recordGuests++;`
		`recordSales = recordSales + triangular(20,30,60);`

sink (Entering Modell)	On enter	`get_Main().recordGuestOverflow++;`

> *Während der Simulation können mit Klick auf die Elemente (zum Beispiel Statistikelemente oder Variablen) weitere Informationen angezeigt werden. Bei Variablen kann sogar ein Diagramm der Wertänderung angezeigt werden.*

9 Diagramme

Für die Auswertung und für die bessere Verständlichkeit werden nun zwei Diagramme gezeichnet. Auf dem Ersten soll die Auslastung der Ressourcen sichtbar sein. Das Zweite soll zeigen, wie viele Gäste an den Tischen, an der Bar oder in Bewegung sind.

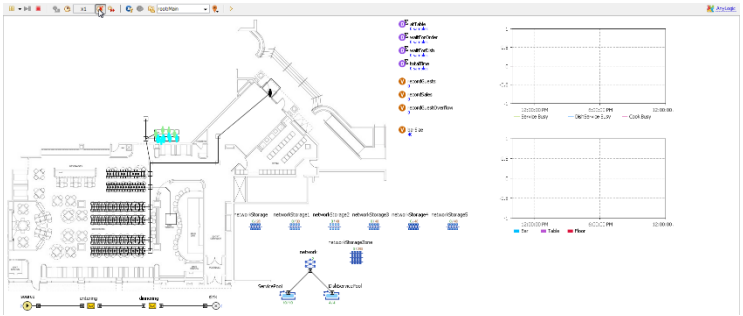


Abbildung 35 Anordnung der Diagramme

1. Ordnen Sie die Elemente auf dem `Main` Modell so an, dass genügend Platz für die zwei Diagramme zur Verfügung steht.
2. Für das erste Diagramm zeichnen Sie ein `Time Plot` aus der `Analysis` Palette. In den Einstellungen unter `General` muss `Time windows` auf `14 * hour()` eingestellt werden, damit der ganze Arbeitstag angezeigt wird. Damit auch genügend Daten vorhanden sind muss bei `Display up to 100 latest samples` der Wert auf 1000 erhöht werden.
3. Mit `Add data item` können Datensätze zum Diagramm hinzugefügt werden. Fügen Sie folgende Daten hinzu:

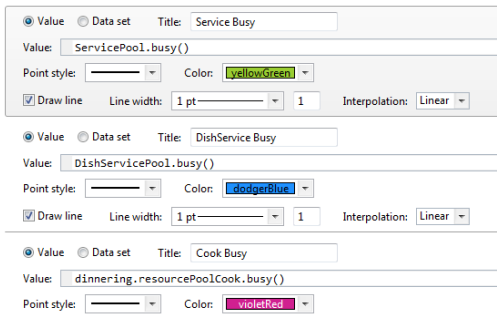


Abbildung 36 Time Plot Daten

Name	Wert
Service Busy	`ServicePool.busy()`
DishService Busy	`DishServicePool.busy()`
Cook Busy	`dinnering.resourcePoolCook.busy()`

4. Fügen Sie nun für die Verteilung der Gäste ein `Time Stack Chart` aus der Palette `Analysis` ins Modell ein. `Time windows` und die Anzahl zu verwendende Datensätze muss wie beim ersten Diagramm konfiguriert werden. Als Datensätze verwenden Sie bitte Folgende:

Name	Wert
Bar	`entering.bar.size()`
Table	`networkStorageZone.size()`
Floor	`entering.restrictedAreaStart.capacity - entering.bar.size() - networkStorageZone.size()`

5. Damit auf der X Achse die Zeit in einem gut leserlichen Format steht, stellen Sie bitte bei beiden Diagrammen unter `Appearance` das `Time axis format` auf `Model date (time only)`.
6. Testen Sie die Simulation und überprüfen Sie, ob Diagramme korrekt funktionieren.

10 Parametrisierung

Damit verschiedene Szenarien einfach simuliert werden können, sollen beim Simulationsstart die wichtigsten Werte vor dem Start einstellbar sein.

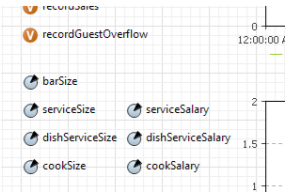


Abbildung 37 Parameter

1. Löschen Sie die Variable `barSize`. Diese wird mit nun ersetzt.
2. Fügen Sie folgende `Parameter` Elemente aus der `General` Palette zum `Main` Modell hinzu:

Name	Type	Default value
barSize	`int`	`40`
serviceSize	`int`	`15`
dishServiceSize	`int`	`10`
cookSize	`int`	`15`
serviceSalary	`double`	`25`
dishServiceSalary	`double`	`20`
cookSalary	`double`	`25`

3. Ändern Sie die `Capacity` von `ServicePool` auf `serviceSize` und diejenige von `DishServicePool` auf `dishServiceSize`. Die `Capacity` von `resourcePoolCook` muss auf `get_Main().cookSize` geändert werden.
4. Da die Variable `barSize` durch einen gleichnamigen Parameter ersetzt wurde, muss in dieser Hinsicht nichts geändert werden.

5. Öffnen Sie `Simulation: Main` im Projektexplorer mit einem Doppelklick.
6. Fügen Sie die gleichen Elemente wie bei Schritt 2, mit den gleichen Werten aber als Variablen und diesmal nicht als Parameter ein.
7. Für die Einstellung der Grössen werden `Slider` verwendet, welche unter `Controls` zu finden sind. Ziehen Sie das Element auf die Arbeitsfläche und geben Sie bei `Minimum value` und `Maximum value` den Range an, in welchem sich der Wert bewegen darf. Bei `Link to` muss die entsprechende Variable angegeben werden. Verwenden Sie folgende Ranges:

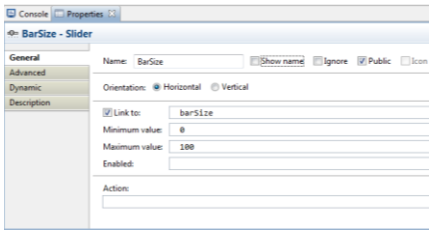


Abbildung 39 Slider Konfiguration

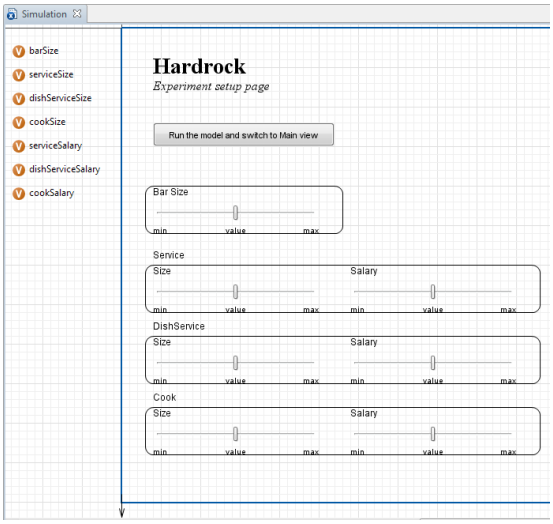


Abbildung 38 Parametrisierung vor Simulationsstart

Slider	Link to	Minimum	Maximum
BarSize	barSize	0	100
ServiceSize	serviceSize	1	30
DishServiceSize	dishServiceSize	1	30
CookSize	cookSize	1	30
ServiceSalary	serviceSalary	0	50
DishServiceSalary	dishServiceSalary	0	50
CookSalary	cookSalary	0	50

8. Damit die Variablen an die Parameter übergeben werden, muss in den Einstellungen von `Simulation` (Mausklick auf eine freie Arbeitsfläche) unter `General` die Variablen angegeben werden. In diesem Beispiel sind die Variablen und Parameter gleichnamig. Geben Sie nun bei jedem Feld den entsprechenden Variablennamen an.
9. Starten Sie die Simulation einige Male und schauen Sie, ob die Parametrisierung funktioniert.
10. Fügen Sie die zwei Variablen `salary` und `saleMinusSalary` hinzu. Beide sollen vom Typen `double` sein und den Initialwert 0 haben.
11. Fügen Sie nun einen `Event`, ebenfalls aus der Palette `General`, hinzu und nennen Sie ihn `updateSalary`. Dieser Event soll jede Minute den aktuell zu bezahlenden Lohn berechnen und in die Variable `salary` abspeichern. Danach soll auch `saleMinusSalary` aktualisiert werden.

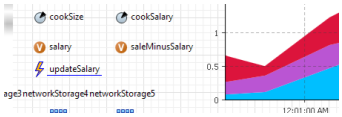


Abbildung 40 Salary Event

12. Der `Mode` des Event muss auf `Cyclic` geändert werden und die erste Ausführung (`First occurrence time`) auf eine Minute (`minute()`) gesetzt werden. Danach soll der Event jede Minute ausgeführt werden. Dies erreicht man, indem man die `Reccurrence time` auf `minute()` setzt. Als `Action` muss der Code der Berechnung angegeben werden.

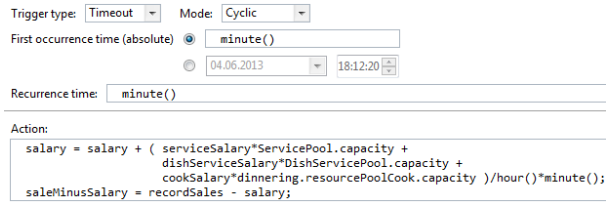


Abbildung 41 Konfiguration des Events

```
salary=salary+(serviceSalary*ServicePool.capacity +
               dishServiceSalary*DishServicePool.capacity +
               cookSalary*dinnering.resourcePoolCook.capacity )/hour()*minute();
saleMinusSalary = recordSales - salary;
```

13. Testen Sie die Simulation und rechnen Sie nach, ob der Event den Lohn richtig ausrechnet.

Abbildung 42 Fertige Simulation in AnyLogic

11 3D Animation

Eine 2D Animation in eine 3D Animation zu ändern ist relativ einfach.

1. Fügen Sie ein `3D Window` aus der `3D` Palette in das Modell ein. Hier wird die 3D Animation während der Simulation angezeigt.
2. Die Bilder der Entitäten und Ressourcen müssen mit einem aus der Palette `3D Objects` ersetzt werden.
3. Alle Elemente welche in der 3D Ansicht erscheinen sollen müssen die `Checkbox Show in 3D scene` aktiviert haben.
4. Um Wände zu zeichnen kann die `Polyline` aus der `3D` Palette verwendet werden. Über die `Z-Height` kann die Höhe und über die `Line width` die Dicke der Wand eingestellt werden. Über `Line color | Textures..` kann der Wand eine Textur angegeben werden, sodass die Wand echter wirkt.

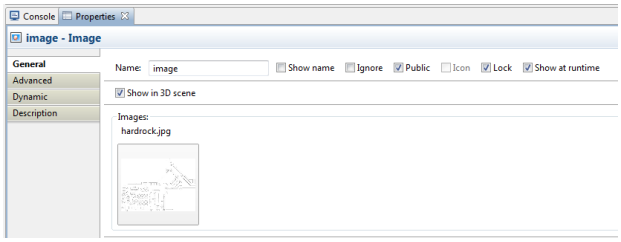


Abbildung 43 Show in 3D scene

> *Mit gedrückter Maustaste kann die 3D Ansicht nach links, rechts, vor und zurück bewegt werden. Mit Hilfe des Scrollrads kann der Zoom verstellt werden und mit gedrückter Alt Taste und gedrückter Maustaste kann der Blickwinkel gedreht werden.*

C. HOW TO SIMIO

In SIMIO erstellen wir ein Modell, um ein Restaurant zu simulieren. Als Vorlage wird das Hardrock Cafe Atlantic City verwendet. Die Bar wird als Queue für die Tische verwendet und zur Vereinfachung wird nur das Essen simuliert. Diese Anleitung hat keinen Anspruch auf die perfekte Lösung und darf nach eigenem Ermessen angepasst werden. Sie soll den Einstieg in die Simulation mit SIMIO erleichtern und zeigt ein paar Aspekte.

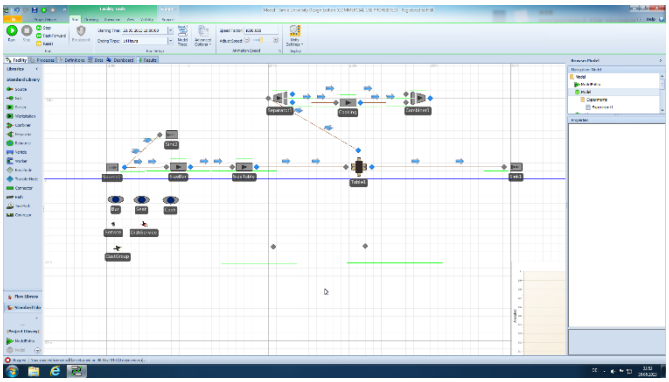


Abbildung 44 SIMIO Projekt

1 Installation

Für dieses Tutorial wurde Simio 5.81.9562 auf einem Windows 7 x64 Betriebssystem verwendet. Als Lizenz wurde der Lizenzserver der HSR angegeben. Da die HSR eine Academic Version besitzt, darf die Software nur auf Hardware der HSR installiert werden.

Es wird empfohlen die Installation mit Administrationsrechten auszuführen. Nach dem die EULA akzeptiert wurde, können Sie die Software an den vorgeschlagenen Ort installieren.

Nach dem Öffnen von Simio kann unter `File | Licesing` der Lizenzserver angegeben werden. Sobald mit dem Lizenzserver verbunden wurde, kann in diesem Dialog mit `Acquire Roaming License` eine Lizenz für eine Zeit ausgeliehen werden. Sobald die Verbindung zum Lizenzserver verloren geht und keine Lizenz ausgeliehen wurde, wechselt der Lizenzstatus auf `unlicensed evaluation version` und die Software hat gewisse Einschränkungen. Wenn das Projekt mehr als 18 Objekte in einem Modell, mehr als 15 Schritte in einem Modell, mehr als 3 Modelle in einem Projekt hat oder wenn Elemente, welche nicht von der Standard Library sind, verwendet werden, kann das Projekt nicht mehr abgespeichert werden.

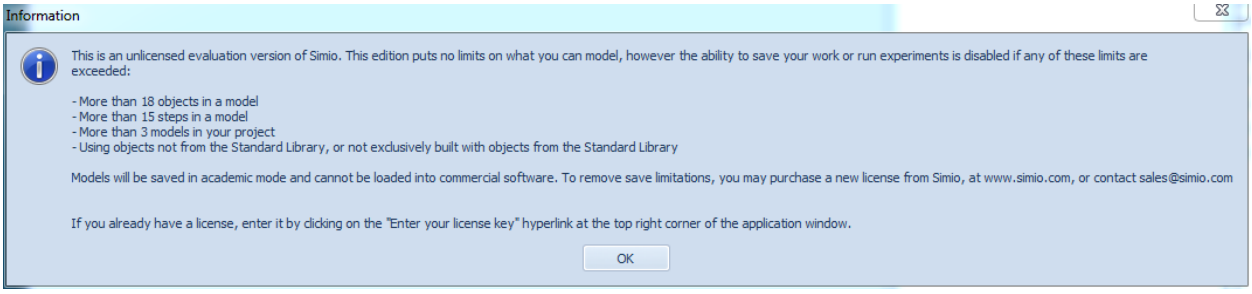


Abbildung 45 Evaluationslizenzeinschränkungen von Simio

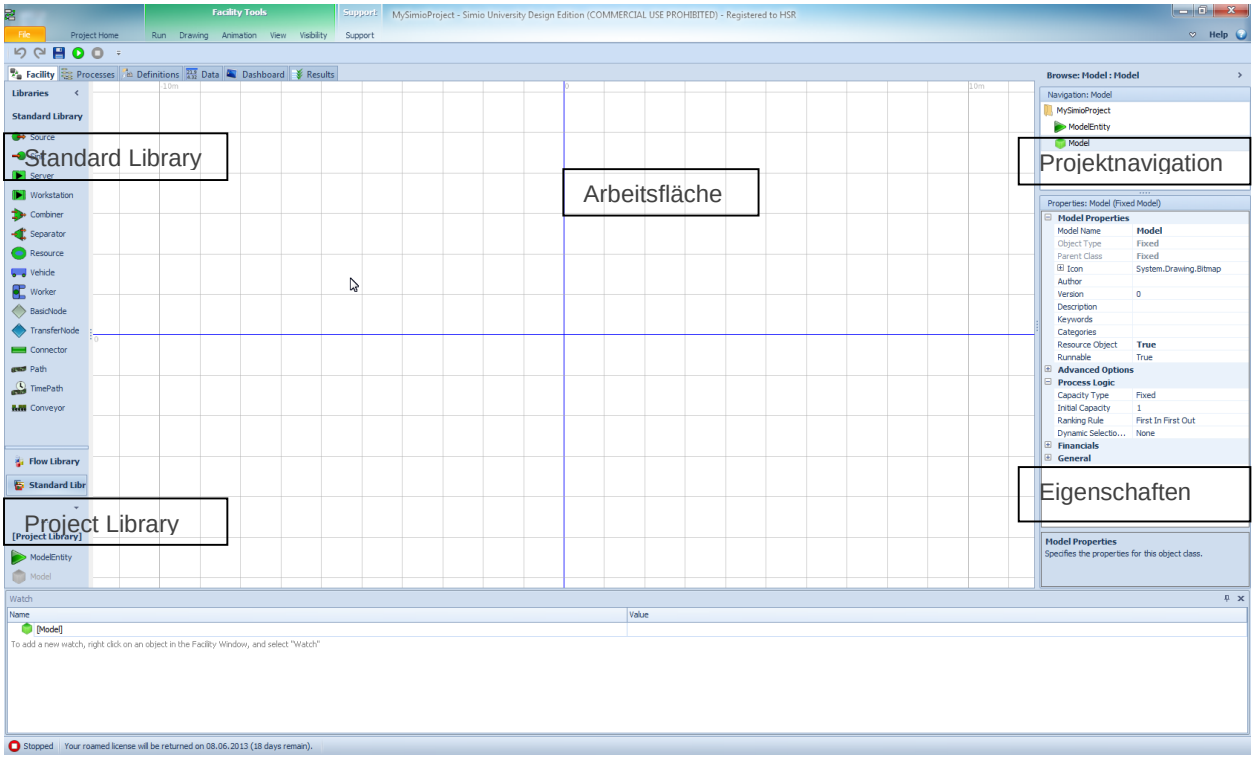


Abbildung 46 Übersicht Arbeitsfenster Simio

Die Arbeitsfläche von Simio ist standardmässig in der 2D Ansicht. Duch Drücken der Taste `3` kann in die 3D Ansicht gewechselt werden. Alle Elemente aus der Standard Library sind in 3D. In der 3D Ansicht kann mit der Taste `W` in den `Walk` Modus gewechselt werden. Mit der Taste `2` kann zurück in den 2D Modus geschaltet werden. Die Maustasten können für das Bewegen, Zoomen und Rotieren der Ansicht verwendet werden.

2 Neues Projekt erstellen

1. Beim Start von Simio wird automatisch ein leeres Projekt gestartet. Speichern Sie über `File | Save As` das Projekt an einem gewünschten Ort ab.
2. Um ein Element aus der Standard Library auf der Arbeitsfläche zu platzieren, klicken Sie auf das Element und es wird an den Mauszeiger geheftet. Mit dem Klick auf den gewünschten Ort in der Arbeitsfläche wird das Element platziert.
3. Platzieren Sie eine `Source`, fünf `Server` und ein `Sink` nebeneinander auf der Arbeitsfläche.
4. Verbinden Sie die Elemente jeweils mit einem `Path`. Ein `Path` ist eine Verbindung, bei welcher Entitäten eine gewisse Zeit brauchen, um die Strecke zurückzulegen. Werden zwei Elemente mit einem `Connector` verbunden, haben die Entitäten beim Transfer keine Verzögerung.
5. Benennen Sie die fünf Server in `Bar`, `Table`, `Ordering`, `Cooking` und `Eating` um.

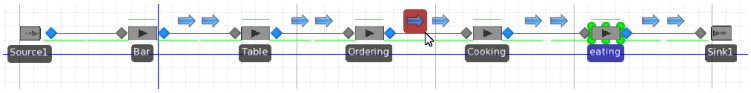


Abbildung 47 Erste Simio Simulation

6. Tauschen Sie die Verbindung zwischen `Ordering` und `Cooking` und `Cooking` und `Eating` mit einem `Connector` aus.
7. Starten Sie die Simulation um zu testen, ob alles fehlerfrei funktioniert.

> *Mit Doppelklick auf `Connector` oder `Path` können (bis die rechte Maustaste gedrückt wird) mehrere Verbindungen gezeichnet werden. Mit Rechtsklick auf eine Verbindung kann via `Convert To Type` die Art gewechselt werden.*

3 Ressourcen

1. Wechseln Sie zur Ansicht `Definitions` und klicken Sie in der `Views` Liste links auf `Properties`.
2. Über Standard `Property > Integer` wird ein Parameter vom Typen Integer erstellt. Erstellen Sie die Integer Parameter `BarSize`, `TableSize`, `ServiceSize`, `DishServiceSize` und `CookSize`. Setzen Sie im Eigenschaftenfenster rechts die entsprechenden `Default Value` von `BarSize` auf 40, `TableSize` auf 250 und die anderen drei auf 10.

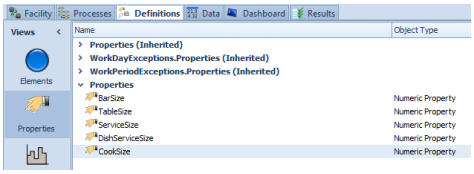


Abbildung 48 Integer Properties

3. Wechseln Sie zurück zur `Facility` Ansicht und benennen Sie die Server `Bar` in `SizeBar` und `Table` in `SizeTable` um.

> *Objekte könne über einen Doppelklick auf den Namen, mit F2 wenn das Element markiert ist oder über die Eigenschaften umbenannt werden.*

4. Fügen Sie eine `Resource` hinzu und setzen Sie den Namen auf `Bar`. Setzen Sie nun in den Eigenschaften `Inital Capacity` auf `BarSize`.

> *Durch das Drücken auf das Symbol mit dem Pfeil nach unten, welches bei manchen aktiven Feldern sichtbar wird, öffnet sich ein grösseres Eingabefeld, um Expressions einzugeben. Dieses Feld unterstützt mit einer Autovervollständigung und zeigt mögliche Befehle oder Elemente an. Die Autovervollständigung kann mit der Pfeiltaste nach unten hervorgehoben werden, wenn Sie nicht bereits dargestellt wird.*

5. Fügen Sie nun zwei `Resource` Elemente hinzu und nennen Sie diese `Table` und `Cook`. Geben Sie die entsprechenden Parameter `TableSize` und `CookSize` als `Inital Capacity` an.
6. Da sich die Servicekräfte und Tellerbringer im Modell bewegen müssen, werden für diese Ressourcen Elemente vom Typ `Worker` verwendet. Fügen Sie zwei Elemente hinzu und nennen Sie diese `Service` und `DishService`. Geben Sie in den Eigenschaften unter `Population` im Feld `Inital Number In System` die entsprechenden Parameter `ServiceSize` und `DishServiceSize` an.

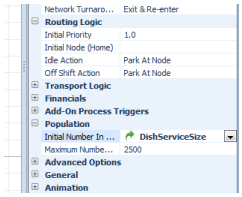


Abbildung 49 Worker

7. Setzen Sie die `Processing Time` des Elements `SizeBar` auf 0 und die `Inital Capacity` auf `Infinity`, da wir die Anzahl der Entitäten welche gleichzeitig bei diesem Prozess stehen dürfen, über die Ressource `Bar` regeln.
8. Klicken Sie auf das Feld `On Entering` unter `Secondary Resources > Other Resource Seizes`. Nun wird ein Button mit drei Punkten sichtbar. Durch Klick auf diesen Button öffnet sich der Dialog `On Entering – Repeating Property Editor`. Klicken Sie auf `Add` um einen neuen Datensatz hinzuzufügen und geben Sie als `Object Name` die Ressource `Bar` an. Mit `Close` bestätigen Sie Ihre Eingabe.
9. Führen Sie den gleichen Schritt für das Objekt `SizeTable` durch verwenden Sie die Ressource `Table`.

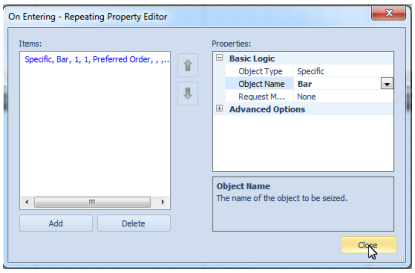


Abbildung 50 Ressource belegen

> *Damit ein Service von mehreren Entitäten belegt werden darf, ist es wichtig, dass die `Inital Capacity` angepasst wird. Dies geht gerne vergessen und verfälscht das Resultat enorm.*

10. Geben Sie bei `Ordering` unter `Secondary Resources > Resource for Processing > Object Name` die Ressource `Service` an und bei `Cooking` dementsprechend die Ressource `Cook`.
11. Überprüfen Sie bei allen Servern, ob die `Inital Capacity` auf `Infinity` gesetzt ist und setzen Sie diese gegebenenfalls.
12. Führen Sie die Simulation aus und schauen Sie, ob alles ohne Fehler läuft.

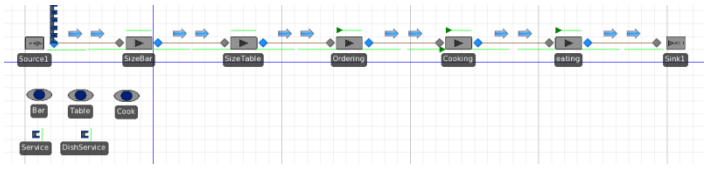


Abbildung 51 Simulation mit Ressourcen

4 Netzwerk

1. Fügen Sie zwei `BasicNode` mit den Namen `HomeService` und `HomeDish` hinzu. Diese Nodes sollen als Standort für die beweglichen Ressourcen dienen. Wenn der `Node` markiert ist, kann über `Draw Queue > ParkingStation.Contents` eine Line gezeichnet werden, auf welcher die Ressourcen aufgereiht sein sollen, wenn sie nicht unterwegs sind. Zeichnen Sie für beide Punkte je eine Queue.

> *Damit die Ressourcen nur bei der gezeichnet Queue angezeigt werden und neben dem `Node` gestapelt werden, darf `Parking Queue` nicht aktiviert sein.*

2. Geben Sie bei den zwei `Worker` `Service` und `DishService` die erstellen Punkte(`HomeService` und `HomeDish`) als `Inital Node (Home)` an. Dieses Feld ist in der Kategorie `Routing Logic` zu finden. Damit die Arbeiter nach dem sie beschäftig waren, wieder zurück zu ihrem Standplatz gehen, muss `Idle Action` und `Off Shift Action` auf `Go To Home` gesetzt werden.
3. Damit kein Netzwerk mit `Node` und `Path` Elementen gezeichnet werden muss, wird bei den beiden `Worker` für `Inital Network` aus dem Dropdown Menü `No Network (Free Space)` ausgewählt.

> *Standardmässig sind Elemente vom Typ `Path` unidirektional. In den Eigenschaften können Sie auf bidirektional umgestellt werden. So muss beim Zeichnen eines Netzwerks nur ein `Path` gezeichnet werden um zwei Punkte zu verbinden.*

4. Damit nun die Servicekraft zum Tisch gehen muss, um die Bestellung aufzunehmen, muss bei `Ordering` unter `Secondary Resources > Resource for Processing` das Feld `Request Move` auf `To Node` umgestellt werden. Bei `Destination Node` muss nun im Dropdown Menü `Input@Ordering` ausgewählt werden.
5. Führen Sie die Simulation aus und überprüfen Sie ob sie sich wie erwartet verhält.

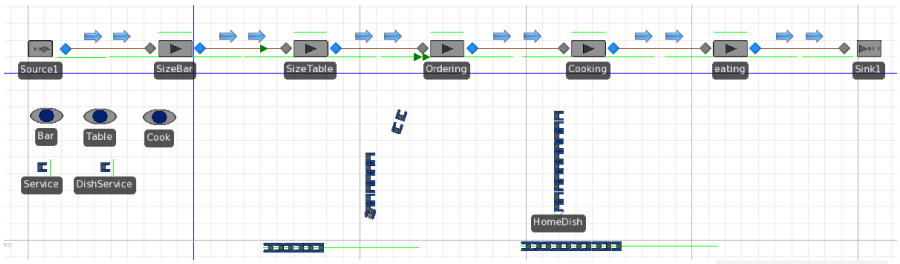


Abbildung 52 Simulation mit bewegten Servicekräften und aktivem `Parking Queue`

5 Neues Modell

Ein Tisch soll als eigenes Modell simuliert werden. In dieser Anleitung wird nur ein Tisch mit 250 Plätzen im Hauptmodell verwendet. In einem weiteren Schritt könnte man verschiedene Tische verwenden. Dann müsste jedoch eine Art Routingtabelle verwendet werden, um zu entscheiden welche Gäste an welchen Tisch sitzen sollen.

1. Erstellen Sie ein neues Modell indem Sie in der Projektnavigation mit der rechten Maustaste auf den Ordner `Model` klicken und `New Model` auswählen.
2. Im Eigenschaftenfenster werden automatisch die Eigenschaften des neuen Modells angezeigt. Ändern Sie dort den `Model Name` auf `Table`.
3. Fügen Sie nun einen `BasicNode` namens `Input`, drei `Server` namens `Ordering`, `Dish`, `Eating` und einen `TranferNode` namens `Output` ein. Verbinden Sie diese Elemente mit dem Connector.

Abbildung 53 Sub Modell Table

4. Setzen Sie bei allen Servern die `Inital Capacity` auf `Infinity`.
5. Wechseln Sie auf `Definitions` und gehen Sie dann auf `External`, um zu sehen, wie das Modell aussieht, wenn es in einem anderen Modell eingebunden wird. Momentan werden alle modellierten Elemente dargestellt.
6. Wechseln Sie zurück auf die `Facility` Ansicht und markieren Sie alle Elemente.

> *Ist die Ctrl Taste gedrückt, kann mit der Maus ein Rahmen gezogen werden um mehrere Elemente zu markieren.*

7. Mit einem Rechtsklick auf ein markiertes Element kann der Punkt `Externally Visible` deaktiviert werden.
8. Wechseln sie wieder zurück zur `External` Ansicht unter `Definitions`. Nun sollten keine Elemente mehr sichtbar sein.
9. Klicken Sie nun im `Drawing` Ribbon auf `Place Symol` und wählen Sie `Download Symbol` aus, um ein 3D Bild aus dem Google 3D Warehouse herunter zu laden.
10. Suchen Sie nach `dinning table` und klicken Sie auf das fünfte Ergebnis. Wenn Ihnen die Grafik gefällt wählen Sie `Downlaod Model`.
11. Überprüfen Sie im Import Dialog, ob die Grösse relativ zur orangen Person realistisch ist und passen Sie gegebenenfalls die `Length`, `Width` oder `Height` an, bevor Sie mit OK den Import bestätigen. Der Wechsel von der 2D in die 3D Ansicht kann sehr hilfreich sein, um die Dimensionen zu sehen.
12. Platzieren Sie nun das Symbol auf die Arbeitsfläche.
13. Da in einem Modell die Begriffe nur einmal verwendet werden können, müssen die beiden `Node Input` und `Output` in der `Facility` Ansicht in `Start` und `Ende` umbenennt werden. Nun wechseln Sie zurück in die `External` Ansicht.

