

Das Evolutionäre Lernspiel-Konzept:

Eine Kombination aus Game-based Learning und Web 2.0

Dissertation

VERFASST AN
DER WIRTSCHAFTSWISSENSCHAFTLICHEN
FAKULTÄT
DER UNIVERSITÄT ZÜRICH

zur Erlangung der Würde
eines Doktors der Informatik

vorgelegt von
TSUYOSHI ITO

von
Zürich

genehmigt auf Antrag von
PROF. DR. H. SCHAUER
PROF. DR. H. GALL

Oktober 2009

Die Wirtschaftswissenschaftliche Fakultät der Universität Zürich, Lehrbereich Informatik, gestattet hierdurch die Drucklegung der vorliegenden Dissertation, ohne damit zu den darin ausgesprochenen Anschauungen Stellung zu nehmen.

Zürich, den 21. Oktober 2009

Der Lehrbereichsvorsteher: Prof. Dr. Harald Gall

Danksagung

Die Entwicklung eines innovativen Konzepts im Bereich Lernspiele findet nicht isoliert statt. Es braucht dazu ein interdisziplinäres Team, welches sich mit verschiedenen Forschungsschwerpunkten aus der Pädagogik, Psychologie und Informatik beschäftigt. Dieses Team wurde unter der Leitung von Prof. Dr. Helmut Schauer am Institut für Informatik über drei Jahre aufgebaut und erarbeitet seither kollaborativ neue Ansätze im Bereich *Game-based Learning* und *E-Assessment*. Dank der engen Kooperation vieler Mitglieder dieser Arbeitsgruppe und deren aufschlussreichen Feedbacks wurde diese Arbeit ermöglicht. Ich möchte mich in erster Linie bei meinem Betreuer Prof. Dr. Helmut Schauer bedanken, welcher mir durch seinen partizipativen Führungsstil die nötige Freiheit liess, eine explorative Arbeit in diesem jungen Forschungsgebiet zu verfassen und mich stets bei der Realisierung meiner Ideen unterstützte. Prof. Dr. Harald Gall hat freundlicherweise das Koreferat übernommen.

Für die vielen Anregungen im Bereich *Game-based Learning* möchte ich mich bei Franziska Spring, sowie bei ihren Betreuern Kurt Squire und Constance Steinkuhler bedanken. Sie haben mir die Grundlagen zu motivierenden Lernspielerlebnissen vermittelt und standen für Fragen in diesem Bereich stets zur Verfügung.

Ein grosses Lob geht an Robert Stoyan, welcher massgeblich das Spieldesign des Prototyps mitbestimmte und die Lerninhalte dazu produzierte. Ausserdem übernahm er die aufwendige und experimentelle Evaluation von *Assessment*-Möglichkeiten, welche wichtige Grundlagen für die Überprüfung von Lernerfolg darstellen.

Die Evaluation des Autorenwerkzeugs, welche Bestandteil dieser Forschungsarbeit war, wurde mit der tatkräftigen Unterstützung von Julian Ebert realisiert, welcher mir sowohl bei Fragen zu Verfahren und Auswertungen wie auch bei der intensiven Phase des Schreibens zur Seite stand.

Für die finanzielle Unterstützung während der letzten zwei Jahre möchte ich mich bei der Hasler Stiftung bedanken.

Des Weiteren bedanke ich mich bei den vielen Studenten für ihre Beiträge zur Verbesserung des Prototyps und zur Aufbereitung der Grundlagen zu dieser Arbeit. Die Entwicklung des Prototyps hat über 3 Jahre in Anspruch genommen und wäre ohne die Unterstützung der Hilfsassistenten Clemens Wilding, Marc Körsgen, Daniel Spicar sowie des Diplomanden Sebastian Burkhart nicht möglich gewesen.

Inhaltsverzeichnis

Tabellenverzeichnis	vii
Abbildungsverzeichnis.....	viii
1. Einleitung.....	1
1.1 Motivation.....	2
1.2 Fragestellung.....	5
2. Grundlagen.....	6
2.1 Web 2.0.....	6
2.1.1 Web 2.0-Prinzipien nach O'Reilly.....	8
2.1.2 Beispiele von Web 2.0-Anwendungen	13
2.1.3 Zusammenfassung.....	29
2.2 Autorenwerkzeuge	31
2.2.1 Anforderungen an Autorenwerkzeuge.....	32
2.2.2 Beispiel: <i>Scratch</i>	39
2.3 Lernplattformen	42
2.3.1 Rollenspezifische Anforderungen an Lernplattformen.....	43
2.3.2 Beispiel: OLAT.....	45
2.4 Digitale Unterhaltungsspiele.....	47
2.4.1 Merkmale von Spielen	48
2.4.2 Geschichte und Entwicklung der Computerspiele.....	50
2.4.3 Genres	52
2.4.4 Beispiele digitaler Unterhaltungsspiele	59
2.4.5 Zusammenfassung.....	67
2.5 Game-based Learning	68
2.5.1 Konstruktivismus im Kontext des <i>Game-based Learning</i>	70
2.5.2 Game Generation	73
2.5.3 Zusammenfassung.....	84
3. Evolutionäres Lernspielkonzept.....	86
3.1 Einleitung.....	86
3.2 Konzept	88
3.3 Komponenten <i>Evolutionärer Lernspiele</i>	93
3.4 Softwarearchitektur <i>Evolutionärer Lernspiele</i>	99
3.5 Zusammenfassung.....	104
4. <i>PM Game</i> : Referenzimplementation des <i>Evolutionären Lernspielkonzepts</i>	105
4.1 Einleitung.....	105
4.2 Komponenten des <i>Evolutionären Lernspiels</i> „ <i>PM Game</i> “	107
4.3 Das Spiel	109
4.4 <i>NPC-Editor</i>	113
4.5 Autorenwerkzeug <i>PEdi</i>	114
4.6 Regeleditor <i>RuEdi</i>	120
4.7 <i>PM Game-Viewer</i>	126
4.8 Verwendete Technologien und Frameworks	126
4.9 Software-Architektur des <i>PM Game</i>	127
4.10 Zusammenfassung.....	131
5. Evaluation des <i>PM Game</i>	133
5.1 Einleitung.....	133
5.2 Akzeptanz <i>Evolutionärer Lernspiele</i>	133

5.2.1	Akzeptanzbegriff.....	134
5.2.2	Akzeptanzmodell: <i>Technology-Acceptance-Model</i>	135
5.3	<i>PEdi</i> -Evaluation.....	137
5.4	Aufgabenstellungen	139
5.5	Organisation & Durchführung der Evaluation.....	141
5.5.1	Qualitätsnachweise	141
5.5.2	Hypothesen & Stichprobe	141
5.5.3	Weitere Hypothesen.....	143
5.5.4	Analysestichprobe.....	143
5.6	Resultate.....	145
5.7	Qualitatives Feedback.....	152
5.8	Zusammenfassung.....	154
6.	Schlussbetrachtung	155
6.1	Empfehlungen zum Betrieb eines <i>Evolutionären Lernspiels</i>	156
6.2	Implikationen für die Forschung.....	158
6.3	Infrastruktur für <i>Evolutionäre Lernspiele</i>	159
7.	Ausblick.....	161
	Literaturverzeichnis	166
	Anhang A: Fragebogen.....	177
	Anhang B: Statistische Daten des Fragebogens.....	184
	Anhang C: Tutorial zum <i>PM Game</i>	205
	Anhang D: Lebenslauf	224

Tabellenverzeichnis

Tab. 1: Wirkungsmechanismen des Long Tail-Effekts nach Anderson (2004).....	9
Tab. 2: Zusammenhang der Identifikation des Nutzers mit Wikipedia und seiner Reputation nach Anthony (2005)	26
Tab. 3: Web 2.0-Beispiele und Web 2.0-Prinzipien	28
Tab. 4: Anforderungen an Autorenwerkzeuge nach Baumgartner (2004).....	32
Tab. 5: rollenspezifische Anforderungen an Lernplattformen	43
Tab. 6: Übersicht der Merkmale von digitalen Spielen	48
Tab. 7: Übersicht über Genres digitaler Spiele nach Ito (2005).....	53
Tab. 8: Übersicht Pseudo-Genres und Subgenres nach Ito (2005).....	55
Tab. 9: Spielziele der letzten Phase von Spore	62
Tab. 10: Zehn kognitive Lernstilveränderungen nach Prensky (2001)	75
Tab. 11: Synergien zwischen Lernelementen und Spielmerkmalen.....	79
Tab. 12: Übersicht über die Phasen des Evolutionären Lernspielkonzepts	91
Tab. 13: Entwurfsmuster für Evolutionäre Lernspiele	101
Tab. 14: Spielobjekte des PM Game	117
Tab. 15: Beschreibung der Attribute von Regeln	125
Tab. 16: Einstellungsakzeptanz und Verhaltensakzeptanz	134
Tab. 17: Korrelationen der untersuchten Bereiche des Akzeptanztests.....	145
Tab. 18: Reliabilitäten der einzelnen Bereiche des Akzeptanztests.....	146
Tab. 19: Mittelwerte und Standardabweichungen der einzelnen Bereiche	147
Tab. 20: Mittelwerte und Standardabweichungen zur Motivation der Studenten (n=22).....	148

Abbildungsverzeichnis

Abb. 1: Hintergrundbild für Line Rider	4
Abb. 2: Long Tail-Effekt, Pfeil (2007).....	8
Abb. 3: Personalisierte Startseite von YouTube™.....	15
Abb. 4: News Feed von Facebook.....	18
Abb. 5: Startseite von Delicious.....	20
Abb. 6: Webseite von Wikipedia.....	24
Abb. 7: Schlagworte des Web 2.0 nach Angermeier (2005)	30
Abb. 8: Komposition eines multimedialen Inhalts auf einer Zeitachse	34
Abb. 9: Hierarchische Komposition.....	34
Abb. 10: Komposition über Referenzpunkte.....	35
Abb. 11: Autorenwerkzeug Scratch.....	40
Abb. 12: Verfügbare, personalisierte Lernressourcen auf OLAT	45
Abb. 13: Spiel vs. Game nach Encyclopaedia Britannica (2003).....	50
Abb. 14: Virtuelle Welt in Spore und drei Editoren	60
Abb. 15: Spielszenario aus Little Big Planet.....	64
Abb. 16: Autorenwerkzeug des Spiels Little Big Planet.....	66
Abb. 17: Klassisches E-Learning vs. Game-based Learning (Prensky, 2003)	69
Abb. 18: Livelong Kindergarten Approach nach Resnick (2007).....	71
Abb. 19: Dales Erfahrungspyramide (1969), Darstellung von Dr. Göbel.....	73
Abb. 20: Phasen des Evolutionären Lernspielkonzepts.....	91
Abb. 21: Komposition der Komponenten.....	98
Abb. 22: Softwarearchitektur Evolutionärer Lernspiele	102
Abb. 23: Sequenzdiagramm des Login-Prozesses	103
Abb. 24: PM Game-Sprechakte.....	110
Abb. 25: PM Game-Spielfläche	111
Abb. 26: NPC-Editor	113
Abb. 27: Rollen und Rechte im PM Game.....	115
Abb. 28: Klassendiagramm der Spielobjekte.....	116
Abb. 29: Klassendiagramm Informationsobjekte.....	117
Abb. 30: Prozessschema der Szenariengenerierung im Rahmen des PM Game	119
Abb. 31: Subsumption-Architektur der PM Game-Regeln.....	122
Abb. 32: pureMVC-Architekturübersicht	127
Abb. 33: PM Game-Architekturübersicht	129
Abb. 34: Entwurfsmuster Business Delegation von Sun Microsystems® (2008).....	131
Abb. 35: Das Technology-Acceptance-Model 2 (Venkatesh & Davis, 2000)	137
Abb. 36: Adaptiertes Modell für den Akzeptanztest des PEdi	138
Abb. 37: Semantic Content Infrastructure	163

1. Einleitung

*“Erzähle mir – und ich vergesse.
Zeige mir – und ich erinnere.
Lass es mich tun – und ich verstehe.”*

Konfuzius, 551–479 v. Chr.

Im Rahmen dieser Arbeit wird aufbauend auf der konstruktivistischen Lernmethode *Game-based Learning* (s. Kapitel 2.5) ein neues *E-Learning*-Konzept erarbeitet und auf seine Akzeptanz und Bedienbarkeit hin evaluiert (s. Kapitel 5). Unter dem Begriff Lernmethode wird ein planmässiges, folgerichtiges Verfahren zum Erlernen, Erhalt oder Ausbau von Kenntnissen, Fähigkeiten und Kompetenz verstanden (Matthias, 2008). Der Fokus der vorliegenden Arbeit liegt dabei auf der informationstechnologischen Umsetzung des innovativen Konzepts. Dessen pädagogische Grundlagen werden überblicksweise in Kapitel 2.5.1 diskutiert.

1.1 Motivation

„In den vergangenen 10 Jahren haben neue Informations- und Kommunikationstechnologien (IKT) in einem atemberaubenden Tempo ihren Einzug in verschiedenste Bereiche des gesellschaftlichen Zusammenlebens gehalten“ (Froidevaux & Täube, 2006, S. 5). Im Jahr 2004 hatten in der Schweiz 61% der Haushalte – bei steigender Tendenz – Zugang zum Internet und nutzten es als Kommunikationsmittel und Informationsquelle. Ein Grund dafür ist neben der technologischen auch die kommerzielle Entwicklung, die sich z.B. in *Flatrate*-Tarifen, mobilem Internetzugriff und benutzerfreundlichen Webapplikationen widerspiegelt (Alby, 2008a).

Organisationen entdeckten das neue Potential des Internets als Werbe- und Absatzkanal nach dem furiosen Scheitern der *New Economy* (Alby, 2008b) und entwickelten rund ums Web neue Geschäftsmodelle. Es entstand u.a. der Trend, Anwender zum Zwecke der Unterhaltung in die Generierung von Inhalt für die von ihnen genutzte Software zu involvieren. Dafür entwickelten Softwarefirmen und Webdienstleister einfach zu bedienende Werkzeuge.

Innert weniger Jahre ist dadurch eine immense Menge an Daten entstanden, die Nutzer im Internet bereitstellen und austauschen. Diese Datenflut ist als potentielle Grundlage für neues Wissen in den Fokus wissenschaftlicher und ökonomischer Forschung gerückt. Die Nutzer stehen jedoch vor der Herausforderung, die für ihr jeweiliges Anliegen relevanten Informationen zu finden, diese mit Bekanntem zu assoziieren und auf diesem Weg Wissen(szuwachs) zu generieren (B.A., 2002). Trotz dieser sprichwörtlichen Suche nach den Nadeln im Heuhaufen wird das Potential benutzergenerierter Information für den kontinuierlichen Wissenserwerb unter Fachspezialisten als gross eingeschätzt. Letzteres, das sogenannte Lebenslange Lernen (Deutsches Bundesministerium für Bildung und Forschung, 2007), wird als unverzichtbar angesehen, um sowohl während der Ausbildung wie auch im Beruf dauerhaft konkurrenzfähig bleiben zu können. Dafür werden nach einhelliger Meinung pädagogischer Forscher neue Lernmethoden benötigt, welche die Lernenden zukünftig besser unterstützen als klassische Lernmethoden (Mandl & Krause, 2001;

Prensky, 2001; Thissen, 2003). Eine solche innovative Lernmethode ist das *Game-based Learning*, welches den Lernenden durch spielerische Komponenten intrinsisch motiviert, sich mit dem Lerninhalt auseinanderzusetzen. Die Entwicklung und Herstellung solcher Lernspiele gilt jedoch als sehr kostspielig und noch nicht etabliert (Bergeron, 2006). Dies führt nicht selten zu Qualitätseinbußen, welche auf zu kleine Budgets für die Umsetzung oder die oberflächliche Zusammenarbeit zwischen Lerninhaltsproduzenten und Spieldesignern zurückzuführen sind (Prensky, 2003). Die Verschmelzung von Spieldesign und didaktischem Konzept ist der kritische Faktor für den Erfolg eines Lernspiels (Gee, 2003).

Im Gegensatz zu Lernspielen sind moderne digitale Unterhaltungsspiele weit verbreitet und verschlingen Budgets von Hollywood-Filmproduktionen (Bergeron, 2006). Die neueste Generation von digitalen Spielen, welche über das Internet kollaborativ gespielt werden, integriert im Stile aktueller Webapplikationen das Konzept der benutzergenerierten Inhalte. Damit wird einerseits auf die (Entwicklungskosten sparende) Inhaltsgenerierung (z.B. von Szenarien) und andererseits auf die (Kunden bindende) Förderung kreativer Aktivitäten abgezielt. Dies geht sogar soweit, dass Nutzer den von den Designern konzipierten Verwendungszweck des Spiels neu interpretieren und dadurch emergente Unterhaltung schaffen. Letzteres bezeichnet eine Unterhaltungsform, welche durch das Einbeziehen der Konsumenten in die Inhaltsgenerierung und die Nutzung deren Kreativität entsteht. Als Beispiel wird folgend das Online-Spiel *Line Rider* von Boštjan Čadež vorgestellt (Čadež, 2006). Dieses wurde im September 2006 auf deviantART veröffentlicht und zählte wenige Wochen später zu den zehn meist gesuchten Stichwörtern in der *Google™*-Suchmaschine. Das Spiel ist ein Autorenwerkzeug (s. Kapitel 2.2), mit welchem der Nutzer Linien zeichnen kann. Auf diesen Linien fährt eine virtuelle Figur mit einem Schlitten. Der Schlitten folgt physikalischen Gesetzen und fährt nur auf Linien, welche von links nach rechts gezogen wurden. Ziel des Spiels ist es, komplizierte Strecken zu konzipieren, welche die Figur abfährt, ohne dabei vom Schlitten zu stürzen. Inzwischen haben die Nutzer den Verwendungszweck jedoch deutlich erweitert: Sie kreieren Bahnen für den Schlitten, welche als Landschaften, Portraits oder abstrakte Objekte getarnt sind und publizieren Videos der Fahrten auf Online-Plattformen für Unterhaltungsmedien, wie

etwa *YouTube*TM (s. Abb. 1). Innerhalb kurzer Zeit ist eine grosse virtuelle Gemeinschaft (s. Kapitel 2.4.4) um das Spiel *Line Rider* entstanden, die Tausende von Videos zu spektakulären Fahrten auf dem Internet publiziert hat.



Abb. 1: Hintergrundbild für *Line Rider*

Auf Lernspiele ist das Konzept der benutzergenerierten Inhalte nicht ohne Weiteres übertragbar, da anhand der erstellten Inhalte nicht nur Unterhaltung erzeugt, sondern darüber hinaus spezifisches Wissen vermittelt werden soll. Trotzdem sollte die Kombination von *Game-based Learning* und benutzergenerierten Inhalten nicht verworfen werden, denn einerseits hat sich das Lernverhalten der heute 15- bis 25-Jährigen in Richtung ausgeprägterer Mediennutzung verändert (Prensky, 2003). Andererseits entspricht das Erstellen von Inhalten dem konstruktivistischen Ansatz der modernen Pädagogik, bei dem sich die Lernenden unter Einbeziehung ihrer Erfahrungen Repräsentationen der Welt konstruieren und diese Konstrukte anderen Lernenden als Grundlage für deren Modelle zur Verfügung stellen können (Lernen durch Synthese). Die Herausforderungen, wie das Schaffen eines motivierenden Spieldesigns oder die Qualitätssicherung der benutzergenerierten Inhalte, bleiben jedoch bei der Integration dieses Konzepts in eine E-Learning-Applikation bestehen.

1.2 Fragestellung

Aus der dargelegten Motivation leitet sich die zentrale Fragestellung dieser Arbeit ab:

Lässt sich das Konzept benutzergenerierter Inhalte mit der Lernmethode *Game-based Learning* in einer Applikation kombinieren?

Die Ausgangsfrage lässt sich wie folgt weiter spezifizieren:

- **Welche Werkzeuge werden benötigt, um Nutzern die Entwicklung von Inhalten für Lernspiele zu ermöglichen?**
- **Welche technischen Vorkenntnisse der Nutzer müssen vorausgesetzt werden, um mit diesen Werkzeugen adäquat arbeiten zu können?**
- **Akzeptieren die Nutzer die Werkzeuge?**
- **Welche qualitätssichernden Massnahmen kommen für die Integration des Konzepts der benutzergenerierten Inhalte ins *Game-based Learning* in Frage?**

Diesen Fragen soll im Rahmen vorliegender Arbeit durch die Analyse bestehender Web 2.0-Applikationen, Unterhaltungsspiele, Lernspiele sowie die Entwicklung und Implementierung eines Prototyps beantwortet und diskutiert werden. Zudem wurde eine Evaluation unter Testnutzern entworfen und durchgeführt, um die entwickelten und später implementierten Werkzeuge zur Erstellung von Lernszenarien zu testen. Dieser Akzeptanztest (s. Kapitel 5.2) soll einerseits Aufschluss über die notwendigen Fertigkeiten und Vorkenntnisse der Nutzer geben und andererseits die Benutzerfreundlichkeit und -akzeptanz der Werkzeuge ermitteln.

2. Grundlagen

2.1 Web 2.0

Ausgelöst durch die rasante technologische Entwicklung der Informationstechnologie hat sich die Vision aus *New-Economy-Zeiten* bewahrheitet, die *AOL/Netscape* 1996 mit der Lancierung des *Netscape Communicator* umsetzen wollte (Alby, 2008a): Applikationen werden nicht mehr nur lokal auf dem Computer des Endbenutzers installiert, sondern stehen auf dem Internet als Dienste zur Verfügung und können wiederum zu neuen Diensten orchestriert werden. Letztere sind als *Web Services* aufrufbar (*Software as a Service*¹) und werden für den täglichen Gebrauch, z.B. zum Publizieren von Inhalten oder als Kommunikationsmittel genutzt. Ein hypothetisches Anwendungsszenario soll dies illustrieren: Der Nutzer sucht mit Hilfe einer Suchmaschine nach einer Online-Tabellenkalkulation, die seinen Anforderungen entspricht. Anschliessend ruft er den Dienst auf, arbeitet online und speichert die Resultate (Daten) direkt auf die dedizierte Speichereinheiten des Diensts sowie zusätzlich als Sicherungskopie lokal auf seinen Computer. Somit ist der Nutzer nicht mehr an einen bestimmten Computer gebunden: Er kann seine Arbeit ortsunabhängig auf verschiedenen Endgeräten mit Internetzugang erledigen.