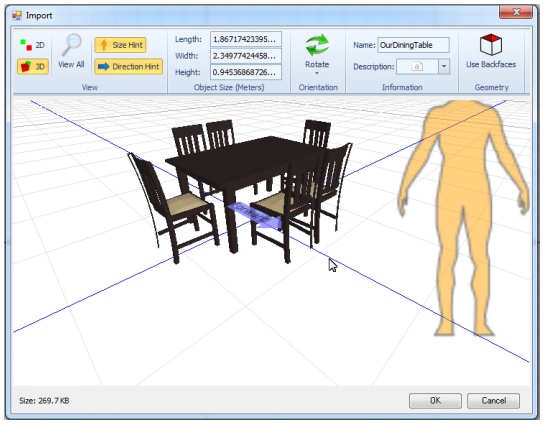


Abbildung 54 3D Grafik Import Dialog

14. Fügen Sie über den Ribbon `Drawing > External Node` einen `Node` vor dem Tisch ein und nennen Sie diesen `Input`. Wählen Sie als `Node Class` die Klasse `BasicNode`. Der `Input Logic Type` muss auf `FacilityNode` gesetzt werden und als `Node` geben Sie `Start` an.
15. Setzen Sie nun nach dem Tisch noch ein `External Node` und nennen Sie diesen `Output`. Hier muss das Feld `Node Class` auf `TransferNode` gesetzt werden. `Input Lgic Type` bleibt auf `None`.

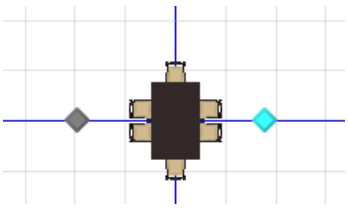


Abbildung 55 Externe Ansicht

16. Gehen Sie zurück zum Hauptmodell, indem Sie in der Projektnavigation auf `Model` klicken.
17. Löschen Sie in der `Facility` Ansicht die Server `Ordering`, `Cooking` und `Eating`, da diese nun im Modell `Table` enthalten sind und dieses Modell hier eingefügt werden soll.
18. Blättern Sie in der `Project Library` nach unten bis `Table` ersichtlich ist und ziehen Sie dieses Element auf die Arbeitsfläche. Fügen Sie nun einen `Path` zwischen `SizeTable` und `Table1` und einen zwischen `Table1` und `Sink1` ein.
19. Da nun das neue Modell und die Ressourcen `Table` heissen wird ein Fehler im unteren Fenster angezeigt. Ändern Sie den Namen der Ressource `Table` auf `Seat`.
20. Speichern Sie die Simulation und führen Sie sie aus.
21. Tritt bei der Simulation der Fehler auf, das von `End` die Entität nicht an einen `outbound link` gesendet werden kann, ist der `TransferNode` `End` nicht sauber an den `External Node` gebunden.
22. Öffnen Sie die die `Facility` Ansicht des Modells `Table` und setzen Sie in den Eigenschaften von `End` unter `Advanced Options` das Feld `Bound External Node` auf `Output`.

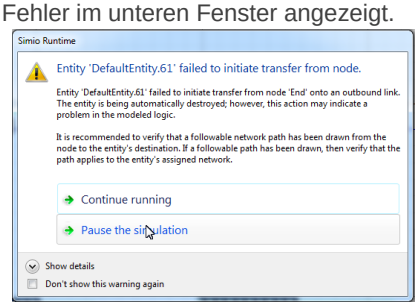


Abbildung 56 Fehlermeldung Initiate transfer

> *Bei einem Rechtsklick auf einen `Node` kann `Bind to New External Input Node` oder `Bind to New External Output Node` ausgewählt werden.*

6 Gruppengrösse

1. Öffnen Sie `ModelEntity` durch einen Doppelklick auf den entsprechenden Eintrag in der Projektnavigation.
2. Fügen Sie unter `Definitions > States` zwei `Integer` Variablen hinzu indem Sie im Ribbon `States` auf `Integer` klicken. Nennen Sie die Variablen `GroupSize` und `GroupID`. `GroupSize` wird für die Gruppengrösse verwendet und `GroupID` um die geklonten Elemente wider zusammenzufügen.

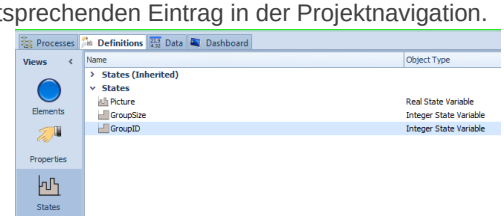


Abbildung 57 ModelEntity

3. Kehren Sie zurück zum `Model` und ziehen Sie das `ModelEntity` Element aus der `Project Library` auf die Arbeitsfläche. Benennen Sie das Element `CustGroup`. Nun muss dieser Entitätstype bei `Source1` unter `Entity Type` angegeben werden.
4. Klappen Sie nun bei `Source1` die Kategorie `State Assignments` auf, klicken Sie in das Feld `Before Exiting`, damit der Button ersichtlich wird und drücken Sie auf diesen.
5. Drücken Sie `Add` und wählen Sie bei `State Variable` im Menü `ModelEntity.GoupSize` aus. Als New Value wird die Expression `Math.Round(Random.Triangular(1,2,4))` angegeben um die Gruppengrösse zu definieren.
6. Fügen Sie einen zweiten Datensatz hinzu. Als `State Variable` soll `ModelEntity.GroupID` und als `New Value` `ModelEntity.ID` verwendet werden.

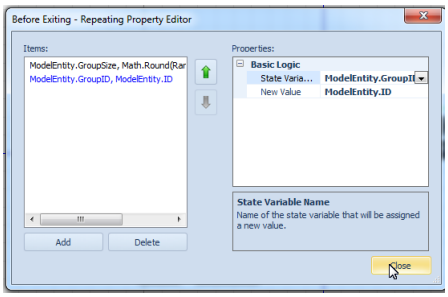


Abbildung 58 Zuweisung Gruppengrösse

7. Damit die richtige Anzahl an `Bar` und `Seat` Ressourcen von einer Gruppe beansprucht wird, muss bei `SizeBar` und `SizeTable` unter `Secondary Resources > Other Resource Seizes` der entsprechende Eintrag angepasst werden. Fügen Sie im Dialogfenster unter `Advanced Options > Units Per Object` die Expression `ModelEntity.GroupSize` ein. Stellen Sie sicher, dass bei der Freigabe der Ressource `Bar` die Gruppengrösse auch berücksichtig wird (bei `SizeTable`).

7 Ressource an das Modell Tisch übergeben

1. Öffnen Sie das Modell `Table` und wechseln Sie zur `Definitions > Properties` Ansicht.
2. Erstellen Sie einen neuen Parameter vom Typen `Object` über den Button `Object Reference` und ändern Sie den Namen auf `ReleaseAfterProcessing`. Setzen Sie nun `Required Value` auf `False` und bei `Category Name` wählen Sie `Process Logic` aus.
3. Fügen Sie einen Parameter vom Typen `Transporter` mit dem Namen `ServiceTransporter` und einen vom Typen `Node` mit dem Namen `TransporterNode` mit den gleichen Anpassungen bei den Einstellungen hinzu. Über diese Parameter werden die benötigten Objekte dem Modell `Table` übergeben.
4. Wechseln Sie nun zu `Facility` und bearbeiten Sie `Eating`. Unter `Secondary Resources > Other Resource Releases > After Processing` soll die Ressource `ReleaseAfterProcessing` mit der Gruppengrösse freigegeben werden, da über dieser Parameter die Ressource `Seat` übergeben werden wird. Bis jetzt wurde die Ressource automatisch freigegeben, wenn die Entitäten über das `Sink` Objekt die Simulation verlassen haben. Nun soll der Platz am Tisch freigegeben werden, sobald der Tisch verlassen wird. Da `Eating` der letzte Prozess am Tisch ist, wird es deshalb nach diesem Prozess konfiguriert.
5. Fügen Sie für die im Dialogfenster für die Freigabe der Ressourcen einen neuen Datensatz ein. Als `Object Name` muss `ReleaseAfterProcessing` von Hand eingegeben werden, da es nicht im Dropdown Menü aufgelistet wird. Der grüne Pfeil vor dem Parameternamen zeigt, dass der Parameter erkannt wurde.
6. Unter `Unit Per Object` muss nun `ModelEntity.GroupSize` als Expression angegeben werden. Sobald nun eine Ressource dem Parameter `ReleaseAfterProcessing` mitgegeben wird, wird diese nachdem der Prozess `Eating` abgeschlossen ist freigegeben.

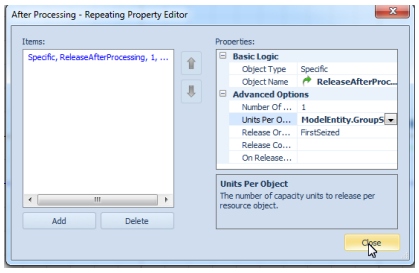


Abbildung 59 Parametrisierte Freigabe

7. Damit die Servicekräfte zum Tisch gehen können, muss auch `Ordering` bearbeitet werden. In das Feld unter `Secondary Resources > Resource for Processing > Object Name` muss `ServiceTransporter` geschrieben werden und bei `Request Move` wählen Sie `To Node` aus. Nun kann unter `Destination Node` in das Feld `TransporterNode` geschrieben werden.
8. Wechseln Sie zurück zu Model und stellen Sie sicher, dass bei `SizeTable` die Gruppengrösse bei der Belegung von `Seat` berücksichtig wird.
9. Übergeben Sie nun bei `Table1` die Objekte an die Parameter. Die erstellten Parameter sind nun wie angegeben in der Kategorie `Process Logic` aufgelistet. Bei `ReleaseAfterProcessing` wählen Sie `Seat` aus und als `ServiceTransporter` geben Sie `Service` an. Mit `TransporterNode` geben Sie an, zu welchem Knoten der Transporter für die Bestellung gesendet werden soll. Wählen Sie `Input@Table1` aus der Liste aus.

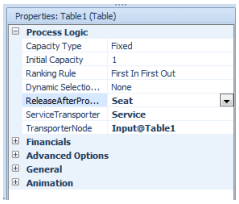


Abbildung 60 Table1

10. Kontrollieren Sie ob die Simulation korrekt abläuft und die Servicekräfte zum richtig Ort laufen um die Bestellung aufzunehmen.

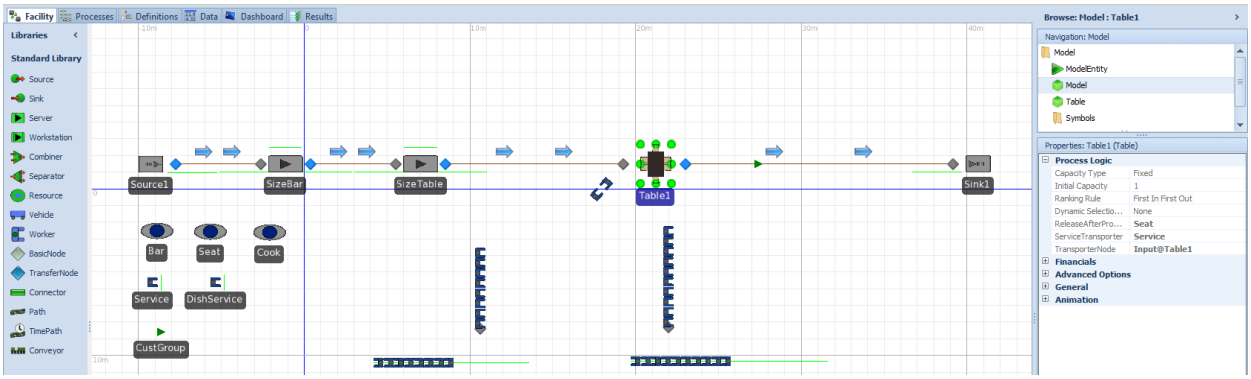


Abbildung 61 Simulation mit Service unterwegs zum Tisch

8 Aufteilung der Bestellung

Damit in der Küche die Gerichte pro Grast gekocht werden können, muss die Bestellung auf die Gruppengrösse aufgeteilt werden, damit im besten Fall für eine Vierergruppe vier Köche die Speisen parallel zubereiten können. Danach werden die Speisen wieder zusammengefügt und vom Tellerbringer gebracht. In diesem Schritt soll die Bestellung erstellt werden, damit diese in die Küche gesendet werden kann.

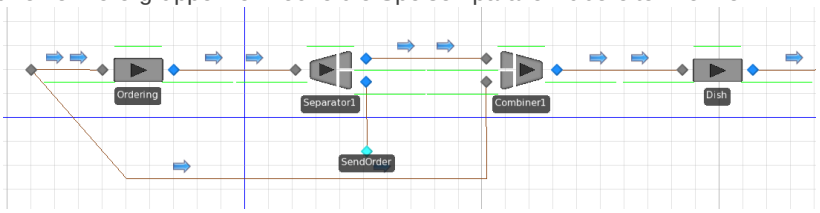


Abbildung 62 Bestellung Senden und Essen Empfangen

1. Öffnen Sie das Modell `Table` und fügen Sie zwischen `Ordering` und `Dish` einen `Separator` und einen `Combiner` ein. Verbinden Sie `Ordering` mit dem `Separator`, den `Separator` mit dem `Combiner` über die oberen Nodes und den `Combiner` mit `Dish`. Verwenden Sie den Verbindungstypen `Connector`.
2. Fügen Sie einen `TransferNode` mit dem Namen `SendOrder` hinzu und verbinden Sie den zweiten Ausgang von `Separator1` mit diesem.
3. Ziehen Sie eine Verbindung von `Start` zum zweiten Eingang von `Combiner1`.
4. Setzen Sie bei `Separator1` den `Seperation Mode` auf `Make Copies`. Zusätzlich muss die `Inital Capacity` auf `Infinity` gesetzt werden. Fügen Sie unter `State Assignments > Before Member Exit` einen Datensatz hinzu, um `ModelEntity.DishInput` auf `TransporterNode` zu setzen. `ModelEntity.DishInput` gibt es noch nicht und wird im nächsten Schritt erstellt. Durch diese Variable weiss der Tellerbringer, wohin das Essen getragen werden muss.
5. Öffnen Sie das `Definitions` Fenster des `ModelEntity`, fügen Sie unter `States` ein `ObjectReference` Objekt vom Typen `Node` hinzu und nennen Sie diesen `DishInput`. Erstellen Sie zusätzlich einen `State` vom Typen `Boolean` und nennen Sie diesen `IsDish`.
6. Kehren Sie nun zurück zur `Facility` Ansicht von `Table` und markieren Sie den `BasicNode` `Start`. Hier muss nun sichergestellt werden, dass das Essen, welche die Tellerbringer an diesen Eingang tragen, direkt zum zweiten Eingang von `Combiner1` weitergeleitet wird. Dies wird mit einem `Add-On Process` sichergestellt. Schreiben Sie `IsDish` in das Feld von `Entered` welche unter `Add-On Process Triggers` zu finden ist. Da der `Add-On Process` `IsDish` noch nicht existiert, muss dieser erstellt werden.
7. Wechseln Sie zu `Processes` und erstellen Sie einen Prozess mit `Create Process` und ändern Sie den Namen auf `IsDish`.
8. Fügen Sie ein `Decide` Element und auf beide Verzweigungen ein `Set Node` Element hinzu. Setzen Sie die `Expression` des `Decide` Elements auf `ModelEntity.IsDish`. Setzen Sie beim `Set Node` Element, welches auf dem `True` Pfad liegt, `Node Name` auf `MemberInput@Combine1` und beim anderen Element auf `Input@Ordering`.

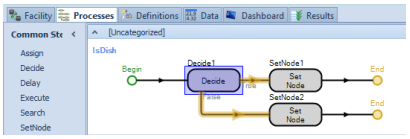


Abbildung 63 Add-On Process

> *Mit einem Klick auf `All Steps (A-Z)` werden alle Elemente angezeigt.*

9. Ändern Sie die `Matching Rule` von `Combiner1` auf `Match Members` und setzten sie die `Member Match Expression` auf `ModelEntity.GroupID`. Zudem muss `Inital Capacity` auf `Infinity` gesetzt werden.
10. Verknüpfen Sie `SendOrder` mit einem `External Output Node`. Klicken Sie mit der rechten Maustaste auf `SendOrder` und wählen Sie `Bind to New External Output Node`. Geben Sie dem `External Node` den Namen `OrderOutput`.
11. Deaktivieren Sie bei allen neu hinzugefügten Elementen mit einem Rechtsklick `Externally Visible`. Wechseln Sie zur `External` Ansicht und platzieren Sie den neuen `External Node` oberhalb des Tisches.

9 Kochen

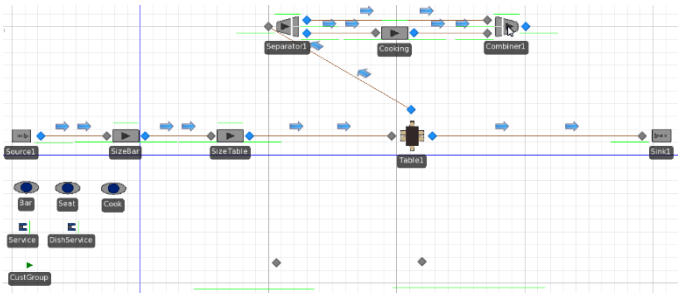


Abbildung 64 Modell mit Küche

1. Fügen Sie einen `Seperator`, einen `Server` und einen `Combiner` zum Hauptmodell hinzu und verbinden Sie diese wie in Abbildung 64 dargestellt mit Connectoren.
2. Setzen Sie bei `Seperator1` den `Separation Mode` auf `Make Copies` wobei `ModelEntity.GroupSize` bei `Copy Quantity` angegeben wird. Als `Copy Entitiy` Type muss `CustGroup` angegeben werden und `Initial Capacity` soll auf `Infinity` gesetzt werden.
3. Ändern Sie den Namen des Servers auf `Cooking` und setzen Sie auch hier die `Initial Capacity` auf `Infinity`. Unter `Secondary Resources > Resource for Processing` muss bei `Object Name` die Ressource `Cook` angegeben werden.
4. Ändern Sie die `Matching Rule` von `Combiner1` auf `Match Members` und setzen sie die `Member Match Expression` auf `ModelEntity.GroupID`. Die `Batch Quantity` soll auf `ModelEntity.GroupSize` gesetzt werden. Zudem muss `Inital Capacity` auf `Infinity` gesetzt werden.
5. Damit die Tellerbringer das Essen von `Output@Combiner1` zum Tisch bringen, muss der `TransferNode` konfiguriert werden. Setzen Sie bei `Output@Combiner1` die `Entity Destination` auf `Specific`, damit `ModelEntity.DishInput` als `Node Name` angegeben werden kann. Dies ist nötig, damit verschiedene Tische verwendet werden könnten. Damit der Tellerbringer die Entität begleitet, muss `Ride On Transporter` auf `True` gesetzt werden. Jetzt kann unter `Transporter Name` der Worker `DishService` angegeben werden. Fügen Sie unter `State Assignments > Before Exiting` ein Datensatz hinzu, damit `ModelEntity.IsDish` auf `True` gesetzt wird.
6. Überprüfen Sie ob bei `DishService` das Netzwerk auf `No Network` gesetzt ist.
7. Testen Sie die Simulation. Die Tellerbringer sollten nun die Teller zum Tisch tragen.

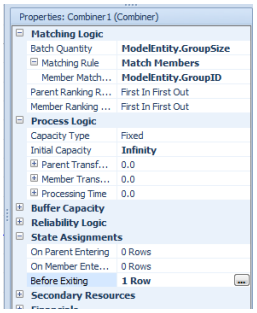


Abbildung 65 Output@Combiner1

> *Ist bei `Combiner1` `Matching Rule` auf `Match Members` eingestellt, kommt es zwar nicht zu einer Fehlermeldung jedoch funktioniert das Zusammenfügen nicht genau wie gewünscht. Deshalb muss dort auf `Match Members And Parent` umgeschalten werden und als `Parent Match Expression` muss ebenfalls `ModelEntity.GroupID` angegeben werden. Wie solche Probleme gefunden werden können sehen Sie im Kapitel* ***Debugging****.*

10 Symbole

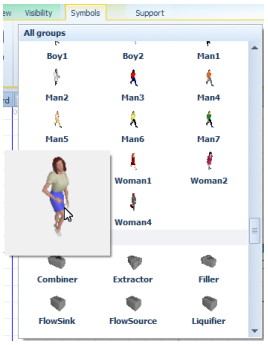


Abbildung 66 Symbole

1. Um einem Element ein Symbol zuzuweisen, markieren sie das Element und drücken im Ribbon `Symbols` im Bereich `Project Symbols` auf den kleinen Pfeil mit dem Oberstrich um alle vorhandenen Grafiken anzuzeigen. Wählen Sie eine Symbol aus und es wird dem Element zugewiesen.
2. Setzen Sie für `Service` das Symbol `Woman3`, für `DishService` `Woman1` und für `CustGroup` `Man7`.
3. Damit CustGroup verschiedene Symbole zugewiesen werden kann, drücken Sie zwei Mal auf `Add Additional Symbol`. Unter `Active Symbol` steht nun `(3 of 3)`. Wenn Sie das Dropdown Menü öffnen, sehen Sie, dass drei Mal das Symbol `Man7` (nummeriert von 0 bis 2) zugewiesen ist. Da momentan das dritte Symbol aktiv ist, klicken Sie auf `Download Symbol` und laden einen Teller vom 3D Warehouse herunter. Danach setzen Sie das zweite Symbol auf aktiv und laden einen Notizzettel herunter. Nun hat `CustGroup` drei verschiedene Symbole.
4. Setzen Sie bei `Combiner1` über `State Assignments > Before Exiting` die Variable `ModelEntity.Picture` auf `2`. Dadurch wird das Symbol 3 angezeigt. Dasselbe machen Sie im Modell `Table` bei `Seperator1` bei `Before Member Exit` und setzen dort die Variable auf `1`.
5. Testen Sie die Simulation.

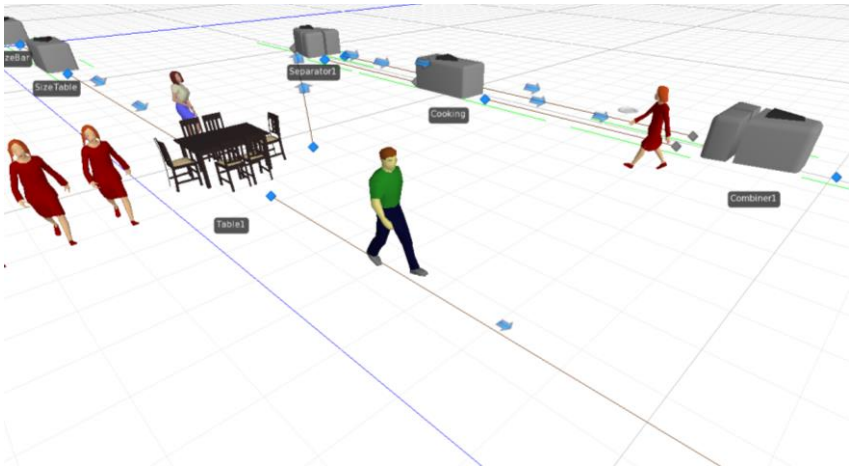


Abbildung 67 3D Simulation mit Symbolen

11 Overflow

1. Fügen Sie dem Modell ein zweites `Sink` Element hinzu und verbinden Sie dieses mit einem `Connector`. Ist die Warteschlange vor dem Restaurant zu gross, sollen die Gäste die Simulation über diesen zweiten `Sink` verlassen ohne das Restaurant je betreten zu haben.
2. Wählen Sie unter `Output@Source1 > Add-On Process Triggers > Entered` im Dropdown Menü `[CreateNew]` aus. Wechseln Sie zum erstellten Prozess und benennen Sie diesen in `IsBareQueueFull` um.
3. Fügen Sie ein `Decide` Element und auf beide Verzweigungen ein `Set Node` Element hinzu. Setzen Sie die `Expression` des `Decide` Elements auf `SizeBar.InputBuffer.EntryQueue.NumberWaiting < 10`. Setzen Sie beim `Set Node` Element, welches auf dem `True` Pfad liegt, `Node Name` auf `Input@SizeBar` und beim anderen Element auf `Input@Sink2`.

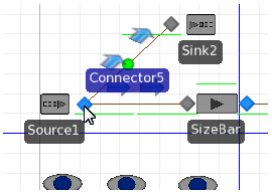


Abbildung 68 Overflow

12 Prozess Verteilfunktionen

1. Gehen Sie im `Model` auf die Ansicht `Data`, klicken Sie im Menü links auf `Rate Tables`. Über den Button `Rate Table` kann ein Scheduler erstellt werden. Benennen Sie die Tabelle in `Arrivals` um und setzen Sie `Number of Intervals` auf 14, da das Restaurant 14 Stunden pro Tag geöffnet ist.
2. Füllen Sie die Tabelle wie in Abbildung 69 dargestellt ab.
3. Stellen Sie bei `Source1` den `Arrival Mode` auf `Time Varying Rate` ein und geben Sie als `Rate Table` die Tabelle `Arrivals` an.
4. Setzen Sie bei folgenden Elementen die entsprechenden Bewegungsgeschwindigkeiten.

Starting Offset	Ending Offset	Rate (events per hour)
Day 1, 00:00:00	Day 1, 01:00:00	53
Day 1, 01:00:00	Day 1, 02:00:00	70
Day 1, 02:00:00	Day 1, 03:00:00	100
Day 1, 03:00:00	Day 1, 04:00:00	60
Day 1, 04:00:00	Day 1, 05:00:00	20
Day 1, 05:00:00	Day 1, 06:00:00	16
Day 1, 06:00:00	Day 1, 07:00:00	51
Day 1, 07:00:00	Day 1, 08:00:00	73
Day 1, 08:00:00	Day 1, 09:00:00	84
Day 1, 09:00:00	Day 1, 10:00:00	100
Day 1, 10:00:00	Day 1, 11:00:00	71
Day 1, 11:00:00	Day 1, 12:00:00	47
Day 1, 12:00:00	Day 1, 13:00:00	20
Day 1, 13:00:00	Day 1, 14:00:00	0

Abbildung 69 Arrivals Tabelle

Element	Inital Desired Speed (Meter per Seconds)
Service	1
DishService	0.5
CustGroup	1

5. Folgende Processing Time müssen bei den entsprechenden Elementen eingestellt werden. Zudem muss überall `Inital Capacity` auf `Infinity` gesetzt werden.

Element	Processing Time
SizeBar	0
SizeTable	0
Cooking	Random.Triangular(1, 5, 10)
Ordering	Random.Triangular(3,6,8)+ModelEntity.GroupSize
Dish	ModelEntity.GroupSize * 0.5
Eating	Random.Triangular(15,40,60)
CookSalary	cookSalary

13 Debugging

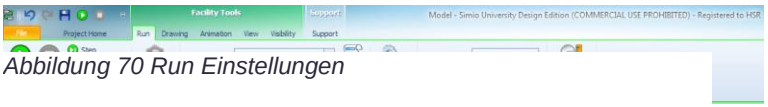


Abbildung 70 Run Einstellungen

In der Run Einstellung kann zum Beispiel die Geschwindigkeit oder der Startzeitpunkt eingestellt werden. Mit `Fast-Forward` kann schnell durch die Simulation gesprungen werden und mit `Breakpoint` könne Breakpoints definiert werden.

Mit Rechtsklick auf ein Element kann `Watch` ausgewählt werden und man kann währenddem die Simulation pausiert ist, die Elemente und deren Eigenschaften / Variablen untersuchen. Das Watch Fenster kann unter `Project Home` manuell angezeigt werden, wenn es nicht bereits angezeigt wird.

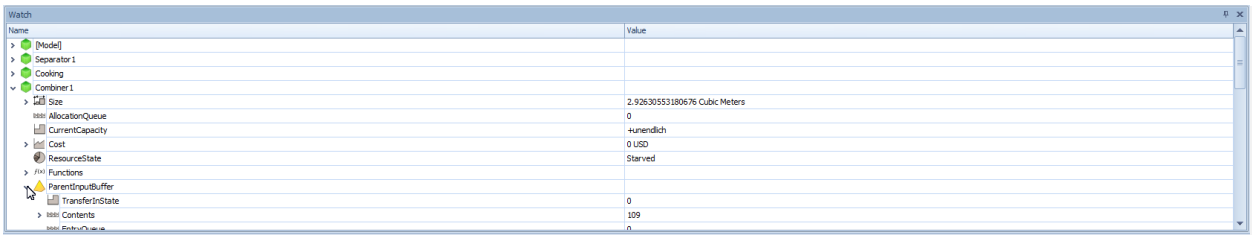


Abbildung 71 Watch Fenster

14 Diagramme

Für die bessere Verständlichkeit werden nun drei Diagramme gezeichnet. Auf dem Ersten soll dargestellt werden wie viele Gäste an den Tischen oder an der Bar sind. Im zweiten Diagramm soll die Auslastung der Ressourcen dargestellt werden. Im dritten Diagramm wird die Gesamtzahl der Gäste angezeigt, welche das Restaurant auf Grund der zu grossen Warteschlange nicht betreten haben. Mit Hilfe dieser Diagramme können schnell Engpässe oder Probleme der Simulationskonfiguration erkannt werden.

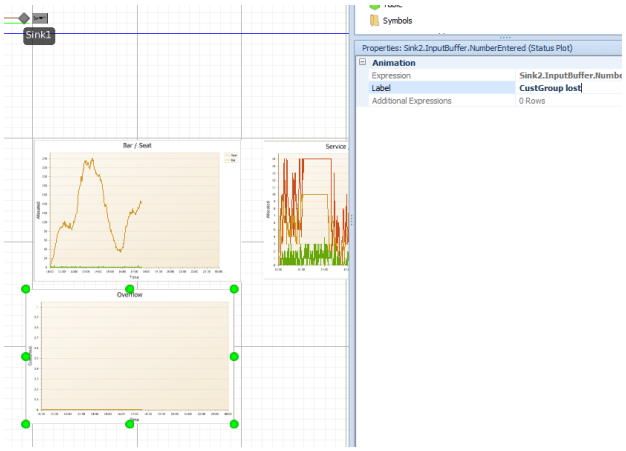


Abbildung 72 Diagramme in Simio

1. Selektieren Sie `Status Plot` unter dem Ribbon `Animation` und zeichnen Sie das erste Diagramm. Geben Sie im Ribbon `Appearance` als `Title` `Bar / Seat` an. Die `X Axis` soll als `Time` und `Y Axis` als `Allocated` angeschrieben sein. Damit der ganze Arbeitstag angezeigt wird, wird der `Time Range` auf `14` Stunden gesetzt. In den Eigenschaften geben Sie als `Expression` `Seat.Capacity.Allocated` an und als `Lable` verwenden Sie `Seat`. Unter `Additional Expressions` soll nun die `Expression` `Bar.Capacity.Allocated` mit dem `Lable` `Bar` angegeben.

2. Erstellen Sie das zweite Diagramm mit der `Y Axis` Beschriftung `Busy` und dem Titel `Service / DishService / Cook`. Die restlichen Einstellungen bleiben gleich. Als Expressions verwenden Sie folgende Angaben:

Lable	Expression
Service	Service.Population.Capacity.Allocated
DishService	DishService.Population.Capacity.Allocated
Cook	Cook.Capacity.Allocated

3. Beim dritten Diagramm ist die `Y Axis` mit `Gruppen` zu beschriften und der Titel soll `Overflow` sein. Als `Expression` wird `Sink2.InputBufer.NumberEntered` verwendet.

15 Zeitmessung

1. Fügen Sie dem `ModelEntity` unter `Definitions` die drei Real States `TimeTable`, `TimeOrder` und `TimeDish` hinzu. Geben Sie bei allen als `Unit Type` `Time` an.
2. Fügen Sie beim `Model` in den `Definitions` unter `Elements` vier `Tally Statistic` Elemente ein. Nennen Sie diese `TallyAtTable`, `TallyWaitForOrder`, `TallyWaitForDish` und `TallyTotalTime`. Setzen Sie bei allen die `Unit Type` auf `Time`.
3. Für die Zählung von Umsatz und Gästen fügen Sie unter `States` ein `Integer` Objekt mit dem Namen `RecordGuests` ein und eines mit dem Namen `RecordOverflowGuests`. Für `RecordSales` verwenden Sie bitte ein `Real Element`.
4. Öffnen Sie das Modell `Table` und fügen Sie bei `Ordering > State Assignments > On Entering` einen neuen Datensatz hinzu. Geben Sie als `State Variable` `ModelEntity.TimeTable` an und als `New Value` `Run.TimeNow`.
5. Wählen Sie `[Create New]` bei `Add-On Process Triggers > Before Processing` des Elements `Ordering` aus und ändern Sie den Namen des erstellten Prozessen unter `Processes` auf `SetTimeOrder`. Fügen Sie dem Prozess den `Step` `Assign` hinzu und setzen Sie analog zu vorher die Variable `ModelEntity.TimeOrder` auf `Run.TimeNow`.

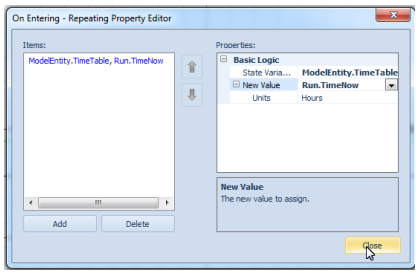


Abbildung 73 Speichere Modellzeit

6. Speichern Sie bei `Dish > On Entering` die aktuelle Zeit in die Variable `ModelEntity.TimeDish`.
7. Öffnen Sie das Element `Input@Sink1` und fügen Sie unter `Tally Statistics > On Entering` einen neuen Datensatz hinzu. Wählen Sie als `Tally Statistic Name` `TallyTotalTime` und als `Value` `ModelEntity.TimeInSystem`.
   Fügen Sie folgende Datensätze hinzu:

Tally Statistic Name	Value
TallyAtTable	ModelEntity.TimeTable – ModelEntity.TimeCreated
TallyWaitForOrder	ModelEntity.TimeOrder – ModelEntity.TimeTable
TallyWaitForDish	ModelEntity.TimeDish – ModelEntity.TimeOrder

8. Fügen Sie unter `Sink1 > State Assignments > On Entering` folgende Datensätze hinzu:

State Variable Name	New Value
RecordGuests	RecordGuests + ModelEntity.GroupSize
RecordSales	RecordSales + Random.Triangular(20,30,60)

9. Fügen Sie unter `Sink2 > State Assignments > On Entering` folgende Datensätze hinzu:

State Variable Name	New Value
RecordOverflowGuests	RecordOverflowGuests + ModelEntity.GroupSize

16 Experimente

1. Klicken Sie im Ribbon `Project Home` auf `New Experiment`. Im neuen Experiment sind die Parameter bereits als Controls eingerichtet.
2. Mit `Add Response` können Sie die Variablen definieren, welche nach dem Durchlauf angezeigt werden sollen.
3. Fügen Sie folgende Responses hinzu:

Neme	Unit Type	Expression
TimeAtTable	Time (Minutes)	TallyAtTable.Average
TimeWaitForOrder	Time (Minutes)	TallyWaitForOrder.Average
TimeWaitForDish	Time (Minutes)	TallyWaitForDish.Average
TotalTime	Time (Minutes)	TallyTotalTime.Average
TotalGuests	Unspecified	RecordGuests
GuestOverflow	Unspecified	RecordOverflowGuests
Sales	Unspecified	RecordSales

4. Nun können Sie verschiedene Szenarien erstellen und danach mit `Run` berechnen lassen.
5. Unter `Response Result`, `Pivot Grid` und `Reports` finden Sie genauere Auswertungen und Grafiken.

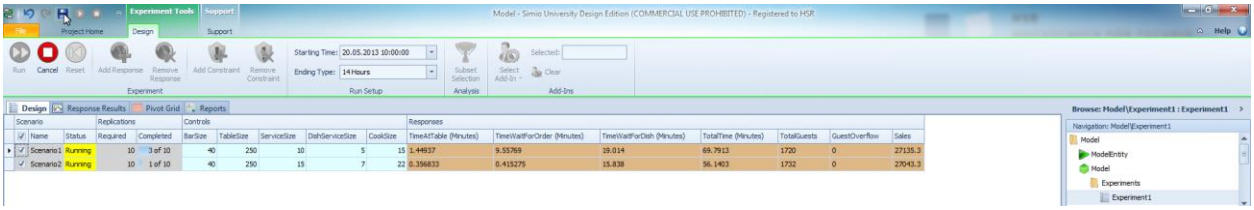


Abbildung 74 Berechnung der Szenarien

17 Kosten

1. Setzen Sie bei der Ressource `Cook` und `Service` `Idle Cost Rate` und `Usage Cost Rate` auf `25`. Bei der Ressource `DishService` auf `20`.
2. Kontrollieren Sie mittels `Watch` ob die Kosten richtig berechnet werden.

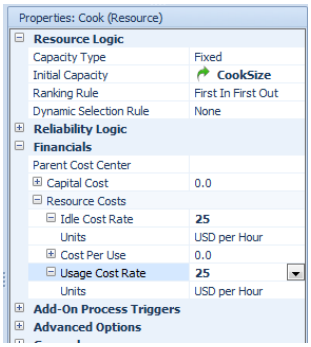


Abbildung 75 Kosten

18 Animation

1. Klicken Sie auf `Place Symbol` unter `Drawing` und wählen Sie dann `Import Symbol`. Geben Sie im Open Dialog das Bild des Gebäudeplans an.
2. Damit die Grössenverhältnisse stimmen ändern Sie die `Length` des Bildes beim Import auf 57.6.
3. Platzieren Sie den Plan auf der Arbeitsfläche und ordnen Sie die Symbole wie gewünscht auf dem Plan an. Die Gehdistanz wird durch die Entfernung auf dem Plan definiert.
4. Damit der Plan nicht aus Versehen verschoben werden kann, wählen Sie nach einem Rechtsklick auf das Bild die Option `Lock Edits`.

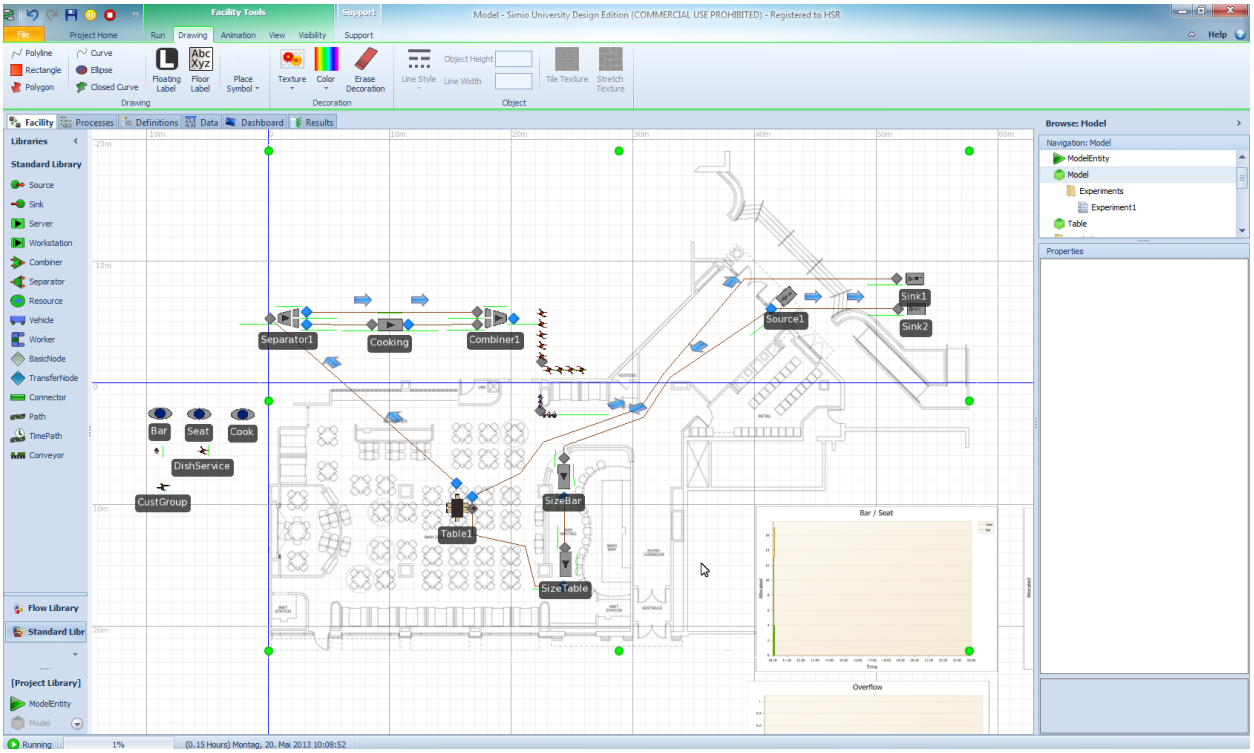


Abbildung 76 Anordnung der Elemente auf dem Gebäudeplan

19 OptQuest

Mit dem Add-In OptQuest kann die optimale Einstellung der Parameter gesucht werden.

1. Öffnen Sie das Experiment und fügen Sie zwei weitere Responses hinzu.

Name	Unit Type	Expression
Cost	Unspecified	Cost
SalesCost	Currency	RecordSales - Cost

2. Da `SalesCost` möglichst gross sein sollte, setzen Sie in den Eigenschaften `Objective` auf `Maximize`.
3. Geben Sie in den Eigenschaften des Experiments `SalesCost` als `Primary Response` an.
4. Wählen Sie über `Select Add-In` den Punkt `ObtQuest for Simio` aus.
5. Setzen Sie bei `BarSize` und `TableSize` das Feld `Include In Optimisation` auf `No`.
6. Setzen Sie bei `ServiceSize` und `CookSize` den `Minimum Value` und den `Maximum Value` auf 1 und 20. Bei `DishServiceSize` setzen Sie `Minimum` auf 1 und `Maximum` auf 10.
7. Über `Run` kann die Optimierung gestartet werden wodurch automatische Szenarien generiert werden.

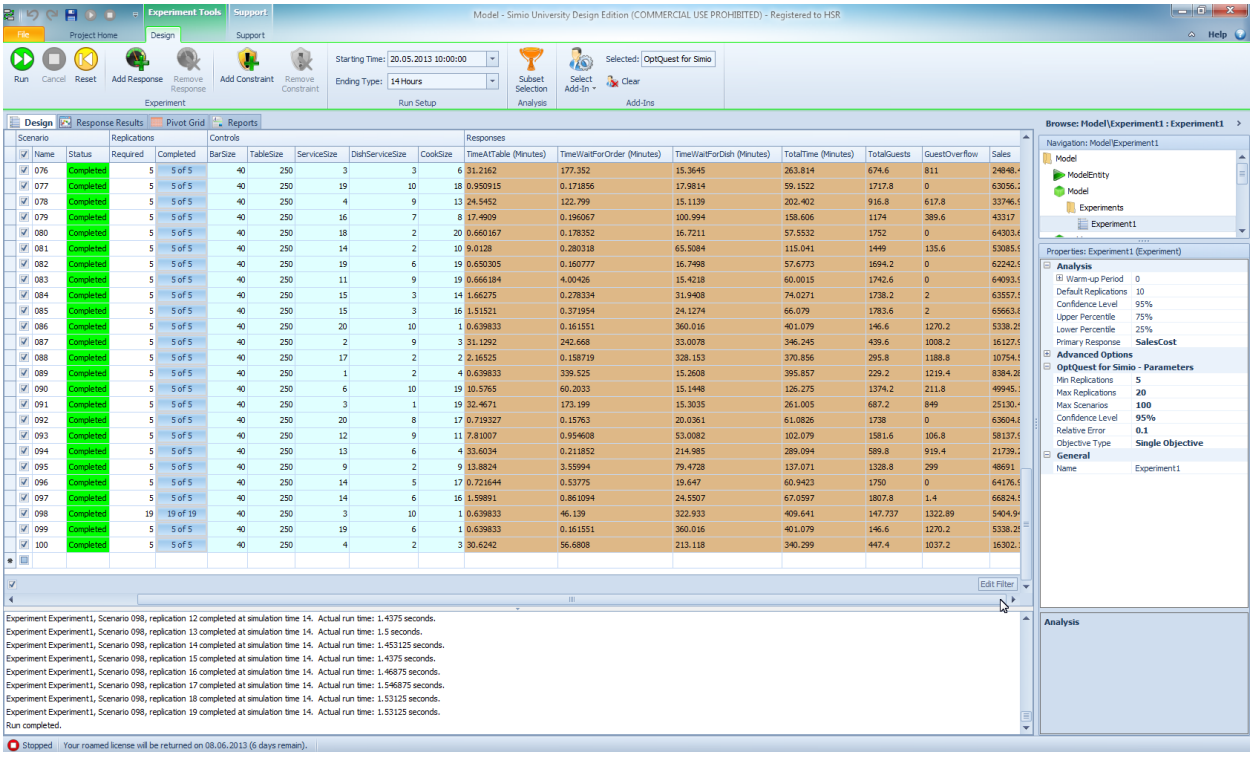


Abbildung 77 OptQuest

D. PROJEKTMANAGEMENT

1 Management Summary

1.1 Ausgangslage

Die Simulation als Hilfe zur Entscheidungsfindung spielt in Geschäfts- und Wirtschaftsbereichen eine immer grössere Rolle, da die zu optimierenden Prozesse oder Phänomene immer komplexer werden. Durch die enthaltene Dynamik kommt man bei diesen Problemen mit den analytischen Verfahren schnell an die Grenzen. An vielen Orten ist ein Ausprobieren oder Erstellen eines Prototyps nicht möglich, zu kostspielig oder zu gefährlich. Hier ist die Simulation das richtige Werkzeug, um die Prozesse zu simulieren und mit verschiedenen Parametern zu optimieren. Es ist wichtig, zu verstehen was genau simuliert wird, um die Resultate zu kontrollieren und auszuwerten. Die Simulation wird immer vielseitiger und in mehreren Bereichen eingesetzt.

Um erstellte Modelle zu simulieren, wird meist eine komplexe Simulationssoftware eingesetzt oder eine massgeschneiderte Applikation entwickelt. Im momentanen Markt gehören AnyLogic von XJ Technologies und SIMIO von SIMIO LLC zu den etabliertesten im Markt.

Diese Bachelor Arbeit befasst sich mit den beiden Produkten AnyLogic und SIMIO und evaluiert jene. In einem Bericht werden die Eckdaten, Eigenschaften sowie Pro und Contras aufgezeigt und die Einsatzgebiete der Software erläutert.

1.2 Vorgehen

Nach dem für beide Tools eine identische Testumgebung vorbereitet ist, wird anhand von Büchern und Tutorials das Wissen über die Applikationen erarbeitet. Gleichzeitig wird im Modul System „Modeling and Simulation“ die Grundlage der Simulation erlernt. In diesem Modul wird momentan noch mit dem Tool Arena gearbeitet. Die Erfahrungen welche mit Arena gemacht wurden fliessen ebenfalls in die Auswertung mit ein.

Um die beiden Applikationen zu vergleichen, wird unter anderem ein universelles Modell, welches es zu erarbeiten gilt, mit beiden Tools simuliert. So lassen sich die Unterschiede der Tools gut herausarbeiten und studieren. Zudem wird für beide Applikationen ein Vorgehensmodell für realisierte Simulation erstellt.

Als weitere Herausforderung wird untersucht, welche Schnittstellen vorhanden sind, um externe Daten zu importieren oder zu bearbeiten.

1.3 Ergebnisse

Der Einstieg in AnyLogic und Arena ist einfacher und die Bedienung intuitiver als bei SIMIO. Hingegen wird der Modellierungsprozess in SIMIO deutlich einfacher wenn die grundlegenden Konzepte verstanden sind. AnyLogic zeichnet sich durch die Möglichkeit aus, verschiedene Modellierungstypen, wie zum Beispiel die agentenbasierte oder diskrete Modellierung zu unterstützen. Ferner erlaubt AnyLogic Javabefehlte zu hinterlegen. Damit können Prozesse sehr agil in AnyLogic realisiert werden.

In SIMIO fehlt bis jetzt der Ansatz von Agentenmodellen. Dafür ist die Library für die diskreten Eventmodelle sehr mächtig und haben viele Konfigurationsmöglichkeiten. Die Elemente können zusätzlich mit Add-on Prozessen, welche wiederum Eventbasiert sind, ergänzt werden. So können komplexe Elemente erstellt werden.

In Arena fehlen die modernen Modellierungsansätze, wie Objektorientierung und Datenkapselung. In SIMIO und in AnyLogic lassen sich schnell 2D/3D Animationen erstellen. Hier ist der Aufwand in Arena deutlich höher. In AnyLogic ist es möglich die Projekte in Java Applet zu exportieren. SIMIO und Arena sind im Bereich von Auswertungen und Generieren von Szenarien gegenüber AnyLogic klar im Vorteil.

2 Projektplan

Das Projekt kann in fünf Phasen aufgeteilt werden.

Nach der Einarbeitung in die beiden Applikationen, wird ein zu erarbeitendes Modell in beiden Tools simuliert. Gleichzeitig wird ein Dokument erstellt, welches die beiden Softwares erklärt und gegenüberstellt. Danach wird untersucht, wie mit den Applikationen auf externe Daten zugegriffen werden kann.

2.1 Projektorganisation

Das Projekt wird als Einzelarbeit von Urs Baumann durchgeführt. Betreuer ist Prof. Dr.-Ing. Andreas Rinkel. Experte ist Dr. Schimpel von IBM Zürich Research Lab.

2.2 Zeitaufwand

Der Projektstart ist der 18.02.2013 und die Abgabe ist am 14.06.2013. Es soll ein Aufwand von mindestens 360 Stunden betrieben werden. Während den ersten 15 Wochen ist gleichzeitig der Unterricht an der HSR zu besuchen. Die Wochen 16 und 17 sind danach komplett für die Bachelor Arbeit reserviert.

2.3 Besprechungen

Es findet wöchentlich eine Sitzung mit dem Betreuer statt. Der Tag und die Uhrzeit sind flexibel. Meistens wird die Sitzung am Dienstag um 08:00 Uhr durchgeführt. Im Anhang befinden sich die Sitzungsprotokolle, welche nach jeder Sitzung erstellt und ausgetauscht wurden.

2.4 Infrastruktur

Die HSR stellt einen Arbeitsplatz mit einem Rechner zur Verfügung. Die Arbeit wird jedoch komplett auf dem Notebook des Studenten erarbeitet und der Rechner dient nur als Notlösung, falls das Notebook technische Probleme haben sollte.

In der ersten Woche werden für beide Applikationen eine virtuelle Umgebung (VMware) eingerichtet, dies dient dazu, dass die zwei Tools die gleichen Voraussetzungen haben und sich gegenseitig nicht beeinflussen können. Zudem kann so bei einem Ausfall schnell auf einen Computer gewechselt werden (dies aber nur, wenn sich ein Backup der VMware auf einer externen Platte verwendet. Schmiert der Laptop (bzw. die HDD) ab, ist auch die VM weg).

2.5 Risikomanagement

Risiko	Auswirkung	Massnahme	Eintrittswahr-scheinlichkeit
Fehleinschätzung des Aufwandes	Zeitplan wird nicht eingehalten	Nicht alle geplanten Funktionen implementieren, Mehraufwand	10%
VMware stürzt ab / startet nicht	Projektumgebung muss neu eingerichtet werden	VMware Kopien erstellen	5%
Simulationssoftwares sind komplizierter als gedacht.	Das Verstehen braucht mehr Zeit	Meilensteine definieren	5%

2.6 Arbeitspakete

Arbeitspakete wurden nach jeder Sitzung für die kommende Woche definiert und geplant.

Start der Arbeitswoche ist jeweils der Dienstag.

2.7 Meilensteine

Zu Beginn des Projekts wurden folgende Meilensteine definiert:

Meilenstein	Beschreibung	Zeitpunkt
MS1	AnyLogic bekannt	Woche 4
MS2	SIMIO bekannt	Woche 7
MS3	Modell für Simulation erstellt	Woche 8
MS4	Modell mit beiden Tools simuliert	Woche 12
MS5	Möglichkeiten für externe Daten sind bekannt Optional: Kleines Projekt	Woche 16
MS6	Abgabe	Woche 17 (14.06.2013)

Alle Meilensteine konnten termingerecht eingehalten werden. Das kleine optionale Projekt bei Meilenstein 5 musste wegen Zeitmangel reduziert werden. So wurde nicht in beiden Tools das gleiche Projekt simuliert sondern nur einzelne Übungen und Tests durchgeführt.

2.8 Zeitplan

Es wurde 385.25 Stunden budgetiert und letztendlich 411.75 Stunden verbucht. Vor allem die zwei Anleitungen waren viel zeitaufwändiger als im Vorfeld geschätzt wurde. Deshalb gegen den Schluss auch die Reservezeit verbraucht. Da die letzten zwei Wochen unterrichtsfrei waren, stellte dies jedoch kein grosses Problem dar.

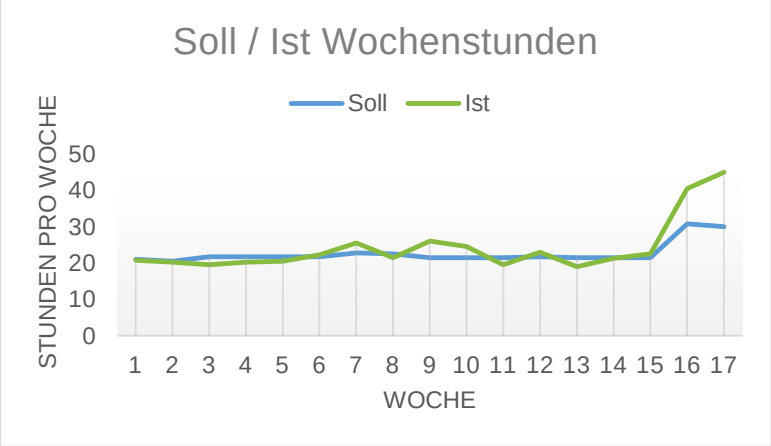


Abbildung 80 Wochenstunden

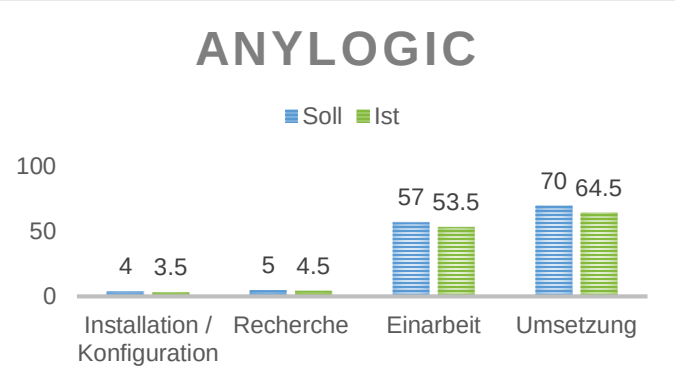


Abbildung 78 Stundenauswertung AnyLogic

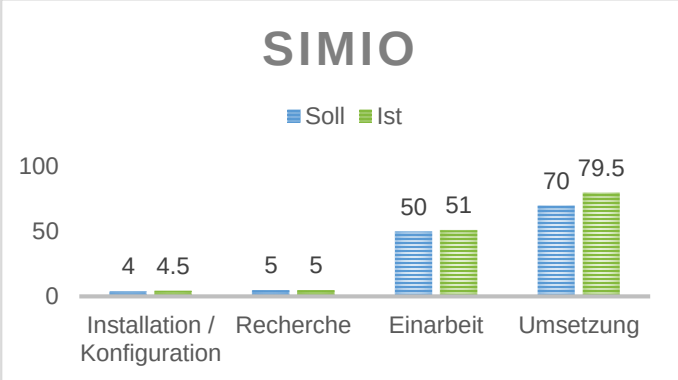


Abbildung 79 Stundenauswertung Simio

Wie in den Diagrammen Abbildung 78 und Abbildung 79 ersichtlich wird, wurde der Aufwand bei AnyLogic eher überschätzt und bei Simio musste tendenziell mehr Aufwand betrieben werden als geplant. Dennoch waren die Schätzungen relativ genau und der Terminkalender konnte eingehalten werden.

2.9 Test

Die Simulationen wurden in sehr kleinen Iterationen durchgeführt und es wurde laufend getestet, ob das Ergebnis auch den Erwartungen entspricht.

3 Selbstreflexion

Am Anfang des Projektes hatte ich grossen Respekt vor ziemlich komplizierten Simulationswerkzeugen. Durch die sorgfältige Durchführung der Tutorials konnte ich aber sehr viel über den Simulator lernen und dieses Wissen während der Arbeit abrufen. Ab dem Erhalt der Fachliteratur konnte ich mich noch besser in das Thema einarbeiten und vor allem "AnyLogic 6 in Three Days" war für mich ein sehr hilfreiches Buch.

Ich war überrascht wie aufwändig das Schreiben einer Anleitung ist und nur dank zusätzlichem Einsatz konnte ich die Arbeit zum geplanten Zeitpunkt abgeben. Das Projekt wurde als Einzelarbeit bearbeitet. Rückblickend denke ich, dass bei den vielen komplexen Aufgabenstellungen ein Teamkollege für das Austauschen der Gedanken und Ideen von Vorteil gewesen wäre. Trotzdem bereue ich die Entscheidung nicht, die Arbeit alleine in Angriff genommen zu haben.

Bei dieser Bachelorarbeit durfte ich viel Neues lernen und bisher unbekannte Erfahrungen sammeln, welche ich in zukünftigen Projekten sicherlich einfliessen lassen kann. Je mehr ich mit dem Thema Simulation vertraut wurde und die verschiedenen Einsatzmöglichkeiten sah, desto motivierter wurde ich. Es fiel mir ab und an schwer mich nur auf das Wesentliche zu konzentrieren und nicht zu tief auf nicht geforderte Fragestellungen einzugehen. Die Verlockung mir zu hohe und zu anspruchsvolle Ziele zu setzen und mich damit zu überfordern, war sehr gross.

Letzen Endes möchte ich mich bei Prof. Dr.-Ing. A. Rinkel herzlich für die gute und sehr kompetente Betreuung bedanken. Meiner Meinung nach herrschte ein sehr gutes Arbeitsklima und die Sitzungen hatten einen grossen inspirativen sowie auch motivierenden Einfluss auf mich.

4 Sitzungsprotokolle

4.1 Protokoll Kick Off Meeting

Ort: 6.11
Datum: 18. Feb. 13
Zeit: 10:00
Anwesend: Rinkel Andreas, Baumann Urs

Traktanden	Beschlüsse / Massnahme
Thema	Als Thema wird die Evaluierung zweier Simulationsprogramme SIMION und AnyLogic definiert. Nach der Einarbeitung in die Tools soll ein Übersicht erstellt werden. Danach werden die verschiedenen möglichen Ansätze untersucht. Mit beiden Tools wird die gleiche Beispielsimulation erstellt. Wie sieht es mit der Anbindung von Fremdsystemen aus? Welche Schnittstellen sind vorhanden? Als Ergebnis wird ein kurzer Bericht mit Stärken, Schwächen, etc erwartet
Meetings	Das wöchentliche Meeting findet wenn möglich am Dienstag um 08:00 Uhr statt.
Termine	Urs Baumann ist selbst für die Einhaltung aller Termine verantwortlich.
Management	Grundplan soll erstellt werden Nach der Grobplanung wird wöchentlich geplant

4.2 1. Sitzung

Ort: 6.11
Datum: 26. Feb. 13
Zeit: 08:00
Anwesend: Rinkel Andreas, Baumann Urs

Traktanden	Beschlüsse / Massnahme
Untersuchungen	Um zu definieren, was genau verglichen wird, muss ein Grundverständnis für die beiden Tools erarbeitet werden. Mögliche Kriterien: Einsatzmöglichkeit, Modelierungsarten, Intuition, Rechenzeit, Mächtigkeit Als Vergleichshilfe wird eine Simulation in beiden Tools simuliert. Sobald ein Grundverständnis aufgebaut ist, kann eine Simulation definiert werden.
Projekt Management	Auf nächste Woche wird der Grobplan erstellt. Verwendete Quellen, Tutorials werden aufgelistet
Vergangene Woche	Installation VmWare für SIMIO und AnyLogic Erste Gehversuche mit AnyLogic Informieren über AnyLogic und SIMIO
Nächste Woche	Auf nächste Woche wird der Grobplan erstellt. Kennenlernen von AnyLogic mit Hilfe Tutorials, Bücher

4.3 2. Sitzung

Ort: 6.11
Datum: 05. Mrz. 13
Zeit: 08:00
Anwesend: Rinkel Andreas, Baumann Urs

Traktanden	Beschlüsse / Massnahme
Vergangene Woche	Erstellung Projektmanagment, Excel für Zeiterfassung, Definition von Meilensteine und Erstellung Grobplan Kennenlernen von AnyLogic, Simulation von 2 Modellen anhand eines Tutorials
Dokumentation	Zeigen der erstellten Dokumente
AnyLogic	Zeigen und erklären der erstellten Simulationen Diskussion über AnyLogic und dessen Bedienbarkeit
Organisatorisches	Termin für die mündliche BA Prüfung kann beim Treffen mit Dr. Schimpel abgemacht werden.
Nächste Woche	Kennenlernen von AnyLogic mit Hilfe Tutorials, Bücher

4.4 3. Sitzung

Ort: 6.11
Datum: 05. Mrz. 13
Zeit: 08:00
Anwesend: Rinkel Andreas, Baumann Urs

Traktanden	Beschlüsse / Massnahme
Vergangene Woche	Kennenlernen von AnyLogic mit Tutorial. Erstellen von Simulation mit Agent Based Modell und Flussdiagramm
AnyLogic	Zeigen und erklären von Simulationen
Simulation	Wenn im Modul SMS für die BA ein Modell entwickelt wird, muss dies erwähnt werden.
Nächste Woche	AnyLogic 6 in Three Days

4.5 4. Sitzung

Ort: 6.11
Datum: 19. Mrz. 13
Zeit: 08:00
Anwesend: Rinkel Andreas, Baumann Urs

Traktanden	Beschlüsse / Massnahme
Vergangene Woche	Komplette Durcharbeit von AnyLogic 6 in Three Days. Diskussion über das Buch und AnyLogic Überlegen eines Modellbeispiels. Favorit ist die Simulation eines Restaurants.
AnyLogic	Zeigen und erklären der Simulationen aus dem Buch.
Modell	Simulation eines Restaurants hat viele Aspekte und verschiedene Modelltypen.
Nächste Woche	Beginn mit der Einarbeit mit SIMIO

4.6 5. Sitzung

Ort: 6.11
Datum: 26. Mrz. 13
Zeit: 08:00
Anwesend: Rinkel Andreas, Baumann Urs

Traktanden	Beschlüsse / Massnahme
Vergangene Woche	Einarbeitung in Simio
Simio	Zeigen und erklären der Simulationen aus dem Buch. Simio nimmt Form an.
Nächste Woche	Einarbeit mit SIMIO

4.7 6. Sitzung

s

Ort: 6.11
Datum: 2. Apr. 13
Zeit: 08:00
Anwesend: Rinkel Andreas, Baumann Urs

Traktanden	Beschlüsse / Massnahme
Vergangene Woche	Einarbeitung in Simio
Simio	Zeigen und erklären der Simulationen aus dem Buch. Simio ist sehr komplex und kompliziert.
Nächste Woche	Beginn mit der Einarbeitung mit SIMIO

4.8 7. Sitzung

Ort: 6.11
Datum: 16. Apr. 13
Zeit: 08:00
Anwesend: Rinkel Andreas, Baumann Urs

Traktanden	Beschlüsse / Massnahme
Vergangene Woche	Einarbeitung in Simio. Buch braucht viel Zeit
Simio	Zeigen und erklären der Simulationen aus dem Buch.
Modell	Prototyp des Modells wird erstellt.
Nächste Woche	Beginn mit der Einarbeitung mit SIMIO