Diese Möglichkeiten zur Nutzung des Internets wurden durch Verbesserungen der Infrastruktur (z.B. Breitbandinternetanschlüsse für eine Vielzahl von Haushalten) ergänzt um *Flatrate*-Tarife verschiedener Telekommunikationsdienstleister sowie durch die oben genannten, innovative Technologien und intuitiven Benutzerschnittstellen ermöglicht (Alby, 2008a). Denn wer würde ein Online-Photoalbum benutzen, wenn jedes Photo einzeln mit einer Datenrate von 56 kbit/s auf einen Server übertragen und zum Betrachten eines hochauflösenden Photos mehrere Minuten Wartezeit in Kauf genommen werden müssten?

¹ Siehe http://de.wikipedia.org/wiki/Software_as_a_Service

Neben niedrigeren Kosten, technologischen Fortschritten und grösserer Geschwindigkeit spielten auch Open-Source-Initiativen eine wichtige Rolle für die Einführung neuer Arbeitsprozesse, Organisationsformen und Geschäftsmodelle (Lakhani & Wolf, 2005). Die Kombination verschiedener Entwicklungen und Trends wird unter dem Schlagwort Web 2.0 zusammengefasst. Dahinter verbirgt sich jedoch weit mehr als lediglich technologische und wirtschaftliche Errungenschaften: Es entstanden organisatorische Innovationen, wie z.B. ortsunabhängige Arbeitsplätze (Neff & Stark, 2003; O'Mahony, 2003). Berners-Lee (1999) konstatiert sogar eine Gesellschaftsveränderung durch den einfachen Zugang zu Informationen. Was das Web 2.0 ausmacht, wird je nach Autor unterschiedlich definiert; nicht wenige Quellen verzichten sogar ganz auf Eingrenzungsversuche, in der irrigen Annahme, das Schlagwort Web 2.0 wäre selbsterklärend genug.

Tim O'Reilly (2007) definiert den Begriff Web 2.0 wie folgt:

Web 2.0 is the network as platform, spanning all connected devices; Web 2.0 applications are those that make the most of the intrinsic advantages of that platform: delivering software as a continually-updated service that gets better the more people use it, consuming and remixing data from multiple sources, including individual users, while providing their own data and services in a form that allows remixing by others, creating network effects through an "architecture of participation," and going beyond the page metaphor of Web 1.0 to deliver rich user experiences.

Nach Tim O'Reilly (2007) spielen nicht nur die verteilten Webapplikationen eine zentrale Rolle im Konzept des Web 2.0, sondern auch die gemeinsamen Nutzung von Inhalten. Das trage dazu bei, dass Informationen und schlussendlich Wissen geteilt würden und neue Applikationen aus der Kombination bestehender Applikationen entstehen könnten. Dies kann als Wandel moderner Softwarearchitekturen weg von statischen Desktopapplikationen hin zu verteilten, datengetriebenen Webapplikationen oder Diensten verstanden werden. So nähert sich die Entwicklung des Internets der ursprünglichen Vision des Webs von Tim Berners-Lee (1999) an: Jeder Nutzer soll Ressourcen auf dem Web betrachten, für seine Zwecke verwenden und verändern können, sofern er über die dafür nötigen Berechtigungen verfügt.

2.1.1 Web 2.0-Prinzipien nach O'Reilly

Was macht das heutige Internet eigentlich so einzigartig, faszinierend und nützlich gegenüber dem Internet zur *New-Economy-Zeiten*? Tim O'Reilly (2007) identifizierte acht Prinzipien, welche das Web 2.0 auszeichnen. Im Folgenden werden diese beschrieben und anschliessend an konkreten Beispielen diskutiert.

1. Der *Long Tail*-Effekt

Der Ausdruck *Long Tail-Effekt* stammt von Chris Anderson und wurde in der amerikanischen Zeitschrift *wired magazine* im Jahr 2004 veröffentlicht (Anderson, 2004). Der Effekt beschreibt das Potential einer Nischenstrategie für Onlinehändler, welche durch ein breites Angebot von verschiedenen Produkten mit einer geringen Nachfrage charakterisiert ist (Pohl, 2006).

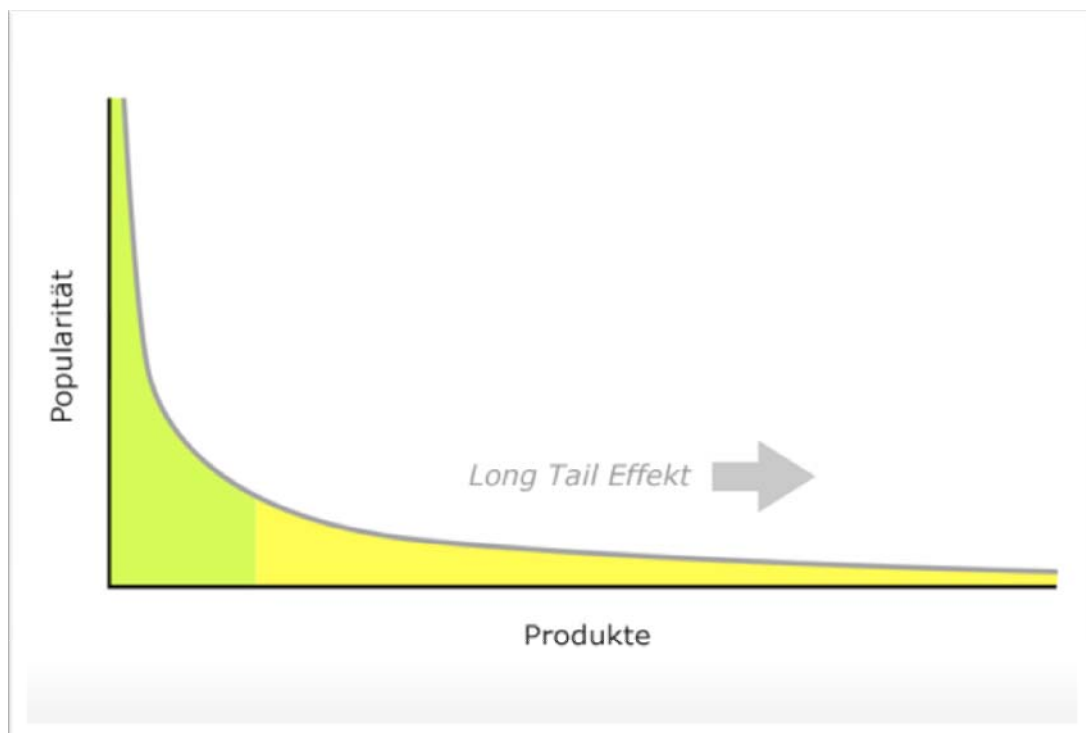


Abb. 2: *Long Tail*-Effekt, Pfeil (2007)

Durch das Internet erreichen Händler ein breiteres Publikum, wodurch die Nachfrage nach ihren Produkten gesteigert werden kann, so dass gesamthaft eine grössere Menge dieser Nischenprodukte verkauft wird als von den wenigen Produkten mit

grosser Nachfrage (s. Abb. 2). Anderson hat drei Wirkungsmechanismen herausgearbeitet, welche den *Long Tail*-Effekt ermöglichen (s. Tab. 1)

Tab. 1: Wirkungsmechanismen des Long Tail-Effekts nach Anderson (2004)

Wirkungsmechanismus	Beschreibung
Demokratisierung der Produktionsmittel	Durch das Internet und die bereitgestellten Werkzeuge kann jeder Nutzer zur Generierung eines Produkts beitragen und damit selbst zum Produzenten werden.
Demokratisierung des Vertriebs	Neben der Demokratisierung des Erstellungsprozesses muss auch die Distribution demokratisiert werden. Online-Plattformen, wie <i>Ebay</i> ² , <i>Amazon</i> ³ oder <i>Apple Store</i> ⁴ , stellen die benötigten Hilfsmittel zur Verfügung, um jeden Nutzer zu einem potentiellen Anbieter werden zu lassen. Somit können mehrere Nischen bedient und Kosten reduziert werden.
Verbindung von Angebot und Nachfrage	Das Zusammenführen von Angebot und Nachfrage erfolgt über Aggregatoren, wie <i>Amazon</i> ⁵ oder <i>Google</i> ⁵ . Diese filtern Informationen für den Nutzer und ermöglichen das effiziente Auffinden von relevanten Inhalten.

Der *Long Tail-Effekt* lässt sich gut am Beispiel der Multimedia-Verwaltungsapplikation *iTunes* (*Apple*) aufzeigen. Neben der Verwaltung von multimedialen Inhalten kann in dieser Applikation online Musik erworben werden. Die Kunden von *iTunes* kaufen neben aktuellen Hits ihre individuell favorisierte Musik, welche unter Umständen nicht dem Mainstream entspricht. Mit Hilfe der vielen Nischen, die durch *iTunes* abgedeckt werden, können Millionen von Nutzern unterschiedlichen Musikgeschmacks angeworben und ihren Wünschen entsprechend bedient werden. Derartige Nischenprodukte bilden somit einen nicht zu vernachlässigenden Teil der

² <http://www.ebay.com>

³ <http://www.amazon.com>

⁴ <http://www.apple.com/iphone/appstore/>

⁵ <http://www.google.com>

verkauften Musik. Die 1000 meist gekauften Lieder generieren für iTunes gerade einmal 2% des Umsatzes. Insgesamt befinden sich über 8 Millionen Lieder im Angebot (5inline.de, 2009).

Dieses Geschäftsmodell ist nur dank des Internets möglich, da die Kosten für den virtuellen Speicherplatz gegenüber den Kosten für physikalischen Platz in einem Ladenlokal vernachlässigbar sind. Der CD-Händler muss in seinem physischen Laden den verfügbaren Platz optimal nutzen, so dass er es sich z.B. nur unter Inkaufnahme finanzieller Risiken erlauben kann, eine CD ohne nennenswerte Absatzchance in sein Angebot aufzunehmen.

Dieses Konzept trifft auch auf andere Phänomene im Internet zu. So werden beispielsweise die meisten auf dem Internet publizierten Inhalte durch einzelne Personen oder kleinere Organisationen zu spezifischen Themen für spezielle Märkte oder Nischen auf dedizierten Webseiten erstellt. Die Inhalte der grossen Online-Plattformen und -Portale stellen nur einen kleinen Teil der verfügbaren Ressourcen dar. Ziel der grossen Online-Plattformen muss es daher sein, diese spezifischen Informationen oder Dienste jedermann einfach zugänglich zu machen, damit die Informationen neben den etablierten Informationsquellen ergänzend oder konkurrierend konsumiert werden können (Surowiecki, 2004). Deshalb stellen Dienste oder Portale, wie *Facebook*⁶ (s. Kapitel 2.1.2) oder *Amazon*® Speicherplatz und standardisierte Schnittstellen zum Publizieren oder Verknüpfen dieser spezifischen, nutzergenerierten Inhalte zur Verfügung, um sie einem breiten Publikum zugänglich zu machen.

2. Inhalte sind entscheidend

Applikationen sind für Nutzer ein austauschbares Werkzeug und stehen nicht im Mittelpunkt ihrer Aktivitäten. Entscheidend sind die Inhalte und Schnittstellen zum Abspeichern und Aufrufen der Inhalte, die von den Applikationen angeboten werden. So gibt es z.B. Hunderte von Diensten wie *YouTube*⁷ (s. Kapitel 2.1.2), doch durch seine spezifischen Inhalte hat sich *YouTube*TM von anderen Applikationen abheben und als weltweit führender *Streaming-Video Service* etablieren können.

⁶ <http://www.facebook.com>

⁷ <http://www.youtube.com>

3. Kollektive Intelligenz

Inhalte durch professionelle Autoren zu erstellen und zu redigieren, ist kostenintensiv. Zudem sind professionelle Autoren meist angehalten, Inhalte den Richtlinien der verantwortlichen Organisation entsprechend zu kreieren. Dies führt zu einseitig interpretierten Inhalten, worauf die Applikationen durch fehlende Objektivität und Vielfalt für ein breites Publikum an Attraktivität verlieren. Werden jedoch Konsumenten aktiv in die Inhaltsgenerierung mit einbezogen, können Themen von verschiedenen Blickwinkeln beleuchtet werden. Neue Nutzer können durch so genanntes virales Marketing⁸ geworben und aufgrund der bestehenden Inhalte angeregt werden, diese zu erweitern oder eigene Inhalte zu publizieren und Meinungen kundzutun.

Nur ein Bruchteil der Internetnutzer wird sich aktiv an der Inhaltsgenerierung beteiligen und seine Kreativität zur Schau stellen (Gosh & Prakash, 2000). Der Grossteil der Nutzer bleibt klassischer Konsument. Dennoch können auch solche Nutzer mit einbezogen werden und ihren Teil zur Qualitätsverbesserung der Applikation beitragen, etwa indem sie Feedback und Aufschluss über ihr Nutzungsverhalten geben. Durch die Analyse des Benutzerverhaltens können z.B. Rückschlüsse auf interessante Inhalte oder Zusammenhänge zwischen verschiedenen Inhalten ermittelt werden, welche wiederum zur Optimierung der Verknüpfung und Anordnung der Inhalte beitragen können (Segaran, 2007; Baur & Kolo, 2002). Solche impliziten Beiträge werden durch kleine, explizite Beiträge, wie etwa Bewertungen oder Weiterempfehlungen ergänzt (s. Kapitel 2.1.2).

Das Konzept der kollektiven Intelligenz lässt sich auch am *PageRank* von *Google*TM demonstrieren: Eine Webseite wird in Abhängigkeit ihrer Verweishäufigkeit im Internet bewertet, wobei die Wertigkeit mit zunehmender Zahl so genannter Links steigt. Somit basiert diese implizite Bewertungsmethode auf den Informationen von Tausenden Personen, welche Verlinkungen zu der entsprechenden Seite erstellt haben (Brin & Page, 1999).

⁸ *“Virales Marketing ist eine Marketingform, die existierende soziale Netzwerke und Medien ausnutzt, um Aufmerksamkeit auf Marken, Produkte oder Kampagnen zu lenken, indem sich Nachrichten epidemisch wie ein Virus ausbreiten sollen. Die Verbreitung der Nachrichten basiert damit letztlich auf Mundpropaganda, also der Kommunikation zwischen den Kunden oder Konsumenten.”* (wikipedia.org, 2008)

4. Netzwerkeffekt

Je mehr Personen eine Applikationen nutzen, desto besser kann diese durch Verhaltensanalysen an die Bedürfnisse der Individuen angepasst werden, indem z.B. Vor- oder Standardeinstellung daraus abgeleitet und für die Definition von Nutzerprofilen verwendet werden. Dies führt dazu, dass die Akzeptanz und Verbreitung der Applikation durch adäquate Personalisierung steigt. Als Beispiel sei das Online-Versandhaus *Amazon*® erwähnt. Nach wenigen Mausklicks auf der Webseite ist die Applikation in der Lage, festzustellen, was den Nutzer interessiert. Dessen Interesse wird anhand eines Vergleichs vergangener Interaktionen anderer Nutzer mit eigenem Nutzungsverhalten identifiziert. Ziel dieser Verhaltensanalyse ist es, dem Nutzer Vorschläge zu anderen ihn wahrscheinlich interessierenden Artikeln zu unterbreiten und dadurch impulsiv weitere Kaufentscheidungen zu fördern.

5. Nicht alle Rechte an der Software und den Inhalten vorbehalten

Das von den Unternehmen häufig vorgenommene Schützen geistigen Eigentums, sowohl bezogen auf Applikation wie auch auf Inhalte, stört die Nutzer bei der Entfaltung ihrer Kreativität und verhindert mitunter deren eigentlich gewünschte Experimentierfreudigkeit. Deshalb sollten die Hersteller von Applikationen deren Schnittstellen offenlegen und das innovative Kombinieren der Inhalte durch die Nutzer erlauben. Lizenzen sollten möglichst so gewählt werden, dass keine Restriktionen zur Wiederverwendung der Applikation und deren Daten entstehen.

6. Andauernde Beta-Phase der Software

Heutige Applikationen werden ständig um neue Funktionen erweitert, um am Markt konkurrenzfähig zu bleiben. Diese Funktionen werden anhand von Benutzeranalysen und Feedbacks identifiziert. Sie werden ganz bewusst als Beta-Version veröffentlicht, um weiteres Feedback von Nutzern bezüglich Brauchbarkeit oder Benutzerfreundlichkeit zu erhalten. Ausserdem können interessierte Nutzer offen als Tester neuer Funktionalitäten verwendet werden, ohne sie zu frustrieren. Falls Funktionen von den Nutzern nicht akzeptiert werden, können sie nach einer kurzen Einführungsphase wieder aus der Applikation entfernt werden. Dadurch entsteht eine rein nutzerbasierte Selektion exakt derjenigen Funktionen, welche den Anforderungen, dem Verhalten und der Akzeptanz der Mehrheit der Nutzer entsprechen.

7. Kooperation statt Kontrolle

Applikationen stellen die für den Endnutzer wichtigsten Funktionen und Inhalte als Dienste zur Verfügung und bieten dadurch anderen Applikationen die Möglichkeit der Integration. Deshalb sollten Applikationen möglichst modular und kontext-unabhängig implementiert werden, damit sie für viele verschiedene Kompositionen, Verwendungszwecke und Domänen nachhaltig einsetzbar sind.

Auf diesem Weg entstehen sogenannte *Mashups*. Darunter werden Kompositionen von verschiedenen Diensten verstanden, welche durch Orchestrierung neuer Funktionalitäten und Dienste entstehen. Der Begriff stammt aus der Musik und bezieht sich auf die Kombination zweier oder mehrerer voneinander unabhängiger Musikstücke (Alby, 2008a; Carl et al., 2008)

8. Geräteunabhängigkeit

Applikationen sollten möglichst geräteunabhängig verwendbar sein, damit Nutzer darauf und auf ihre Daten überall und mit ihren bevorzugten Endgeräten zugreifen können (Dobusch & Forsterleitner, 2009).

Die beschriebenen Prinzipien von O'Reilly sollen in den nächsten Abschnitten anhand etablierter Web 2.0-Applikationen besprochen werden. Diese werden wegen ihrer Charakteristiken mitunter auch als *Social Software* bezeichnet. Unter dem Begriff *Social Software* wird ein System verstanden, mit welchem mehrere Nutzer kommunizieren, zusammenarbeiten oder auf andere Arten interagieren können (Koch & Richter, 2007). Im Rahmen des Web 2.0 kann dieser Definition mit der Förderung des Aufbaus und der Unterstützung des Selbstmanagements virtueller Gemeinschaften entsprochen werden. Letztere sind für die Qualitätssicherung der benutzer-generierten Inhalte verantwortlich (Surowiecki, 2004).

2.1.2 Beispiele von Web 2.0-Anwendungen

YouTube™

YouTube™ ist ein Videoportal, welches 2005 von Chad Hurley, Steve Chen und Jawed Karim gegründet und 2006 von Google™ gekauft wurde. Nutzer dieses

Videoportals können kostenlos Video-Clips ansehen und bereitstellen. Hauptsächlich werden Film- und Fernsehausschnitte, Musikvideos sowie selbst gedrehte Filme auf *YouTube*TM publiziert. Seit der Gründung wurde eine beachtliche Menge von Inhalten hochgeladen, welche Millionen von Nutzern täglich anlocken (Kirkpatrick, 2006). Von den einhundert Millionen täglich angeschauten Videos wird nur ca. 1% mehr als 500'000 mal im Monat aufgerufen. Dies führt zu einer Häufigkeitsverteilung von Videoaufrufen, welche dem bereits erläuterten *Long Tail*-Effekt entspricht.

Das Erfolgsrezept von *YouTube*TM sind die einfachen, übersichtlichen Prozesse für das Auffinden, Bewerten, Kommentieren und Bereitstellen von Videos. Diese können mit wenigen Mausklicks über eine intuitive Kategorisierung und dank einer "intelligenten" Verschlagwortung gefunden werden. Des Weiteren haben sich drei spezielle Kategorisierungen etabliert, die auf der Startseite des Portals angezeigt werden:

1. Hoch bewertete Videos werden von *YouTube*TM speziell gekennzeichnet und in der Kategorie "featured videos" berücksichtigt (s. Abb. 3, mitte).
2. Videos, welche von vielen Nutzern mit ähnlichem Benutzerprofil angeschaut wurden, werden dem Suchenden als Empfehlung angezeigt ("Recommended Videos for you", s. Abb. 3, oben)
3. Häufig angeschaute Videos werden über die Kategorien "most viewed" (absolute Häufigkeit) oder "rising videos" (Häufigkeit pro Tag) abgebildet (s. Abb. 3, unten).

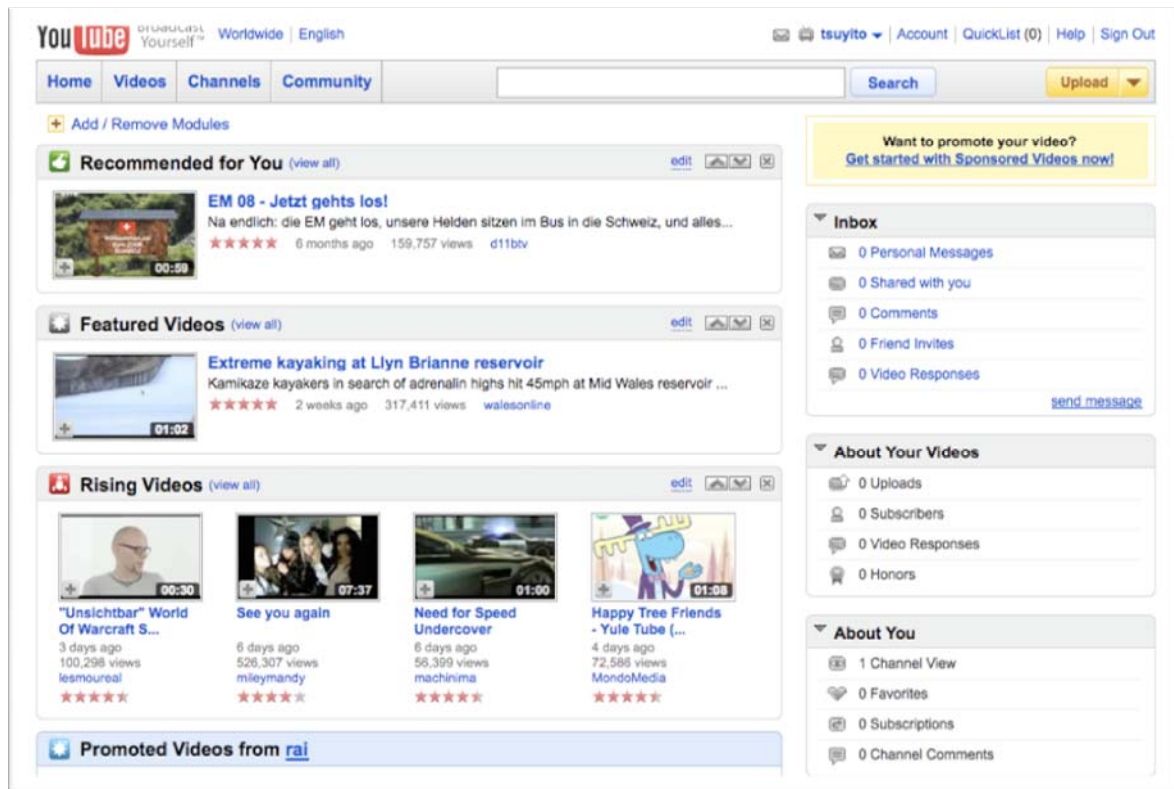


Abb. 3: Personalisierte Startseite von YouTube™

Falls unangebrachte (d.h. rassistische, hetzerische oder pornographische) Videoinhalte publiziert wurden, können die Nutzer diese als nicht geeignet markieren. YouTube™-Mitarbeiter werden daraufhin diesen Inhalt begutachten und im Falle von Verstößen gegen die allgemeinen Geschäfts- und Nutzungsbedingungen nicht mehr zugänglich machen.

Trotz dieser Qualitätskontrolle kämpft YouTube™ einen nahezu aussichtslosen Kampf gegen fragwürdige Inhalte und Urheberrechtsverletzungen: Die manuelle Kontrolle der Menge an minütlich hochgeladenen neuen Inhalten wäre nur durch eine massive, kostenintensive Aufstockung des Personals zu bewältigen, was die Effizienz der Plattform erheblich beeinträchtigen würde. YouTube™ wurde schon mehrmals ermahnt und verklagt, urheberrechtlich geschützte Inhalte zu entfernen. Das Aufspüren sämtlicher illegalen Inhalte erweist sich jedoch als ein schwieriges, aufwendiges Unterfangen.

Neben Urheberrechtsverletzungen stellen fehlende Massnahmen zur Altersverifizierung zur Betrachtung nicht jugendfreier Inhalte ein Problem für YouTube™

dar: Jeder registrierte Nutzer erhält automatisch Zugang zu allen Videoinhalten, ohne dass bei der Registrierung falsche Altersangaben festgestellt werden könnten.

*YouTube*TM bietet nicht nur eine Plattform, um Videos anzuschauen, sondern kann ebenfalls als Speicherplatz für multimediale Inhalte genutzt werden: Nutzer laden ihr Videomaterial auf die Server und können die Inhalte als Dienst in andere Seiten einbetten. Somit müssen die Produzenten sich keine Gedanken über die Formatierung der Videos, Bandbreite, Performanz und Speicherplatz machen, sondern können ihr Videomaterial ohne grösseren Aufwand in ihre Seiten integrieren. Des Weiteren haben *YouTube*TM resp. *Google*TM verschiedene Programmierschnittstellen (*API*)⁹ veröffentlicht, mit denen Entwickler diverse Dienste von *YouTube*TM ansteuern können. Webapplikationen, die diese *API* benutzen, sind dadurch in der Lage, die Abspieleinrichtung zu personalisieren und Suchfunktionalitäten, Bewertungs- und Feedbackfunktionalitäten zu integrieren.

*YouTube*TM hat sich als weltweit führendes Online-Videoportal durchgesetzt und wird von Millionen von Menschen täglich genutzt: Die Plattform rangierte 2007 unter den fünf meist genutzten Online-Plattformen. Der Dienst ist sowohl für Produzenten wie auch für Konsumenten kostenlos und deckt die entstehenden Kosten für Bandbreite, Personal- und Mietkosten zum Teil durch dezente Werbeeinblendungen in ausgewählten Videos von Partnern sowie mit Hilfe von Pay-Per-Click-Anzeigen (vgl. Ads.youtube.com, 2008; Faz.net, 2007). Der Grossteil der Kosten wird jedoch durch *Google*TM gedeckt. Vor der Übernahme durch *Google*TM wurde *YouTube*TM vom Silicon-Valley-Risikokapitalgeber *Sequoia Capital* finanziert.

Die Online-Plattform bezieht ihre Popularität aus der Kreativität ihrer Nutzer: Täglich werden neue, mitunter aussergewöhnlich einfallsreiche Videos publiziert. Einige dieser Videos und deren Produzenten werden daraufhin weltweit bekannt (vgl. Google Press Center, 2009). Dabei ist anzumerken, dass *YouTube*TM sich erlaubt, ausgewählte Inhalte für eigene Zwecke weiterzuverwenden, zu verkaufen oder zu lizenzieren.

⁹ englisches Akronym für *Application Programming Interface*

Die Video-Clips werden zudem in verschiedene Formate konvertiert, sodass sie auf unterschiedlichen Endgeräten problemlos abrufbar sind und jederzeit überall angeschaut werden können.

Facebook

Facebook ist eine Online-Kontaktwebseite, welche 2004 von Mark Zuckerberg zur Bildung sozialer Netzwerke an der Harvard University entwickelt wurde. *Facebook* war zuerst nur für Studenten dieser Hochschule konzipiert. Mittlerweile ist der Dienst jedoch weltweit verfügbar und eignet sich für eine Vielzahl unterschiedlicher Anwendungszwecke: Die Nutzer registrieren sich bei *Facebook*, erstellen Benutzerprofile, mit denen sie sich anderen registrierten Nutzern vorstellen, und laden Freunde, Bekannte und Verwandte ein, um (Kontakt-)Informationen auszutauschen. Zudem können Fotos bereitgestellt und einem ausgewählten Kreis an Nutzern als Fotogalerien zur Verfügung gestellt werden. Die Fotos können mit Schlagworten versehen und mit den Profilen anderer Nutzer, welche auf den Fotos zu sehen sind, verknüpft werden. Ergänzt werden die persönlichen Profile durch einen *Blog*, ein Nachrichtensystem und eine Pinwand, auf welcher eingeladene Nutzer Nachrichten hinterlassen können (s. Abb. 4). Zudem können Nutzer auf einem Marktplatz Kleinanzeigen aufgeben und einsehen sowie Netzwerke resp. Gruppen bilden.

Bis 2008 haben sich mehr als 120 Millionen Nutzer bei *Facebook* registriert (Facebook.com, 2009). Die Nutzung der Applikation geht weit über eine Verwaltung der Kontakte hinaus. Die Nutzer kombinieren Inhalte von anderen Webapplikationen, wie z.B. *YouTube*TM, um im *Blog* oder auf der Pinwand über verfügbare Schnittstellen Videos zu publizieren. Ausserdem wurde 2007 ein *Facebook-API* veröffentlicht, welches den Nutzern das selbständige Entwickeln von Applikationen für *Facebook* oder die Integration von *Facebook*-Daten in andere Applikationen ermöglicht (Developers.facebook.com, 2009). Daraufhin wurden innerhalb eines Jahres über 10'000 solcher Applikationen für *Facebook* entwickelt (s. Abb. 4, rechts), jedoch nur ca. zehn Applikationen werden von den Nutzern regelmässig gebraucht (Businessshakers.com, 2007). Marshall Kirkpatrick (2007) schrieb auf ReadWriteWeb.com:

“... less than 20 Facebook [applications] that are used on a daily basis by more than 25% of the users who have installed that app. The vast majority of these 10,000 apps are installed and forgotten about.

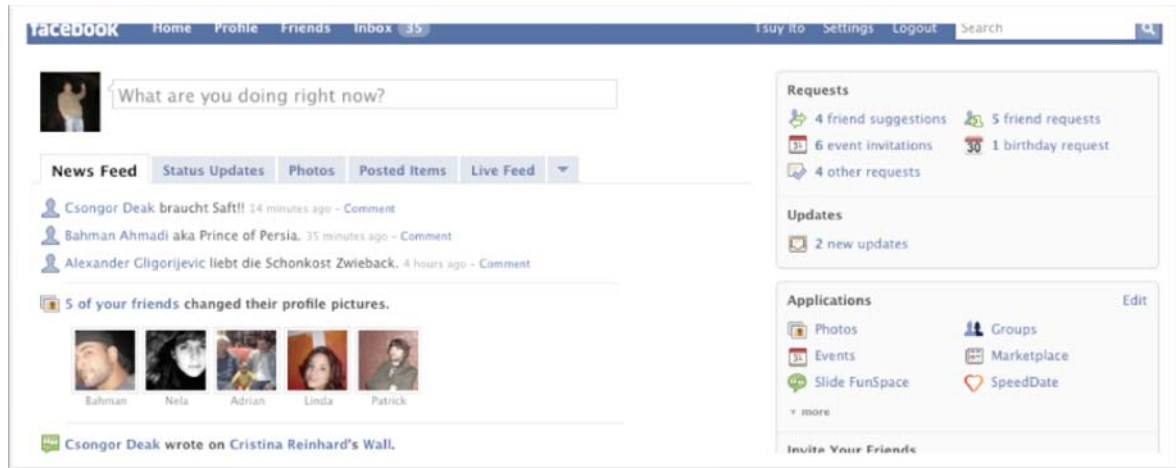


Abb. 4: News Feed von Facebook

Dennoch ist durch die Offenlegung der Schnittstelle der Bekanntheitsgrad der Applikation gestiegen, da nun neue Benutzerschnittstellen für diverse Endgeräte (z.B. *iPhone*) entwickelt, neue *Mashups* generiert (Alby, 2008a; Alby, 2008b) und Kooperationen mit anderen Online-Plattformen möglich wurden. Des Weiteren erscheint im Januar 2009 die überarbeitete Fotoverwaltungs- und -bearbeitungsapplikation *iPhoto* von *Apple*, welche sich ebenfalls der Schnittstelle zu *Facebook* bedient, um direkt Fotos von *iPhoto* nach *Facebook* zu publizieren und eventuelle Metainformationen der Bilder gegenseitig auszutauschen.

Die Nutzungsmöglichkeiten von *Facebook* werden durch die Kreativität der Nutzer und Entwickler stetig erweitert. So werden die persönlichen Profile für Statusberichte über Arbeit, Freizeit oder Ferien genutzt. Die Gruppenbildungsfunktionalität wurde ausgebaut, so dass es als Einladungssystem mit An- und Abmeldefunktionalitäten für Partys oder andere Anlässe genutzt werden kann.

Zwischen besonders engagierten *Facebook*-Nutzern ist in den letzten Jahren ein Wettbewerb entstanden, wer über die meisten Kontakte verfügt. Nach Meinung dieser Nutzer widerspiegelt die Anzahl eigener Kontakte den Bekanntheitsgrad (sowie die Beliebtheit?) des jeweiligen Nutzers. Dies führte zu einer Masseneinladungshysterie,

wodurch die Bezeichnung "Freund" im Kontext von *Facebook* nicht mehr dem allgemeinen Sprachgebrauch entspricht und neu definiert werden muss. Erwähnenswert ist, dass eine wissenschaftliche Studie einen (positiven?) Zusammenhang zwischen der Nutzung von Facebook und der Steigerung des subjektiven Wohlbefindens bei Personen mit geringer Lebenszufriedenheit und geringer Selbstachtung feststellen konnte (vgl. Ellison et al., 2007).

Facebook finanziert sich grösstenteils durch die Schaltung personalisierter Werbung. Dafür wurden sensible persönliche Daten, wie demographische Angaben und persönliche Beziehungen der Nutzer Werbe-Interessenten zur Verfügung gestellt, um ihnen personalisierte Werbung auf *Facebook* zu ermöglichen.

Delicious

Delicious wurde 2003 von Joshua Schachter entwickelt und gilt als eines der ersten nicht-hierarchischen, kollaborativen Verschlagwortungssysteme (*Social Bookmarking Applikation*). Unter dem Begriff *Tagging* wird im Internet der Vorgang verstanden, (einen) Inhalt (Video, Audio, Bild, Text oder Webseite) mit einem oder mehreren kennzeichnenden Schlüsselwörtern (engl. *Tag*) zu markieren. Im *Browser* werden diese Verknüpfungen (Links) zu Webseiten mitunter auch Lesezeichen (engl. *Bookmarks*) genannt und in einer Lesezeichen-Sammlung verwaltet. In dieser können Links individuell kategorisiert werden. Dabei hat es sich als nachteilhaft erwiesen, dass die Links einerseits nur lokal auf dem Computer im *Internetbrowser* gespeichert und somit nicht von anderen Applikationen und Endgeräten aufrufbar sind und dass andererseits jeder Link nur jeweils einer Kategorie zugeordnet werden kann.

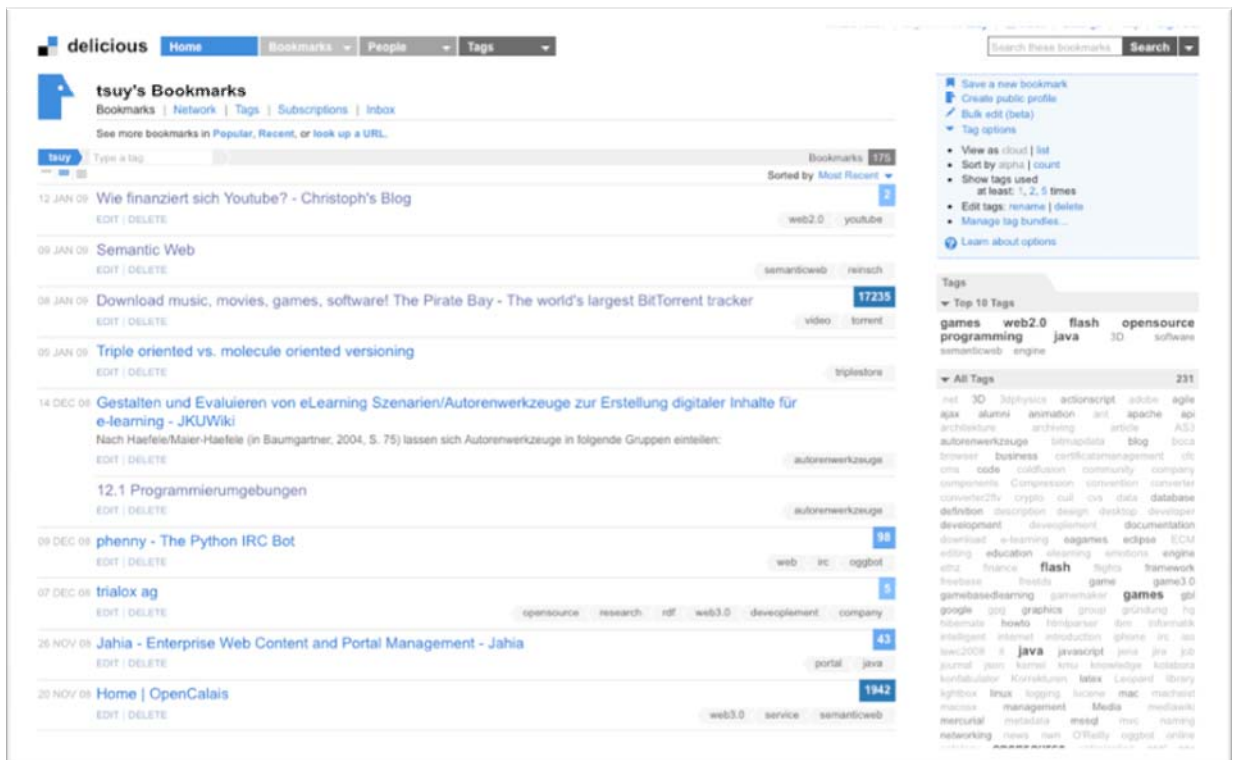


Abb. 5: Startseite von Delicious

Delicious stellt eine Erweiterung dieser browser-integrierten Lesezeichen-Sammlung dar und behebt die aufgezählten Nachteile der bestehenden Lösung, indem Schlagwörter anstatt einer hierarchischen Struktur zum Ablegen und Auffinden von Inhalten verwendet werden und die Links zu den Inhalten auf dem Internet auf einem öffentlich zugänglichen *URI-Space* auch anderen Nutzern zur Verfügung gestellt werden.

Die Links und Schlagwörter werden von der Applikation *Delicious* analysiert, um dem Suchenden die meist verbreiteten Schlagworte für ihn interessierende Links vorzuschlagen. Auf diesem Weg entsteht eine Taxonomie, die auf den Kategorisierungen der Nutzer basiert und den Ursprung des Begriffs *Folksonomy* darstellt. *Folksonomy* kombiniert die englischen Wörter *Folks* (= die Leute) und *Taxonomy* (= Taxonomie) und betont die Verbindung zwischen Experten- und Laienwissen. Die *Tags* werden zudem je nach Gebrauchshäufigkeit mit grösserer oder kleinerer Schriftgröße in Form einer sogenannten *Tag-Cloud* visualisiert (s. Abb. 5, rechts unten).

An einem typischen Anwendungsszenario sollen zuerst die Schwächen der herkömmlichen Systeme verdeutlicht und anschliessend die Nutzungsmöglichkeiten von *Delicious* erläutert werden. Der Berater eines fiktiven Internetdienstleisters recherchiert auf dem Web nach Rahmenwerken, welche das Architekturmuster der Modell-Repräsentations-Steuerung (*MVC*)¹⁰ unterstützen und wird nach längerer Suche fündig. Da er das Rahmenwerk für ein PHP-Projekt braucht, ordnet er es unter dem Schlagwort "*MVC-Rahmenwerk*" in der Hauptkategorie "Softwareentwicklung" unter der Programmiersprache "PHP" ein. Am nächsten Tag bespricht der Berater mit einem Kunden die Möglichkeiten der Realisierung eines anderen Projektes. Da der Berater die Chance sieht, auch im Rahmen dieses Vorhabens das zuvor gefundene Rahmenwerk einzusetzen, möchte er es dem Kunden anhand einer Übersichtsgraphik vorstellen, welche auf der Entwickler-Webseite hinterlegt ist. Das Lesezeichen zu dieser Webseite hat er jedoch auf seinem Desktopcomputer im Büro gespeichert, so dass er nun über eine Suchmaschine erneut nach dem Rahmenwerk suchen muss und eine gewisse Zeit braucht, die Übersichtsgraphik zu finden. Ein halbes Jahr später arbeitet der Berater an einem neuen Projekt, das in Java realisiert werden soll. Er recherchiert und durchsucht seine Lesezeichen im Browser unter der Kategorie „Softwareentwicklung“ unter der Programmiersprache „Java“, findet aber keine Anhaltspunkte für ein geeignetes Rahmenwerk, da er den Link zum *MVC*-Architekturmuster in einer andern Kategorie gespeichert hat und es ihn dazumal nicht interessierte, dass das Rahmenwerk ebenfalls in Java implementiert wurde.

Mit dem herkömmlichen System mussten sich die Nutzer bereits zum Zeitpunkt der Speicherung des Lesezeichens bewusst sein, unter welcher Kategorie sie dieses ablegen sollen, damit die Wahrscheinlichkeit möglichst gross ist, den Link später wiederzufinden.

Mit der Verwendung von *Delicious* würde dasselbe Szenario wie folgt ablaufen. Der Berater hat sich beim Dienst *Delicious* registriert und ein Online-Konto auf einem eigenen URI-Space für seine Lesezeichen eröffnet. Er installiert zudem noch ein *Plugin* im *Browser*, um die Links direkt von den jeweils interessanten Webseiten/-ressourcen in sein Online-Konto zu speichern und somit den gewohnten Lesezeichen-

¹⁰ englisches Akronym für Model View Control (vgl. Freeman et al., 2004)

Prozess beizubehalten. Der Berater sucht nun nach Rahmenwerken, welche für das PHP-Projekt in Frage kommen und findet ein geeignetes. Er möchte das Lesezeichen nun speichern und klickt auf die Schaltfläche des installierten *Plugins*, kann daraufhin die Webseite/-ressource mit Hilfe mehrerer Schlagworte kennzeichnen und sie so in mehreren unterschiedlichen Kategorien ablegen. Falls ihm keine geeigneten Schlagworte einfallen, liefert *Delicious* Empfehlungen, wie die Ressource verschlagwortet werden könnte, basierend auf den Schlagworten, die andere Nutzer dieser Ressource bisher gegeben haben. In der Empfehlung steht unter anderem das Schlagwort „Java“ und weist den Nutzer daraufhin, dass das Rahmenwerk ebenfalls in Java implementiert wurde.