4.9 8. Sitzung

Ort: 6.11
Datum: 23. Apr. 13
Zeit: 08:00
Anwesend: Rinkel Andreas, Baumann Urs

Traktanden	Beschlüsse / Massnahme
Vergangene Woche	Erstellung Modells
Modell	Modell wird besprochen
Nächste Woche	Beginn Umsetzung des Prototypen

4.10 9. Sitzung

Ort: 6.11
Datum: 30. Apr. 13
Zeit: 08:00
Anwesend: Rinkel Andreas, Baumann Urs

Traktanden	Beschlüsse / Massnahme
Vergangene Woche	Prototyp in Simio noch wage
Simio	Zeigen des Prototyps. Sehr kompiliert
Nächste Woche	Umsetzung des Prototypen

4.11 10. Sitzung

Ort: 6.11
Datum: 7. Mai. 13
Zeit: 08:00
Anwesend: Rinkel Andreas, Baumann Urs

Traktanden	Beschlüsse / Massnahme
Vergangene Woche	Prototyp in Simio Externe Daten mit Simio
Simio	Zeigen des Prototyps in Simio Interaktion mit externen Daten
Nächste Woche	Beginn Umsetzung des Prototypen AnyLogic

4.12 11. Sitzung

Ort: 6.11
Datum: 21. Mai. 13
Zeit: 08:15
Anwesend: Rinkel Andreas, Baumann Urs

Traktanden	Beschlüsse / Massnahme
Vergangene Woche	Prototyp in AnyLogic
AnyLogic	Zeigen des Prototyps
Nächste Woche	Umsetzung des Prototypen

4.13 12. Sitzung

Ort: 6.11
Datum: 28. Mai. 13
Zeit: 08:15
Anwesend: Rinkel Andreas, Baumann Urs

Traktanden	Beschlüsse / Massnahme
Vergangene Woche	Prototyp in AnyLogic Externe Daten in AnyLogic
AnyLogic	Zeigen des Prototyps. Interaktion mit externen Daten
Nächste Woche	Umsetzung des Prototypen

4.14 13. Sitzung

Ort: 6.11
Datum: 4. Jun. 13
Zeit: 08:15
Anwesend: Rinkel Andreas, Baumann Urs

Traktanden	Beschlüsse / Massnahme
Vergangene Woche	Umsetzung in AnyLogic und Simio Viele PrintScreens für Anleitungen erstellt
Simulation	Zeigen der Simulationen. Resultat stimmen überein
Nächste Woche	Umsetzung des Prototypen

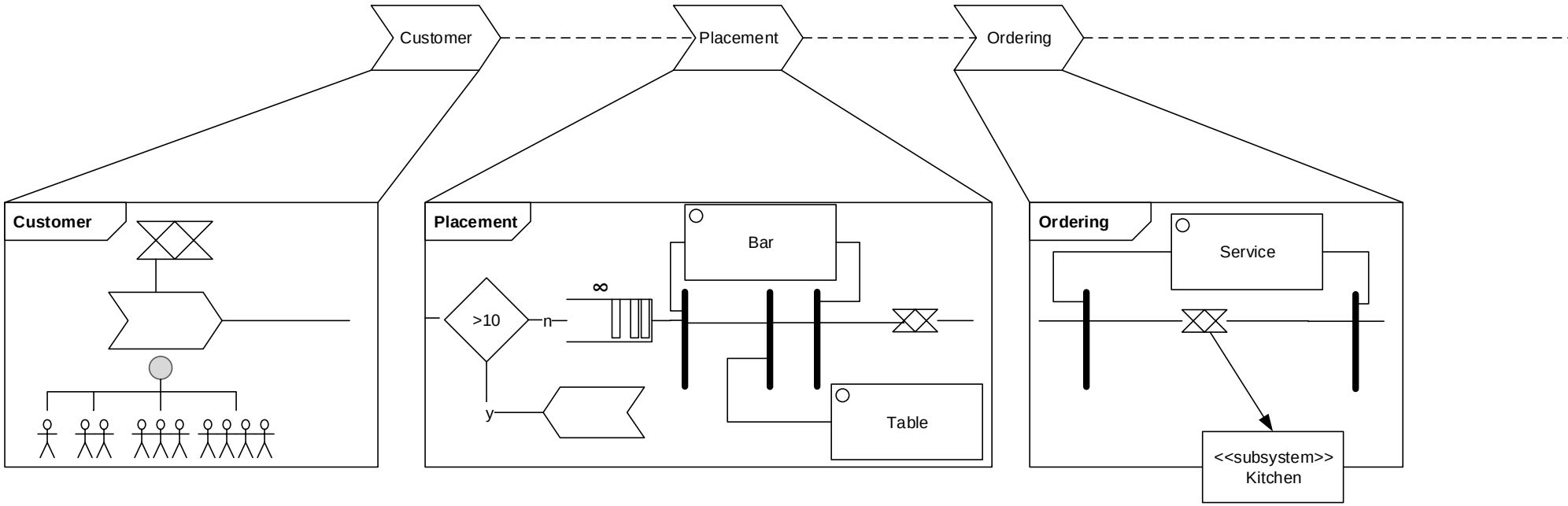
E. ANHANG

1 Abbildungsverzeichnis

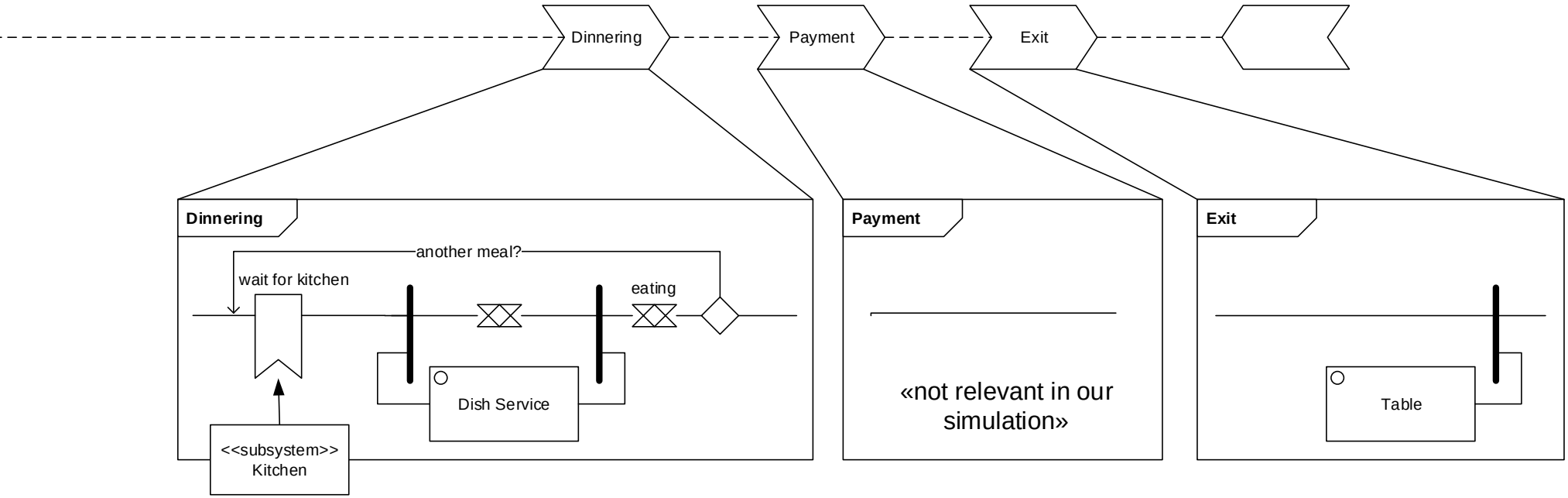
Abbildung 1 Modellierungsarten AnyLogic 11
Abbildung 2 Diagramm in Simio 12
Abbildung 3 AnyLogic Projekt 15
Abbildung 4 Eingabe Evaluation Key 15
Abbildung 5 Anforderung Key 15
Abbildung 6 AnyLogic Hauptfenster 15
Abbildung 7 Erste Simulation 16
Abbildung 8 Submodel 17
Abbildung 9 Modell mit Submodel 17
Abbildung 10 Laufende Simulation 18
Abbildung 11 Netzwerkplan 19
Abbildung 12 Konfiguration Netwok Storage 20
Abbildung 13 Eintritt Netzwerk 21
Abbildung 14 Konfiguration Network Enter 21
Abbildung 15 Storage Put 21
Abbildung 16 Austritt Netzwerk 22
Abbildung 17 Restriced Area End 22
Abbildung 18 Fehler network Auflösung 22
Abbildung 19 Simulation Netzwerk 22
Abbildung 20 Netzwerk Ressourcen 23
Abbildung 21 Network Resource Pool 23
Abbildung 22 Sende Ressourcen zum Tisch 23
Abbildung 23 Service Station Kochen 24
Abbildung 24 Simulation Tellerbringer 24
Abbildung 25 Entering Restricted Area 25
Abbildung 26 Restrict Area Simulation 25
Abbildung 27 Konfiguration Scheduler 26
Abbildung 28 Konfiguration Source 27
Abbildung 29 Ordering 27
Abbildung 30 Simulationszeit Einstellungen 27
Abbildung 31 Eigene Klasse 28
Abbildung 32 Generierung Cust Entitäten 28
Abbildung 33 Statistikelemente 28
Abbildung 34 Zeitmessung 28
Abbildung 35 Anordnung der Diagramme 29
Abbildung 36 Time Plot Daten 29
Abbildung 37 Parameter 30
Abbildung 38 Parametrisierung vor Simulationsstart 31
Abbildung 39 Slider Konfiguration 31
Abbildung 40 Salary Event 31
Abbildung 41 Konfiguration des Events 32
Abbildung 42 Fertige Simulation in AnyLogic 32
Abbildung 43 Show in 3D scene 32
Abbildung 44 SIMIO Projekt 33
Abbildung 45 Evaluationslizenzeinschränkungen von Simio 33
Abbildung 46 Übersicht Arbeitsfenster Simio 34
Abbildung 47 Erste Simio Simulation 34
Abbildung 48 Integer Properties 35

Abbildung 49 Worker 35
Abbildung 50 Ressource belegen 35
Abbildung 51 Simulation mit Ressourcen 36
Abbildung 52 Simulation mit bewegten Servicekräften und aktivem `Parking Queue` 36
Abbildung 53 Sub Modell Table 37
Abbildung 54 3D Grafik Import Dialog 37
Abbildung 55 Externe Ansicht 37
Abbildung 56 Fehlermeldung Initiate transfer 38
Abbildung 57 ModelEntity 38
Abbildung 58 Zuweisung Gruppengrösse 38
Abbildung 59 Parametrisierte Freigabe 39
Abbildung 60 Table1 39
Abbildung 61 Simulation mit Service unterwegs zum Tisch 39
Abbildung 62 Bestellung Senden und Essen Empfangen 40
Abbildung 63 Add-On Process 40
Abbildung 64 Modell mit Küche 41
Abbildung 65 Output@Combiner1 41
Abbildung 66 Symbole 42
Abbildung 67 3D Simulation mit Symbolen 42
Abbildung 68 Overflow 43
Abbildung 69 Arrivals Tabelle 43
Abbildung 70 Run Einstellungen 44
Abbildung 71 Watch Fenster 44
Abbildung 72 Diagramme in Simio 44
Abbildung 73 Speichere Modellzeit 45
Abbildung 74 Berechnung der Szenarien 46
Abbildung 75 Kosten 47
Abbildung 76 Anordnung der Elemente auf dem Gebäudeplan 47
Abbildung 77 OptQuest 48
Abbildung 80 Wochenstunden 52
Abbildung 78 Stundenauswertung AnyLogic 52
Abbildung 79 Stundenauswertung Simio 52

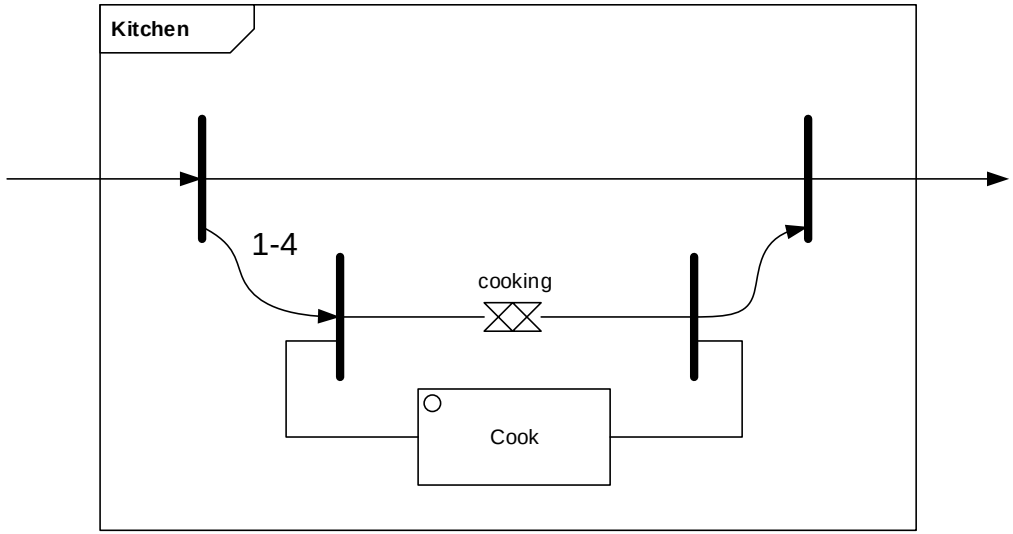
Hardrock Flow Diagramm



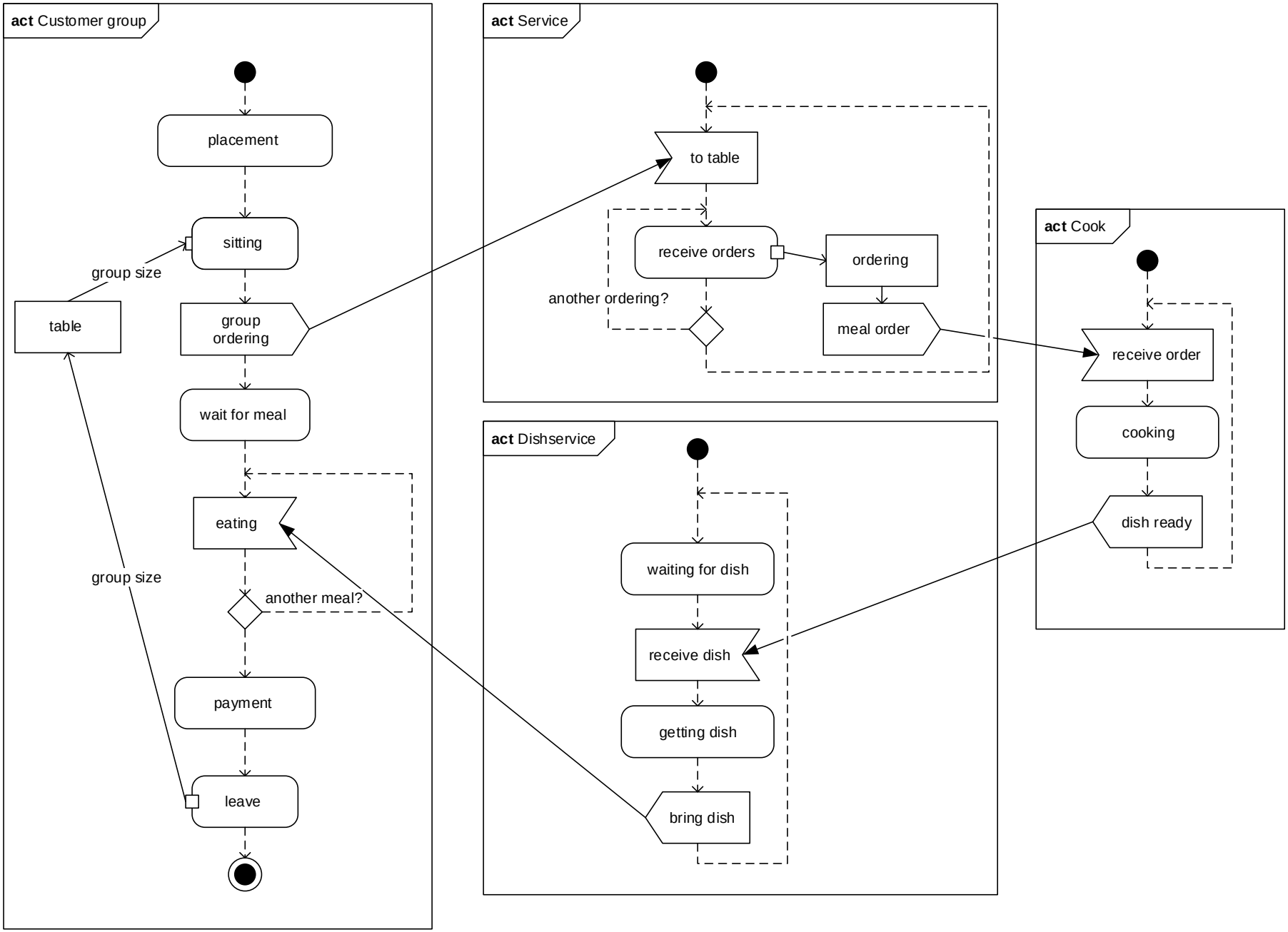
Hardrock Flow Diagramm



Hardrock Flow Diagramm



Hardrock Activity Diagramm



Zeitplan

Projekt: **Evaluierung der Simulationssoftware AnyLogic, Arena und SIMIO**
Verantwortliche Pers. Urs Baumann

			18. Feb - 24. Feb		25. Feb - 3. Mrz		4. Mrz - 10. Mrz		11. Mrz - 17. Mrz		18. Mrz - 24. Mrz		25. Mrz - 31. Mrz		1. Apr - 7. Apr		8. Apr - 14. Apr	
Meilensteine									MS1						MS2		MS3	
	Total		Woche 1		Woche 2		Woche 3		Woche 4		Woche 5		Woche 6		Woche 7		Woche 8	
Arbeitspakete:	**SOLL**	**IST**	SOLL	IST	SOLL	IST	SOLL	IST	SOLL	IST	SOLL	IST	SOLL	IST	SOLL	IST	SOLL	IST
Projektmanagement	**100.25**	**127.75**	4	3.75	7.5	7.25	1.75	1.5	1.75	2.25	1.75	1.5	1.75	1.25	7.75	9.5	2.5	3.5
Sitzungen	10.25	10.75	1	1	0.5	0.5	0.75	0.75	0.75	1	0.75	0.5	0.75	0.75	0.75	1	0.5	0.5
Projektplan / Zeitplan	2	2.25			2	2.25												
Dokumentation	69	100			5	4.5	1	0.75	1	1.25	1	1	1	0.5	1	1.5	2	3
sonstiger Projektaufwand	19	14.75	3	2.75											6	7		
AnyLogic	**136**	**126**	13	12.5	13	13	20	18	20	18	0	0	0	0	0	0	0	0
Installation / Konfiguration	4	3.5	*4*	3.5														
Recherche	5	4.5	5	4.5														
Einarbeit	57	53.5	4	4.5	13	13	20	18	20	18								
Umsetzung	70	64.5																
SIMIO	**129**	**140**	4	4.5	0	0	0	0	0	0	20	19	20	21	15	16	0	0
Installation / Konfiguration	4	4.5	4	4.5														
Recherche	5	5									5	5						
Einarbeit	50	51									15	14	20	21	15	16		
Umsetzung	70	79.5																
Simulation	**20**	**18**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	18
Erarbeitung des Modelles	20	18															*20*	18
Wochentotal			21	20.75	20.5	20.25	21.75	19.5	21.75	20.25	21.75	20.5	21.75	22.25	22.75	25.5	22.5	21.5
Gesamt	385.25	411.75																

Soll Stunden = 360

Meilensteine

MS1: AnyLogic bekannt	Woche 4
MS2: SIMIO bekannt	Woche 7
MS3: Modell für Simulation erstellt	Woche 8
MS4: Modell mit beiden Tools simuliert	Woche 12
MS5: Möglichkeiten für externe Daten sind bekannt	Woche 16
MS6: Abgabe	Woche 17 (16.06.2013)

15. Apr - 21. Apr		22. Apr - 28. Apr		29. Apr - 5. Mai		6. Mai - 12. Mai		13. Mai - 19. Mai		20. Mai - 26. Mai		27. Mai - 2. Jun		3. Jun - 9. Jun		10. Jun - 16. Jun	
						MS4								MS5		MS6 (14.06.2013)	
Woche 9		**Woche 10**		**Woche 11**		**Woche 12**		**Woche 13**		**Woche 14**		**Woche 15**		**Woche 16**		**Woche 17**	
SOLL	**IST**	SOLL	**IST**	SOLL	**IST**	SOLL	**IST**	SOLL	**IST**	SOLL	**IST**	SOLL	**IST**	SOLL	**IST**	SOLL	**IST**
1.5	2	1.5	1.5	1.5	1	1.75	2	1.5	2	1.5	1.75	1.5	1.5	30.75	40.5	30	45
0.5	0.75	0.5	0.5	0.5	0.5	0.75	0.75	0.5	0.75	0.5	0.5	0.5	0.5	0.75	0.5		
1	1.25	1	1	1	0.5	1	1.25	1	1.25	1	1.25	1	1	30	40	20	40
																10	5
0	0	0	0	20	18.5	20	21	20	17	10	8	0	0	0	0	0	0
				20	18.5	20	21	20	17	10	8						
20	24	20	23	0	0	0	0	0	0	10	11.5	20	21	0	0	0	0
20	24	20	23							10	11.5	20	21				
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21.5	26	21.5	24.5	21.5	19.5	21.75	23	21.5	19	21.5	21.25	21.5	22.5	30.75	40.5	30	45

Projekt Hardrock

Dokumentation

Reto Schelbert, Simon Schreiber, Tobias Blaser, Urs Baumann

Änderungsnachweis

Version	Änderung	Autor	Datum
1.0	Dokumentenentwurf	Tobias Blaser	24.03.2013
1.1	Entwurf Konzept	Tobias Blaser	28.03.2013
1.2	Ergänzungen und Korrekturen	Reto Schelbert	03.04.2013
1.3	Überarbeitung Konzept (Word Model)	Tobias Blaser	08.04.2013
1.4	Modell & Verteilungsfunktionen	Tobias Blaser	03.05.2013
1.5	Simulation	Alle	10.05.2013
1.6	Modellbeschreibungen	Tobias Blaser	11.05.2013
1.7	Plausibilitätscheck, Szenarios	Alle	13.05.2013
1.8	Szenarien, Auswertung	Urs Baumann, Tobias Blaser	17.05.2013
1.9	Auswertung, Überarbeitungen	Alle	24.05.2013
2.0	Überarbeitungen	Alle	27.05.2013

Inhaltsverzeichnis

1. Konzept **5**
1.1. Model . . . 5
1.1.1. Eingangsbereich . . . 5
1.1.2. Bar . . . 6
1.1.3. Tische . . . 6
1.1.4. Küche . . . 6
1.2. Bewegungsabläufe . . . 7
1.2.1. Gäste . . . 7
1.2.2. Personal . . . 8
1.2.3. Realität und Simulation . . . 8
1.3. Warum soll simuliert werden? . . . 8
1.4. Objectives . . . 9
1.4.1. Gewinn . . . 9
1.4.2. Utilization . . . 9
1.4.3. Dauer des Aufenthalts . . . 9
1.5. Spezifikationen . . . 9
1.5.1. Variable Grössen . . . 10
1.5.2. Resultatgrössen . . . 10
1.5.3. Zusatzprojekt (optional) . . . 10

2. Modellierung **11**
2.1. Activities . . . 11
2.1.1. Gastgruppe . . . 11
2.1.2. Service . . . 12
2.1.3. Köche . . . 12
2.1.4. Tellerbringer . . . 12
2.2. Hardrock Guest Flow . . . 13
2.2.1. Customer . . . 13
2.2.2. Placement . . . 13
2.2.3. Ordering . . . 14
2.2.4. Dinnering . . . 14
2.2.5. Payment . . . 15
2.2.6. Exit . . . 15
2.2.7. Kitchen . . . 15
2.2.8. Entitäten . . . 16

3. Simulation mit Arena **17**
3.1. Customer . . . 17
3.1.1. Verteilung der ankommenden Gäste . . . 17

3.2. Placement . . . 18
3.2.1. Überlauf am Eingang . . . 18
3.2.2. Assignement und Release von Warteschlange, Bar und Tisch . . . 18
3.2.3. Laufweg der Gäste von der Bar zum Tisch . . . 18
3.3. Ordering . . . 19
3.3.1. Bestellvorgang . . . 19
3.4. Preparation . . . 19
3.4.1. Verteilung der Zubereitungszeit . . . 19
3.4.2. Gruppenbestellungen . . . 19
3.5. Dinnering . . . 19
3.5.1. Bestellungsauslieferung . . . 19
3.5.2. Essen . . . 20
3.6. Szenarien . . . 20
3.7. Plausibilitäts-Checks . . . 20
3.8. Umsatz . . . 21

4. Auswertung **22**
4.1. Szenarien . . . 22
4.1.1. Szenarienoptimierung . . . 22
4.1.2. erste Selektion . . . 22
4.1.3. zweite Selektion . . . 23
4.1.4. Tellerbringer . . . 24
4.1.5. Köche . . . 24
4.1.6. Dritte Selektion . . . 24
4.1.7. Verifikation . . . 25
4.2. Fragestellungen (If Questions) . . . 27

A. Simulationsdaten **28**

B. Arena Report **31**

C. Präsentation Projektvorstellung **39**

D. Materialnachweise **43**
D.1. Titelblatt . . . 43

1. Konzept

1.1. Model

Als Ausgangskonzept wird das Hardrock in Atlantic City verwendet.

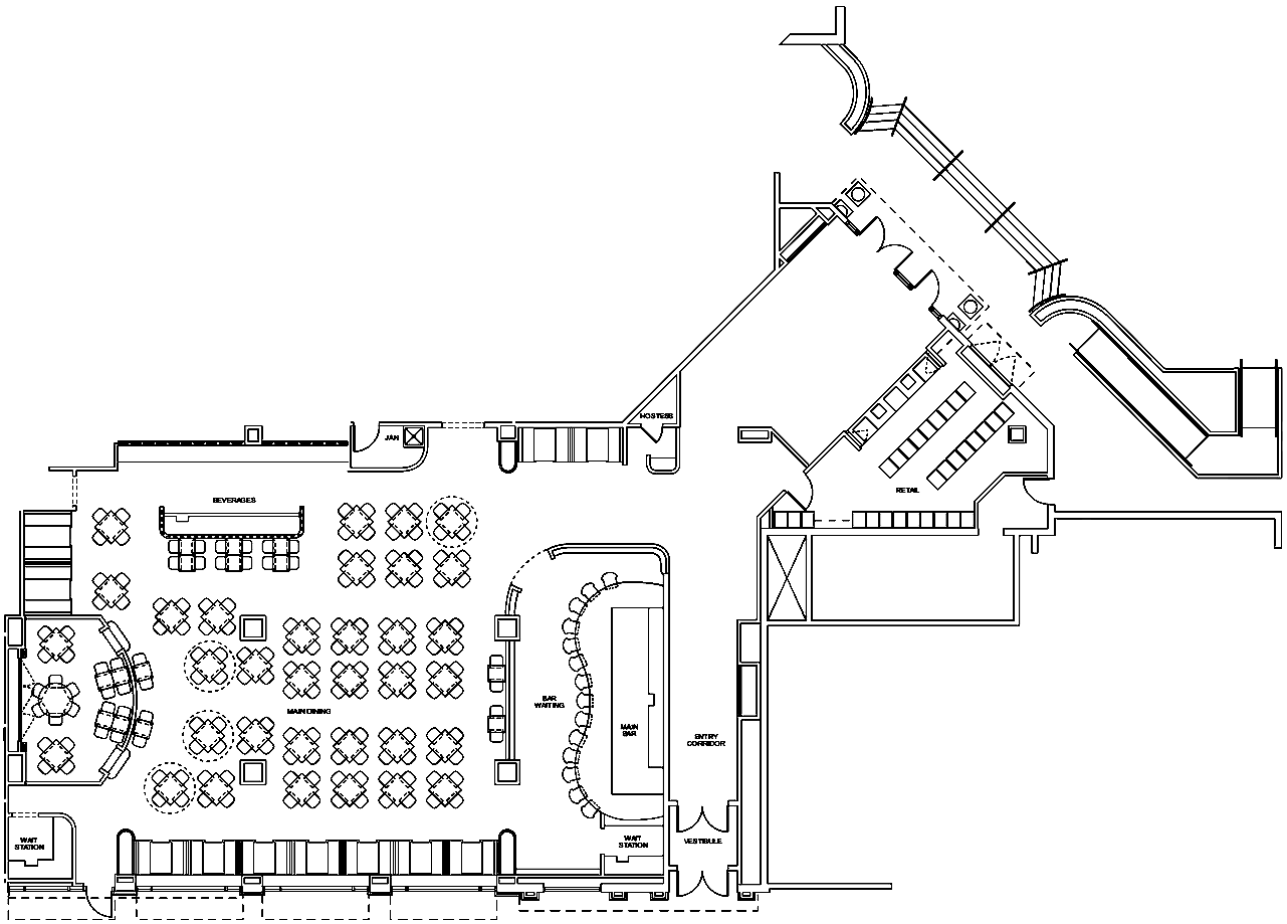


Abbildung 1.1.: Gebäudeplan Hardrock Cafe, Atlantic City (Quelle: www.hardrock.com, 25.03.13)

1.1.1. Eingangsbereich

Vor dem Eingangsbereich warten die Gäste in einer Schlange, bis an der Bar Plätze frei werden. Ziel ist, diese Warteschlange so kurz wie möglich zu halten.

Ist die Schlange vor dem Eingang zu lang, verlassen die Gäste das Restaurant wieder ohne Konsumation. Dies soll möglichst vermieden werden, da diese Gäste keinen Umsatz

bescheren. Sind dagegen Plätze an der Bar frei, so bewegen sich die Gäste direkt dort hin.

1.1.2. Bar

Die Bar dient als Warteschlange für die Gäste, bis Plätze an einem Tisch frei werden. Gäste werden an der Bar mit Getränken bedient.

In der Realität wäre das Ziel, dass die Bar optimal besetzt ist, weil dies zusätzliche Einnahmen durch Konsumation generiert. Gleichzeitig sollte vor der Türe die Schlange möglichst klein sein oder gar nicht existieren, damit keine Gäste an der Kälte warten müssen, oder sogar wieder gehen. Die Wartezeit an der Bar sollte nicht mehr als ein bis zwei Drinks betragen, damit niemand wieder geht, bevor ein Tisch frei wird.

Das Barpersonal soll in unserer Simulation nicht berücksichtigt werden. Grund dafür ist die Unabhängigkeit des Barbereichs vom Rest des Lokals. Das heisst: Das Barpersonal ist nicht in der Lage, kurzzeitig die Bar zu verlassen und Tische zu bedienen. Weil das Barpersonal nur für die Ausgabe der Getränke verantwortlich ist, kann es in der Simulation vernachlässigt werden, ohne das System zu verfälschen. Zudem gehen wir von einer fixen Bargrösse aus und beziehen auch den durch die Bar erzielten Umsatz nicht in die Simulation mit ein.

1.1.3. Tische

Am Tisch werden ein bis drei Gänge gegessen. In einem ersten Schritt soll dies sehr einfach abgebildet werden. Auf eine komplexe Menüsimulation in der Granularität der einzelnen Gänge wird verzichtet.

Das Servicepersonal bewegt sich ein einziges Mal zum Tisch und nimmt Bestelllungen auf. Das Servicepersonal ist somit pro Tisch ein Mal für eine bestimmte Zeit belegt und anschliessend wieder frei.