Im Szenario besucht der Berater anschliessend einen Kunden und möchte ihm das Rahmenwerk vorstellen. Dafür loggt sich der Berater bei *Delicious* ein, gibt im Suchfeld die Schlagwörter *MVC* und PHP ein und findet den gewünschten Link mit zwei bis drei Mausklicks. Der Kunde sieht die Liste der Lesezeichen unter den Schlagworten *MVC* und PHP und möchte aus Eigeninteresse ebenfalls noch einige Webseiten dazu anschauen. Der Berater gibt dem Kunden daraufhin den diesbezüglichen Link zu all seinen öffentlich verfügbaren, gesammelten Lesezeichen.

Ein halbes Jahr später wird der Berater die Webseite für das initiierte Java-Projekt ebenfalls finden, da er das Rahmenwerk mit dem Schlagwort „Java“ versehen hat. Zudem wird er auf weitere Lesezeichen aufmerksam gemacht, welche von anderen Nutzern unter demselben Schlagwort gespeichert wurden. Vielleicht sind zwischenzeitlich bessere Rahmenwerke für die Anforderungen an die zu erstellende Software verfügbar. Falls der Nutzer auch in Zukunft Informationen erhalten möchte, welche neuen Lesezeichen zu den ausgewählten Schlagworten hinzugefügt wurden, kann er bei *Delicious* einen entsprechenden Benachrichtigungsdienst abonnieren.

Die *Social-Bookmarking*-Applikation *Delicious* ist ein hervorragend geeignetes Beispiel, um den Mehrwert einer Applikation aufzuzeigen, die über viele Nutzer verfügt: Je mehr Nutzer *Delicious* verwenden, indem sie Webseiten/-ressourcen mit Schlagworten versehen, desto grösser wird der Nutzen für alle Anwender, da sie Zusatzinformationen und Hilfestellung durch das Wissen Anderer bekommen. Die Applikation ist aus technologischer Sicht nichts Besonderes, die Inhalte und die

Nutzer geben diesem System den Mehrwert gegenüber herkömmlichen Lesezeichen-Sammlungen.

Andere Applikationen, wie z.B. *YouTube*TM, haben auf ihren Webseiten einen direkten Link auf den *Delicious*-Dienst, welcher es dem Nutzer erlaubt, Inhalte von *YouTube*TM direkt mit einem *Delicious*-Lesezeichen zu versehen. Dieser Dienst wurde zudem für viele verschiedene Endgeräte zugänglich gemacht, welche nicht auf *Browser* zurückgreifen. Damit wird den Nutzern von überall der Zugriff auf ihre Lesezeichen-Sammlung ermöglicht.

Wikipedia

Die letzte ausgewählte Web 2.0-Applikation gilt gleichzeitig als die bekannteste Plattform und wird im folgenden Abschnitt anhand der zuvor beschriebenen Muster von O'Reilly (2007) vorgestellt und diskutiert.

*Wikipedia*¹¹ ist eine benutzergenerierte Online-Enzyklopädie. Der Name *Wikipedia* setzt sich aus den beiden Begriffen *Wiki* (der Name einer Webapplikation zur effizienten Editierung von Online-Webseiten/-ressourcen, basierend auf dem Hypertext-System) und *Encyclopedia* (Enzyklopädie) zusammen. Der Hauptunterschied zu herkömmlichen Online-Enzyklopädien besteht darin, dass sämtliche redaktionellen Inhalte von Nutzern erstellt und kontrolliert werden. Auf bezahlte Autoren wird bewusst verzichtet. Beiträge mit rassistischen, hetzerischen oder sexistischen Inhalten können von den Nutzern per Löschantrag zur Überprüfung gesperrt werden. Kritische Inhalte werden von ausgewählten Nutzern überwacht, die über Administratorenrechte verfügen. Diese Rechte werden von engagierten Nutzern, welche sich durch ein grosses Engagement in die Qualitätssicherung und der Inhaltsgenerierung für ein Gremium qualifizierten, an eine Autorengemeinschaft vergeben.

¹¹ <http://de.wikipedia.org>, <http://en.wikipedia.org>

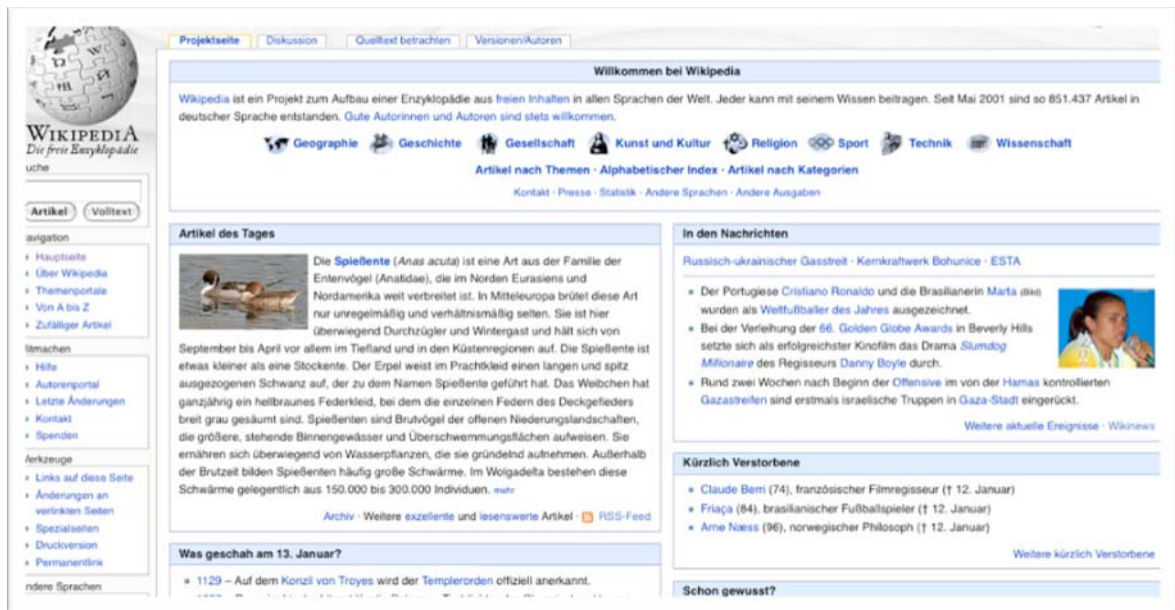


Abb. 6: Webseite von Wikipedia

Das *Open Source*-Projekt *Wikipedia* ging im Jahre 2001 aus dem von Jimmy Wales und Larry Sanger initiierten Projekt *Nupedia* hervor. Ziel dieses Projekts war es, eine konventionelle Online-Enzyklopädie mit Larry Sanger als Chefredakteur zu erstellen. 2001 wurden die Initianten auf das Wiki-Projekt aufmerksam und wollten das Wiki als Instrument zum kollaborativen Erstellen von Inhalten nutzen. Da das Wiki auf dem Internet frei verfügbar war – das Wiki-Prinzip gestattete die Mitarbeit ohne Registrierung – entstand eine Eigendynamik bei der Erstellung der Inhalte. In kürzester Zeit wurden Hunderte Artikel von Freiwilligen publiziert.

Einzelne Inhalte oder ganze Artikel können durch *Hyperlinks* miteinander verbunden werden. Die Autoren können diese Links generieren, indem sie eine doppelte eckige Klammer um einen Begriff setzen. Der Begriff wird dadurch als Verknüpfung im Artikel ersichtlich. Existiert zum verknüpften Begriff noch kein Artikel, erscheint der Link in roter Farbe und ein entsprechender Beitrag kann von freiwilligen Autoren erstellt werden. Zur Erstellung und Pflege der Inhalte gibt es in *Wikipedia* pro Beitrag eine Diskussionsseite, auf der die Nutzer über erstellte oder zu erstellende Inhalte diskutieren können. Des Weiteren können empfehlenswerte Artikel durch die *Wikipedia*-Nutzer gekennzeichnet werden. Derart häufig markierte Beiträge werden in der Liste „Exzellente Artikel“ aufgeführt und von der *Wikipedia*-Gemeinschaft

eingehender überprüft und gegebenenfalls auf einer dedizierten Seite als empfohlener Artikel aufgelistet.

Die grössten Herausforderungen, denen sich *Wikipedia* stellen muss, sind die Motivierung der Nutzer zur Generierung von Inhalten (Stoeckl et al., 2007), die Qualitätssicherung der Inhalte und die Verhinderung von Urheberrechtsverletzungen. Das Phänomen der Inhaltsgenerierung ohne persönlichen Nutzen ist v.a. von *Open Source*-Softwareprojekten bekannt. In Bezug auf *Wikipedia* wird die Hemmschwelle zur Beteiligung an der Inhaltsgenerierung durch einen einfachen Zugang zum System und zur Erstellung der Inhalte mit Hilfe eines einfach zu bedienenden Texteditors ermöglicht. Neben den genannten technischen Voraussetzungen zur aktiven Beteiligung sind Wissensbildung und Reputationsgewinn (etwa durch Nennung des Autors auf dedizierten Seiten der Artikel) als motivationsfördernde Faktoren zu beachten (Anthony et al., 2005). Angespornt durch potentiellen Reputationszuwachs, verfassen registrierte Autoren häufig qualitativ gute Beiträge (s. Kapitel 3.3). Des Weiteren steigt die Qualität der benutzergenerierten Inhalte mit der Anzahl der verfassten Beiträge eines registrierten Autors. Bemerkenswerter Weise aber werden die qualitativ besten Artikel anonym bereitgestellt, was in offensichtlichem Widerspruch zur Reputations-Hypothese steht (Anthony et al., 2005). Der Erstellungsprozess der Inhalte widerspiegelt das Prinzip der kollektiven Intelligenz, welches Surowiecki (2004) beschrieben hat. Ihm zufolge wird die Qualität der Beiträge durch die Anzahl der Autoren sowie deren Kompetenzen bestimmt.

Tabelle 2 zeigt die Zusammenhänge zwischen der Art des Benutzers (Nutzeridentifikation mit der Plattform *Wikipedia*), der Qualität der Inhalte und der Motivation der aktiven Nutzer.

Tab. 2: Zusammenhang der Identifikation des Nutzers mit *Wikipedia* und seiner Reputation nach Anthony (2005)

		Interesse an Reputation	
		Stark	Schwach
Grad der Identifizierung	Stark	Registrierte Nutzer Fanatiker & Experten Viele Beiträge	Anonyme Nutzer Viele Beiträge Schlechte Qualität
	Schwach	Registrierte Nutzer Wenige Beiträge Schlechte Qualität?	Anonyme Nutzer Wenige Beiträge Gute Samariter Hohe Qualität?

Das bereits erläuterte *Long Tail*-Prinzip findet sich auch bei *Wikipedia*: Viele Nutzer erstellen anonym einzelne Beiträge, relativ wenige Experten generieren viele Beiträge, sodass durch die Demokratisierung des Produktionsmittels *Wiki* und den Vertrieb der Informationen über das Internet der *Long Tail*-Effekt erzeugt wird .

Untersuchungen bezüglich des *Long Tail*-Effekts in *Wikipedia* haben gezeigt, dass nur ein kleiner Bruchteil der Nutzer sich aktiv an der Inhaltsgenerierung beteiligt (Ghosh & Prakash, 2000; Lerner & Tirole, 2002; Lakhani & von Hippel, 2002). Der Grossteil der Nutzer bleibt Konsument. Die aktiven Nutzer von *Wikipedia* können als Experten für die Inhaltsgenerierung angesehen werden und kennzeichnen sich durch eine idealistische Einstellung zu *Open Source*-Projekten sowie durch ähnliche Vorstellungen bezüglich der Qualität des öffentlichen Guts *Wikipedia* aus (Anthony et al., 2005).

In den letzten Jahren hat das Forschungsinteresse an *Wikipedia* stark zugenommen. Unterschiedliche Disziplinen haben eine Vielzahl von Untersuchungen durchgeführt, um u.a. herauszufinden, ob und wie selbstorganisierende Online-Enzyklopädien

funktionieren (Lih, 2004; Anthony et al., 2005; Lanier, 2006). Im Speziellen wurden die Motivation der Nutzer zur Mitwirkung an den Beiträgen, ihr Vertrauen in die Korrektheit der Beiträge sowie die Qualitätssicherung in einem selbstorganisierenden System untersucht.

Wiegand (2007, S. 136 ff.) formuliert dazu folgendes:

“Der erste gross angelegte Vergleich der deutschsprachigen Wikipedia mit den etablierten digitalen Nachschlagewerken Microsoft Encarta Professional 2005 und Brockhaus multimedial 2005 Premium erschien im Oktober 2004 in der Computer-Fachzeitschrift c’t (Ausgabe 21/04). Wikipedia erzielte dort im Inhaltstest die höchste durchschnittliche Gesamtpunktzahl, in der Kategorie Multimedia schnitt die freie Enzyklopädie dagegen schlecht ab [...] 2007 führte die c’t einen weiteren Vergleich zwischen Brockhaus, Bertelsmann, Encarta und Wikipedia durch, bei dem Wikipedia hinsichtlich Fehlerzahl nicht schlechter als die beiden kommerziellen Enzyklopädien abschnitt, in den Kategorien „Vollständigkeit“ und „Aktualität“ sogar vorne lag. Allerdings empfanden die Tester die Artikel der Wikipedia vielfach als „zu lang“, manchmal sogar „geschwafelt“.“

Im folgenden Abschnitt werden Ausschnitte aus *Wikipedia* (2008) bezüglich der Qualitätsvergleiche zwischen *Wikipedia* und anderen Enzyklopädien zusammengefasst:

In der Zeitschrift *Nature* wurde 2005 ein Vergleich zwischen der englischen *Wikipedia* und der *Encyclopædia Britannica* publiziert. Es wurden von 50 Experten je ein Artikel aus den beiden Werken auf Fehler geprüft. *Wikipedia* lag mit durchschnittlich vier Fehlern pro Artikel knapp hinter der *Encyclopædia Britannica*, in der im Durchschnitt drei Fehler gefunden wurden.

Im Dezember 2007 veröffentlichte die Zeitschrift Stern einen Test zwischen *Wikipedia* und der Online-Ausgabe des Brockhaus. Es wurden 50 zufällig ausgewählte Einträge zu diversen Themen, wie z.B. Politik, Wirtschaft und Sport vom Informationsdienst Wind GmbH überprüft. *Wikipedia* erreichte eine Durchschnittsnote von 1,7 nach deutschen Schulnoten. Die Brockhaus-Einträge zu den gleichen Stichworten kamen auf eine Durchschnittsnote von 2,7. Dabei wurden 43 Artikel der *Wikipedia* besser benotet als die kostenpflichtige Konkurrenz.

Die Qualität der Inhalte auf *Wikipedia* kann offenbar mit der anderer konventionell erstellter Enzyklopädien mithalten, jedoch sollten *Wikipedia*-Nutzer die dort publizierten Informationen mit gebotener Vorsicht verwenden und zusätzlich andere Informationsquellen zur Bestätigung heranziehen.

Zur Übersicht werden die diskutierten Applikationen anhand der von Tim O'Reilly postulierten Prinzipien vergleichend gegenübergestellt (Tab. 3).

Tab. 3: Web 2.0-Beispiele und Web 2.0-Prinzipien

	1	2	3	4	5	6	7	8
YouTube	x	x	x	x	x	x	x	x
Facebook		x		x	x	x	x	x
Delicious		x	x	x			x	x
Wikipedia	x	x	x	x	x			x

1. *Long Tail*-Effekt
2. Inhalte sind entscheidend
3. Kollektive Intelligenz
4. Netzwerkeffekt
5. Nicht alle Rechte an der Software und den Inhalten vorbehalten
6. Andauernde Beta-Phase der Software
7. Kooperation statt Kontrolle
8. Geräteunabhängigkeit

Den ausführlich erläuterten Vorteilen der beschriebenen Web 2.0-Applikationen stehen diverse Nachteile gegenüber, welche folgend überblicksweise diskutiert werden.

Das Zurverfügungstellen von Daten und Schnittstellen kann zu Datenschutzproblemen führen: Informationen werden miteinander verbunden und dadurch die Privatsphäre der Nutzer gefährdet. So könnten z.B. falsche Informationen über einen Nutzer in *Facebook* öffentlich publiziert und so dessen Ruf geschädigt werden. Es ist aufwändig, derartige Information vom Internet zu entfernen und praktisch unmöglich,

den guten Ruf des betroffenen Nutzers wiederherzustellen. In vielen Applikationen mit nutzergenerierten Inhalten ist zudem die Qualität der Inhalte mit Skepsis zu betrachten.

2.1.3 Zusammenfassung

Der aktuelle Aufschwung des Internets (Stichwort Web 2.0) wird u.a., aber nicht ausschliesslich durch die technologische Entwicklung der letzten Jahre ermöglicht. Die von Tim O'Reilly definierten Prinzipien deuten daraufhin, dass das Potential des Internets erst durch die Nutzung der Applikationen voll ausgenutzt werden kann (*Attention Economy*). Eine langfristige Bindung der Nutzer an entsprechende Applikationen wird durch Geräteunabhängigkeit und deren einfache Bedienbarkeit ermöglicht, sodass jederzeit Inhalte editiert und anderen Nutzern zur Verfügung gestellt werden können. Des Weiteren werden Inhalte miteinander verknüpft, damit aus Informationen Wissen entstehen kann. Dieser Trend zeichnet sich unter dem Schlagwort *Semantic Web* (Web 3.0) ab und wird auch als Web der nächsten Generation bezeichnet (s. Kapitel 7). Eine weitere wichtige Eigenschaft des Web 2.0 ist dessen Möglichkeit zur aktiven und passiven Personalisierung von Applikationen, um die Nutzer optimal zu unterstützen (Spring-Keller, 2006; Mödritscher et al., 2004; Foerster, 2004). Unter aktiver Personalisierung wird die Bedienungseinstellungsmöglichkeit der Applikation hinsichtlich der Präferenzen des Nutzers durch den Nutzer selbst verstanden, wohingegen in der passiven Personalisierung das System versucht, sich dem Nutzer und dessen Verhalten anzupassen.

Durch die Öffnung der Schnittstellen der Applikationen und das Zurverfügungstellen von Diensten ist es möglich, verschiedene Applikationen zu orchestrieren (*Mashups*) und daraus neue Applikationen/Dienste zu entwickeln. Markus Angermeier (2005) fasst die Prinzipien des Web 2.0 und ihre Zusammenhänge in der folgenden *Tag-Cloud* zusammen (Angermeier, 2005).



Abb. 7: Schlagworte des Web 2.0 nach Angermeier (2005)

In den genannten Beispielen bilden benutzergenerierte Inhalte den zentralen Teil der Plattformen, welcher über deren Erfolg oder Misserfolg bestimmt. Denn diese Inhalte ermöglichen durch die Kreativität der Nutzer emergente Unterhaltung und fördern die Reputation der Nutzer. Tom Alby (2008a, S. 109) formuliert dies folgendermassen:

„Wer etwas produziert/ausgräbt/als Erster entdeckt, das auch andere interessant finden, der darf stolz auf sich sein. Er bekommt Punkte. Sein Wort gilt in Zukunft mehr.“

2.2 Autorenwerkzeuge

"Die Technik soll nicht in den Mittelpunkt der Betrachtung gerückt werden, sondern letztendlich nur als Instrument dienen, die Lerninhalte zu transportieren, das Lernziel zu erreichen, die Kommunikation zu ermöglichen und die geeigneten Lehrmethoden zu implementieren."

Bremer, 2006

Mittels Autorenwerkzeugs sollen Nutzer Inhalte erfassen, editieren und anschließend publizieren können. Sie bilden einen wichtigen Teil zur Realisierung des *Long Tail*-Effekts (s. Kapitel 2.1.1). Beispielsweise erleichtern WYSIWYG-Editoren (What You See Is What You Get) das Editieren und Formatieren von Textbausteinen, sodass auch technisch wenig versierte Benutzer relativ leicht (Texte für) eigene Webseiten gestalten können.

Bislang ist es Forschern und Entwicklern offenbar noch nicht gelungen, sich auf eine allgemein gültige Definition des Begriffs Autorenwerkzeug zu verständigen. Dies wird u.a. auch dadurch erschwert, dass Autorenwerkzeuge stark vom jeweiligen Kontext und Einsatzzweck abhängen. Im Zusammenhang mit multimedialem E-Learning definiert Wache (2003) den Begriff wie folgt:

„Autorenwerkzeuge sind Programme zur multimedialen Aufbereitung von Lerninhalten und Entwicklung von Lernsoftware. Auch ohne besondere Programmierkenntnisse lassen sich damit einfache Anwendungen erstellen [...].“

Somit werden Autorenwerkzeuge im E-Learning-Kontext als Hilfsmittel verstanden, mit denen digitale Inhalte für E-Learning-Formate erstellt werden können. Zu derartigen Autorenwerkzeugen zählen sowohl Spezialprogramme für die elektronische Aufbereitung von Lerninhalten, welche lokal auf dem Computer (Computer-Based Trainings/CBT) oder via Internet (Web-Based Trainings/WBT) konsumiert werden können, als auch Software zur Erstellung multimedialer Inhalte (eLearn jku, 2006). Die derzeit gängigsten, meist genutzten Lernplattformen (s. Kapitel 2.3) verfügen nur selten über Funktionen zur Erstellung multimedialer Inhalte. Hierfür

werden externe Autorenwerkzeuge eingesetzt, die es den Autoren erlauben, Inhalte ohne besonderes Vorwissen in bestimmte Auszeichnungssprachen wie HTML oder XML einzupflegen (Baumgartner et al., 2002).

2.2.1 Anforderungen an Autorenwerkzeuge

Die nachfolgend vorgestellten Anforderungen an Autorenwerkzeuge sind von generellen Anforderungen an Software abgeleitet. Baumgartner (2004, S. 99) identifiziert dazu fünf zentrale Aspekte (Tab. 4).