Ein Szenario, in dem das Servicepersonal mehrfach an einem Tisch vorbeikommt – z.B. um das Gedeck, Teller und Karte zu bringen und den Tisch am Ende wieder abzuräumen – soll erst in einem zweiten Schritt modelliert werden, je nach verfügbarer Projektzeit.

In der realen Welt bestehen die Tische aus Vierertischen. Ankommende Gruppen müssen entweder aufgeteilt oder Tische zusammengeschoben werden, damit Gruppen zusammen sitzen können. Bei Gruppen von weniger als 4 Personen kann es unbesetzten Plätzen an den Tischen kommen: Ein einzelner Gast setzt sich selten zu einer Gruppe von 3 Personen an einem Vierertisch. Um die Simulation nicht unnötig aufzublähen, soll dies nicht berücksichtigt werden. Die Anzahl Plätze werden fix gewählt anhand des Beispiels von Atlantic City (250 Plätze).

Nach dem Esssen verlassen die Gäste das Hardrock Cafe und der Tisch wird frei für neue Gäste. In der Realität kann es durchaus vorkommen, dass Gäste nach dem Essen nochmals an die Bar zurückkehren. Auch dies soll in der Simulation nicht berücksichtigt werden.

1.1.4. Küche

In der Küche werden die verschiedenen Mahlzeiten gekocht und angerichtet. Die Köche arbeiten gleichzeitig an mehreren Menüs und versuchen, alle zu einer Gruppe von Gästen gehörenden Mahlzeiten gleichzeitig fertigzustellen.

In einem ersten Schritt soll die Küche sehr rudimentär simuliert werden. Es existieren Köche, die eine bestimmte Zeit benötigen, um Essen zuzubereiten. Paralleles Kochen von Mahlzeiten soll nur durch entsprechende Verkürzung der Gesammtkochzeit abgebildet werden.

In einem zweiten Schritt könnnten verschiedene Köche, die pro Mahlzeit und Gang für eine bestimmte Zeit gebunden sind, modelliert werden.

1.2. Bewegungsabläufe

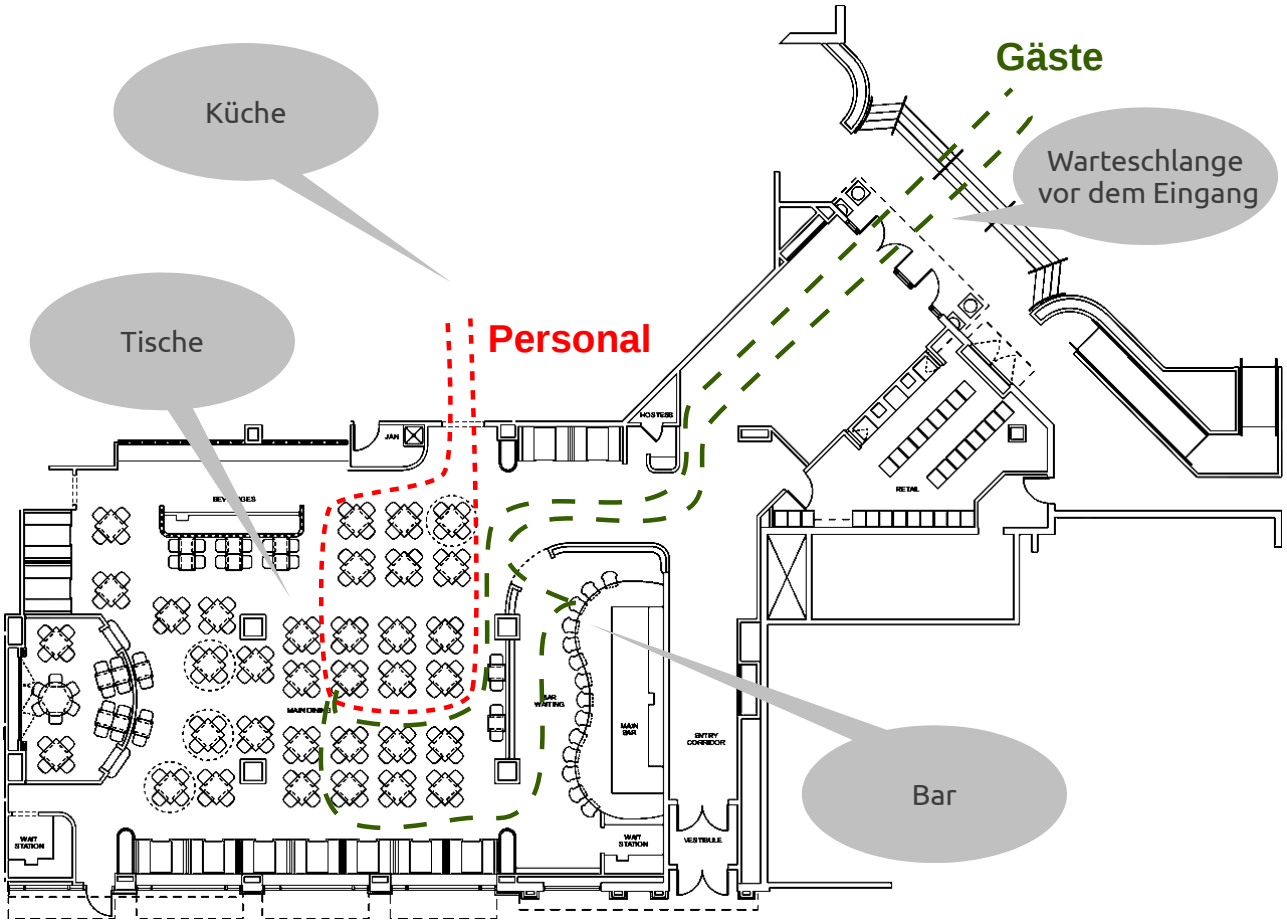


Abbildung 1.2.: Bewegungsschema im Hardrock Cafe, Atlantic City (Beispiel Bewegungsablauf an einem Tisch)

Die verschiedenen Aktivitäten finden an unterschiedlichen Orten im Café statt. Die daraus resultierenden Wegzeiten variieren stark, unterliegen sehr komplexen Verteilungen und können nur empirisch ermittelt werden.

1.2.1. Gäste

- Gäste benötigen eine variierende Zeit um vom Eingang an die Bar zu gelangen.

- Gäste benötigen eine variierende Zeit, um von der Bar zum zugewiesenen Tisch zu gelangen. Diese kann von der Gruppengrösse, der Anordnung der Tische und der Belegung des Restaurants (verschiedene "Gästeströme") abhängen. Diesen Umstand berücksichtigen wir in der Simulation jedoch nur durch eine Variation der Wegzeit – unabhängig von den beeinflussenden Faktoren.
- Gäste benötigen eine variierende Zeit, um das Lokal wieder zu verlassen.

1.2.2. Personal

- Köche bewegen sich nur innerhalb der Küche
- Das Servicepersonal benötigt eine variierende Zeit einerseits für den Weg von Tisch zu Tisch und andrerseits für den Weg von der Küche zu den Tischen.
- Die Tellerbinger benötigen je nach Tisch unterschiedlich viel Zeit, um von der Küche zu den Tischen zu gelangen.
- Servicepersonal, Tellerbringer und Gäste beeinflussen sich gegenseitig in den Wegzeiten: Eine grosse Anzahl von Personen im Raum führt zu Verzögerungen bei allen Laufwegen. Theoretisch kann es auch zu Zusammenstössen kommen. Dies wird in der Simulation nicht berücksichtigt.

1.2.3. Realität und Simulation

Für die Simulation sollen sinnvolle Annahmen über das Verhalten der Wegzeiten getroffen werden und die gegenseitige Beeinflussung der Gäste und des Personals gegenseitig und untereinander vernachlässigt werden. Eine zusätzliche 2D/3D Animation wäre sehr aufwändig, was den Rahmen dieses Projektes sprengen würde. Zudem kann aus einer Animation kein grosser Mehrwert gezogen werden, da die Prozesse auch ohne gut vorstellbar sind.

1.3. Warum soll simuliert werden?

- Das Szenario ist zu komplex, um alle einfliessenden Faktoren in statischen Berechnungen zu berücksichtigen. Es sind zu viele Variablen im Spiel.
- Gegenseitige Beeinflussung der Faktoren: Das System Restaurant eignet sich für die Simulation, da die Werte wie Umsatz und Auslastung von verschiedenen Faktoren abhängen und eine einfache Berechnung deshalb nicht möglich ist. Die Wahl und Anzahl der Ressourcen wie Anzahl Köche, Servicepersonal und Tellerbringer ist entscheidend für das Funktionieren des Systems. Zudem beeinflussen sich die Ressourcen gegenseitig und stehen in Abhängigkeit zueinander.
- Gebäudebau: Aus Kostengründen ist es nicht möglich einen Prototypen zu bauen und darin zu testen.

- Ist das Gebäude einmal gebaut, so sind architektonische Veränderungen, die sich aus einer falschen Annahme der Anzahl von Gästen und Mitarbeitern ergeben (Anzahl Kochplätze, Grösse der Bar, Abstand der Tische zueinander) nur mit grossem Kostenaufwand realisierbar.
- Testen durch Reallife-Simulation ist nicht möglich, weil Statisten beim Warten sterben würden, vom vielen Essen dick würden oder verhungern würden beim Warten auf die Bestellung. Möglicherweise würden Sie auch an der Bar zu viele Drinks nehmen und wären zu betrunken um an einen Tisch zu wechseln.
- Zudem tauchen Unsicherheiten auf durch Verteilungen, Streuungen und Variabilitäten in der Anzahl der Gäste, Serviceagents und deren Servicetime, Warteschlangen und Wartezeiten.

1.4. Objectives

1.4.1. Gewinn

Ziel ist die Gewinnmaximierung durch gute Belegung des Restaurants. Zu optimieren ist die Anzahl des Personals im Hinblick auf dieses Ziel. Die Anzahl des Personals im Service, in der Küche und im Tellerbringen ist voneinander abhängig. Hautpziel ist, die optimale Anzahl für den jeweiligen Bereich zu bestimmen.

1.4.2. Utilization

Indirekt abhängig von der Utiliziation der beteiligten Ressourcen ist das Ziel der Gewinnmaximierung. So ist zum Beispiel ein nicht ausreichend ausgelastetes Servicepersonal mit Kosten verbunden, denen keine Einnahmen in Form von verkauften Mahlzeiten gegenüber steht.

1.4.3. Dauer des Aufenthalts

Ebenfalls ein Hinweis auf geringen Umsatz ist eine zu grosse mittlere Aufenthaltszeit am Tisch oder sehr lange Wartezeiten der Gäste auf bestellte Mahlzeiten. Wir gehen davon aus, dass die Aufenthaltsdauer bei ca. 1.5 Stunden liegt und die mittlere Wartezeit für die Bedienung 15 Minuten nicht übersteigt.

1.5. Spezifikationen

Gearbeitet wird mit fiktiven Daten, ausgenommen die Anzahl der Plätze im Saal. Diese werden vom Hardrock Atlantic City übernommen.

1.5.1. Variable Grössen

Es sollen verschiedene Szenarien simuliert werden. Die dazu benötigten Grössen sollen Variabel sein:

- Queuelänge
- Anzahl Servicepersonal
- Anzahl Tellerbringer
- Anzahl Köche
- Gruppen mit unterschiedlichen Anzahlen von Gästen

1.5.2. Resultatgrössen

Durch die Simulation sollen die Aufenthaltszeiten ermittelt werden, z.B. die **Dauer des Aufenthaltes und Wartezeiten am Tisch**, die **Gesamtaufenthaltsdauer im Restaurant** und die **Auslastung der Agents**. Zusätzlich soll anhand von festgelegten Kosten der Gewinn berechnet werden.

1.5.3. Zusatzprojekt (optional)

In einem weiteren Projekt kann die Simulation ausgebaut und verfeinert werden, indem weitere Parameter hinzugefügt und detailliertere Daten erhoben werden:

- Anzahl Personal an der Bar
- Verschiedene Essen/Menüs mit unterschiedlichen Servicezeiten für die Köche.
- Gruppen müssten zusammen an einem Tisch untergebracht werden, keine Leerplätze.
- Weitere Resultate: Wie lange dauert die Bestellaufnahme, wie lange muss ein Gast durchschnittlich auf die Bestellung warten und wie lange wartet er auf die Rechnung? Wie oft muss das Servicepersonal an einen Tisch gehen pro Gast?
- Optional: Kleine Animation

Im grunde kann die Simulation beliebig verfeinert werden.

2. Modellierung

2.1. Activities

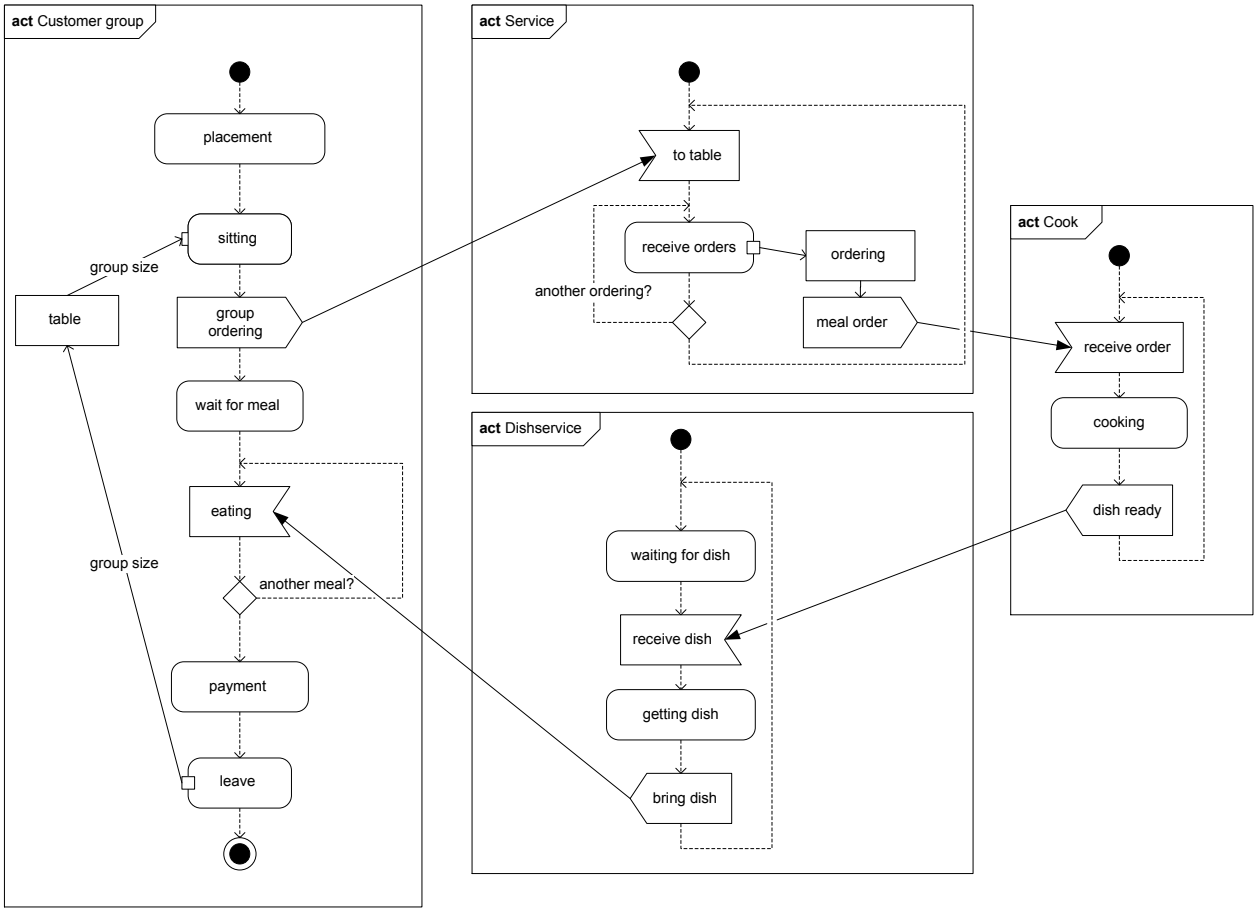


Abbildung 2.1.: Activity Diagramm

2.1.1. Gastgruppe

Eine Gastgruppe belegt während dem Aufenthalt am Tisch Plätze. Anschliessend wird Servicepersonal benachrichtigt um eine Bestellung zu erstellen. Entsprechend den Anzahl Gängen werden soviele Gänge angeliefert. Vor den Verlassen des Lokals wird Bezahlt, anschliessend werden die Plätze wieder freigegeben.

2.1.2. Service

Das Servicepersonal befindet sich in einem geschlossenen Zyklus. Es nimmt am Tisch eine oder mehrere Bestellungen auf und bringt diese in die Küche.

2.1.3. Köche

Die Köche befinden sich ebenfalls in einem geschlossenen Zyklus. Sie nehmen Bestellungen entgegen und richten Teller an.

2.1.4. Tellerbringer

Auch die Tellerbringer befinden sich in einem geschlossenen Zyklus. Sie holen angerichtete Teller ab und bringen Sie an den Tisch.

2.2. Hardrock Guest Flow

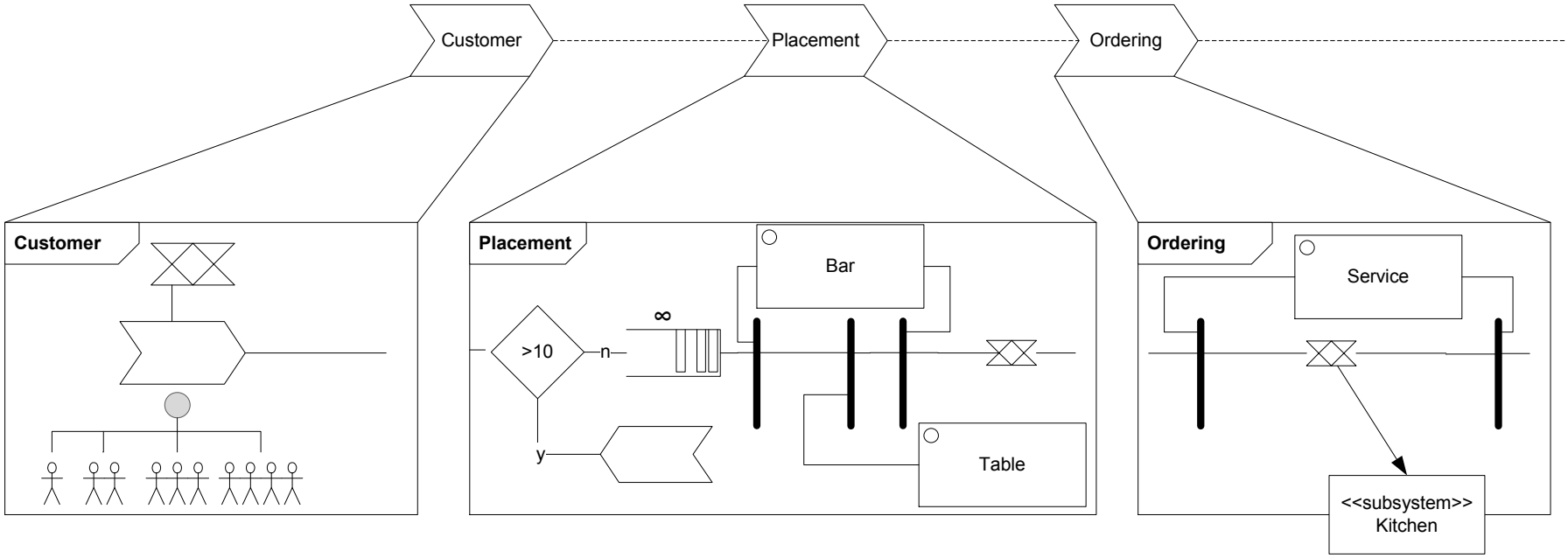


Abbildung 2.2.: Hardrock Flow Diagramm Teil1

2.2.1. Customer

Gäste werdem in Gruppen von eins bis vier Personen erzeugt. Die Erzeugung unterliegt einer definierten Verteilung.

2.2.2. Placement

Gäste warten in einer Queue, bis eine Platzressource zur Belegung an der Bar frei wird. Sollte diese Queue die Länge von 10 Gruppen übsteigen, was im Mittel ca. 20 Leuten entspricht, werden neu ankommende Gäste das Hardrock wieder verlassen.

Die Anzahl verlorener Gäste wird am Schluss ermittel. Die Platzressource wird freigegeben, sobald eine freie Tischressource belegt werden kann.

2.2.3. Ordering

Während einer Bestellung wird die Bedienung als Ressource belegt und anschliessend wieder freigegeben.

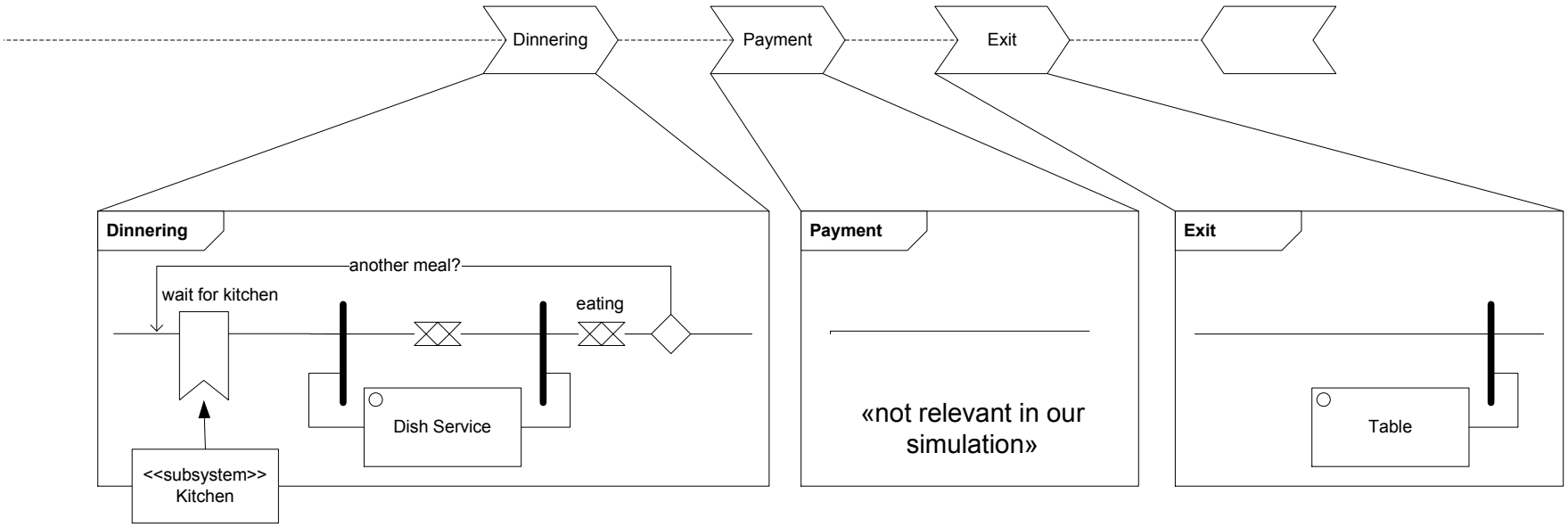


Abbildung 2.3.: Hardrock Flow Diagramm Teil2

2.2.4. Dinnering

Gäste warten, bis die Bestellung zubereitet wurde (Verteilfunktion). Für die Auslieferung der Teller wird die Ressource Tellerbringer für eine bestimmte Zeit (Verteilfunktion) belegt. Die Dauer des Essens wird ebenfalls mittels einer Verteilfunktion modelliert.

Wenn Gäste mehrere Gänge bestellt haben, so kommen auch mehrmals die Tellerbringer und bringen das Gekochte.

2.2.5. Payment

Für die Bezahlung ist bis jetzt kein Szenario definiert, sie wird hier nur der Vollständigkeitshalber aufgeführt.

2.2.6. Exit

Gäste geben die belegte Ressource Tisch wieder frei und verlassen das Restaurant.

2.2.7. Kitchen

Die Bestellung der Gruppe wird in einzelneUnterbestellungen aufgesplittet (je nach Gruppengrösse 1-4) und kann dadurch parallel verarbeitet werden. Eine Unterbestellung belegt eine Ressource Koch für die Dauer der Zubereitung, welche mittels einer Verteilfunktion bestimmt wird.

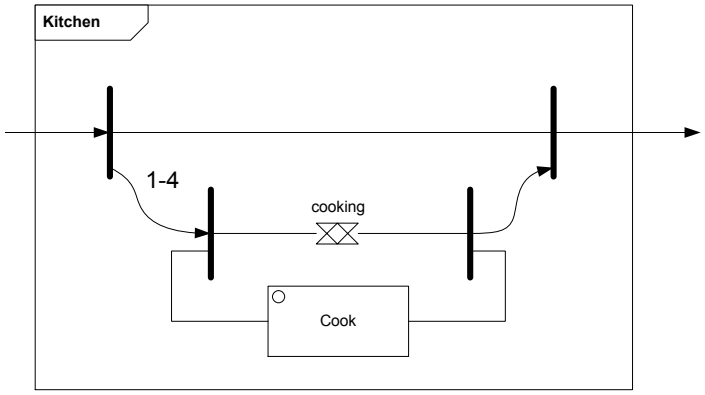


Abbildung 2.4.: Hardrock Flow Diagramm Kitchen

2.2.8. Entitäten

Zusammengefasst sind in der Modellierung des Systems die folgenden Entitäten vorhanden.

- Vor der Türe:
  - Queue: Ankommende Gäste
- Küche
  - Agents: Köche
- Bedienung
  - Agents: Servicepersonal, Tellerbringer
- Bar
  - Queue: Anzahl Warteplätze
- Tische
  - Ressourcen: Anzahl Plätze
- Gäste
  - Entities: Gruppen von 1-4 Personen

3. Simulation mit Arena

3.1. Customer

3.1.1. Verteilung der ankommenden Gäste

Um die Verteilung des Andrangs, insbesondere die beiden Spitzenzeiten am Mittag und am Abend, zu simulieren, nutzen wir einen Schedule. Die Ankunftsrate im Scheduler wird durch eine negative Exponentialverteilung bestimmt.

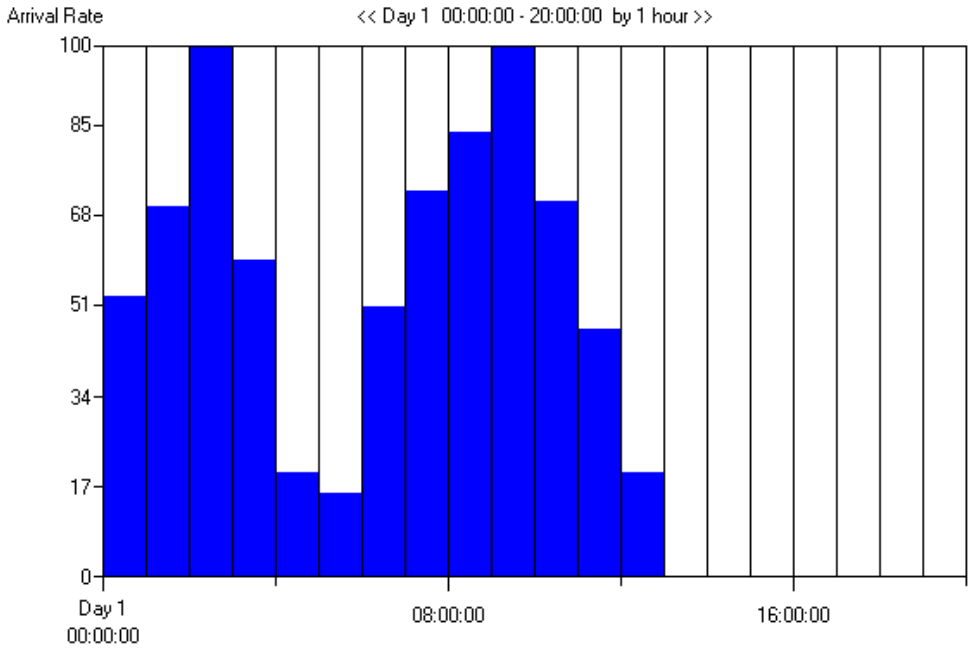


Abbildung 3.1.: Ankunfts Sceduler in Arena

Gruppengrösse

Einer angekommenden Entität wird anschliessend eine zufällige Gruppengrösse zwischen eins und vier Personen zugeteilt. Dazu wird eine Triangularfunktion mit einem Mittelwert von 2 verwendet und der Ausgabewert auf die nächste ganze Zahl gerundet. So entsteht eine diskrete Verteilung.

3.2. Placement

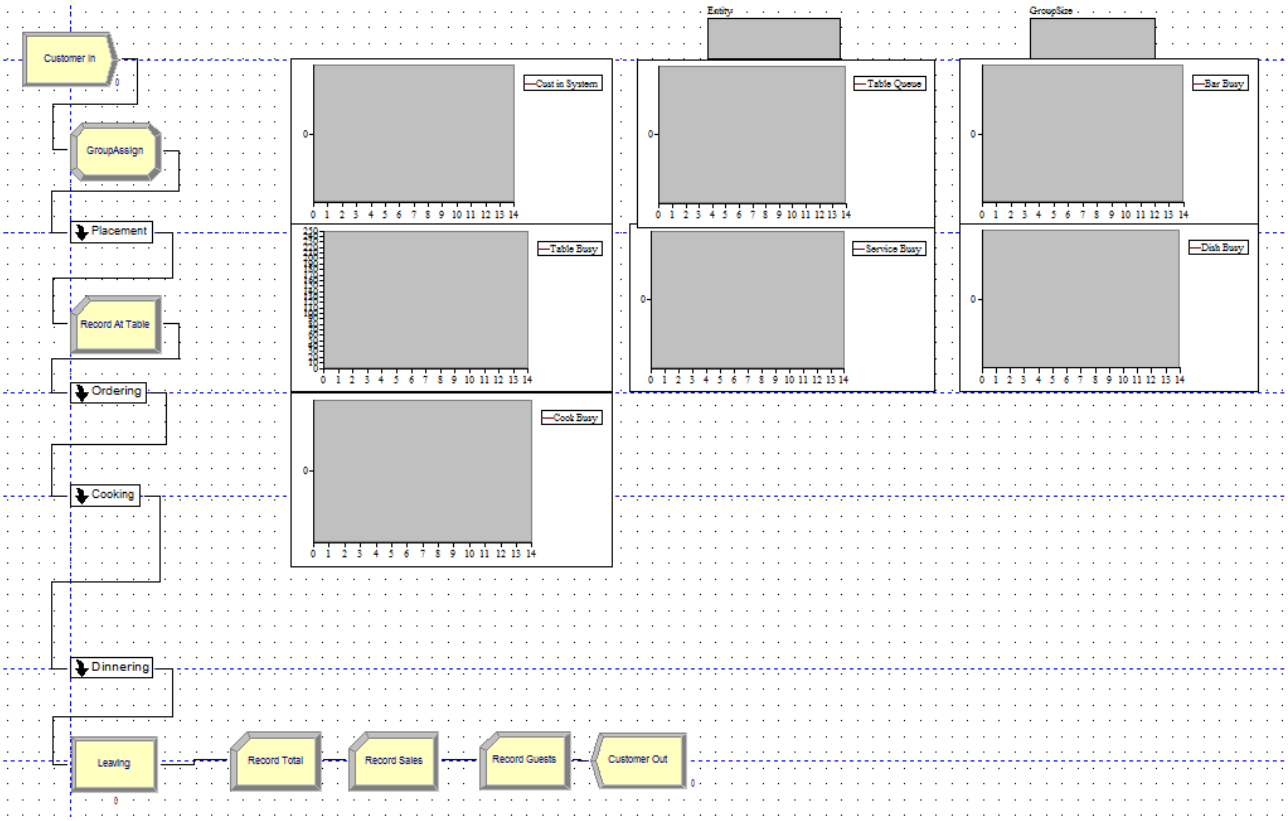


Abbildung 3.2.: Aufbau einer ersten Simulation in Arena mit Submodels

3.2.1. Überlauf am Eingang

Warten zu viele Gäste vor dem Eingang, werden weitere Gruppen abgeschreckt und kehren wieder um.

Für die Simulation wird eine fixe Grösse verwendet: Befinden sich in der EntryQueue weniger als 10 Gruppen, so stellt sich die neue Gruppe an, andernfalls nicht. Mittels einem alternativen Dispose und Record werden die abgeschreckten Kunden gezählt.

3.2.2. Assignement und Release von Warteschlange, Bar und Tisch

In Arena wird das Assign/Release mittels Ressourcen umgesetzt. Als erstes wird die Bar zugewiesen. Sobald die ganze Gruppe freie Tischplätze hat, wird die der Tisch assigned. Erst wenn der Platz am Tisch assigned ist, wird die Bar released.

3.2.3. Laufweg der Gäste von der Bar zum Tisch

Für den Weg von der Bar zum Tisch wird eine Uniformverteilung von eins bis fünf Minuten eingesetzt. Der Grund liegt darin, dass sich die Gruppen im Restaurant oft weit auseinander setzen. Es entstehen unterschiedliche Laufzeiten, da Tische näher oder weiter entfernt von

der Bar platziert sind. Da die Entfernung der Tische keine Häufung aufweist und wir von einer gleichmässigen Auslastung aller Tische ausgehen, wird die Wegverteilung als uniform angenommen.

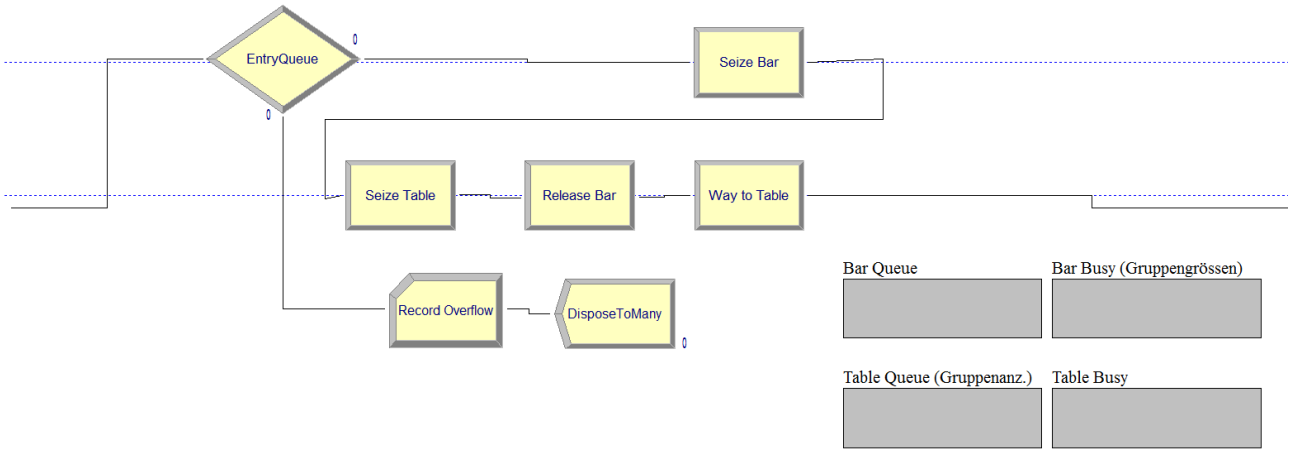


Abbildung 3.3.: Placement Submodule in Arena

3.3. Ordering

3.3.1. Bestellvorgang

Das Erreichen des Tisches durch die Bedienung und die Bestellungsaufnahme werden zusammengefasst und mittels einer Triangularverteilung mit einem Mittelwert von 6 sowie einem Minimum von 3 und Maximum von 8 abgebildet. Pro Person am Tisch wird eine Minute aufgeschlagen, um die Abhängigkeit der Bestellzeit von der Gruppengrösse zu realisieren.

Zudem ist in dieser grosszügigen Zeitberechnung der Anteil für einen zweiten Besuch am Tisch und die zusätzlichen Aufgaben wie Bereitmachen und Abräumen des Tisches enthalten.

3.4. Preparation

3.4.1. Verteilung der Zubereitungszeit

Für die Verteilung der Kochzeit wählen wir eine Triangularverteilung von eins bis zehn Minuten und einem Mttelwert von fünf Minuten. Da im Hardrock verschiedenste Speisen zubereitet werden, von Salaten über Hauptgängen bis Desserts, ist die Verteilung hier die sinnvollste Abbildung. Unsere erste Idee war eine Normalverteilung mit einem Mittel von fünf und einer Standardabweichung von drei. Jedoch giebt es bei dieser Verteilung auch Werte im Minusbereich was nicht Realitätsgetreu ist. Die Kochzeit wird so kurz gewählt, weil ein guter Koch durch kochen von mehreren Menüs gleichzeitig Synergien nutzen kann.

3.4.2. Gruppenbestellungen

Gruppen bestellen gemeinsam und erhalten ihr Essen zum gleichen Zeitpunkt, gekocht wird es jedoch unabhängig. Um dies abzubilden, verwenden wir folgendes Vorgehen: Die Bestellung der Gruppe wird in der Küche in einzelne Bestellungen unterteilt und damit werden bis zu vier Unterbestellungen simuliert. Vor dem Ausliefern werden die Unterbestellungen wieder zu einer Gesamtbestellung zusammengefügt.

3.5. Dinnering

3.5.1. Bestellungsauslieferung

Die Tellerbringer liefern immer nur Gesamtbestellungen an eine Gruppe aus. Darum wird Bestellung nur ein Tellerbringer benötigt, der bis zu vier Teller (maximale Gruppengrösse) zum Tisch bringt. Zur Simulation wird die Laufzeit von der Theke bis zum Tisch in einer Uniformverteilung von einer halben bis vier Minuten abgebildet.

3.5.2. Essen

Das Essen dauert zwischen 15 Minuten und einer Stunde. Einzelne Gänge des Essens werden nicht konkret unterschieden, sondern nur implizit über die Dauer des Aufenthalts am Tisch in einer Verteilfunktion simuliert. Genutzt wird eine Triangularverteilung mit einem Mittelwert von 40 Minuten.

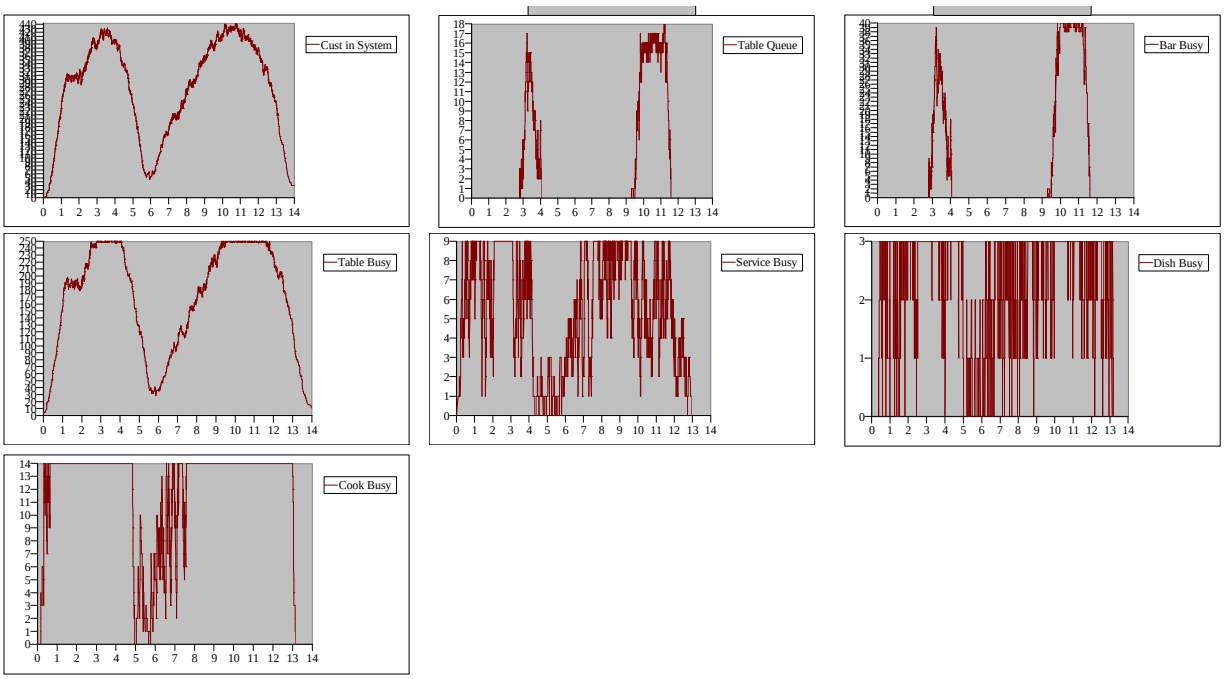


Abbildung 3.4.: Überwachung der Simulation in Arena mit Diagrammen

3.6. Szenarien

Zur einfacheren Realisierung der Szenarien wird in Arena das Tool Process Analyzer genutzt. Damit können Szenarien definiert werden, die die Simulation mit den definierten Parametern beliebig viele Male aufrufen und die Resultate wieder zusammenführen.

Die Szenarien sollen in einem ersten Schritt aufzeigen, in welcher Grössenordnung sich die Anzahl der benötigten Servicekräfte, Tellerbringer und Köche zueinander bewegt. In weiteren Schritten sollen die Szenarien ähnlich einer Lawinensuche auf den jeweils besten Bereich eingeschränkt und verfeinert werden.

3.7. Plausibilitäts-Checks

Zur Überprüfung der Plausibilität des aufgebauten Arena Modells experimentierten wir mit verschiedenen Parametern. Dabei sind einige Dinge aufgefallen:

- Die minimale Grösse der Bar muss der grösst möglichen Gruppengrösse (in unserem Fall 4) entsprechen. Andererseits können Gruppen, die grösser als die Bar sind, gar nicht ins Restaurant eintreten. Dieses Szenario entspricht nicht ganz der Realität. In der Realität bestehen grosse Gruppen aus kleineren Gruppen. Daher wird ein Teil einer grossen Gruppe bereits das Restaurant betreten, während die Restlichen noch draussen warten.
  → Da wir eine Bar zwischen 10 und 50 Plätzen anstreben und die Gruppen eine maximale Grösse von vier Personen ausweisen, spielt diese Gegebenheit keine Rolle.

- Fünf Angestellte im Service und drei Tellerbringer sind trotz 250 Gästen zu wenig ausgelastet. Das scheint für die Realität wenig plausibel.
  → Als Korrektur werden Lauf- und Bestellzeiten für das Servicepersonal und die Tellerbringer angepasst.

- Den Flaschenhals in der Simulation stellen die Köche dar. Die tiefe Utilization des Servicepersonals ergibt sich daraus, dass die Zubereitung der Essen zu lange dauert. Um dieses Verzerrung zu beheben, muss die Anzahl der Köche im Verhältnis zur Anzahl Servicepersonal erhöht werden.

  Gleichzeitig bringt das Servicepersonal in der Realität noch Getränke, Speisekarte, Gedecke, weist dem Gast den Tisch, räumt ab und steht für Fragen zur Verfügung.
  → Wir verzichten in unserer Simulation darauf, dies abzubilden. Da die Zeit für die Bedienung aber grosszügig berechnet ist, können die zusätzlichen Zeiten als darin enthalten betrachtet werden.

- Die benutzten Zeitvariablen zur Erfassung von Zeitstempeln im System basieren auf TNOW, wobei TNOW scheinbar kein Stempel zurückgibt, mit dem gerechnet werden kann.
  → TNOW muss zuerst in Basetime konvertiert werden. Wobei noch beachtet werden muss, dass die Basetime vom Run Setup abhängig ist.

3.8. Umsatz

Um den generierten Umsatz zu berechnen, setzen wir die folgenden Werte ein: Die Kosten für Angestellte sind konstant und nicht von der Anzahl Bestellungen / Besucher abhängig. Deshalb wird wird der Lohn den Ressourcen als Kosten fix zugeteilt.

Bezeichnung	**Preis CHF**	**Verteilung**
Menu	20-60	TRIA(20,30,60)
Stundenlohn Koch	25	konstant (Mittelwert)
Stundenlohn Tellerbringer	20	konstant (Mittelwert)
Stundenlohn Servicepersonal	25	konstant (Mittelwert)

4. Auswertung

4.1. Szenarien

Gestartet wird mit 24 Szenarien, die sich verdoppelden Anzahlen von Angestellten in jedem Bereich (2, 4, 8, 16) und Variationen beinhalten.

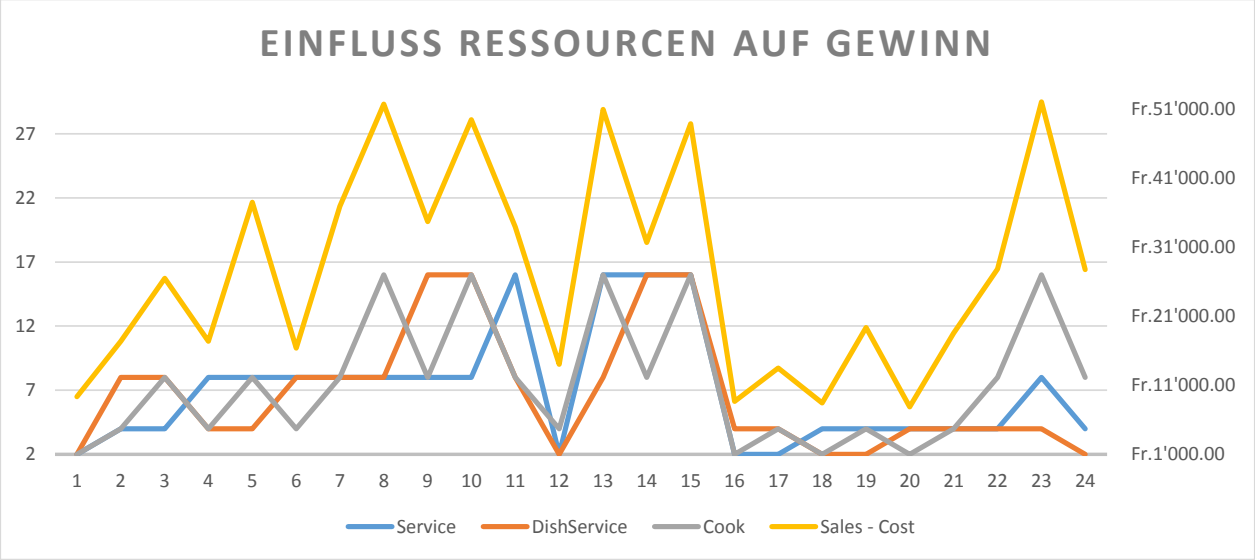


Abbildung 4.1.: Grobszenarien (Y_{left}: Mitarbeiter / X: Szenario / Y_{right}: Gewinn)

4.1.1. Szenarienoptimierung

Nach einer ersten Simulation hat sich herausgestellt, dass einige der oben definierten Szenarien gleich von Anfang an ausgeschlossen werden können. So zeigte sich, dass die Anzahl der Tellerbringer und Servicekräfte nicht über der Anzahl Köche liegen kann. Dies wurde ersichtlich, weil sich der Umsatz und damit der Durchsatz (Servicezeiten) im System nicht änderte wenn die Anzahl der Tellerbringer und Servicekräfte variiert wurde und diese überhalb der Anzahl Köche lag. Lediglich deren Utilization verringerte sich bei einer Erhöhung der Anzahl.

4.1.2. erste Selektion

Bezogen auf die Umsatzoptimierung lassen sich aus den besten vier und den schlechtesten vier Szenarien die folgenden Tendenzen ableiten:

- Es werden viele Köche benötigt.

- Tellerbringer braucht es sehr wenige.
- Beim Servicepersonal werden etwa halb so viele wie Köche und doppelt so viele wie Tellerbringer benötigt.

Aus diesen Daten entnehmen wir anschliessend den besten Fall (8 Servicepersonal, 4 Tellerbringer und 16 Köche) und analysieren diesen detaillierter:

- Der Service ist mit fast 90% zu hoch ausgelastet. Konkret ist das Personal nicht nur zu den im Scheduler definierten Spitzenzeiten komplett ausgelastet, sondern über den ganzen Zeitraum mit Ausnahme in den Nachmittagsstunden.
- Die Tellerbringer sind zwar über die ganze Simulationszeit regelmässig ausgelastet, insgesamt jedoch zu wenig.
- Die Köche sind sehr konstant über die gesamte Simulationszeit ausgelastet.

4.1.3. zweite Selektion

Als erstes erhöhen wir die Anzahl Simulationsdurchgänge von 10 auf 100, um stabilere und vertrauenswürdigere Resultate zu erhalten.

Servicepersonal

In einer erneuten Simulation wird nur die Anzahl Servicepersonal zwischen 8 und 14 variiert, während die Anzahl Tellerbringer bei 4 und die Anzahl Köche bei 16 bleibt.

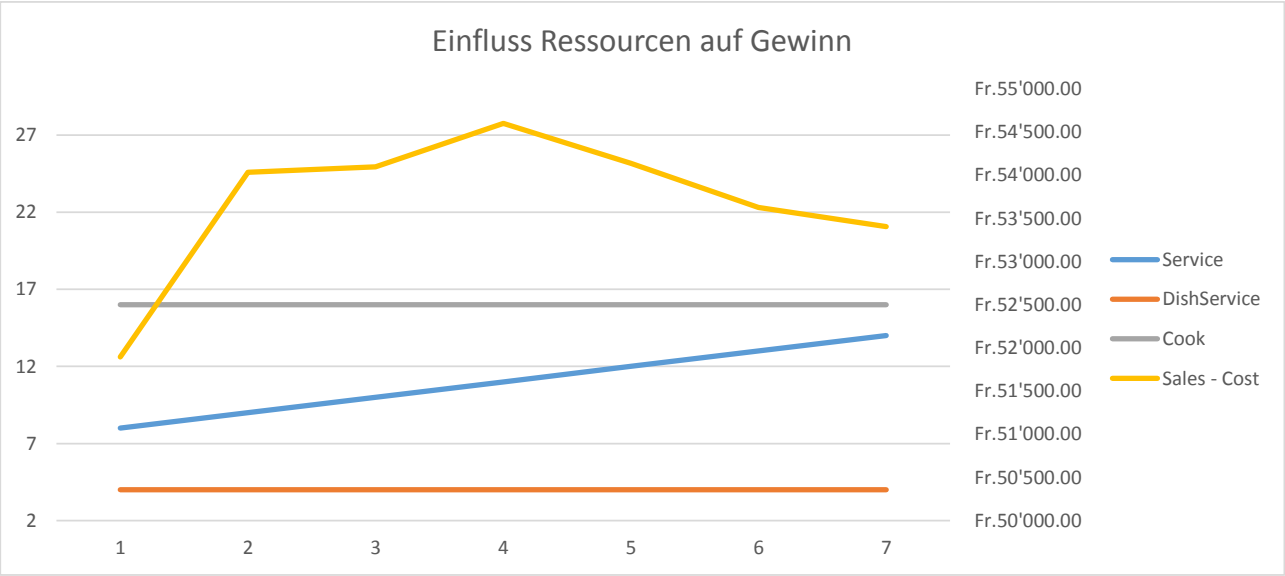


Abbildung 4.2.: Serviceoptimierungsszenarien

Aus der erneuten Simulation geht hervor, dass sich die optimale Anzahl der Servicekräfte um 11 bewegt.

4.1.4. Tellerbringer

In einem weiteren Schritt fixieren wir das Servicepersonal bei 11 und die Köche bei 16 Personen, variieren aber die Tellerbringer zwischen 2 und 6.

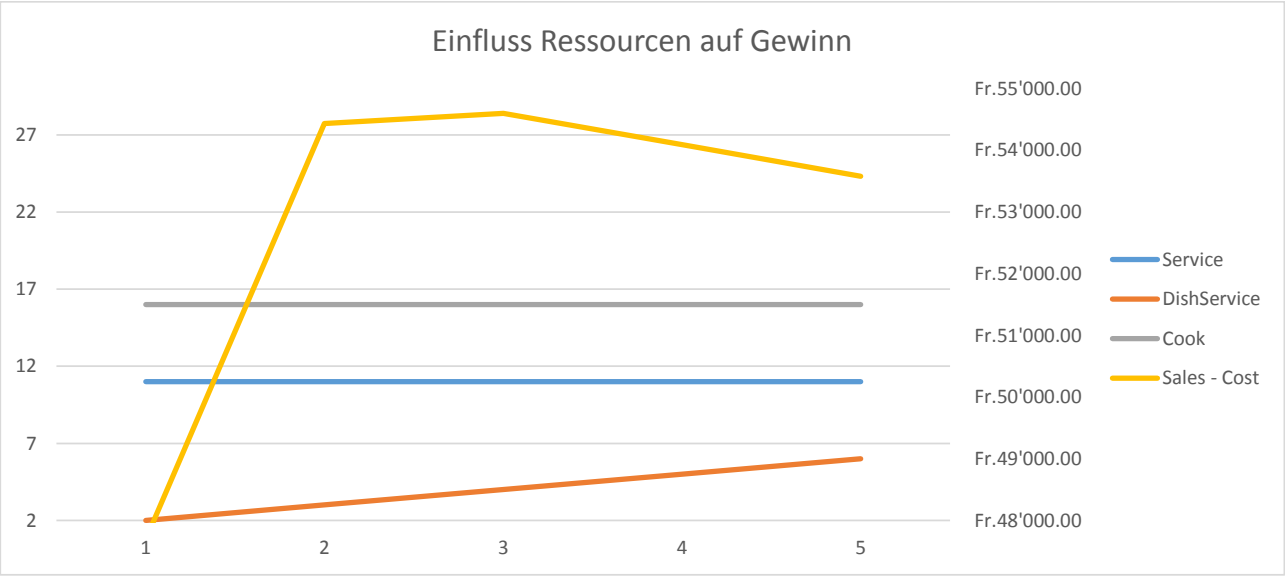


Abbildung 4.3.: Tellerbringeroptimierungsszenarien

Aus den Ergebnissen der Simulationsläufe geht hervor, dass die optimale Anzahl Tellerbringer bei 3 liegt. Wir wählen für das weitere Vorgehen trotzdem eine Anzahl von 4, um auch mit einer grösseren Anzahl an Köchen die Menüs noch rauszubringen.

4.1.5. Köche

Im letzen Schritt fixieren wir die Tellerbringer und das Servicepersonal, variieren stattdessen die Köche zwischen 11 und 18 Personen.

Die optimale Anzahl Köche liegt bei 14. Daraus ergibt sich ein Verhältnis von 11:4:14, bzw. 0.8 Servicekräfte auf 0.3 Tellerbringer auf einen Koch.

4.1.6. Dritte Selektion

Nach der Selektion muss überprüft werden, ob die einzelnen Optimierungen zu globalen Veränderungen geführt haben.

Dazu erstellen wir 27 Szenarien, die das Servicepersonal zwischen 10 und 12, die Tellerbringer zwischen 3 und 5 sowie die Köche zwischen 13 und 15 variieren.

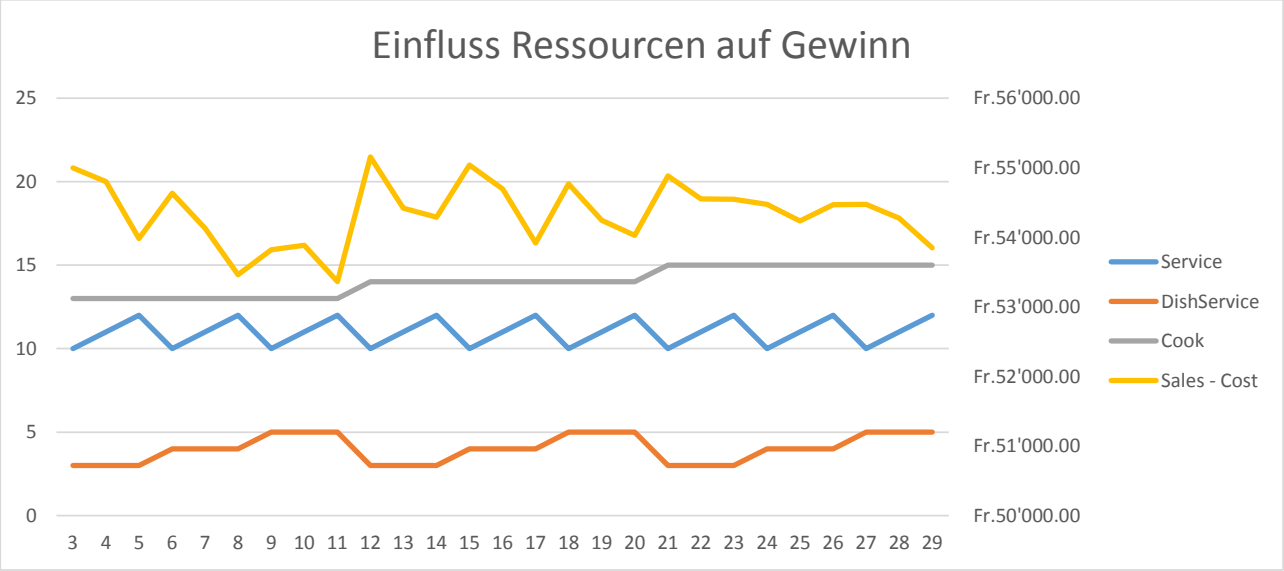


Abbildung 4.4.: Endoptimierung

Die vier besten Resultate dieser Simulation erweisen sich als die allerbesten Resultate über alle bisherigen Szenarien hinaus. Alle diese Szenarien benötigen 10 Servicekräfte, die Tellerbringer variieren zwischen 3 und 4 und die Köche zwischen 13 und 15. Denr besten Fall erhalten wir mit 10 Servicekräften, 3 Tellerbringer und 14 Köchen.

In einem nächsten Schritt müssten wir noch die Fälle auf zwei Tellerbringer und 9 oder 8 Servierkräfte ausdehnen, um sicher zu gehen, dass der optimale Fall gefunden wurde. Auch dann kann jedoch nicht ausgeschlossen werden, dass nur ein Suboptimum gefunden wurde. Wir entscheiden uns jedoch für eine Verifikation mit den in Arena eingebauten Tools.

4.1.7. Verifikation

Zur Verifikation der Resultate nutzen wir OptQuest von Arena. Wir lassen OptQuest die Personalgrössen nach dem Gewinn optimieren. Arena liefert nach zwei Tagen Simulation beinahe das gleiche Resultat welches wir weiter oben erhalten haben. Lediglich für die Servicekräfte schlägt Arena eine Person weniger vor.

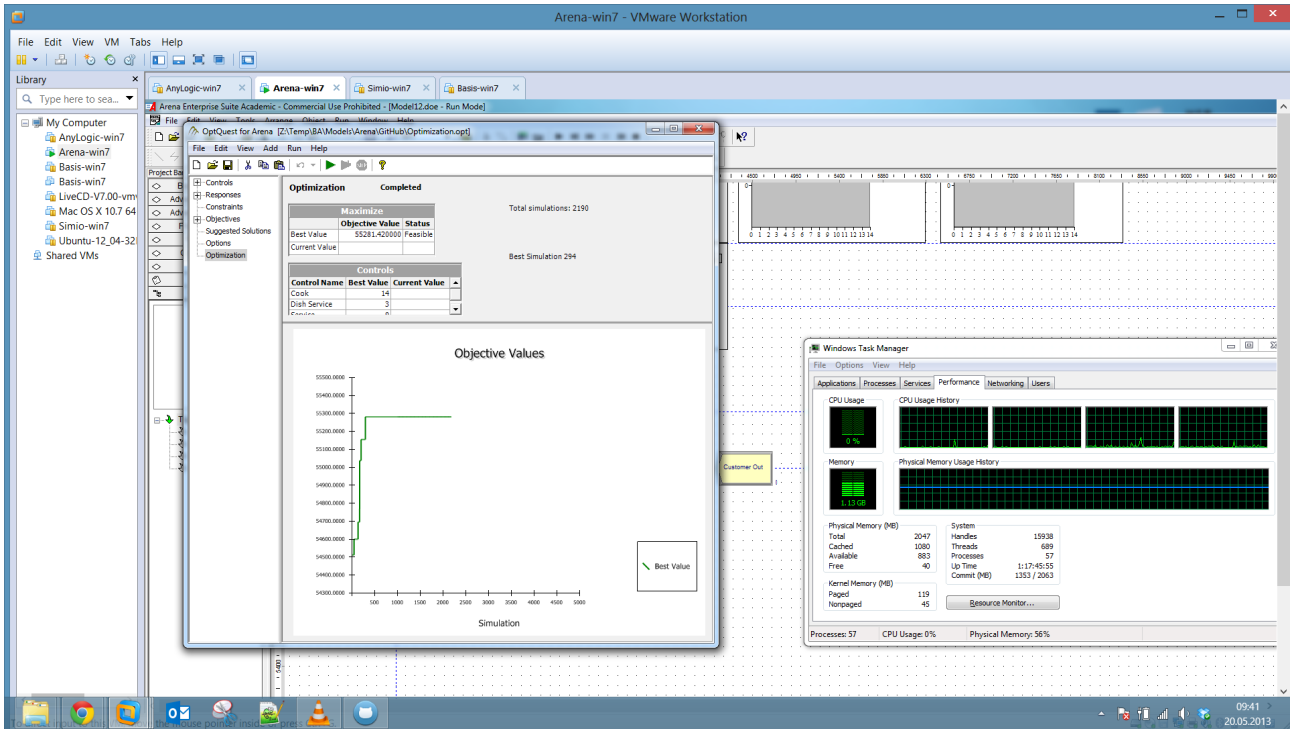


Abbildung 4.5.: OptQuest in Arena

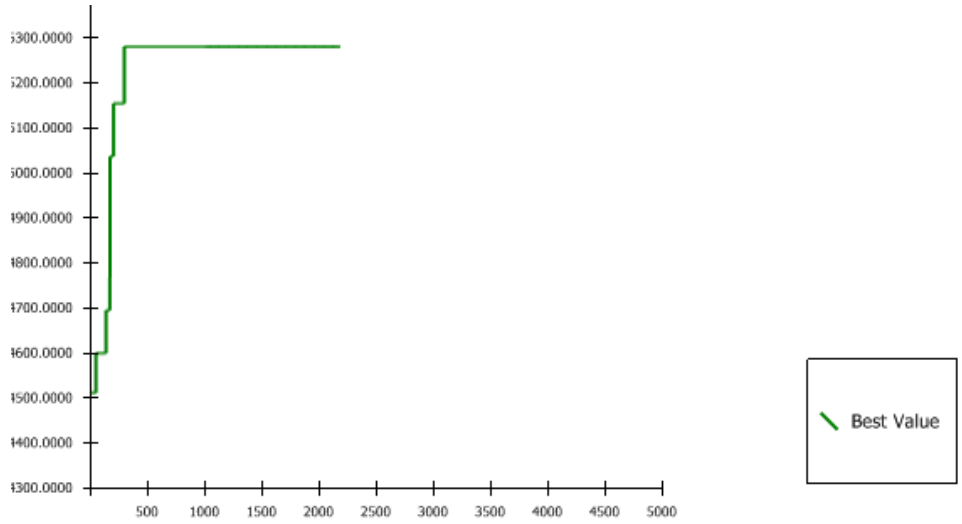


Abbildung 4.6.: OptQuest Suchverlauf

Damit erhalten wir ein endgültiges Resultat von 14 Köchen, 9 Servierkräften und 3 Tellerbringer, die pro Tag über 55’200 CHF erwirtschaften (Umsatz - Löhne) und dabei über 1’700 Essen ausliefern.