Tab. 4: Anforderungen an Autorenwerkzeuge nach Baumgartner (2004)

Anforderung	Kernfragestellung
Interoperabilität	Wie reibungslos funktioniert die Zusammenarbeit zwischen dem Autorenwerkzeug und anderen Systemen? Wie gut lässt es sich in eine Lernumgebung integrieren?
Wiederverwendbarkeit	Können erstellte Inhalte auch problemlos in anderem Kontext und anderen Systemen verwendet werden?
Verwaltbarkeit	Wie können die Inhalte verwaltet werden?
Zugang	Können Nutzer jederzeit auf bestimmte Inhalte zugreifen?
Nachhaltigkeit	Funktioniert das Autorentool technologieunabhängig?

Bevor nun detaillierter auf die aufgelisteten Anforderungen an Autorenwerkzeuge eingegangen wird, soll der Begriff "multimedialer Inhalt" erklärt werden. So bezeichneter Inhalt setzt sich aus Informationseinheiten zusammen, welche Informationstypen, wie z.B. Text, Graphik, Animation, Audio oder Video aufweisen. Diese Informationseinheiten stehen in einer gewissen Abfolge zueinander, die allerdings je nach Interaktion des Nutzers von dem System individuell gesetzt werden kann. In multimedialen Beziehungsnetzwerken werden Informationseinheiten als Knoten und Verbindungen als Kanten definiert. Das Beziehungsnetzwerk resp. die Komposition

der Informationseinheiten legt somit die geometrische Struktur, den zeitlichen Verlauf und den Inhalt einer multimedialen Applikation fest. Es existieren nach Steinmetz (1993) drei Arten der Medienkomposition – räumliche, konfigurale und zeitliche:

Die **räumliche Komposition** beschreibt die Anordnung graphischer Elemente der Benutzerschnittstelle. Die Umsetzung erfolgt in ein konkretes Layout, welches die räumliche Ausprägung der Informationseinheiten erfordert.

Die **konfigurale Komposition** stellt die Kopplung von Informationseinheiten über Zugriffspunkte bereit.

Die **zeitliche Komponente** legt temporale Beziehungen zwischen Informationseinheiten fest. Zu ihrer Definition wird eine einheitliche Behandlung von diskreten und kontinuierlichen Informationseinheiten benötigt. Dies wird realisiert, indem diskrete Einheiten einem Zeitintervall zur Präsentation zugeordnet werden (Präsentationsintervall). Die Präsentationsintervalle sind zeitlich begrenzt. Die zeitliche Komponente kann auf drei verschiedene Arten realisiert werden. Dietrich Boles (1998) definiert diese wie folgt:

1. Komposition auf einer Zeitachse

Im Rahmen dieses Ansatzes werden die Informationseinheiten relativ zu einer Zeitachse (*timeline*) mit Start- und Endpunkt angeordnet, welche die Präsentationszeit darstellt. Die Informationseinheiten können nebeneinander auf der Achse platziert und so synchronisiert werden (Abb. 8). Die temporale Komposition auf einer Zeitachse wird auch absolute zeitliche Komposition bezeichnet.

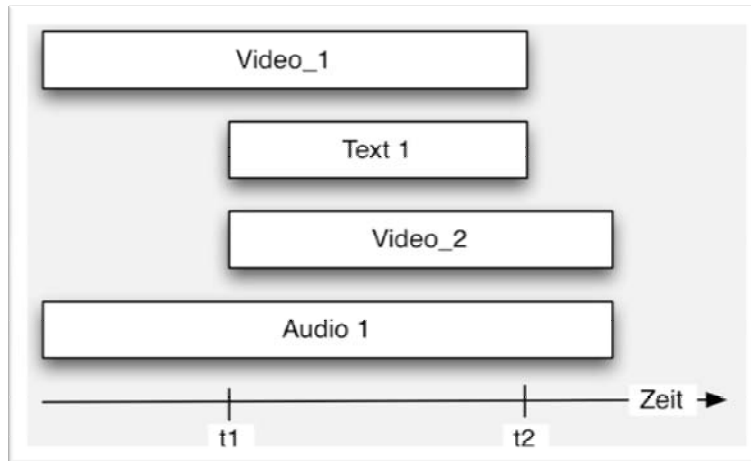


Abb. 8: Komposition eines multimedialen Inhalts auf einer Zeitachse

2. Hierarchische Komposition

Dietrich Boles (ebd.) beschreibt die hierarchische Komposition wie folgt:

„Bei diesem Ansatz werden sowohl die Informationseinheiten wie auch Zeitpunkt und Präsentationsintervalle in einer Baumstruktur angeordnet (Abb. 9). Die Wurzel und die inneren Knoten bestimmen dabei die zeitliche Anordnung der Präsentationsintervalle untergeordneter Knoten über die Intervallrelationen. Die Blätter symbolisieren die diskreten und kontinuierlichen Ausgaben der Informationsobjekte. Zudem können in Blättern und äußeren Knoten Zeitintervalle zur Beschreibung zeitlicher Verzögerungen stehen. Die hierarchische Komposition ist eine relative zeitliche Komposition.“

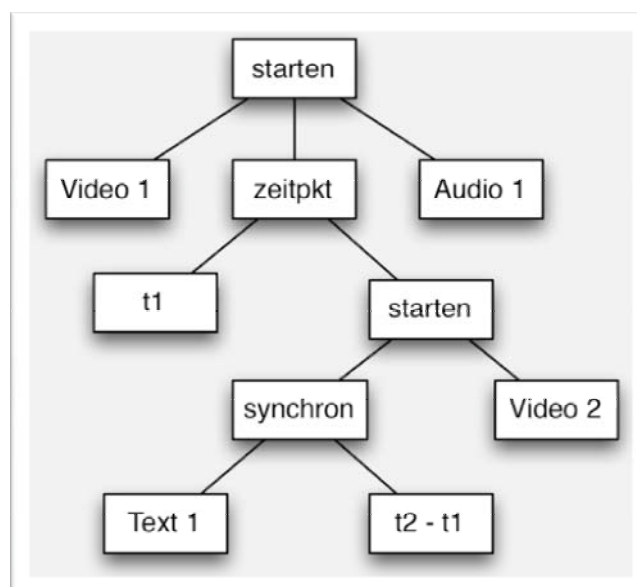


Abb. 9: Hierarchische Komposition

3. Komposition über Referenzpunkte

Dietrich Boles (ebd.) erläutert die Komposition über Referenzpunkte (Abb. 10) folgendermassen:

„Dieser Ansatz benutzt Synchronisations- oder Referenzpunkte (beliebige Zeitpunkte) der Präsentationsintervalle zur Bestimmung zeitlicher Beziehungen. Dabei werden Referenzpunkte der Informationseinheiten, welche gleichzeitig zu erreichend sind, in eine Liste eingetragen. Sobald nun ein solcher Zeitpunkt erreicht wird, werden die Informationseinheiten synchronisiert. Somit werden die Informationseinheiten resp. Datenströme in Beziehung zueinander gesetzt. Z.B. wird durch das Gleichsetzen der Startzeitpunkte von zwei Informationseinheiten der gleichzeitige Beginn zweier Präsentationen beschrieben. Die Komposition über Referenzpunkte stellt ebenfalls eine relative zeitliche Komposition dar.“

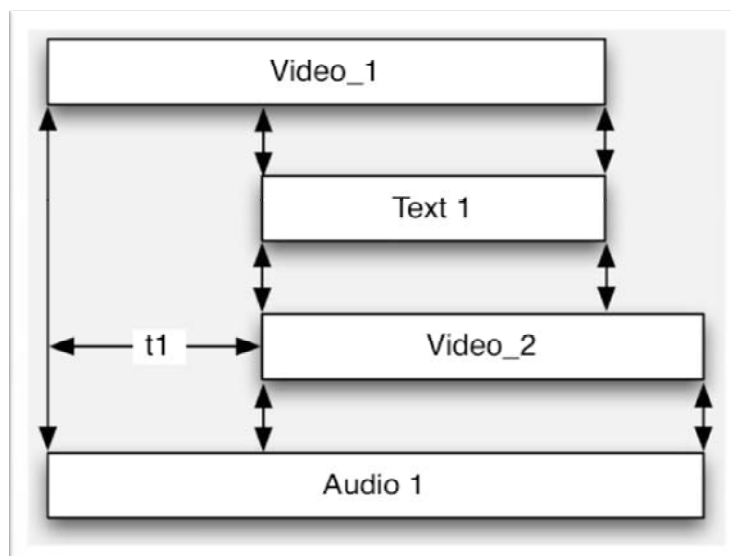


Abb. 10: Komposition über Referenzpunkte

Im Folgenden werden Autorenwerkzeuge zur Erstellung multimedialer Inhalte in Anlehnung an die Arbeit von Haefele & Maier-Haefele (2003) kategorisiert, die daraus resultierenden Anforderungen identifiziert und anschliessend im Zusammenhang mit E-Learning-Applikationen diskutiert. Ziel dieses Kapitels ist es somit nicht, einen vollständigen Vergleich der verschiedenen Eigenschaften von Werkzeugen durchzuführen. Vielmehr soll dieses Kapitel dazu dienen, die Konzepte für diese

Arbeit geeigneter Autorenwerkzeuge und deren Eigenschaften zu präsentieren. Anzumerken ist weiterhin, dass die meisten vorgestellten Autorenwerkzeuge nicht vom Autor persönlich getestet und evaluiert, sondern sämtliche Informationen zu Bedienung und Funktionalitäten weitgehend von Webseiten, Diplomarbeiten sowie aus Broschüren der Hersteller zusammengesucht wurden. Es wird kein Anspruch auf Vollständigkeit erhoben.

Nach Haefele & Maier-Haefele (2003) lassen sich Autorenwerkzeuge zur Generierung multimedialer Inhalte in sechs Kategorien einteilen. Fünf davon sind für den Kontext E-Learning relevant und werden darum detailliert beschrieben.

Professionelle Werkzeuge

Professionelle Werkzeuge wurden ursprünglich für komplexe Multimedia-produktionen, wie z.B. CD-ROM-Produktionen entwickelt. Sie zeichnen sich durch eine integrierte Skript- oder Programmiersprache aus, um das Werkzeug möglichst flexibel einsetzen und jeglichen multimedialen Inhalt erstellen zu können. Professionelle Werkzeuge werden v.a. für das Erstellen von Quizzes, Multiple-Choice-Aufgaben, Lernspielen, standardisierten Lerninhalten und Videos verwendet. Die Mächtigkeit der Werkzeuge geht jedoch auf Kosten der vom Autor/Nutzer aufzuwendenden Einarbeitungszeit und einer einfachen Bedienbarkeit.

Eines der bekanntesten professionellen Autorenwerkzeuge ist *Adobe® Flash®*¹², welches den Industriestandard bei der Erstellung und Bereitstellung wirkungsvoller Rich-Media-Inhalte und -Applikationen für Desktopcomputer und zahlreiche andere Endgeräte verkörpert. Mit Hilfe von *Flash®* können Inhalte auf einer Zeitleiste komponiert und mit Ereignissen kombiniert oder mit der mächtigen, dynamischen, objekt-orientierten Skriptsprache *ActionScript 3* implementiert werden. *Flash®* war ursprünglich nur als Daten-Format für vektorbasierte Graphiken im Internet entwickelt worden und wurde durch das *Flash® Player-Plugin* für Internetbrowser bekannt. Dieses wurde in den letzten Jahren deutlich erweitert und verbessert und gilt aktuell als Quasi-Standard für die Präsentation interaktiver, multimedialer Inhalte auf dem Internet. Heute wird die Technologie v.a. für animierte Online-Werbung sowie

¹² <http://www.adobe.com/products/flash>

Online-Spiele verwendet. Zudem wurde ein proprietäres Format für Videoinhalte entwickelt, welches sich in *Flash*®-basierten Applikationen einbetten lässt. U.a. konvertiert *YouTube*™ hochgeladene Videos in dieses Format. Das *Flash*® *Plugin* 8 ist mittlerweile weltweit auf über 98% der Computer mit Internetzugang installiert (Adobe.com, 2008).

Standardeditoren mit *Plugins*

Standardeditoren sind z.B. WYSIWYG-Editoren zur Erstellung von Webseiten. Sie stellen dem Nutzer eine graphische Schnittstelle ähnlich der eines herkömmlichen Textbearbeitungsprogramms zur Verfügung und konvertieren die Eingaben des Nutzers im Hintergrund zu Code. Der Nutzer muss somit z.B. nicht über HTML-/XML-Kenntnisse verfügen, um Webseiten-Quellcode generieren zu können.

Die meisten Editoren können um *Plugins* erweitert werden, die weitere Funktionalitäten bereitstellen, z.B. um standardisierte Inhalte mit Hilfe von Vorlagen zu Erstellen. So können SCORM-Inhalte¹³, Quizzes und Bewertungsvorlagen einfach konfiguriert werden.

Solche Standardeditoren liefern jedoch meist nur eingeschränkte Möglichkeiten, multimediale Inhalte zu editieren. Die Integration von Bildern, Audio und Video ist in den meisten Fällen jedoch möglich. Ein Beispielprogramm dafür ist *Adobe's*® *Dreamweaver*®, welches das Editieren, Publizieren und Verwalten von Webseiten unterstützt. Inhalte können entweder per Drag & Drop oder via integriertem Editor für HTML, Javascript oder andere Skriptsprachen erstellt werden. Zudem unterstützen diverse *Plugins* für *Dreamweaver*® den Nutzer z.B. bei der Generierung dynamischer Webseiten.

¹³ Akronym für Shareable Content Object Reference Model (SCORM, 2006)

Autorensysteme der neuen Generation

Werkzeuge der neuen Generation zeichnen sich durch einen geringen Aufwand zur Einarbeitung aus und werden darum auch *Rapid-Development-Tools* genannt. Wie im Falle anderer Editoren auch, brauchen Autoren keine Programmierkenntnisse zur Erstellung der multimedialen Inhalte und können dank integrierter Vorlagen innert kürzester Zeit ansprechende interaktive Inhalte erstellen (Fikentscher & Hlava-Landeck, 2005).

Diese Autorenwerkzeuge übernehmen auch den Export und das Publizieren und entlasten dadurch die Autoren bei der Distribution der Inhalte.

Content Converter

Content Converter sind Werkzeuge, welche Inhalte von einem Format in ein anderes umwandeln. Dabei werden die Inhalte zuerst hinsichtlich ihrer Struktur und ihres Aufbaus analysiert. Die Informationen über die Präsentation und den Inhalt werden extrahiert und in ein geeignetes Zwischenformat gespeichert. Anschliessend werden die Inhalte mit den Informationen zur neuen Präsentation/Formatierung verknüpft. Folgendes Beispiel soll dies illustrieren: Ein Nutzer verfasst einen Text im *Rich Text Format*. Der Converter analysiert diese Datei und extrahiert ihre Struktur und Daten in eine XML-Datei. Diese wird daraufhin mit XSL-Transformation zu einem HTML-Dokument und somit in die ideale Publikationsform für das Internet konvertiert.

Live-Recording-Systeme und Screen Movie-Recorder

Unter Live-Recording-Systemen werden Autorenwerkzeuge zusammengefasst, mit denen Präsentation aufgenommen, aufbereitet und bereitgestellt werden können. Dazu erstellt der Nutzer eine Folienpräsentation (z.B. mit *Microsoft® Powerpoint* oder *Apple Keynote*), welche während der Präsentation vom Autorenwerkzeug aufgezeichnet wird. Der Vortragende trägt ein Mikrofon, um verbale Erläuterungen synchron aufzunehmen. Zudem können weitere Informationsquellen, wie z.B. Video oder ein White-/Smartboard miteinbezogen werden. Nach der Aufnahme kann der Inhalt weiter verarbeitet oder exportiert werden.

Screen Movie-Recorder werden in den letzten Jahren von immer mehr Internetnutzern/-produzenten verwendet, um z.B. Tutorials, Howtos, Produktpräsentationen

oder Vorträge aufzuzeichnen. Dabei wird die Ausgabe des Bildschirms aufgezeichnet. Zusätzlich kann der Nutzer seine Aktionen über ein Mikrofon kommentieren. Das Werkzeug konvertiert anschliessend das Bild- und Audiomaterial zu einem Video. Dieses kann der Nutzer über einen Dienst wie z.B. *YouTube*TM veröffentlichen oder auf eine DVD brennen.

Die grösste Herausforderung, welcher sich die Entwickler und Designer heutiger Autorenwerkzeuge stellen müssen, ist die Identifizierung des Zielpublikums der Applikation. Denn informationstechnologie-affine Anwender verfügen über ein anderes konzeptuelles Verständnis von der Produktion und Anwendung von IT als nicht-affine. Somit ist ein Gleichgewicht zwischen möglichst vielen Funktionalitäten, der Mächtigkeit des Werkzeugs und der Benutzerfreundlichkeit für die Zielgruppe herzustellen, damit keine langen Einarbeitungszeiten entstehen (Boles, 1998).

2.2.2 Beispiel: *Scratch*

Im folgenden Abschnitt wird mit *Scratch* ein Autorenwerkzeug vorgestellt, welches am Massachusetts Institute of Technology (MIT) unter der Leitung von Mitchel Resnick erstellt worden ist (vgl. Resnick, 2003, Maloney et al., 2003) und ein Bindeglied zwischen Autorenwerkzeug und E-Learning darstellt. Ziel dieses Werkzeugs ist das Erlernen von Algorithmen, Programmierparadigmen und die Sensibilisierung auf Informationstechnologie (s. Kapitel 2.5.1). Es gilt als Grundlage für die Entwicklung evolutionärer Lernspiele. Das Adjektiv „evolutionär“ soll dabei – ganz im Sinne des Web 2.0-Trends – die iterative Erweiterung und Verbesserung der Lernszenarien durch die Nutzer ausdrücken.

Scratch integriert eine dynamische, visuelle Programmiersprache und gehört in die Kategorie „Autorensysteme der neuen Generation“. Dieses Werkzeug dient zur Aufbereitung von multimedialen und interaktiven Inhalten, wie z.B. Animationen oder Computerspielen. Dafür können vom Inhaltsersteller eigene multimediale Ressourcen importiert, bestehende Objekte aus der *Scratch*-Bibliothek verwendet oder per integriertem Zeichnungswerkzeug erstellt und weiterverarbeitet werden. Somit ist es möglich, eigene Ressourcen wiederzuverwerten und mit Ressourcen von

Scratch zu kombinieren. Diesen Kombinationsmöglichkeiten verdankt das Werkzeug seinen Namen: *Scratch* beschreibt im Hip-Hop eine Technik zum Mixen von Musik.

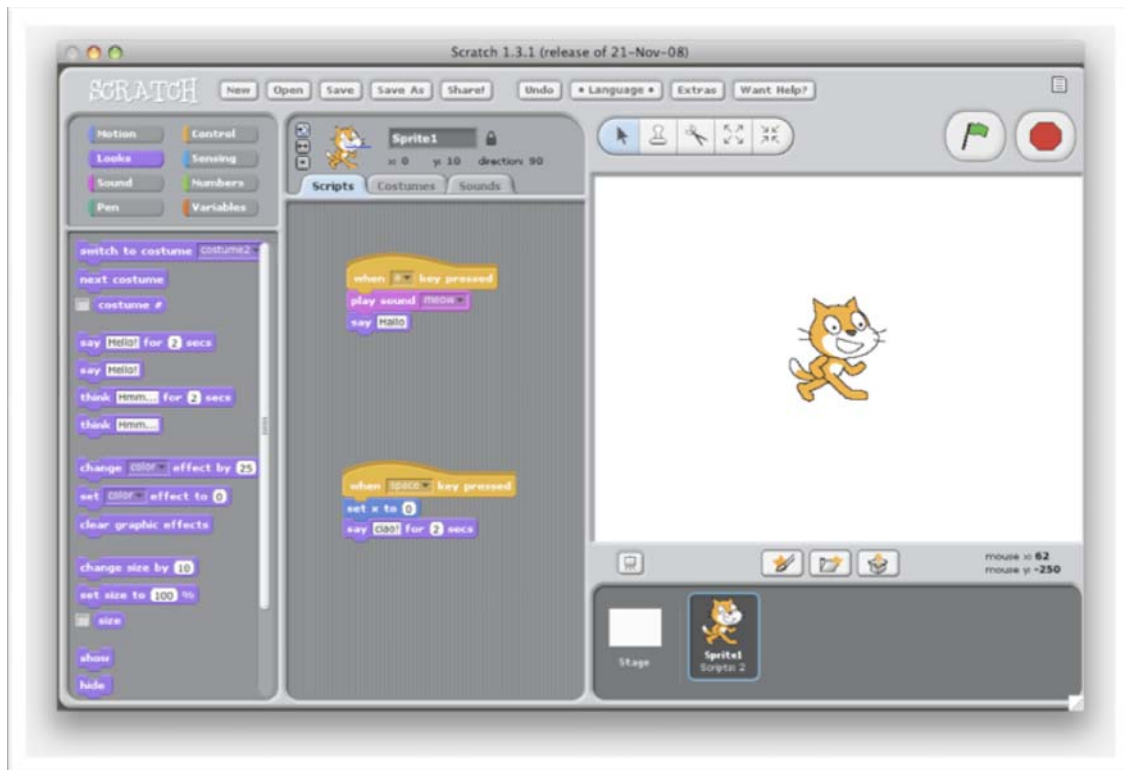


Abb. 11: Autorenwerkzeug *Scratch*

Mit Hilfe visueller Programmierbausteine, die entweder grundlegende Programmieranweisungen (z.B. If/else-Blöcke oder While-Schleifen) oder Ereignisse beinhalten, können interaktive Animationen erstellt werden (Abb. 11). Jedem importierten graphischen Element können solche Programmierbausteine zugeordnet und diese via Schlüssel-Schloss-Prinzip zusammengestellt werden. So entstehen kombinierte Multimedia-Objekte (Boles, 1998), welche miteinander komponiert werden können. In Abbildung 11 ist rechts auf dem Bildschirm eine Katze zu sehen. Durch Drücken der Taste "A" auf der Tastatur (= Ereignis) werden zunächst ein Ton und anschließend eine Sprechblase mit dem Text "Hallo" ausgegeben. Diese Animation, ausgelöst durch eine Benutzereingabe, wird durch das Zusammenfügen dreier Bausteine ermöglicht: Der erste Baustein definiert das Ereignis. Der zweite Baustein spielt nach eingetretenem Ereignis den zuvor als Parameter definierten Ton ab, und der letzte Baustein gibt die Sprechblase mit dem Text als Parameter aus. Die Bausteine werden in das Skriptfeld gezogen (Abb. 11, Mitte) und können in der aufgezählten

Reihenfolge wie Puzzleteile in einem Legespiel angeordnet und miteinander verknüpft werden, sodass die beschriebene Animation durch das Drücken der Taste "A" initiiert wird.

2.3 Lernplattformen

Lernplattformen sind Softwareapplikationen, welche das Bereitstellen und Verwalten von Lerninhalten sowie die Organisation von Lernvorgängen unterstützen (Baumgartner et al., 2002). Als wichtiger Bestandteil einer Lernplattform (engl. *Learning Management System*) gilt eine Benutzerverwaltung mit diversen Auswertungsmöglichkeiten bezüglich des Lernverhaltens und Wissenszuwachs' der Nutzer. Zu den Schlüsselfunktionalitäten gehören die Kommunikations- und Kollaborationsmöglichkeiten zwischen Lernenden und Lehrenden: Die Kommunikation verläuft in den meisten Fällen asynchron entweder über ein Diskussionsforum oder über ein integriertes E-mailsystem. Dadurch können die Lehrkräfte im Gegensatz zum Frontalunterricht persönlich auf eine grössere Anzahl von Lernenden eingehen und allfällige Fragen beantworten.

Ein Anwendungsfall könnte wie folgendes *Blended Learning*-Szenario aussehen: Ausgangspunkt ist eine Lehrveranstaltung, die sich aus klassischen Frontalunterrichts- sowie Online-Lehreinheiten zusammensetzt. Die Lehrperson publiziert ihre Lerninhalte und Tests auf der Lernplattform. Die Kursteilnehmer werden automatisch vom System über neue Inhalte benachrichtigt, rufen die bereitgestellten Lernmaterialien auf und setzen sich mit ihnen auseinander. Im Anschluss daran können sie im integrierten Forum, Chat oder per E-Mail gegebenenfalls anderen Lernenden oder der Lehrperson Fragen stellen. Mit Hilfe schriftlicher Dokumentationen oder Online-Tests der Nutzer kann die Lehrperson danach das Lernergebnis oder gar den Lernzuwachs jedes einzelnen Kursteilnehmers begutachten und im Frontalunterricht auf individuelle Verständnisprobleme eingehen.

Lernplattformen eignen sich zusätzlich zur Kollaboration unter Lernenden, um z.B. Gruppenarbeiten durchzuführen oder zu verwalten. Ziel der Nutzer ist es, gemeinsam, virtuell Aufgaben bearbeiten können. Ausserdem sollen erstellte Inhalte kommentiert, annotiert oder bewertet werden können.

In der Zusammenfassung machen folgende generellen Eigenschaften eine gute Lernplattform aus:

- Benutzerverwaltung mit Evaluationsmöglichkeiten
- Lerninhalts- und Kursverwaltung
- Rollen- und Rechtevergabe für einzelne Nutzer oder Gruppen
- Möglichkeit zur Kommunikation mit anderen Lernern sowie mit Lehrenden
- Unterstützungswerkzeuge für die Lernverankerung (z.B. Online-Tests inkl. Auswertungen)
- Präsentation der Inhalte

2.3.1 Rollenspezifische Anforderungen an Lernplattformen

Die Nutzer von Lernplattformen nehmen unterschiedliche Rollen ein, deren adäquate Ausübung neben den zuvor genannten generellen Aspekten auch die Erfüllung rollenspezifischer Anforderungen voraussetzt. So unterscheiden sich die Forderungen der Lehrenden an eine Lernplattform mitunter von denen der Lernenden, wie aus Tabelle 5 ersichtlich ist.

Tab. 5: rollenspezifische Anforderungen an Lernplattformen

Anforderung	Lernende	Lehrende
Intuitive Benutzerschnittstelle und Werkzeuge ohne lange Einarbeitungszeiten	notwendig	notwendig
Einfache oder gar keine Installation von Software client-seitig, damit die Hemmschwelle zur Nutzung des Systems möglichst tief ist und ortsunabhängig zugegriffen werden kann	wünschenswert	wünschenswert
Schnelles Auffinden von relevanten Inhalten	notwendig	notwendig
Personalisierung des Systems in Abhängigkeit des Lerntyps	wünschenswert	wünschenswert
Kommunikationsmöglichkeiten mit anderen Nutzern	notwendig	notwendig

Evaluationsmöglichkeiten der einzelnen Nutzer/Lernenden	-	notwendig
Standardisierung der Lerninhaltsformate, um diese in anderen Lernplattformen wiederverwenden zu können	-	wünschenswert
Kategorisierung der Informationen zum einfachen Auffinden der Inhalte	wünschenswert	wünschenswert
Verwaltung von mehreren Lehrveranstaltungen und die Zuordnung der Teilnehmer zu den Veranstaltungen	-	notwendig
Qualitätsmanagement-System zur Abbildung von Kontrollmechanismen	-	notwendig

Lernplattformen werden insbesondere an Hochschulen eingesetzt, um Lerninhalte für eine grosse Anzahl von Studenten bereitstellen zu können. Die Studenten können so Informationen, welche sie während des Frontalunterrichts erhalten, repetieren oder sich auf Prüfungen vorbereiten. Zudem können Online-Tests durchgeführt werden, welche sowohl den Lehrenden, als auch den Lernenden selbst Anhaltspunkte geben, wie gut die Lerninhalte verstanden worden sind. Immer häufiger werden die Lehrveranstaltungen auch auf Video aufgezeichnet und anschliessend als Ressourcen auf der Lernplattform publiziert, damit Nutzer die Lehrveranstaltungen zeit- und ortsunabhängig konsumieren können.

Wissenschaftliche Studien konnten zeigen, dass die Nutzer den Einsatz von Lernplattformen nicht im gewünschten Masse akzeptieren, wenn diese einerseits nicht adäquat in die entsprechenden Lehrveranstaltungen integriert sind und andererseits das System als Informationsquelle Schwächen aufweist (vgl. Bürg & Mandl, 2005). Des Weiteren haben sich in den letzten Jahren neue Lernmethoden, wie z.B. *Game-based Learning* entwickelt, welche Lernende besser in ihren individuellen Lernprozessen unterstützen sollen als konventionelle Methoden (Prensky, 2003). Diese

neuen Methoden können jedoch nur zum Teil durch konventionelle Lernplattformen unterstützt werden (s. Kapitel 2.5).

2.3.2 Beispiel: OLAT

OLAT ist eine webbasierte *Open Source*-Lernplattform, welche an der Universität Zürich entwickelt wurde und an mehreren Hochschulen in Europa im Einsatz steht. Das Projekt für die Entwicklung von OLAT (Online Learning And Testing) wurde 1999 initiiert. Unterdessen liegt sie als modulare Plattform mit Unterstützungsmöglichkeiten für diverse E-Learning-Standards, wie z.B. SCORM, IMS Content Packaging oder IMS QT vor.

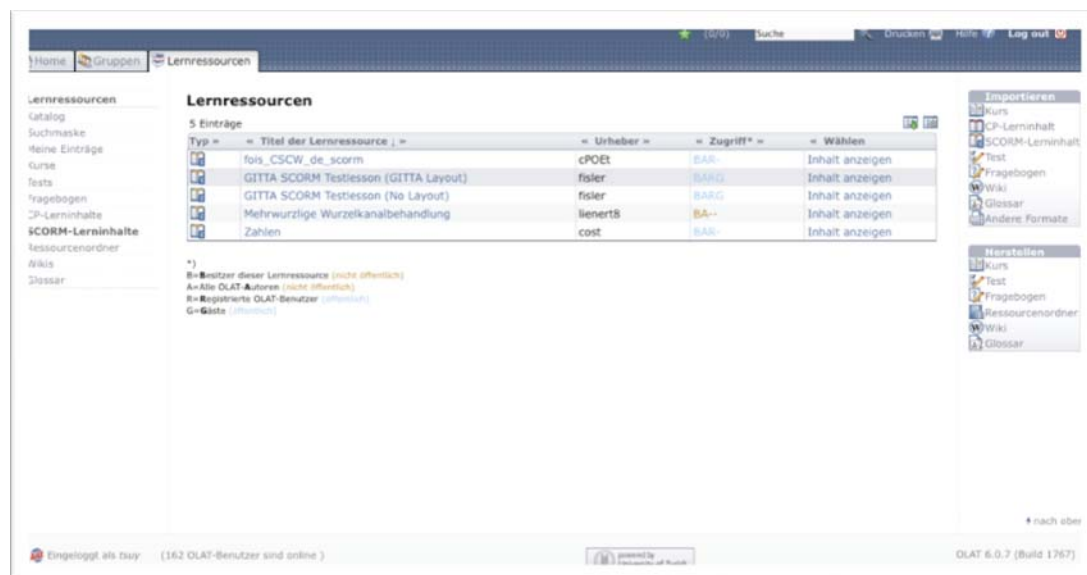


Abb. 12: Verfügbare, personalisierte Lernressourcen auf OLAT

OLAT wird an der Universität Zürich als Werkzeug zur Online-Unterstützung von Lehrveranstaltungen eingesetzt und erlaubt Studierenden, sich in Gruppen zusammenzuschliessen, Diskussionen in Foren zu führen und den integrierten Chat zu nutzen. Die Lehrbeauftragten können OLAT für die Materialdistribution, Aufgabenverteilung, Lösungsabgaben und Verwaltung von Leistungsnachweisen verwenden. Die Inhalte, z.B. einfache interaktive Materialien, Folien für die Vorlesungen, Seminare oder Übungen, können ebenfalls mit OLAT verwaltet werden. Des Weiteren stehen Schnittstellen zur Verfügung, die Anbindungen an andere Plattformen ermöglichen.

Im Überblick bietet OLAT folgende Funktionalitäten (OLAT, 2008):

- Einfache, personalisierte Benutzeroberfläche
- Didaktische Freiheit mit einem flexiblen Kurssystem inkl. Kurseditor
- Tests, Selbsttests und Fragebogen können mit OLAT-internem Editor erstellt und den Teilnehmern zur Verfügung gestellt werden
- Online-Bewertungsmöglichkeiten der Teilnehmer inkl. Leistungsnachweis für gesamten Kurs
- Einschreibung der Teilnehmer in Gruppen inkl. Gruppenmanagement
- Zugangsregelung (Anbindung an externe Informationssysteme)
- Groupwaretools, wie Diskussionsforen, E-Mailformulare, Chat, Dokumentenablage und Wiki
- Benachrichtigungsservice via E-Mail oder RSS
- Mehrsprachige Benutzerführung zum Einsatz im In- und Ausland

OLAT ist gegenüber anderen Lernplattformen einerseits dadurch eingeschränkt, dass Autoren ihre Online-Dokumente ausserhalb von OLAT z.B. mit HTML-Editoren erstellen müssen und andererseits dadurch, dass die Erstellung von Webseiteninhalten nicht unterstützt wird, da das System keine Funktionalitäten eines *Content Management Systems (CMS)* zur Verfügung stellt.

2.4 Digitale Unterhaltungsspiele

“Computer games have done more for the propagation of computers than any other software.”

Nolan Bushnell, Gründer von Atari (1972)

Digitale Unterhaltungsspiele haben sich in den letzten Jahren stark verbreitet. Dementsprechend wird erwartet, dass das weltweite Marktvolumen für Video- und Computerspiele in den nächsten Jahren um durchschnittlich 11,4 Prozent auf 46 Milliarden US-Dollar jährlich zunimmt (PriceWaterhouseCoopers, 2006). Bereits die bisherige Umsatzentwicklung ermöglichte eine rasante technologische Entwicklung der Videospielekonsolen und PCs, weshalb immer grössere Budgets für die Spielehersteller zur Verfügung stehen mussten, um die verbesserte Hardware ausreizen zu können. Es entstanden innovative Spielkonzepte und Geschäftsmodelle, welche die Umsätze nochmals steigern sollten. Zu Illustrationszwecken kann das *Massive Multiplayer Online Role-playing Game* (MMORPG) „World of Warcraft“ von *Blizzard Entertainment*® herangezogen werden (s. Kapitel 2.4.3): Das Spiel kann vom Internet heruntergeladen oder für ein vergleichsweise geringes Entgelt als CD-ROM gekauft werden. Für die Nutzung der virtuellen Online-Welt des Spiels muss der Spieler jedoch eine monatliche Gebühr bezahlen. Darüber hinaus bietet *Blizzard*® kostenpflichtige Erweiterungen der Grundversion an. Zurzeit spielen circa 11 Millionen Menschen weltweit dieses Spiel und zahlen monatliche Gebühren, folglich ist „World of Warcraft“ aktuell eines der lukrativsten Spiele auf dem Markt (Mmog-welt.de, 2008). Derzeit besonders gefragte Unterhaltungsspiele haben sich aus den ursprünglichen „Einzelspieler-Spielen“ (Spiele, welche nicht mit bzw. gegen andere Spieler, sondern mit bzw. gegen computergesteuerte/n Spielpartner/n bzw. -gegner/n) hin zu Online-Multiplayer-Spielen weiterentwickelt.

In den letzten Jahren ist die Unterhaltungsspielbranche dem Trend des Internets gefolgt und steht am Anfang der Entwicklung einer neuen Generation von Spielen, welche von Sony unter dem Schlagwort *Game 3.0* vermarktet wird. Phil Harrison, Präsident der weltweiten Sony-Spieleentwicklungsstudios und Vize-Präsident des Creative-Developments beim Sony-Konzern, versteht unter der dritten Generation

von Spielen – in Analogie zu Web 2.0 – die Kombination aus Multiplayer-Online-spielen und benutzergenerierten Inhalten. Somit wird der Konsument (Spieler) zum Co-Produzenten seiner eigenen Spiele und übernimmt einen Teil der Aufgaben (und Verantwortung) des Spieledesigners (Wonderlandblog.com, 2007).

2.4.1 Merkmale von Spielen

Digitale Spiele sind – wie analoge Spiele auch – durch Regeln, Ziele, Feedback, Herausforderung, Interaktion und eine Repräsentation definiert (Prensky, 2003; s. Tab. 6; Crawford 1984; Bateman & Boon, 2005). Beim Spieldesign ist darauf zu achten, diese Merkmale in Einklang zueinander zu bringen. Eine zu hohe Gewichtung einzelner Merkmale auf Kosten anderer machen Spiele unausgeglichen und dadurch mit der Zeit potentiell langweilig. Wird etwa auf Kosten der Hintergrundgeschichte die Repräsentation eines Spiels ausgebaut, um dem Spieler eine realitätsnahe, virtuelle Welt zu präsentieren, kann der Spieler unter Umständen das Interesse am Spiel infolge der fehlenden Verankerung der Hintergrundgeschichte verlieren (Björk & Holopainen, 2005).

Tab. 6: Übersicht der Merkmale von digitalen Spielen

Regeln	Regeln geben die Rahmenbedingungen und Grenzen des Spiels vor. Des Weiteren legen sie fest, welche Interaktionen erlaubt sind. Sie stellen den Hauptunterschied zwischen organisiertem und spontanem Spielen dar (Encyclopaedia Britannica, 2003).
Ziele	Ziele stellen für einen Grossteil der Spieler den wichtigsten Motivationsfaktor dar. Der Spieler kann bei angemessener Herausforderung bei der Zielverfolgung in einen als „Flow“ bezeichneten Zustand versetzt werden, welcher das absorbierte Eintauchen in die virtuelle Welt ermöglicht und die Bindung des Spielers an das Spiel verankert (Csikszentmihalyi et al., 1993).
Feedback	Im Spiel bekommt der Nutzer direktes Feedback vom System. Des Weiteren werden Auswertungen und Bewertungen nach dem Erreichen der Ziele präsentiert. Somit ist der Nutzer in der Lage, gemachte Fehler zu erkennen und zu korrigieren.

	Durch die <i>Trial-and-Error</i> -Methode kann sich der Nutzer im Ausschlussverfahren der optimalen Lösung zur Erreichung des gestellten Ziels annähern und aus seinen Fehlern lernen.
Herausforderung	Einer der wichtigsten strukturellen Faktoren digitaler Spiele ist deren Herausforderung. Der Spieler darf nicht über- oder unterfordert und somit frustriert werden. Durch Adaptivität und Personalisierung können die Herausforderungen an das individuelle Spielerprofil angepasst werden (Spring-Keller & Ito, 2007).
Interaktion	Digitale Computerspiele zeichnen sich durch ihre Interaktionsmöglichkeiten aus: Der Nutzer kann aktiv Entscheidungen fällen und dadurch den weiteren Verlauf des Spiels bestimmen. Im Gegensatz dazu ist der Nutzer beim Medium Fernsehen ein passiver Konsument. Die Interaktionen mit dem Spiel erlauben ein exploratives Lernen des Spiels (und möglicherweise dessen Lerninhalts) und lassen Spieler Erfahrungen sammeln.
Repräsentation	Ein digitales Spiel braucht eine Repräsentation, welche als Interaktionsschnittstelle zwischen Spieler und System zur Anzeige des aktuellen Spielzustandes dient.

In der *Encyclopaedia Britannica* wird – anders als im Deutschen – zwischen *play* und *game* unterschieden. *Play* bezeichnet dabei den Obergriff für die Aktivität des Spielens, welche wiederum in spontanes und organisiertes Spielen unterteilt wird. Ersteres zeichnet sich durch die fehlende Formulierung klarer Ziele und Regeln aus, wohingegen diese Merkmale bei organisiertem Spielen vorhanden sind, weshalb dieses dem Begriff *Game* gleichgesetzt wird. Des Weiteren können *Games* mit Wettbewerbscharakter in intellektuelle und physische Wettbewerbe unterteilt werden. Letzteren werden z.B. Sportarten zugeordnet (s. Abb. 13).

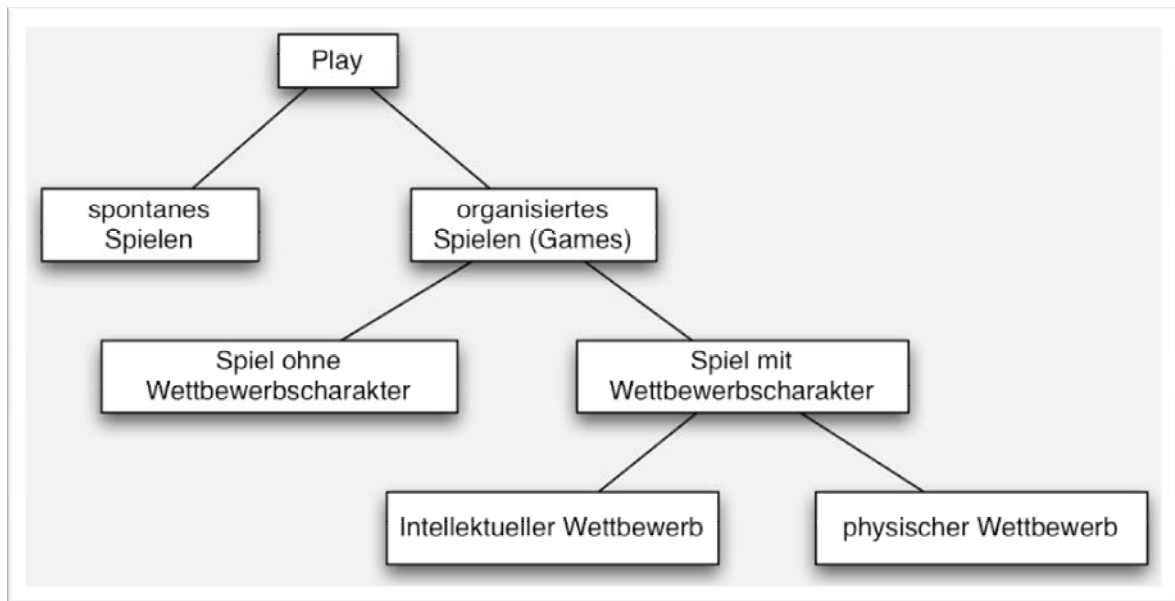


Abb. 13: Spiel vs. Game nach Encyclopaedia Britannica (2003)

Im Rahmen dieser Arbeit wird die Bezeichnung „Spiel“ im Sinne des englischen Begriffs *Game* verwendet.

2.4.2 Geschichte und Entwicklung der Computerspiele

Im Folgenden wird zunächst auf die Geschichte der Personal Computer (PC) und Spielkonsolen eingegangen und anschliessend ein Überblick über die verschiedenen Typen von Computerspielen gegeben. Die Geschichte der Computerspiele ist in sieben Phasen gegliedert. Dies repräsentiert jedoch nur eine von vielen möglichen Einteilungen und dient als grober Überblick über die Entwicklungen in der Computerspielbranche.