Von diesem Betrag müssten natürlich noch sämtliche anfallende Kosten wie Nahrungsmitteleinkäufe, Infrastrukturkosten, Sozialabgaben und Personalausfälle abgezogen werden, wollte man mit den ermittelten Daten einen ernsthaften Businessplan aufstellen.

4.2. Fragestellungen (If Questions)

- **Was passiert, wenn die Anzahl Köche erhöht wird?**
  Die Köche bringen eine grössere Menge an Bestellungen durch in der gleichen Zeit. Das Servicepersonal ist besser ausgelastet. Sobald das Verhältnis von Köchen zu Servicepersonal 1.4:1 übersteigt, sind die Servicerkräfte komplett ausgelastet, bzw. überlastet. Die Tellerbringer sind überlastet, sobald das Verhältnis von Köchen zu Tellerbringern 4.7:1 übersteigt.

- **Was passiert, wenn die Anzahl Köche verringert wird?**
  In der Küche stauen sich die Bestellungen. Die Gäste müssen zu lange auf das Essen warten. Zu wenige neue Gäste können in das Lokal eintreten, das Servicepersonal und die Tellerbringer sind ab den genannten Grenzen nicht mehr ausgelastet.

- **Was passiert, wenn die Anzahl Tellerbringer erhöht wird?**
  Sind zu viele Tellerbringer vorhanden, sind diese nicht ausgelastet und stehen herum.

- **Was passiert, wenn die Anzahl Tellerbringer verringert wird?**
  Sind weniger Tellerbringer als oben erwähnt vorhanden, stauen sich die gekochten Bestellungen in der Küche. Die Gäste erhalten wiederum die Bestellungen verspätet, belegen das Lokal länger und insgesammt sinkt die Anzahl Bestellungen, die durchgebracht werden können.

- **Was passiert, wenn die Anzahl Servicekräfte erhöht wird?**
  Werden die Servicekräfte erhöht, können mehr Bestellungen aufgenommen werden. Sobald das oben genannte Verhältnis zu den Köchen überstiegen wird, stauen sich die Bestellungen in der Küche und das Servicepersonal ist unterausgelastet.

- **Was passiert, wenn die Anzahl Servierkräfte verringert wird?**
  Die Servierkräfte können weniger Bestellungen aufnehmen. Die Küche und die Tellerbringer sind weniger stark ausgelastet.

- **Was passiert, wenn die Anzahl Plätze an der Bar verändert werden?**
  Wird die Platzzahl erhöht, so ist die Schlange vor der Türe kleiner und weniger Leute gehen wieder nach Hause. Das Restaurant ist länger besetzt und mehr Umsatz wird generiert. Ist die Bar zu gross, führt der Verzögerungseffekt dazu, dass Gäste noch nach Betriebsschluss im Lokal sind.

  Ist die Bar zu klein, gehen viele Leute wieder nach Hause, weil die Schlange vor der Tür zu lang ist. Weniger Gäste durchlaufen das Lokal und weniger Umsatz wird generiert.

- **Was passiert, wenn die Grösse der akzeptablen Schlange variiert wird, bis Gäste wieder nach Hause gehen?**
  Wird die Grösse angehoben, gehen weniger Leute nach Hause. Es braucht Mehr Personal im Lokal und die Gäste verbleiben durch den Verzögerungseffekt wie bei der Bar möglicherweise bis nach Betriebsschluss.

  Wir die Grösse verkleinert, so gehen mehr Leute nach Hause. Es wird weniger Personal benötigt und weniger Umsatz generiert.

A. Simulationsdaten

Szenario: Abschätzung wo ein Optimum vermutet wird

Controls						Responses																Excel
Replication	Bar	Table	Service	Dish Service	Cook	Bar.Utilization	Cook.Utilization	DishService.Utilization	Service.Utilization	Table.Utilization	Record At Table	Record Wait For Dish	Record Sales	Resources.TotalCost	Cook.BusyCost	Cook.IdleCost	DishService.BusyCost	DishService.IdleCost	Service.BusyCost	Service.IdleCost	Record Overflow	Sales - Cost
10	40	250	2	2	2	0.835	0.985	0.189	0.995	0.925	1.199	2.279	11295	1952.174	685.703	10.387	105.493	454.259	692.546	3.786	1172	9'342.83
10	40	250	4	8	4	0.81	0.985	0.094	0.88	0.923	0.803	2.319	22431	5029.973	1373.356	21.292	210.049	2029.781	1227.715	167.781	923	17'401.03
10	40	250	4	8	8	0.702	0.724	0.136	0.992	0.912	0.538	0.156	32933	6422.829	2018.635	771.526	305.261	1934.382	1382.08	10.945	665	26'510.17
10	40	250	8	4	4	0.8	0.985	0.187	0.442	0.924	0.803	3.309	22707	5307.644	1372.791	21.184	209.477	910.332	1232.042	1561.818	942	17'399.36
10	40	250	8	4	8	0.529	0.982	0.372	0.705	0.883	0.345	1.398	44219	6707.101	2736.83	50.965	415.971	703.335	1974.545	825.455	386	37'511.90
10	40	250	8	8	4	0.8	0.985	0.094	0.443	0.924	0.805	3.316	22810	6427.084	1372.518	21.184	210.222	2029.587	1235.348	1558.225	939	16'382.92
10	40	250	8	8	8	0.506	0.982	0.188	0.718	0.88	0.33	1.366	44774	7828.792	2738.711	50.965	419.417	1819.699	2010.611	789.389	388	36'945.21
10	40	250	8	8	16	0.229	0.66	0.254	0.896	0.726	0.14	0.154	62347	10640	3697.216	1902.784	569.615	1670.385	2508.836	291.164	54	51'707.00
10	40	250	8	16	8	0.506	0.982	0.094	0.718	0.88	0.33	1.366	44774	10068.792	2738.711	50.965	419.417	4059.699	2010.611	789.389	388	34'705.21
10	40	250	8	16	16	0.229	0.66	0.127	0.896	0.726	0.14	0.154	62347	12880	3697.216	1902.784	569.615	3910.385	2508.836	291.164	54	49'467.00
10	40	250	16	8	8	0.498	0.981	0.182	0.354	0.878	0.326	1.454	44573	10628.063	2735.441	53.102	408.076	1831.445	1981.347	3618.653	386	33'944.94
10	40	250	2	2	4	0.832	0.733	0.275	0.995	0.929	1.004	0.165	16671	2651.407	1021.579	374.132	153.995	405.748	692.167	3.786	1057	14'019.59
10	40	250	16	8	16	0.018	0.667	0.25	0.451	0.547	0.056	0.248	64415	13440	3737.747	1862.253	560.608	1679.392	2527.973	3072.027	1	50'975.00
10	40	250	16	16	8	0.498	0.981	0.091	0.354	0.878	0.326	1.454	44573	12868.063	2735.441	53.102	408.076	4071.445	1981.347	3618.653	386	31'704.94
10	40	250	16	16	16	0.025	0.67	0.125	0.453	0.552	0.058	0.253	64554	15680	3749.532	1850.468	562.161	3917.839	2537.257	3062.743	1	48'874.00
10	40	250	2	4	2	0.835	0.985	0.095	0.995	0.925	1.201	2.295	11197	2513.15	686.107	10.387	105.808	1013.954	693.107	3.786	1177	8'683.85
10	40	250	2	4	4	0.836	0.724	0.136	0.995	0.93	1.012	0.163	16728	3212.711	1010.653	385.835	151.955	967.456	693.026	3.786	1071	13'515.29
10	40	250	4	2	2	0.839	0.986	0.183	0.567	0.927	1.236	4.746	11081	2654.755	687.673	9.733	102.508	457.342	790.607	606.893	1187	8'426.25
10	40	250	4	2	4	0.802	0.985	0.38	0.885	0.924	0.791	2.236	22722	3349.619	1373.785	21.292	212.701	346.932	1233.306	161.604	895	19'372.38
10	40	250	4	4	2	0.839	0.986	0.092	0.569	0.927	1.234	4.745	11091	3214.94	687.642	9.733	103.247	1016.548	794.053	603.716	1187	7'876.06
10	40	250	4	4	4	0.81	0.985	0.188	0.881	0.923	0.804	2.321	22453	3910.353	1373.838	21.292	210.293	909.534	1228.231	167.166	923	18'542.65
10	40	250	4	4	8	0.692	0.724	0.276	0.992	0.912	0.532	0.155	33161	5304.414	2021.052	771.409	308.917	810.599	1381.493	10.945	666	27'856.59
10	40	250	8	4	16	0.189	0.649	0.5	0.883	0.708	0.122	0.154	61572	9520	3633.093	1966.907	560.193	559.807	2473.347	326.653	45	52'052.00
10	40	250	4	2	8	0.715	0.719	0.549	0.992	0.915	0.553	0.16	32519	4745.264	2006.233	785.972	307.007	252.634	1382.474	10.945	681	27'773.74

Szenario: Einfluss des Service bei 4 Tellerbringer und 16 Köchen

Controls					Responses																Excel	
Replication	Bar	Table	Service	Dish Service	Cook	Bar.Utilization	Cook.Utilization	DishService.Utilization	Service.Utilization	Table.Utilization	Record At Table	Record Wait For Dish	Record Sales	Resources.TotalCost	Cook.BusyCost	Cook.IdleCost	DishService.BusyCost	DishService.IdleCost	Service.BusyCost	Service.IdleCost	Record Overflow	Sales - Cost
100	40	250	8	4	16	0.204	0.65	0.496	0.882	0.704	0.131	0.155	61415	9519.928	3640.973	1958.961	555.664	564.33	2470.177	329.823	56	51'895.07
100	40	250	9	4	16	0.108	0.661	0.504	0.798	0.634	0.089	0.157	63904	9870	3702.626	1897.374	564.422	555.578	2512.195	637.805	13	54'034.00
100	40	250	10	4	16	0.051	0.665	0.508	0.721	0.579	0.067	0.164	64315	10220	3726.092	1873.908	568.883	551.117	2524.806	975.194	3	54'095.00
100	40	250	11	4	16	0.041	0.674	0.514	0.664	0.564	0.064	0.191	65170	10570	3772.198	1827.802	575.123	544.877	2557.215	1292.785	1	54'600.00
100	40	250	12	4	16	0.034	0.677	0.515	0.609	0.56	0.061	0.231	65058	10920	3790.375	1809.625	576.498	543.502	2556.502	1643.498	1	54'138.00
100	40	250	13	4	16	0.034	0.673	0.511	0.56	0.555	0.061	0.244	64898	11270	3768.917	1831.083	572.564	547.436	2547.037	2002.963	1	53'628.00
100	40	250	14	4	16	0.032	0.675	0.513	0.521	0.557	0.061	0.253	65024	11620	3777.68	1822.32	574.954	545.046	2554.297	2345.703	2	53'404.00

Szenario: Einfluss der Tellerbringer bei 11 Servicekräfen und 16 Köchen

Controls					Responses																Excel	
Replication	Bar	Table	Service	Dish Service	Cook	Bar.Utilization	Cook.Utilization	DishService.Utilization	Service.Utilization	Table.Utilization	Record At Table	Record Wait For Dish	Record Sales	Resources.TotalCost	Cook.BusyCost	Cook.IdleCost	DishService.BusyCost	DishService.IdleCost	Service.BusyCost	Service.IdleCost	Record Overflow	Sales - Cost
100	40	250	11	2	16	0.301	0.631	0.953	0.62	0.773	0.179	0.783	57705	10009.521	3532.638	2067.362	533.416	26.105	2388.361	1461.639	119	47'695.48
100	40	250	11	3	16	0.055	0.671	0.682	0.661	0.575	0.069	0.23	64725	10290	3755.146	1844.854	572.633	267.367	2544.073	1305.927	4	54'435.00
100	40	250	11	4	16	0.041	0.674	0.514	0.664	0.564	0.064	0.191	65170	10570	3772.198	1827.802	575.123	544.877	2557.215	1292.785	1	54'600.00
100	40	250	11	5	16	0.034	0.67	0.41	0.663	0.559	0.061	0.184	64945	10850	3753.701	1846.299	573.584	826.416	2551.643	1298.357	1	54'095.00
100	40	250	11	6	16	0.034	0.671	0.341	0.662	0.559	0.061	0.186	64710	11130	3759.579	1840.421	573.201	1106.799	2546.923	1303.077	2	53'580.00

Szenario: Variation der Ressourcen um +-1

Controls					Responses																Excel	
Replication	Bar	Table	Service	Dish Service	Cook	Bar.Utilization	Cook.Utilization	DishService.Utilization	Service.Utilization	Table.Utilization	Record At Table	Record Wait For Dish	Record Sales	Resources.TotalCost	Cook.BusyCost	Cook.IdleCost	DishService.BusyCost	DishService.IdleCost	Service.BusyCost	Service.IdleCost	Record Overflow	Record Guests	Sales - Cost
100	40	250	10	3	13	0.149	0.817	0.673	0.719	0.663	0.106	0.376	63889	8890	3718.228	831.772	565.015	274.985	2514.909	985.091	26	1751	54'999.00
100	40	250	11	3	13	0.146	0.819	0.675	0.655	0.662	0.104	0.441	64039	9240	3725.603	824.397	566.722	273.278	2520.04	1329.96	25	1754	54'799.00
100	40	250	12	3	13	0.147	0.816	0.67	0.598	0.659	0.105	0.464	63573	9590	3711.244	838.756	562.864	277.136	2509.785	1690.215	28	1745	53'983.00
100	40	250	10	4	13	0.144	0.819	0.506	0.722	0.66	0.103	0.365	63805	9170	3727.106	822.894	567.061	552.939	2526.667	973.333	24	1750	54'635.00
100	40	250	11	4	13	0.137	0.814	0.503	0.653	0.654	0.101	0.427	63650	9520	3701.431	848.569	563.485	556.515	2513.201	1336.799	25	1746	54'130.00
100	40	250	12	4	13	0.128	0.811	0.502	0.596	0.647	0.097	0.445	63333	9870	3692.275	857.725	562.524	557.476	2503.46	1696.54	24	1739	53'463.00
100	40	250	10	5	13	0.133	0.81	0.401	0.714	0.648	0.099	0.356	63271	9450	3683.268	866.732	561.246	838.754	2497.997	1002.003	23	1736	53'821.00
100	40	250	11	5	13	0.152	0.817	0.403	0.654	0.658	0.107	0.431	63687	9800	3719.58	830.42	563.761	836.239	2517.822	1332.178	28	1748	53'887.00
100	40	250	12	5	13	0.139	0.814	0.403	0.597	0.651	0.102	0.449	63513	10150	3703.59	846.41	563.711	836.289	2508.745	1691.255	25	1743	53'363.00
100	40	250	10	3	14	0.102	0.764	0.678	0.723	0.625	0.086	0.265	64395	9240	3744.931	1155.069	569.882	270.118	2531.398	968.602	12	1768	55'155.00
100	40	250	11	3	14	0.082	0.757	0.672	0.652	0.61	0.079	0.332	64009	9590	3709.163	1190.837	564.859	275.141	2511.564	1338.436	8	1754	54'419.00
100	40	250	12	3	14	0.093	0.76	0.677	0.602	0.617	0.083	0.368	64232	9940	3723.354	1176.646	568.577	271.423	2526.364	1673.636	12	1763	54'292.00
100	40	250	10	4	14	0.088	0.764	0.508	0.725	0.619	0.081	0.246	64558	9520	3745.841	1154.159	569.227	550.773	2536.857	963.143	7	1769	55'038.00
100	40	250	11	4	14	0.085	0.765	0.509	0.658	0.615	0.08	0.325	64566	9870	3746.145	1153.855	569.828	550.172	2534.816	1315.184	12	1771	54'696.00
100	40	250	12	4	14	0.088	0.761	0.508	0.601	0.611	0.081	0.355	64138	10220	3727.253	1172.747	568.869	551.131	2525.659	1674.341	10	1762	53'918.00
100	40	250	10	5	14	0.088	0.765	0.409	0.726	0.619	0.081	0.245	64569	9800	3748.061	1151.939	572.102	827.898	2539.467	960.533	9	1770	54'769.00
100	40	250	11	5	14	0.092	0.764	0.407	0.657	0.612	0.082	0.324	64395	10150	3741.926	1158.074	570.343	829.657	2530.883	1319.117	10	1765	54'245.00
100	40	250	12	5	14	0.096	0.765	0.408	0.604	0.617	0.084	0.361	64527	10500	3746.45	1153.55	571.092	828.908	2535.424	1664.576	12	1771	54'027.00
100	40	250	10	3	15	0.07	0.712	0.677	0.723	0.598	0.074	0.201	64472	9590	3736.793	1513.207	568.999	271.001	2530.872	969.128	6	1766	54'882.00
100	40	250	11	3	15	0.061	0.712	0.68	0.658	0.588	0.071	0.269	64490	9940	3739.98	1510.02	570.928	269.072	2533.9	1316.1	6	1770	54'550.00
100	40	250	12	3	15	0.066	0.716	0.682	0.606	0.594	0.073	0.309	64839	10290	3757.513	1492.487	573.149	266.851	2544.521	1655.479	6	1778	54'549.00
100	40	250	10	4	15	0.07	0.711	0.507	0.722	0.592	0.074	0.182	64343	9870	3730.401	1519.599	568.394	551.606	2526.868	973.132	5	1763	54'473.00
100	40	250	11	4	15	0.049	0.712	0.509	0.657	0.576	0.066	0.242	64457	10220	3736.172	1513.828	569.895	550.105	2529.628	1320.372	3	1767	54'237.00
100	40	250	12	4	15	0.049	0.717	0.514	0.608	0.58	0.066	0.279	65040	10570	3762.299	1487.701	575.141	544.859	2552.987	1647.013	2	1783	54'470.00
100	40	250	10	5	15	0.069	0.717	0.409	0.727	0.596	0.074	0.182	64623	10150	3766.14	1483.86	572.066	827.934	2542.78	957.22	7	1776	54'473.00
100	40	250	11	5	15	0.051	0.715	0.41	0.662	0.579	0.067	0.238	64778	10500	3753.929	1496.071	573.924	826.076	2546.885	1303.115	3	1777	54'278.00
100	40	250	12	5	15	0.054	0.717	0.409	0.606	0.58	0.068	0.281	64699	10850	3765.314	1484.686	572.677	827.323	2545.146	1654.854	4	1776	53'849.00

B. Arena Report

14-3-9/category overview.pdf

14:16:25 **Category Overview** Mai 24, 2013

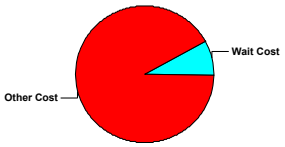
Values Across All Replications

HardRock

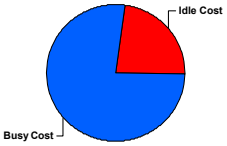
Replications: 100 Time Units: Hours

Key Performance Indicators

All Entities	Average
Non-Value Added Cost	0
Other Cost	6,261
Transfer Cost	0
Value Added Cost	0
Wait Cost	569
Total Cost	6,830



All Resources	Average
Busy Cost	6,830 *
Idle Cost	2,060
Usage Cost	0
Total Cost	8,890



* *these costs are included in Entity Costs above.*

System	Average
Total Cost	8,890
Number Out	767

14-3-9/category overview.pdf

14:16:25 **Category Overview** Mai 24, 2013

Values Across All Replications

HardRock

Replications: 100 Time Units: Hours

Entity

Time

VA Time	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Client	0.04924605	0.00	0.04714677	0.05115268	0.00	0.08333329

NVA Time	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Client	0.00	0.00	0.00	0.00	0.00	0.00

Wait Time	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Client	1.2597	0.01	1.1240	1.3606	0.00	2.8863

Transfer Time	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Client	0.00	0.00	0.00	0.00	0.00	0.00

Other Time	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Client	0.3745	0.00	0.3581	0.3903	0.00	0.8939

Total Time	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Client	1.3708	0.01	1.2436	1.4726	0.00	2.4747

Cost

VA Cost	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Client	0.00	0.00	0.00	0.00	0.00	0.00

NVA Cost	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Client	0.00	0.00	0.00	0.00	0.00	0.00

Wait Cost	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Client	0.7388	0.00	0.7022	0.7744	0.00	1.3333

Other Cost	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Client	8.1283	0.03	7.7666	8.5262	0.00	21.0833

Transfer Cost	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Client	0.00	0.00	0.00	0.00	0.00	0.00

Model Filename: Z:\Temp\BA\Models\Arena\GitHub\Model12 Page 2 of 8

14-3-9/category overview.pdf

14:16:25 **Category Overview** Mai 24, 2013

Values Across All Replications

HardRock

Replications: 100 Time Units: Hours

Entity

Cost

Total Cost	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Client	8.8671	0.03	8.4829	9.2663	0.00	22.2700

Other

Number In	Average	Half Width	Minimum Average	Maximum Average
Client	3299.05	20.80	3047.00	3526.00

Number Out	Average	Half Width	Minimum Average	Maximum Average
Client	3285.12	20.23	3030.00	3509.00

WIP	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Client	225.87	1.88	203.45	245.45	0.00	362.00

Model Filename: Z:\Temp\BA\Models\Arena\GitHub\Model12 Page 3 of 8

14-3-9/category overview.pdf

14:16:25 **Category Overview** Mai 24, 2013

Values Across All Replications

HardRock

Replications: 100 Time Units: Hours

Queue

Time

Waiting Time	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
CookBatch.Queue	0.06361087	0.00	0.05963287	0.06954629	0.00	0.3872
Seize Bar.Queue	0.00818974	0.00	0.00	0.02193006	0.00	0.3476
Seize Cook.Queue	0.01766144	0.00	0.00785642	0.03994014	0.00	0.1860
Seize Dish Service.Queue	0.01080312	0.00	0.00734090	0.01454572	0.00	0.1548
Seize Service.Queue	0.2839	0.01	0.1906	0.3702	0.00	0.7749
Seize Table.Queue	0.04511126	0.00	0.00020227	0.08380691	0.00	0.4443

Cost

Waiting Cost	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
CookBatch.Queue	0.00	0.00	0.00	0.00	0.00	0.00
Seize Bar.Queue	0.00	0.00	0.00	0.00	0.00	0.00
Seize Cook.Queue	0.00	0.00	0.00	0.00	0.00	0.00
Seize Dish Service.Queue	0.00	0.00	0.00	0.00	0.00	0.00
Seize Service.Queue	0.00	0.00	0.00	0.00	0.00	0.00
Seize Table.Queue	0.00	0.00	0.00	0.00	0.00	0.00

Other

Number Waiting	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
CookBatch.Queue	11.4937	0.12	10.0962	13.0374	0.00	32.0000
Enter Cafe.Queue	0.00	0.00	0.00	0.00	0.00	0.00
Seize Bar.Queue	0.4481	0.07	0.00	1.2281	0.00	10.0000
Seize Cook.Queue	2.2388	0.16	0.9562	4.9126	0.00	31.0000
Seize Dish Service.Queue	0.5858	0.02	0.3990	0.7979	0.00	11.0000
Seize Service.Queue	15.4264	0.47	9.8045	20.8618	0.00	52.0000
Seize Table.Queue	2.4645	0.23	0.01040224	4.6932	0.00	21.0000

Model Filename: Z:\Temp\BA\Models\Arena\GitHub\Model12 Page 4 of 8

14-3-9/category overview.pdf

14:16:25 **Category Overview** Mai 24, 2013

Values Across All Replications

HardRock

Replications: 100 Time Units: Hours

Resource

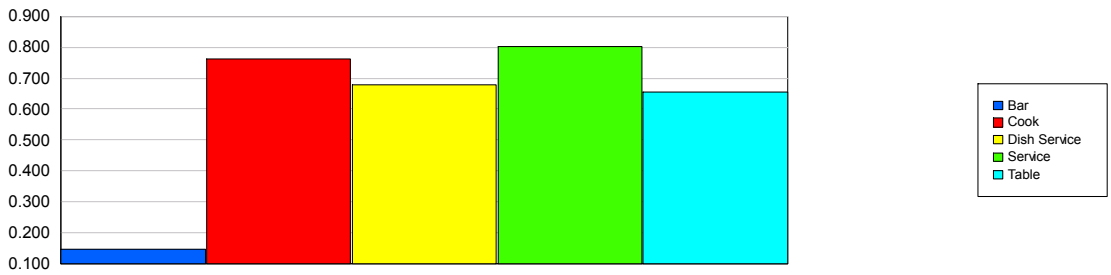
Usage

Instantaneous Utilization	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Bar	0.1446	0.01	0.00052092	0.2839	0.00	1.0000
Cook	0.7613	0.01	0.6862	0.8161	0.00	1.0000
Dish Service	0.6772	0.00	0.6169	0.7245	0.00	1.0000
Service	0.8033	0.00	0.7420	0.8591	0.00	1.0000
Table	0.6550	0.01	0.5715	0.7250	0.00	1.0000

Number Busy	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Bar	5.7848	0.53	0.02083689	11.3560	0.00	40.0000
Cook	10.6587	0.07	9.6070	11.4254	0.00	14.0000
Dish Service	2.0317	0.01	1.8506	2.1736	0.00	3.0000
Service	7.2293	0.04	6.6776	7.7316	0.00	9.0000
Table	163.75	1.76	142.88	181.26	0.00	250.00

Number Scheduled	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Bar	40.0000	0.00	40.0000	40.0000	40.0000	40.0000
Cook	14.0000	0.00	14.0000	14.0000	14.0000	14.0000
Dish Service	3.0000	0.00	3.0000	3.0000	3.0000	3.0000
Service	9.0000	0.00	9.0000	9.0000	9.0000	9.0000
Table	250.00	0.00	250.00	250.00	250.00	250.00

Scheduled Utilization	Average	Half Width	Minimum Average	Maximum Average
Bar	0.1446	0.01	0.00052092	0.2839
Cook	0.7613	0.01	0.6862	0.8161
Dish Service	0.6772	0.00	0.6169	0.7245
Service	0.8033	0.00	0.7420	0.8591
Table	0.6550	0.01	0.5715	0.7250



Model Filename: Z:\Temp\BA\Models\Arena\GitHub\Model12 Page 5 of 8

14-3-9/category overview.pdf

14:16:25 **Category Overview** Mai 24, 2013

Values Across All Replications

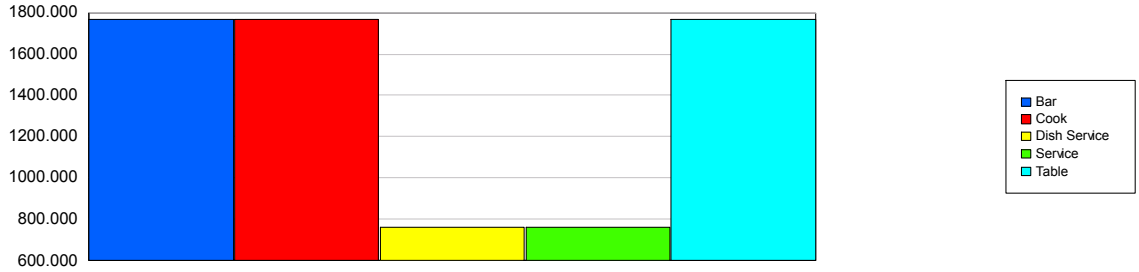
HardRock

Replications: 100 Time Units: Hours

Resource

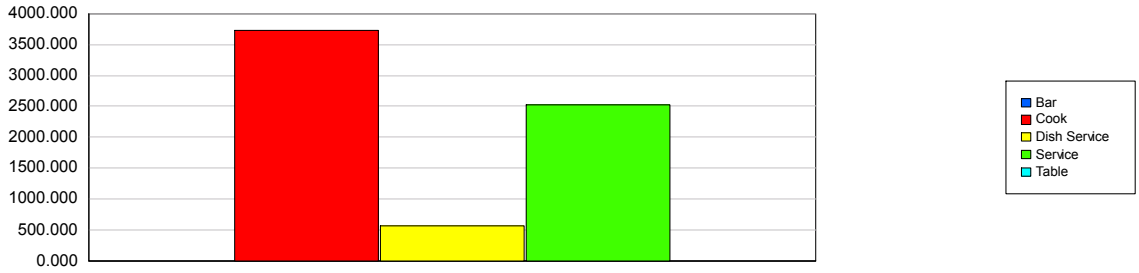
Usage

Total Number Seized	Average	Half Width	Minimum Average	Maximum Average
Bar	1770.17	11.16	1635.00	1861.00
Cook	1770.17	11.16	1635.00	1861.00
Dish Service	758.74	4.53	700.00	816.00
Service	758.74	4.53	700.00	816.00
Table	1770.17	11.16	1635.00	1861.00



Cost

Busy Cost	Average	Half Width	Minimum Average	Maximum Average
Bar	0.00	0.00	0.00	0.00
Cook	3730.54	25.52	3362.45	3998.87
Dish Service	568.87	3.50	518.18	608.61
Service	2530.27	15.34	2337.17	2706.05
Table	0.00	0.00	0.00	0.00



Model Filename: Z:\Temp\BA\Models\Arena\GitHub\Model12 Page 6 of 8

14-3-9/category overview.pdf

14:16:25 **Category Overview** Mai 24, 2013

Values Across All Replications

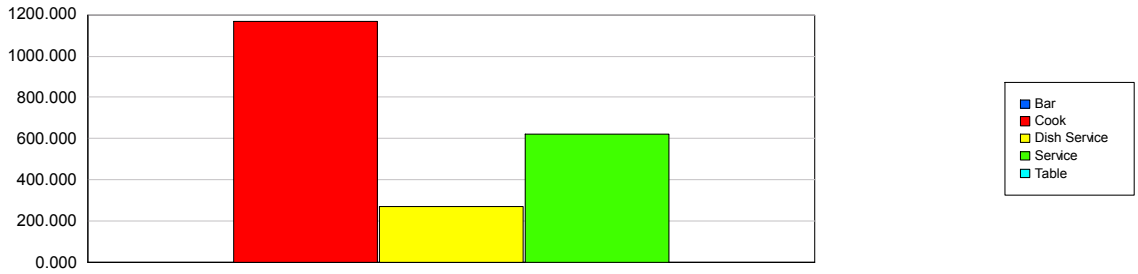
HardRock

Replications: 100 Time Units: Hours

Resource

Cost

Idle Cost	Average	Half Width	Minimum Average	Maximum Average
Bar	0.00	0.00	0.00	0.00
Cook	1169.46	25.52	901.13	1537.55
Dish Service	271.13	3.50	231.39	321.82
Service	619.73	15.34	443.95	812.83
Table	0.00	0.00	0.00	0.00



Usage Cost	Average	Half Width	Minimum Average	Maximum Average
Bar	0.00	0.00	0.00	0.00
Cook	0.00	0.00	0.00	0.00
Dish Service	0.00	0.00	0.00	0.00
Service	0.00	0.00	0.00	0.00
Table	0.00	0.00	0.00	0.00

Model Filename: Z:\Temp\BA\Models\Arena\GitHub\Model12 Page 7 of 8

14-3-9/category overview.pdf

14:16:25 **Category Overview** Mai 24, 2013

Values Across All Replications

HardRock

Replications: 100 Time Units: Hours

User Specified

Tally

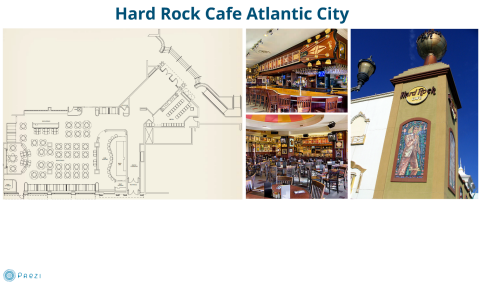
Interval	Average	Half Width	Minimum Average	Maximum Average	Minimum Value	Maximum Value
Record At Table	0.1035	0.01	0.05106868	0.1563	0.01667029	0.6205
Record eating time	0.6393	0.00	0.6270	0.6510	0.2508	0.9973
Record Ordering	0.1334	0.00	0.1315	0.1353	0.06749797	0.1992
Record Total	1.3914	0.01	1.2436	1.5073	0.5349	2.4747
Record Wait For Dish	0.1808	0.00	0.1674	0.1985	0.00893856	0.4561