Die **erste Phase** datiert etwa im Zeitraum zwischen 1971 bis 1979. *Commodore®-PET* (engl. Anronym für *Personal Electronic Transactor*) und *Atari®-Konsolen* machten die ersten Gehversuche in der Computerspielbranche und legten einen wichtigen Grundstein für alle noch folgenden Konsolen und Spiele. Der *Commodore®-PET* war der erste Computer der Firma *Commodore®*, welche fast zehn Jahre lang eine sehr erfolgreiche Reihe von Bürocomputern herstellte. Sie wurde am Ende durch den *Amiga®* und die *Commodore®-PC-Serie* abgelöst. *Atari®* wurde 1972 von Nolan Bushnell gegründet, der zuvor mit *Nutting Associates®* die ersten Videospiele-

Automaten entwickelt hatte. Kurz darauf veröffentlichte *Atari*® das erste digitale Unterhaltungsspiel namens *Pong*, das seit 1977 als Heimvideospiele mit austauschbaren ROM-Modulen vermarktet wurde.

Die **zweite Phase** (1980 bis 1985), auch Goldenes Zeitalter genannt, beginnt mit der Veröffentlichung des *ZX Spectrum*, einem Heimcomputer, welcher von der Firma *Sinclair*® auf den Markt gebracht wurde und sich besonders in Grossbritannien grosser Beliebtheit als Spielcomputer erfreute. In dieser Phase wurde mit dem C64 auch einer der erfolgreichsten Heimcomputer dieser Zeit entwickelt. Die meisten Käufer des C64 wollten damit keine Programme schreiben, sondern Computerspiele spielen. Ebenso wurde in dieser Phase der Grundstein für das Geschäft mit den Spielhallenautomaten gelegt. Letztere brachten ihren Besitzern als ausgezeichnete Publikumsmagneten viel Geld ein.

Zwischen 1985 bis 1990 begann zudem die kommerzielle Erfolgsgeschichte von *Nintendos*® *Gameboy*™. Computerspielehersteller entdeckten in zunehmendem Masse den Markt der Heimcomputer (PC) als Plattform für digitale Spiele. Infolge der rasanten technologischen Entwicklung der Hardware, welche durch neue Verfahren und Technologien kontinuierlich preiswerter und leistungsstärker wurde, konnte der PC immer stärker als Spielcomputer genutzt werden.

Von 1990 bis 1993 fand der erste martialisch als „Konsolenkrieg“ bezeichnete Konkurrenzkampf zwischen *Nintendo*®'s *SNES*™ und *Sega*®'s *Megadrive* /*Genesis*™ statt, welcher nahtlos in einen zweiten Konsolenkrieg der Hersteller *Nintendo*®, *Sony*® und *Sega*® überging. Ab 1999 versuchte *Sega*® mit dem *Dreamcast*™ auf dem hart umkämpften Konsolenmarkt Marktanteile zu gewinnen. Jedoch war neben den Produkten von *Nintendo*® (*Game Cube*™), *Sony*® (*Playstation*™2) und *Microsoft*® (*Xbox*™) offenbar kein Platz für weitere Anbieter, weshalb sich dieses Produkt in der Gunst der Kunden nicht langfristig durchsetzen konnte.

Ab 2005 stellten alle drei grossen Konsolenhersteller ihre jeweils neue Generation von Konsolen vor. Dabei setzten sie sowohl in puncto Graphik, als auch in puncto

Interaktionsmöglichkeiten neue Massstäbe und bieten erstmals Online-Anbindungen an das Internet. *Nintendo*®'s *Wii*™ fokussiert v.a. auf Party-Games (s. Abschnitt „Genre“ dieser Arbeit), welche via eines innovativen Eingabegeräts gespielt werden. Das Eingabegerät kommuniziert per *Bluetooth*-Technologie mit der Konsole und verfügt über einen Beschleunigungssensor, der Bewegungen und Drehungen des Eingabegeräts erfasst und eine mechanische Rückmeldung über Vibrationen im Eingabegerät erzeugt. In die Konsole wurde dafür eine Infrarotkamera integriert, welche die Position und Lage des Eingabegeräts ermittelt und die so gewonnenen Informationen in spezielle graphische und funktionelle Spieleigenschaften umsetzt. Ein derartiges Eingabegerät ermöglicht innovative Spielkonzepte und lässt bisher ungeahnte Benutzerinteraktionen zu. Auf eine möglichst realitätsnahe Darstellung der virtuellen Welten wurde hingegen bewusst verzichtet, um sich dadurch eindeutiger vom grössten Konkurrenten *Sony*® abgrenzen zu können. Dessen Konsole *Playstation*™ 3 setzt zwar auf Altbewährtes in Sachen Eingabegerät, allerdings auch neue Massstäbe bezüglich Prozessor- und Graphikleistung in Konsolen. Mit der *Playstation*™ 3 wird eine detailgetreue Annäherung der virtuellen Welten an die Realität bezüglich Photorealismus und physikalischer Eigenschaften versucht. Das Produkt *XBox 360*™ von *Microsoft*® hat sich als dritte Konsole auf dem Markt etabliert, besitzt jedoch keine besonders hervorzuhebenden Eigenschaften gegenüber seinen Konkurrenten.

2.4.3 Genres

Trotz der grossen Auswahl inhaltlich verschiedener Computerspiele gibt es sehr viele Spiele ähnlichen Konzepts (engl. *Game Play*), so dass sich Ansätze zur Kategorisierung von Spielen als sinnvoll herausgestellt haben (Björk & Holopainen, 2005). Neben der Einordnung nach *Game Plays*, kann auch noch nach *Genres* unterschieden werden, wobei *Genre* auf eine Beschreibung des narrativen Inhalts des Spieles abzielt (Grace, 2005). In dieser Arbeit wird, wie in der Unterhaltungsspielbranche üblich, das *Game Play* und das *Genre* unter dem Begriff *Genre* zusammengefasst. In der folgenden Tabelle werden die wichtigsten *Genres* dargestellt. Die Namen der *Genres* sind in englischer Sprache belassen, da diese auch im Deutschen verwendet werden.

Tab. 7: Übersicht über Genres digitaler Spiele nach Ito (2005)

Name des Genres		Beschreibung
Diese Genres werden oft auch als Action Genre zusammengefasst (s. Subgenre)	Bat and Ball	Der Spieler kontrolliert ein Objekt, mit welchem er einen hüpfenden Ball durch ein Spielfeld manövrieren soll. Bat and Ball ist eines der ältesten Genres der digitalen Computerspiele. Beispiele: Pong, Arkanoid
	Fighting	In <i>Fighting Games</i> kämpft der Spieler mit seiner virtuellen Figur meist eins-gegen-eins gegen Kontrahenten. Dafür kann er verschiedene Bewegungsabläufe über das Eingabegerät erzeugen, um seine Gegner zu besiegen. Dies stellt Ansprüche an die motorischen Fertigkeiten (z.B. Auge-Hand-Koordination) und daran, die optimale Entscheidung darüber zu treffen, wann und wie eine bestimmte Bewegungskombination ausgeführt werden sollte. Beispiele: Street Fighter, Tekken, Dead or Alive
	Platform	Der Spieler hat die Aufgabe mit einer virtuellen Figur, von Plattform zu Plattform zu springen. Verfehlt er eine Plattform, wird ihm ein virtuelles Leben abgezogen. Beispiel: Pitfall Harry
	Shooter	Die Aufgabe des Spielers besteht darin, Punkte zu sammeln, indem eine grosse Zahl gegnerischer Einheiten mit einer virtuellen Schusswaffe zerstört wird. Beispiel: Attack of the Mutant Camels
	Video Pinball	Das Spiel simuliert einen Flipperautomaten.
Adventure:	Der Spieler schlüpft in die Rolle einer fiktiven Figur, die Fortschritte im Spiel durch das Lösen diverser Rätsel erzielt, die in eine epische Hintergrundgeschichte eingebettet sind. Kampfszenen spielen, wenn überhaupt, nur eine untergeordnete Rolle. Beispiele: Myst, The Longest Journey	
Collector:	Ziel des Spiels ist es, möglichst viele Punkte durch	

	<p>Einsammeln in begrenzter Anzahl vorhandener Gegenstände zu erzielen.</p> <p>Beispiel: Commander Keen</p>
Puzzle:	<p>Ziel dieses Spiels ist es, Puzzlespiele oder Rätsel zu lösen. Diese Spiele werden auch als extrinsische Spiele bezeichnet, da der Inhalt nicht direkt an die Spielstruktur gekoppelt ist.</p> <p>Beispiele: Tetris, Minesweeper, Lemmings</p>
Racer:	<p>Ziel des Spielers ist es, einen Parcours oder einen Rundkurs zu absolvieren, entweder innerhalb einer gewissen Zeit oder schneller als seine Gegenspieler. Der Schwierigkeitsgrad kann durch Hindernisse bei der Steuerung oder auf der Strecke sowie durch schnellere Gegner erhöht werden.</p> <p>Beispiel: Need For Speed</p>
Rhythm-Dance:	<p>Ziel des Spiels ist, im Rhythmus der Musik Aktionen auszuführen.</p> <p>Beispiele: Guitar Hero, Dance Dance Revolution</p>
Role Playing Game (RPG)	<p>Der Spieler schlüpft in die Rolle einer fiktiven Figur und erlebt selbst handelnd soziale Situationen. Während des Spiels sammelt seine Figur Erfahrungspunkte und „erlernt“ neue Fähigkeiten, mit welchen der Spieler seiner zugeordneten Rolle bestmöglich gerecht werden sollte. Das Spiel beinhaltet Kampfszenen und virtuelle Welten, welche es zu erforschen gilt.</p> <p>Beispiele: Eye of the Beholder, Diablo</p>
Simulation	<p>Das Spiel simuliert bestimmte Aspekte einer virtuellen Welt, die real oder fiktiv sein kann. Durch Veränderungen von Parametern dieser Welt kann der Spieler Einfluss auf den weiteren Spielverlauf nehmen. Er kontrolliert mehrere Akteure oder Ressourcen.</p> <p>Beispiele: SimCity, The Sims</p>
Strategy:	<p>Der Spieler sammelt verschiedene Einheiten von begrenzten Rohstoffen und muss Entscheidungen treffen, wie diese</p>

	verwendet werden sollen, um Gegner zu besiegen. Beispiel: Civilization V
Sports:	Das Spiel simuliert eine reale oder fiktive Sportart. Beispiele: FIFA 08, NBA 08
Video Boardgame:	Das Spiel ist die digitale Umsetzung eines Brettspiels. Beispiele: Monopoly, Schach

Neben den genannten *Genres* gibt es sogenannte *Pseudo-Genres*. Diese kombinieren mehrere der soeben beschriebenen *Genres* und haben sich im Laufe der Zeit bei Spielherstellern und Spielern als Hauptkategorien durchgesetzt. Des Weiteren gibt es *Subgenres* (= Unterkategorie eines *Genres*), welche sich durch spezielle Elemente von den anderen *Subgenres* unterscheidet. In der folgenden Tabelle werden die wichtigsten *Pseudo-* resp. *Subgenres* aufgezählt und beschrieben.

Tab. 8: Übersicht *Pseudo-Genres* und *Subgenres* nach Ito (2005)

Pseudo-Genre/Subgenre	Beschreibung
Action	Der Ausdruck Action bezieht sich auf ein rasantes, abwechslungsreiches Spielgeschehen. Es ist meist eine Mischung aus Shooter-, Plattform- und Adventure-Spiel. Beispiele: Legend of Zelda, Tomb Raider
Jump and Run	Dieses Pseudo-Genre ist eine Mischung aus Plattform- und Adventure-Spiel. Beispiele: Sonic the Hedgehog, Super Mario
Driving Simulation	Driving Simulation ist ein Subgenre aus dem Racer-Genre. Das Fahrverhalten, die Autos und die Umgebung sollen möglichst realistisch gestaltet sein.

	Beispiele: Need for Speed, Gran Turismo
First Person Shooter (FPS)	<p>Der Spieler sieht die Umgebung aus den Augen einer virtuellen Repräsentation des Spielers, also aus der Egoperspektive. Meist werden noch Arme oder Hände des zusteuernden virtuellen Akteurs gezeigt. Aufgabe des Spielers ist es, gegnerische Einheiten ausser Gefecht zu setzen und eine Mission zu erfüllen, ohne dabei selbst von den Kotraherenten ausser Gefecht gesetzt zu werden.</p> <p>Beispiele: Far Cry, Call of Duty</p>
Interactive Fiction	<p>Dieses <i>Genre</i> gehört zu den Adventures und wird auch Text-Adventure genannt. Die virtuelle Welt wird nicht graphisch dargestellt, sondern die Interaktion erfolgt nur mittels textbasierter Kommunikation.</p> <p>Beispiel: Zork I</p>
God Games	<p>In diesem <i>Subgenre</i> von Simulationen hat der Spieler die Aufgabe, als höhere Gewalt mit virtuellen Figuren zu interagieren.</p> <p>Beispiele: Black and White, Spore</p>
Light Gun Shooter	<p>Light Gun Shooter gehören zum Genre der Shooter und stellen das präzise Zielen und Feuern auf Gegenstände in den Vordergrund. Es geht also weniger um die Zerstörung gegnerischer Einheiten.</p> <p>Beispiele: Time Crisis, Duck Hunt</p>
Massively Multiplayer Online RPG (MMORPG)	<p>Dieses <i>Pseudo-Genre</i> gehört zu den Computer-RPG. Das Spiel wird online in einer virtuellen Welt mit oder gegen andere Spieler(n) gespielt. Der Schwer-</p>

	<p>punkt liegt jedoch auf der Kollaboration zwischen den Spielern. Die virtuelle Welt ist meist persistent auf einem Server verfügbar¹⁴, sodass die Spieler jederzeit gemeinsam die gleiche Instanz der virtuellen Welt betreten können.</p> <p>Beispiele: World of Warcraft, Guild Wars</p>
<p>Real Time Strategy (RTS)</p>	<p>Diese Spiele gehören dem <i>Genre</i> Strategy an, jedoch können auch Elemente von Simulation einfließen. Bei Real-Time Strategy-Spielen werden die Aktionen der Spieler resp. Gegner synchron ausgeführt. Im Gegensatz dazu wird bei Turn-Based Strategy-Spielen jedem Spieler ein Zug pro Runde gestattet. Die Pausen zwischen den Runden erlauben dem Spieler, zu reflektieren und seine nächsten Aktionen zu planen.</p> <p>Beispiele: SimCity™, Age of Empires</p>
<p>Survival Horror</p>	<p>Dieses Pseudo-Genre ist eine Mischung aus Shooter und Adventure-Spiel. Der Spieler hat die Aufgabe, den Weltuntergang zu verhindern. Die Umgebung ist sehr düster und unheimlich.</p> <p>Beispiele: Resident Evil, Silent Hill</p>
<p>Party Games</p>	<p>Party Games, sind eine Sammlung von Minispielen, welche gemeinsam, gegeneinander, nacheinander, in Gruppen oder allein gespielt werden. Ein Minispiel dauert im Durchschnitt nur</p>

¹⁴ Beim Spiel „Guildwars“ wird hingegen für jedes Team, das ein Szenario startet, eine eigene virtuelle Welt instanziiert.

	<p>einige Minuten. Die Inhalte der Spiele können Geschicklichkeits-, Reaktionsübungen oder Puzzles beinhalten</p> <p>Beispiel: Rayman Raving Rabbits</p>
X-Games	<p>Dieses Subgenre aus dem Bereich Sport bezieht sich auf aktuelle Trendsportarten. Der Spieler hat die Aufgabe, als virtuelle Figur verschiedene Wettbewerbe zu gewinnen. Als Preise winken neue Ausrüstungen und neue Strecken oder Parcours.</p> <p>Beispiel: Tony Hawk's Underground</p>

Wie zu Beginn dieses Kapitels erwähnt, wurden als Begleiterscheinung der enormen technologischen Entwicklung in der Unterhaltungsspielbranche und im Web 2.0 innovative digitale Spiele entwickelt. Im Mittelpunkt derartiger Spiele stehen neben der Kollaboration der Spieler auch deren Einbindung in den Erstellungs- und Weiterentwicklungsprozess der Spiele und somit die Nutzung ihrer Kreativität (Evans, 2007). Die Einsatzmöglichkeiten der Nutzer als Produzenten gehen von der Kombination von *Social Software* (s. Kapitel 2.1.2) mit RPG, wie z.B. Second Life, über die Gestaltung von Spielszenarien (z.B. Little Big Planet) bis hin zu explorativen Spielen mit freiwählbarem Ziel (z.B. SimCity™).

Der Prozess der Inhaltserstellung muss somit als essentieller Teil des Spiels angesehen werden. Er findet entweder parallel zum Spiel, als Ziel des Spiels oder ausserhalb des Spiels (jedoch für das Spiel) statt. Autorenwerkzeuge für letzteres sind schon seit der ersten Generation digitaler Spiele verfügbar. Bislang waren sie allerdings schwierig zu bedienen (s. Kapitel 2.2), sodass sie meist nur von Spielern mit Informatikkenntnissen verwendet werden konnten.

Inzwischen wurden intuitive Werkzeuge für die Inhaltsgenerierung entwickelt. Zudem integrieren die Spielhersteller ihre Werkzeuge zunehmend in die Spiele und schaffen so ein abwechslungsreiches, motivierendes Spielerlebnis. Die Motivierung der

Spieler, selbst aktiv zum Inhalt ihrer Spiele beizutragen, sollte dabei auf den gleichen Prinzipien wie bei *Open Source*-Projekten oder Web 2.0-Applikationen basieren.

2.4.4 Beispiele digitaler Unterhaltungsspiele

Im Folgenden sollen nun die wichtigsten Eigenschaften dieser neuesten Generation digitaler Spiele an zwei Beispielen (Spore und Little Big Planet) prototypisch erläutert und mit den von O'Reilly (2007) postulierten und zuvor erläuterten Prinzipien verglichen werden (s. Kapitel 2.1.1).

Spore



Abb. 14: Virtuelle Welt in Spore und drei Editoren

Spore ist das neueste Spiel von Will Wright und folgt dem explorativen Charakter seiner Vorgänger wie The Sims (Wright, 2000) oder SimCity™. Es verbindet die Genres Simulation und Real Time Strategy. Das Ziel des Spielers besteht darin, eine Spezies über verschiedene Phasen von einem Einzeller zu einem hochentwickelten Raumfahrer zu entwickeln. Die Entwicklung von einer Phase in die nächste wird nach Erreichen einer gewissen Anzahl von Spielpunkten ermöglicht. Nachfolgend werden die insgesamt fünf Phasen des Spiels erläutert.

1. Zellphase:

In der Zellphase kann der Spieler einen Pflanzen oder Fleisch fressenden Mikroorganismus auswählen. In einer 2-dimensionalen Umgebung muss der Spieler schwächere Organismen fressen und den stärkeren ausweichen, um Punkte zu sammeln und diese einlösen zu können. Mit zunehmender Zahl eingelöster Punkte wird der vom Spieler gesteuerte Organismus schneller und/oder angriffsstärker. Jeder erzielte Punkt lässt die Mikrobe wachsen. Sobald sie eine definierte Größe erreicht hat, wechselt der Spieler automatisch in die Kreaturenphase.

2. Kreaturenphase:

In dieser Phase befindet sich der Spieler mit seiner Kreatur in einer 3-dimensionalen virtuellen Welt. Die Kreatur muss für ihre Weiterentwicklung Nahrung zu sich nehmen. Je nach Typ eignen sich andere Nahrungsquellen. Des Weiteren erfüllt der Spieler Missionen und erhält dafür weitere Punkte. Letztere werden wiederum für die Verbesserung der Kreatur eingesetzt, z.B. kann ab einer gewissen Punktzahl ein geeigneter Artgenosse für die Paarung gesucht werden. Es entstehen neue Generationen, basierend auf dem Erbmateriale der Kreaturen. Zudem wirken sich Entscheidungen auf mehrere Generationen aus und bestimmen letztendlich die Kultur der entstehenden Gemeinschaft resp. Zivilisation.

3. Stammesphase:

Ist die Evolution der Kreatur abgeschlossen, beginnt die Stammesphase, die zur Bildung eines Stammes genutzt werden muss. Dafür steuert der Spieler nicht mehr nur eine Figur, sondern eine ganze Gruppe. Somit nimmt das Spiel *Game Play*-Eigenschaften vom *Genre* God Games an. Die Kreaturen können nun auch miteinander interagieren und kommunizieren. Der Spieler kauft Gegenstände und Gebäude, welche zur Gründung von Allianzen, Führung von Kriegen oder Nahrungsaufbewahrung benutzt werden können.

4. Zivilisationsphase:

In der Zivilisationsphase kontrolliert der Spieler weiterhin seinen Stamm und mittlerweile auch dessen Stadt. Auf dem Planeten befinden sich jedoch gegnerische Einheiten, welche derselben Spezies angehören. Dem Spieler steht es frei, über Strategien resp. Mittel der Interaktion mit den gegnerischen Kreaturen zu entscheiden. Mit dem Editor kann er weitere Fortbewegungsmittel und Gebäude erschaffen. Nach der Konstruktion eines Raumschiffs beginnt für den Spieler die letzte Phase des Spiels.

5. Weltraumphase:

In der letzten Phase werden die erstellten Planeten mit den entworfenen Kreaturen, Gebäuden, Zivilisationen anderen Spielern über das Internet zur Verfügung gestellt. Der Spieler fliegt mit seinem Raumschiff durch die Galaxien und besucht, erobert oder kolonialisiert die anderen Planeten. Diese werden kopiert und für jeden Spieler

instanziiert, damit die Eroberung eines Planeten keinen Einfluss auf den Spielverlauf für dessen Erschaffer hat.