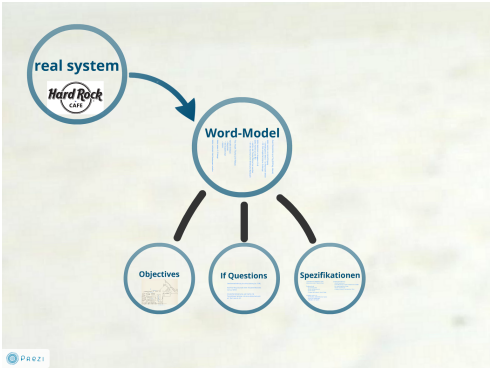
Counter

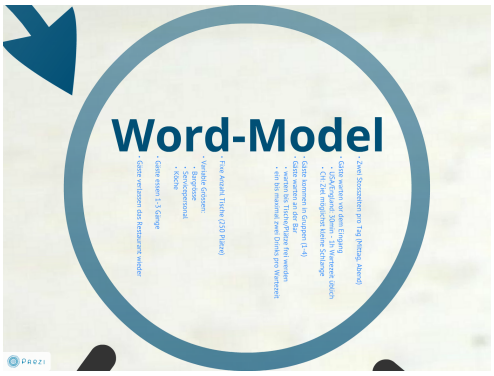
Count	Average	Half Width	Minimum Average	Maximum Average
Record Guests	1762.46	10.89	1626.00	1851.00
Record Overflow	26.8000	5.74	0.00	97.0000
Record Sales	64171.42	409.80	59433.00	68040.00

Model Filename: Z:\Temp\BA\Models\Arena\GitHub\Model12 Page 8 of 8

C. Präsentation Projektvorstellung



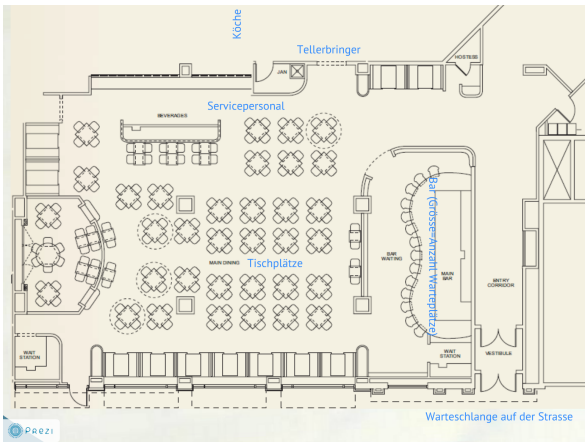




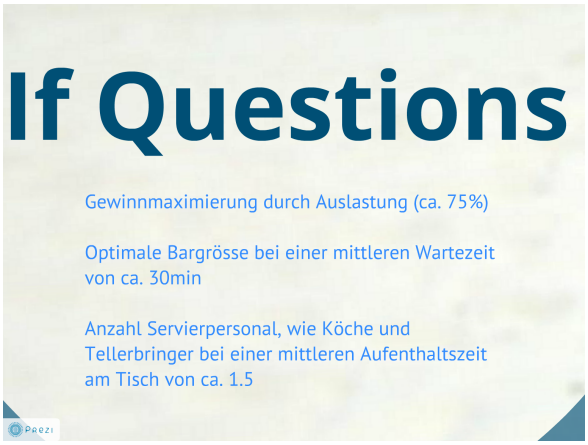
-Model
• Zwei Stosszeiten pro Tag (Mittag, Abend)
• Gäste warten vor dem Eingang
• USA/England: 30min - 1h Wartezeit üblich
• CH: Ziel möglichst kleine Schlange
• Gäste kommen in Gruppen (1-4)
• Gäste warten an der Bar
• warten bis Tische/Plätze frei werden
• ein bis maximal zwei Drinks pro Wartezeit

Word-
• Fixe Anzahl Tische (250 Plätze)
• Variable Grössen:
• Bargrösse
• Servicepersonal
• Köche
• Gäste essen 1-3 Gänge
• Gäste verlassen das Restaurant wieder

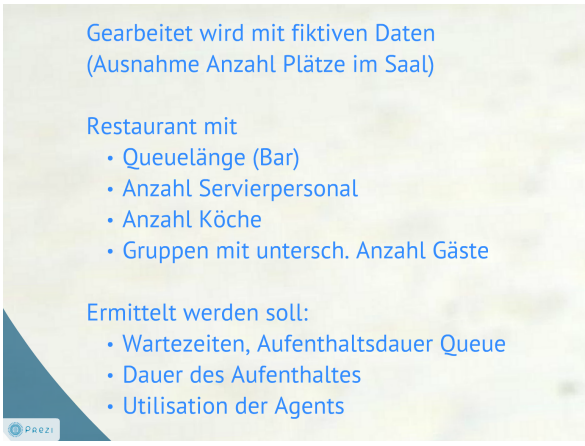
Objectives

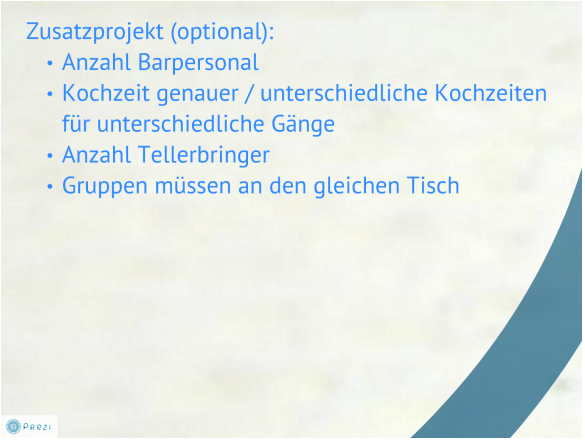

Köche
Tellerbringer
Servicepersonal
Tischplätze
Bargrösse/Anzahl Wartepläze
Warteschlange auf der Strasse

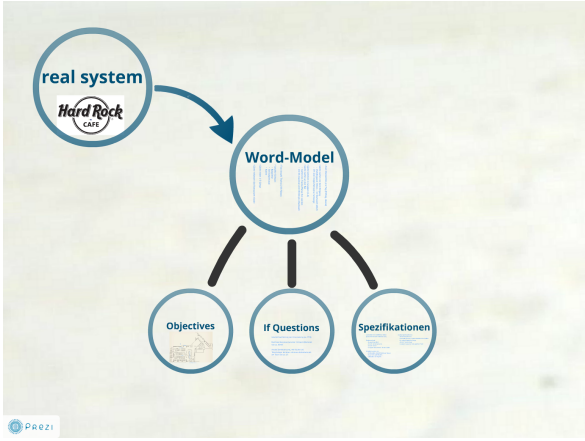
If Questions
Gewinnmaximierung durch Auslastung (ca. 75%)
Optimale Bargrösse bei einer mittleren Wartezeit von ca. 30min
Anzahl Servierpersonal, wie Köche und Tellerbringer bei einer mittleren Aufenthaltszeit am Tisch von ca. 1.5


If Questions
Gewinnmaximierung durch Auslastung (ca. 75%)
Optimale Bargrösse bei einer mittleren Wartezeit von ca. 30min
Anzahl Servierpersonal, wie Köche und Tellerbringer bei einer mittleren Aufenthaltszeit am Tisch von ca. 1.5

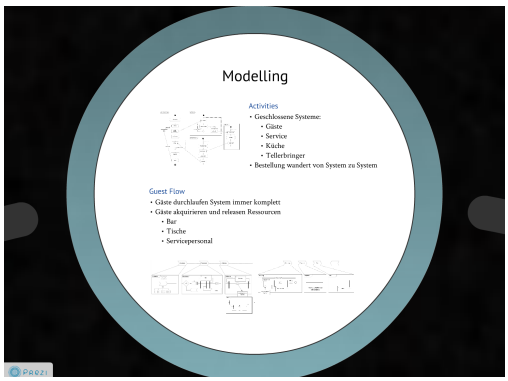
Spezifikationen
Gearbeitet wird mit fiktiven Daten
(Ausnahme Anzahl Plätze im Saal)
Restaurant mit
• Queuelänge (Bar)
• Anzahl Servierpersonal
• Anzahl Köche
• Gruppen mit untersch. Anzahl Gäste
Ermittelt werden soll:
• Wartezeiten, Aufenthaltsdauer Queue
• Dauer des Aufenthaltes
• Utilisation der Agents
Zusatzprojekt (optional):
• Anzahl Barpersonal
• Kochzeit genauer / unterschiedliche Kochzeiten
für unterschiedliche Gänge
• Anzahl Tellerbringer
• Gruppen müssen an den gleichen Tisch
Prezi

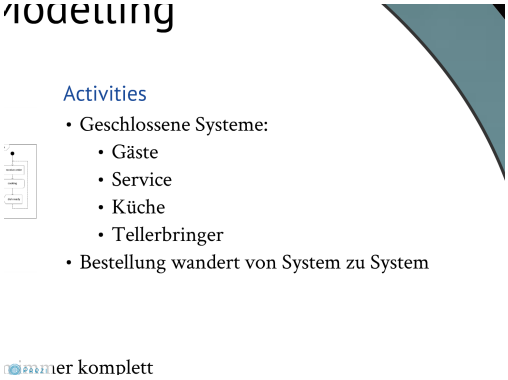

Gearbeitet wird mit fiktiven Daten
(Ausnahme Anzahl Plätze im Saal)
Restaurant mit
• Queuelänge (Bar)
• Anzahl Servierpersonal
• Anzahl Köche
• Gruppen mit untersch. Anzahl Gäste
Ermittelt werden soll:
• Wartezeiten, Aufenthaltsdauer Queue
• Dauer des Aufenthaltes
• Utilisation der Agents
Prezi

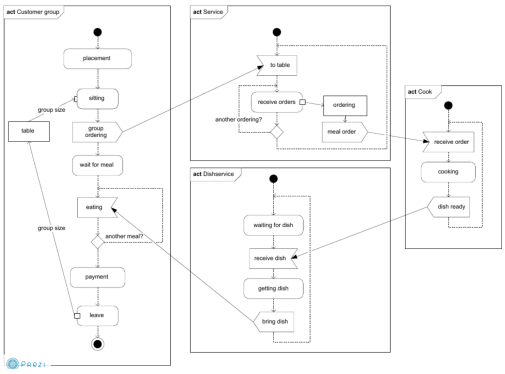

Zusatzprojekt (optional):
• Anzahl Barpersonal
• Kochzeit genauer / unterschiedliche Kochzeiten
für unterschiedliche Gänge
• Anzahl Tellerbringer
• Gruppen müssen an den gleichen Tisch
Prezi

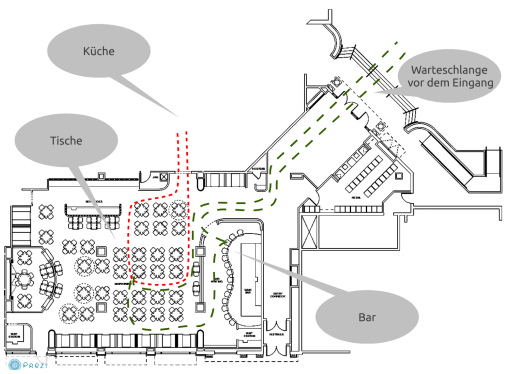

real system
Hard Rock
CAFE
Word-Model
Objectives
If Questions
Spezifikationen
Prezi

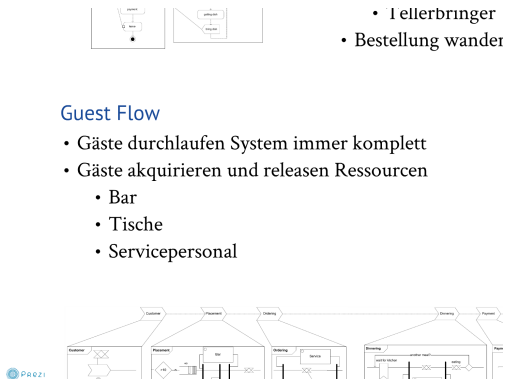
D. Abschlusspräsentation Projekt

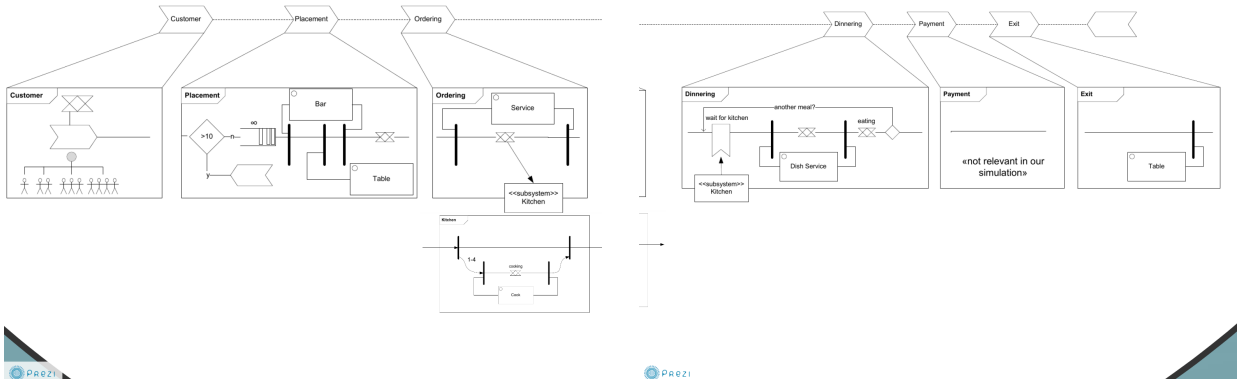


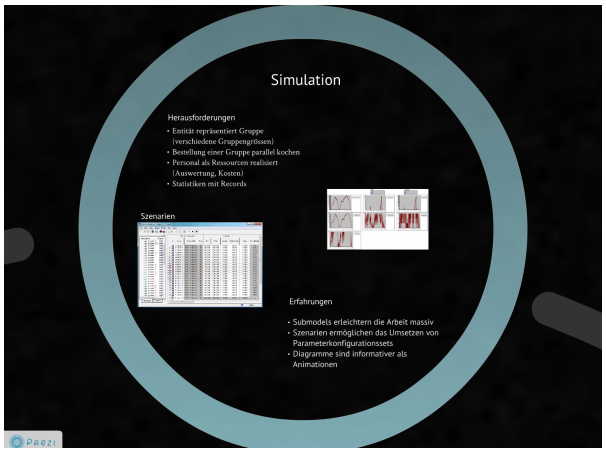






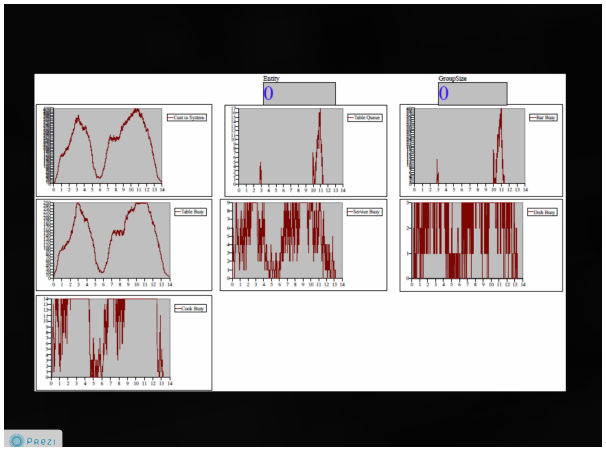




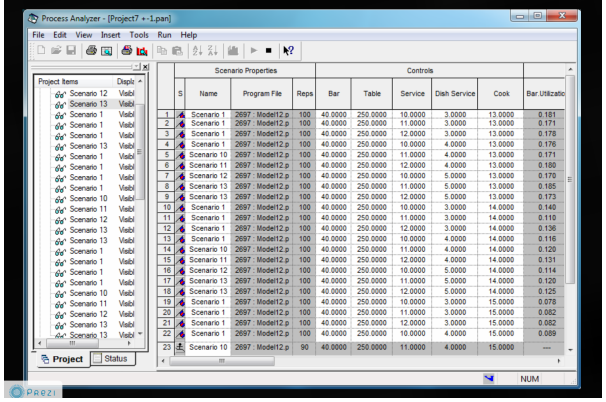


Herausforderungen

- Entität repräsentiert Gruppe (verschiedene Gruppengrössen)
- Bestellung einer Gruppe parallel kochen
- Personal als Ressourcen realisiert (Auswertung, Kosten)
- Statistiken mit Records



Szenarien



Erfahrungen

- Submodels erleichtern die Arbeit massiv
- Szenarien ermöglichen das Umsetzen von Parameterkonfigurationssets
- Diagramme sind informativer als Animationen

Objectives

Ziele der Simulation

- Gewinnoptimierung: vereinfacht als Umsatz-Kosten
- Maximieren des Gäste-Durchsatzes: Auslastung aller Ressourcen so tief, dass im System kein Stau entsteht

Auslastung aller Ressourcen so hoch, dass keine Ressourcen überflüssig.

What if...

- Gegenseitige Abhängigkeit der Ressourcen: Festhalten der Anzahl einer Ressource bei Variation der anderen:
  - z.B.: Fixe Anzahl Köche bei vielen Tellerbringern und wenig Servicepersonal
- Suchen der optimalen Verhältnisses.

Ziele der Simulation

- Gewinnoptimierung: vereinfacht als Umsatz-Kosten
- Maximieren des Gäste-Durchsatzes: Auslastung aller Ressourcen so tief, dass im System kein Stau entsteht

Auslastung aller Ressourcen so hoch, dass keine Ressourcen überflüssig.

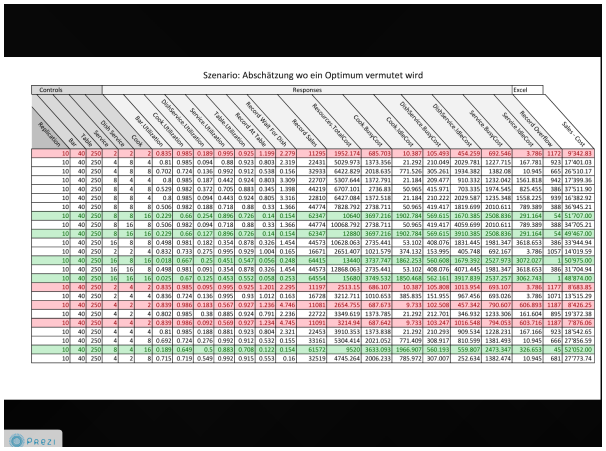
What if...

- Gegenseitige Abhängigkeit der Ressourcen Festhalten der Anzahl einer Ressource bei Variation der anderen:
  - z.B.: Fixe Anzahl Köche bei vielen Tellerbringern und wenig Servicepersonal
- Suchen der optimalen Verhältnisses.

Results

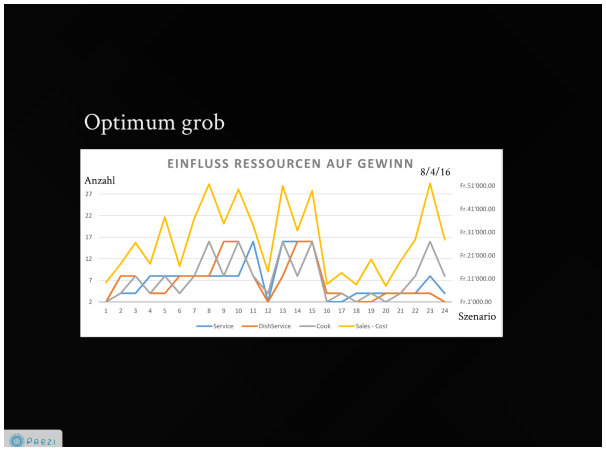
Szenarien und Optimierungen

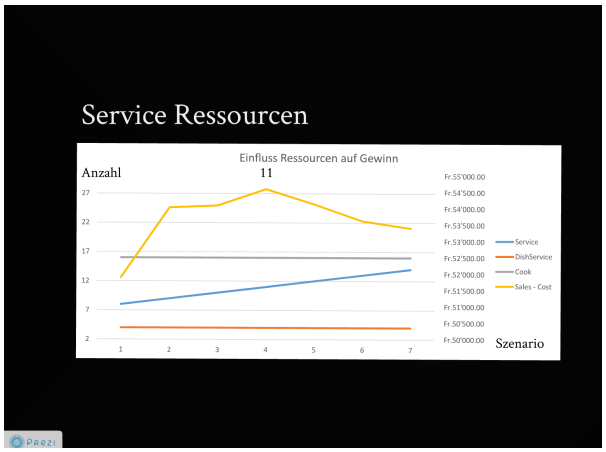
- Verschiedene Szenarien
- Process Analyzer
- Kosten einführen bei Ressourcen
- Zeitstempel für Waitingtimes
- OptQuest

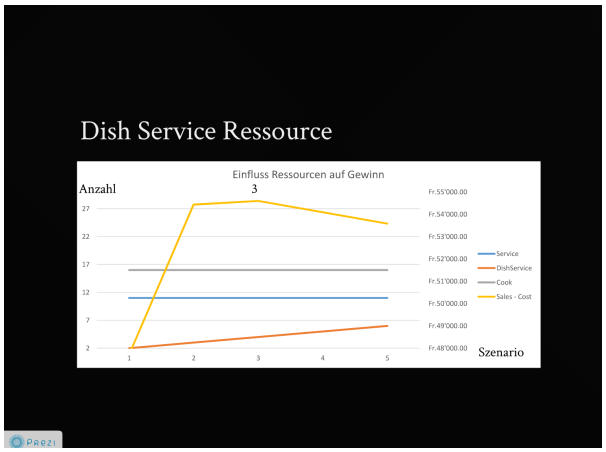


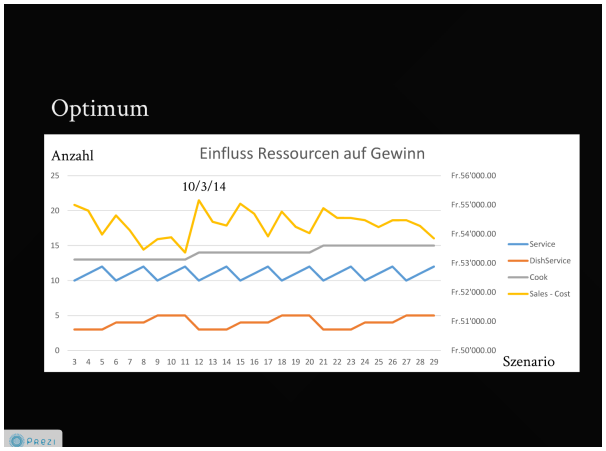
Szenario: Abschätzung wo ein Optimum vermutet wird

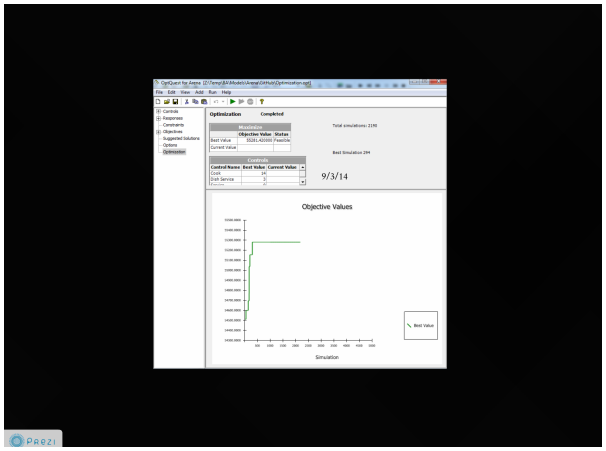
Controls						Responses																Excel	
Replications	Dish Service	Cook	Dish Service	Cook	Sales Price	Service Utilization	DishService Utilization	Cook Utilization	Bar Utilization	Service Utilization	DishService Utilization	Cook Utilization	Bar Utilization	Revenue Wait for Food	Revenue Leave	Revenue Total	Cost Service	Cost Dish	DishService Wages	Cook Wages	Service Wages	Revenue Operating	Sales - Cost
10	40	250	2	2	2	0.835	0.985	0.189	0.995	0.925	1.199	2.270	11295	1952.174	685.703	10.387	105.493	454.259	693.540	3.786	1177	9'342.84
10	40	250	4	8	4	0.81	0.985	0.094	0.88	0.923	0.803	2.319	22431	5029.973	1373.356	21.292	210.049	2029.781	1227.715	167.781	923	17'401.03
10	40	250	4	8	8	0.702	0.724	0.136	0.992	0.912	0.538	0.156	32933	6422.829	2018.635	771.526	305.261	1934.382	1382.08	10.945	665	26'510.17
10	40	250	8	4	4	0.8	0.985	0.187	0.442	0.924	0.803	3.309	22707	5307.644	1372.791	21.184	209.477	910.332	1232.042	1561.818	942	17'399.36
10	40	250	8	4	8	0.529	0.982	0.372	0.705	0.883	0.345	1.398	44219	6707.101	2736.83	50.965	415.971	703.335	1974.545	825.455	386	37'511.90
10	40	250	8	8	4	0.8	0.985	0.094	0.443	0.924	0.805	3.316	22810	6427.084	1372.518	21.184	210.222	2029.587	1235.348	1558.225	939	16'382.92
10	40	250	8	8	8	0.506	0.982	0.188	0.718	0.88	0.33	1.366	44774	7828.792	2738.711	50.965	419.417	1819.699	2010.611	789.389	388	36'945.21
10	40	250	8	8	16	0.229	0.66	0.254	0.896	0.726	0.14	0.154	62347	10640	3697.216	1902.784	569.615	1670.385	2508.836	291.164	54	51'707.00
10	40	250	8	16	8	0.506	0.982	0.094	0.718	0.88	0.33	1.366	44774	10068.792	2738.711	50.965	419.417	4059.699	2010.611	789.389	388	34'705.21
10	40	250	8	16	16	0.229	0.66	0.127	0.896	0.726	0.14	0.154	62347	12980	3697.216	1902.784	569.615	3910.385	2508.836	291.164	54	49'467.00
10	40	250	16	8	8	0.498	0.981	0.182	0.354	0.878	0.326	1.454	44573	10628.063	2735.441	53.102	408.076	1831.445	1981.347	3618.653	386	33'944.94
10	40	250	2	2	4	0.832	0.733	0.275	0.995	0.929	1.004	0.165	16671	2651.407	1021.579	374.132	153.995	405.748	692.167	3.786	1057	14'019.59
10	40	250	16	8	16	0.018	0.667	0.25	0.451	0.547	0.056	0.246	64415	13440	2737.747	1862.253	560.608	1679.392	2527.973	3072.027	3	50'975.00
10	40	250	16	16	8	0.498	0.981	0.091	0.354	0.878	0.326	1.454	44573	12868.063	2735.441	53.102	408.076	4071.445	1981.347	3618.653	386	31'704.94
10	40	250	16	16	16	0.025	0.67	0.125	0.453	0.552	0.058	0.253	64554	15680	3749.532	1850.468	562.161	3917.839	2537.257	3062.743	3	48'874.00
10	40	250	2	4	2	0.835	0.985	0.095	0.995	0.925	1.201	2.295	11197	2513.15	686.107	10.387	105.808	1013.954	693.107	3.786	1177	8'683.85
10	40	250	2	4	4	0.836	0.724	0.136	0.995	0.93	1.012	0.163	16728	3212.711	1010.653	385.835	151.955	967.456	693.026	3.786	1071	13'515.29
10	40	250	4	2	2	0.839	0.986	0.183	0.567	0.927	1.236	4.746	11081	2654.755	687.673	9.733	102.508	457.342	790.607	606.893	1187	8'426.25
10	40	250	4	2	4	0.802	0.985	0.38	0.885	0.924	0.791	2.236	22722	3349.619	1373.785	21.292	212.701	346.932	1233.306	161.604	895	19'372.38
10	40	250	4	4	2	0.839	0.986	0.092	0.569	0.927	1.234	4.745	11093	3214.94	687.642	9.733	103.247	1016.548	794.053	603.716	1183	7'876.06
10	40	250	4	4	4	0.81	0.985	0.188	0.881	0.923	0.804	2.321	22453	3910.353	1373.838	21.292	210.293	909.534	1228.231	167.166	923	18'542.65
10	40	250	4	4	8	0.692	0.724	0.276	0.992	0.912	0.532	0.155	33161	5304.414	2021.052	771.409	308.917	810.599	1381.493	10.945	666	27'856.59
10	40	250	8	4	16	0.189	0.649	0.5	0.860	0.708	0.122	0.154	61572	9520	3631.093	1865.907	560.193	559.807	2473.347	326.653	43	52'053.00
10	40	250	4	2	8	0.715	0.719	0.549	0.992	0.915	0.553	0.16	32519	4745.264	2006.233	785.972	307.007	252.634	1382.474	10.945	681	27'773.74

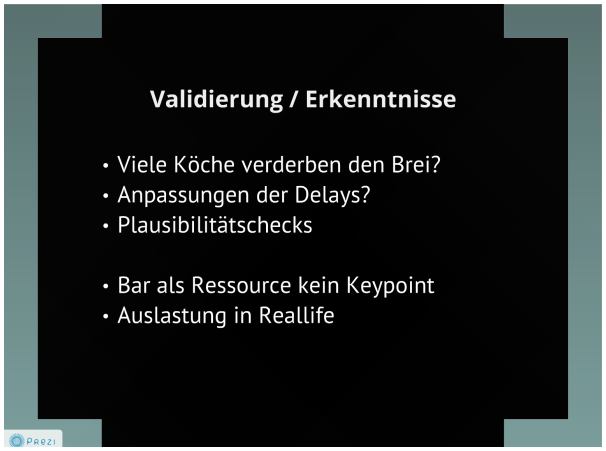


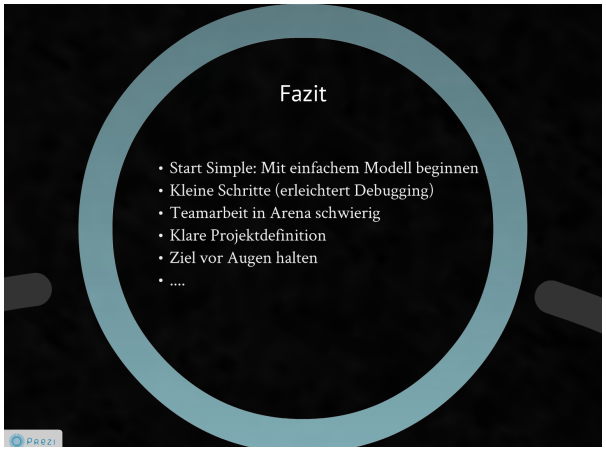









Validierung / Erkenntnisse
• Viele Köche verderben den Brei?
• Anpassungen der Delays?
• Plausibilitätschecks
• Bar als Ressource kein Keypoint
• Auslastung in Reallife


Fazit
• Start Simple: Mit einfachem Modell beginnen
• Kleine Schritte (erleichtert Debugging)
• Teamarbeit in Arena schwierig
• Klare Projektdefinition
• Ziel vor Augen halten
• ....

Abbildungsverzeichnis

1.1. Gebäudeplan Hardrock 5
1.2. Bewegungsschema Hardrock 7

2.1. Activity Diagramm 11
2.2. Hardrock Flow Diagramm Teil1 13
2.3. Hardrock Flow Diagramm Teil2 14
2.4. Hardrock Flow Diagramm Kitchen 15

3.1. Arrival Sceduler in Arena 17
3.2. Simulationsaufbau in Arena 18
3.3. Diagramme in Arena 20

4.1. Grobszenarien 22
4.2. Serviceoptimierungsszenarien 23
4.3. Tellerbringeroptimierungsszenarien 24
4.4. Endoptimierung 25
4.5. OptQuest in Arena 26
4.6. OptQuest Suchverlauf 26

E. Materialnachweise

E.1. Titelblatt

Logo Hardrock	www.hardrock.com	25.03.2013