Wie auch bei den übrigen Spielen von Will Right üblich, gibt es in dieser letzten Phase kein übergeordnetes Ziel. Vielmehr bietet es dem Spieler vier verschiedene offene Spielziele:

Tab. 9: Spielziele der letzten Phase von Spore

Mögliche Ziele	Beschreibung
Suche nach Artefakten	Auf abgelegenen Planeten können einzigartige und dadurch wertvolle Gegenstände gefunden werden, welche auf dem Marktplatz versteigert werden können. Des Weiteren befinden sich auf versteckten Planeten neue Gestaltungs- und Farbwerkzeuge, welche es dem Spieler erlauben, die Planeten individuell zu dekorieren.
Gestalten der Umgebung (Terraforming)	Mit Hilfe des Raumschiffs können leblose Planeten bewohnbar gemacht werden. Dafür muss Wasser auf den Planeten gebracht, Flora angepflanzt und Fauna ausgesetzt werden.
Besiedlung	Zu Beginn der Kolonialisierung müssen die Kreaturen unter riesigen Kuppeln angesiedelt werden. Die Kolonien können Gewürze anpflanzen und Tauschhandel betreiben. Des Weiteren können sie durch Kinos, Fabrik- und Wohnhäusern aufgebessert werden. Dabei ist zu beachten, dass die Kreaturen bei Laune gehalten werden, damit sie motiviert und zufrieden sind.
Eroberung/Diplomatie	Neben der Besiedlung und Erkundung des Universums können interstellare Zivilisationen entdeckt werden. Mit diesen

	kann Handel betrieben werden, jedoch lässt es sich nicht ausschliessen, dass diese Zivilisationen die Planeten des Spielers angreifen.
--	--

Die Phasen 1 bis 4 des Spiels spiegeln das klassische Design von Computerspielen wider: Spieldesigner konzipieren das Spiel und entwickeln sowohl die Geschichten, als auch Herausforderungen des Spiels. Die Spieler entscheiden sich zwischen vorgegebenen Handlungssträngen und lösen die gestellten Herausforderungen. Ergänzt wird dieses klassische Spieldesign um sieben verschiedene Autorenwerkzeuge, mit denen Kreaturen, Gebäude, Fortbewegungsmitteln, UFOs, Flora, Fauna und das Terrain erstellt werden können. Somit stellt die Inhaltsgenerierung einen integralen Bestandteil des Spiels dar. Allerdings sind die Entwickler von Spore in Phase 5 noch einen Schritt weiter gegangen: Erschaffene Kreaturen können auf dem Internet publiziert und somit anderen Mitspielern präsentiert und von diesen bewertet, in ihre Welten importiert oder als Grundlage für eigene Kreationen verwendet werden. Besonders gelungene Kreaturen werden auf einer dedizierten Webseite ausgestellt.

Die Autorenwerkzeuge sind so konzipiert, dass in wenigen Konfigurationsschritten ein Fortbewegungsmittel, ein Gebäude oder eine animierte 3-dimensionale Kreatur, bestehend aus Körper, Kopf, Extremitäten, Muster und Farbe, experimentell erstellt werden kann. Bilder und Videosequenzen der Inhalte können exportiert und entweder per Mail an Interessierte verschickt oder auf *YouTube*TM bereitgestellt werden. Die konfigurierten Kreaturen, Gebäude und Fortbewegungsmittel werden zudem als Grundlage zur Erweiterung des virtuellen Universums des Spiels verwendet.

Spore nutzt ganz offensichtlich einige von Tim O'Reillys (2007) Prinzipien des Web 2.0. Die erstellten Inhalte beleben das Spiel durch die Kreativität der Nutzer und schaffen eine vielfältige virtuelle Welt. Des Weiteren werden die Inhalte von den Nutzern bewertet und diese sorgen dadurch für eine Qualitätssicherung in Form eines Empfehlungssystems. Erweiterungen und Verbesserungen werden kontinuierlich entwickelt, um den Nutzer längerfristig an das Spiel zu binden. Durch das offene Spielziel in Phase 5 können neue Verwendungszwecke gefunden werden, wie z.B. die

Herstellung von Kurzfilmen – sogenannte *Machinima* (vgl. Machinima.org, 2002; Mprem.com, 2008).

Zusammenfassend entsprechen die fünf Phasen folgenden Prinzipien des Web 2.0:

- Inhalte sind entscheidend
- Kollektive Intelligenz
- Andauernde Beta-Phase der Software

Little Big Planet



Abb. 15: Spielszenario aus Little Big Planet

Dieses Spiel kombiniert das *Genre* Platform mit dem Adventure-*Genre* und unterstützt kollaboratives Spielen. In Abbildung 15 ist ein Ausschnitt aus einem Multiplayer-Szenario aus Little Big Planet zu sehen: Ein Spieler muss mit einem „Düsenrucksack“ einen Mechanismus auslösen, damit der andere Spieler ihm folgen kann. Zudem setzt Little Big Planet auf virtuelle Gemeinschaften für die kollaborative Generierung von Spielszenarien (s. Abb. 16). Virtuelle Gemeinschaften haben sich durch die heutige Informations- und Kommunikationstechnologie in unserer Gesell-

schaft als Ausdruck von Interessensgemeinschaften etabliert (Prestipino, 2008). Infolge neuer Interaktionsformen und Möglichkeiten des Wissenstransfers innerhalb virtueller Gemeinschaften ist das Forschungsinteresse von verschiedenen Disziplinen, wie z.B. der Soziologie, Psychologie, Informatik, Sprachwissenschaften und Mathematik geweckt und der Begriff fachspezifisch unterschiedlich definiert worden (Prestipino, 2008). Wellman u. Gulia (1999) definieren virtuelle Gemeinschaften wie folgt:

„Communities are networks of interpersonal ties that provide sociability, support, information, a sense of belonging, and social identity.“

Howard Rheingold (1993) definiert den Begriff *Virtual Community* aufgrund seiner Erfahrungen mit der virtuellen Gemeinschaft "THE WELL" (Rheingold, 1993; S. 413):

„Virtual communities are social aggregations that emerge from the Net when enough people carry on those public discussions long enough, with sufficient human feeling, to form webs of personal relationships in cyberspace.“

In der virtuellen Gemeinschaft sollen Szenarien für Little Big Planet entwickelt und anhand der im Kapitel Web 2.0 diskutierten Bewertungs-, Feedback- und Sharing-Funktionalitäten ergänzt werden. Im Gegensatz zu den meisten Social Network-Webseiten werden die erstellten Inhalte bei Little Big Planet von einem Expertenteam begutachtet und allenfalls den Erstellungsrichtlinien angepasst. Im Jahr 2008 wurden aufgrund dieser Filterung rund 2% der von Nutzern erstellten Spielszenarien wegen pornographischer, hetzerischer oder rassistischer Inhalte nicht publiziert (k-videogames.de, 2008).

Die Einführung der Spieler resp. Produzenten in die Inhaltsgestaltung wurde so konzipiert, dass diese Inhalte in Abhängigkeit ihres Erfahrungsgrads mit Hilfe verschiedener Techniken erstellen können (Küster, 2008). Davon verspricht sich der Hersteller eine hohe Benutzerakzeptanz. So können z.B. Novizen zuerst bereits bestehende Welten graphisch gestalten, ihre virtuelle Spielfigur (genannt *Sackboy*) personalisieren oder Inhalte von anderen Spielern bewerten und Feedback geben. Falls Spieler sich mehr engagieren wollen, haben sie die Gelegenheit, Vorlagen von

Szenarien auszubauen oder schlussendlich eigene Szenarien von Grund auf selbst zu erstellen. Ausführliche Tutorials und jederzeit abrufbare Hilfsfunktionen unterstützen die Spieler während des gesamten Entwicklungsprozesses.

Das innovative Moment des integrierten Autorenwerkzeugs von Little Big Planet ist das kollaborative Erstellen der 3-dimensionalen Objekte und Hindernisse in einer Art Zeichnungsmodus, welcher eine Aufnahmefunktion bereitstellt, sodass die Erstellung selbst Spielcharakter erhält (s. Abb. 16). Das Wechselwirkungsprinzip, welches durch eine fortschrittliche *Physik-Engine* unterstützt wird und das Verhalten der Objekte, Materialien und Formen aufgrund physikalischer Gesetze vorhersehbar macht, kann in einem Abspiel- und Pausemodus getestet werden.



Abb. 16: Autorenwerkzeug des Spiels Little Big Planet

Das Spiel baut auf dem Konzept "Play, Create, Share" von Alex Evans (2007) auf. Im Spielmodus macht sich der Spieler mit der Umgebung vertraut und lernt die Interaktionsmuster kennen. Des Weiteren wird der Spieler zur Gestaltung seiner eigenen virtuellen Welten inspiriert. Er hat die Möglichkeit, das Spiel zur Laufzeit zu personalisieren und den Produzenten Feedback zu geben. So können die Spiel-szenarien kontinuierlich optimiert und mehrere Nutzer durch hohe Bewertungen auf das jeweilige Szenario aufmerksam gemacht werden. Wie in Spore auch wird auf

diesem Weg eine oberflächliche Qualitätssicherung nach dem Prinzip kollektiver Intelligenz gewährleistet. Im Kreativmodus kann der Spieler eigene Inhalte je nach Fertig- und Fähigkeiten erstellen und schlussendlich anderen Nutzern zur Verfügung stellen (jedoch behalten sich die Macher von Little Big Planet vor, bestimmte Inhalte, wie z.B. die Kleidung des *Sackboys*, kostenpflichtig im Online-Shop anzubieten und nicht von den Nutzern generieren zu lassen). Durch die bereits erstellten Szenarien werden die Spieler inspiriert, eigene Szenarien basierend auf bereits realisierten Ideen zu generieren. Zudem fördert das Streben nach einem guten Ruf in der virtuellen Gemeinschaft die Bereitschaft, als Gutachter und/oder Produzent aktiv zu werden. Somit treffen folgende Prinzipien von O'Reilly (2007) auf Little Big Planet zu:

- Inhalte sind entscheidend
- Kollektive Intelligenz
- Netzwerkeffekt
- Andauernde Beta-Phase der Software

2.4.5 Zusammenfassung

Die neuste Generation digitaler Spiele kombiniert die Funktionalitäten der Web 2.0-Applikationen, innovative Eingabegeräte und kollaboratives Spielen. Ziel der Entwickler ist es, den Nutzer resp. Spieler in die Inhaltsgenerierung zu involvieren, emergente Unterhaltung durch Förderung und Forderung ihrer Kreativität zu schaffen und ein neues Spielerlebnis mit intuitiv bedienbaren, integrierten Autorenwerkzeugen zu ermöglichen.

2.5 Game-based Learning

E-Learning wird u.a. an vielen Hochschulen genutzt, um kostengünstig Lerninhalte für eine grosse Anzahl von Studenten zur Verfügung zu stellen. Leider erbringen nicht alle eingesetzten Lösungen auch den erwarteten Nutzen, denn ohne die vollständige Integration der Softwarelösungen in die Lerneinheiten (z.B. Kurse oder Vorlesungen) sinkt die Motivation zur Nutzung des Angebots (Bürg & Mandl, 2005). Darum wurde in den letzten Jahren nach innovativen E-Learning-Konzepten geforscht, welche den Lernenden motivieren sollen, sich ausdauernder und tiefgründiger mit dem Lerninhalt auseinanderzusetzen. *Game-based Learning* gilt als vielversprechendes E-Learning-Konzept, das den Lerneffekt durch spielerische Elemente steigern und die Nutzungseinstellung der Lernenden positiv beeinflussen soll (Michael & Chen, 2005; Squire, 2003). Dafür spricht beispielsweise der mehrfach replizierte Befund, dass spielerische Elemente das Eintauchen in den Flow-Zustand (Csikszentmihalyi & Csikszentmihalyi, 1988) erleichtern. Davon verspricht man sich eine Unterstützung nachhaltiger Wissensvermittlung.

Ein Spiel motiviert¹⁵ den Nutzer intrinsisch durch Herausforderungen, die Verknüpfung des Inhalts mit einer Hintergrundgeschichte, Interaktionen und die aktive Beeinflussung des Spielgeschehens (Prensky, 2003; vgl. Adelsberger et al., 2000; s. auch Kapitel 2.4). Es soll sich dadurch als Grundlage für ein Lernmodell eignen (vgl. Malone 1980). So identifizierte etwa Clark N. Quinn (2005) Synergien zwischen Lernelementen und Spielmerkmalen (s. Kapitel 2.4.1), welche im Falle einer Aktivierung durch ein Lernspiel motivierende Lernerlebnisse ermöglichen können (Garris et al., 2002). Auch im *Game-based Learning* erscheint es für den Lernenden nützlich, wenn Lerninhalte kontextabhängig präsentiert und klare Ziele definiert werden sowie angemessene Herausforderung gegeben und die Relevanz für den Lernenden erkennbar ist, zudem der Lernende aktiv Entscheidungen treffen kann und

¹⁵ In den letzten Jahrzehnten wurden diverse Modelle zu intrinsischer Motivation publiziert, welche sich hauptsächlich in der Emphase und der Konstruktion unterscheiden. Im Folgenden wird nicht genauer auf Motivationsforschung eingegangen. Der interessierte Leser wird auf das expectancy/valence Model (Mathieu et al., 1992) und Kellers Attention, Relevancy, Confidence, and Satisfaction model (ARCS; 1983) verwiesen.

Feedback erhält (Wechselberger, 2005). Diese Elemente gelten als die Grundlagen für Spass und Vergnügen der Lernenden bei der Nutzung von E-Learning-Applikationen.

Im Zuge der zunehmenden Verbreitung von Heimcomputern, wachsender Bedeutung von Computerspielen und rasanter technologischer Entwicklung der letzten Jahre erkannten u.a. Schulbuchverlage das Potential digitaler Lernspiele und brachten einige Titel auf den Markt. Diese Lernspiele richteten sich an Personen aller Altersgruppen und versuchten – mehr oder minder erfolgreich –, das didaktische und insbesondere das motivationale Design von unterhaltungsbezogenen, digitalen Spielen zur Anregung von Lernprozessen zu nutzen (Bopp, 2006). Jedoch erwiesen sich die ersten Lernspiele bisher suboptimaler Weise entweder als digitale Spiele mit wenig Lerninhalt oder als klassische E-Learning-Applikation mit rudimentär implementierten Spielmerkmalen. Die optimale Verschmelzung von Spieldesign und didaktischem Konzept stellt ganz offensichtlich eine grosse, wenn nicht sogar *die* zentrale Herausforderung für den Erfolg von *Game-based Learning* dar, wenn der Lernende längerfristig an die Applikationen gebunden und sich sein Lerneffekt vergrössern soll. (Abb. 17).

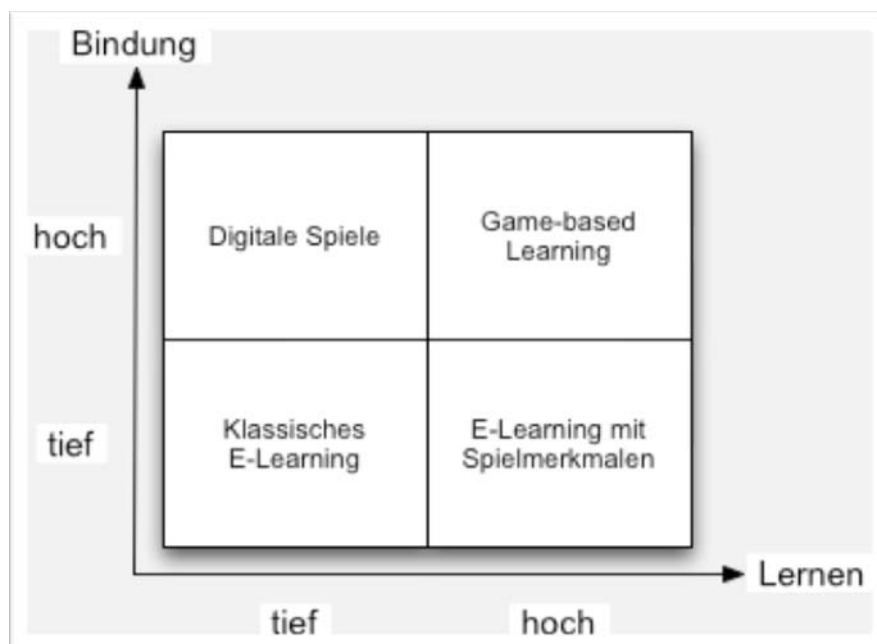


Abb. 17: Klassisches E-Learning vs. Game-based Learning (Prensky, 2003)

2.5.1 Konstruktivismus im Kontext des *Game-based Learning*

Lernforscher sehen das Potential von *Game-based Learning* in den vom Spiel implizierten Interaktionsmöglichkeiten, der Steigerung des Lerneffekts durch die optimierende Nutzung des Flow-Zustands unter der Voraussetzung adäquater Forderung der Fähigkeiten des Nutzers (Csikszentmihalyi, 1988) sowie im Ansatz des konstruktivistischen Lernens (vgl. Reich et al., 2005; Papert, 1998). Konstruktivistisches Lernen beruht darauf, „*dass menschliches Erleben und Lernen Konstruktionsprozessen unterworfen ist, die durch sinnesphysiologische, neuronale, kognitive und soziale Prozesse beeinflusst werden*“ (wikipedia.org, 2008). Demnach erstellen Lernende während des Lernprozesses eine subjektive Repräsentation der Lerninhalte und ihrer Assoziationen, was zur Folge hat, dass das Erlernte stark von den Erfahrungen des Lernenden abhängt.

Funderstanding.com (2008) definiert den Begriff Konstruktivismus wie folgt:

„Constructivism is a philosophy of learning founded on the premise that, by reflecting on our experiences, we construct our own understanding of the world we live in. Each of us generates our own "rules" and "mental models," which we use to make sense of our experiences. Learning, therefore, is simply the process of adjusting our mental models to accommodate new experiences.,,

Piaget (1973, S. 20) formuliert zur konstruktivistischen Lernmethode folgendes:

„To understand is to discover, or reconstruct by rediscovery, and such conditions must be complied with if in the future individuals are to be formed who are capable of production and creativity and not simply repetition“

Dementsprechend beschreibt Kersten Reich den von ihm geprägten, interaktionistischen Konstruktivismus als eine Kombination aus Rekonstruktion, Konstruktion und Dekonstruktion: Rekonstruktion im Sinne von Entdecken der Welt, Konstruktion als Kreation einer subjektiven Realität und Dekonstruktion durch Kritik oder Bewertung der Welt. Re-, De- und Konstruktion sind an die Aktivitäten der Lernenden geknüpft und somit wirkt deren subjektives Erleben wechselwirkend auf die Lernumgebung ein. Schlussfolgernd erachtet der Interaktionistische Konstruktivi-

vismus diejenigen Lernprozesse als am effektivsten, die der Lernende selbst steuern kann. Dies setzt allerdings die zeitintensive Aneignung entsprechender Methodenkompetenz in Form von Lerntechniken voraus (Reich et al., 2005).

Ähnlich dem Ansatz von Reich geht Resnick (2007) davon aus, dass die Fähigkeit kreativ zu denken neben Faktenwissen und sozialen Kompetenzen ein Schlüssel zum Erfolg im Beruf und im privaten Umfeld sei (Resnick 2007, Sawyer 2006). Sein Modell *Lifelong Kindergarten* basiert auf den konstruktivistischen Ansätzen von Papert und Piaget und macht sich das spielerisch-explorative und kollaborative Lernverhalten von Kleinkindern zunutze (vgl. Fulghum, 1986), wurde in der Lehre („After-School-Programs“, Vorlesungen, Seminare) bereits erfolgreich eingesetzt sowie in Software, wie *StarLogo*, *Scratch* und *Programmable Bricks* integriert.

Im folgenden Abschnitt werden die fünf Phasen des "Lifelong Kindergarten Approach" beschrieben (Resnick, 2007).

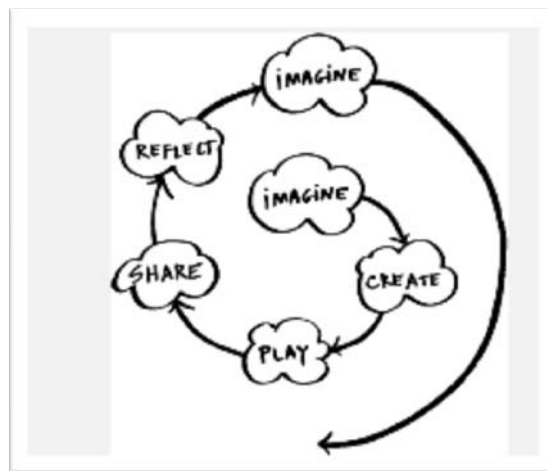


Abb. 18: *Lifelong Kindergarten Approach* nach Resnick (2007)

Während des Imaginierens (*imagine*) konstruiert der Lernende seine Wirklichkeit mit dem Ziel, sein bestehendes Wissen durch die Aneignung von Theorien und die Nutzung seiner Phantasie zu erweitern. Hierfür kann er auf bestehende Erfahrungen und Erlebnisse zurückgreifen. Anschliessend werden im Rahmen des Kreierens (*create*) explorativ die Inhalte, basierend auf den Ideen und Gedankenspiele der vorherigen Phase, erarbeitet bzw. entwickelt und mit seiner Vorstellung abgeglichen. Diese Phase gilt als Wurzel kreativen Denkens. Aus diesem Grund sollen dem

Lernenden alle Möglichkeiten zur Kreativitätsentfaltung innerhalb der geltenden Rahmenbedingungen offen gelassen werden. Beim Spielen (*play*) sollen die zuvor entwickelten Konstruktionen ausprobiert und getestet werden, bevor sie in der nächsten Phase (*share*) Anderen zur Verfügung gestellt, kollaborativ ausprobiert und diskutiert werden. Die Diskussionen werden für den nächsten Schritt des individuellen Reflektierens zur Einbindung in das eigene Erfahrungsrepertoire genutzt, welches wiederum als Grundlage für zukünftige Ideen oder Phantasien dient. Die aufgenommenen Informationen werden in dieser Phase verarbeitet und das Gelernte verankert. Somit schliesst sich der Zyklus mit der Reflexionsphase und kann mit der Entwicklung neuer Ideen erneut gestartet werden.

Ziel des Kurationsprozesses ist das Lernen durch Synthese, welches nach Edgar Dale (1969) den grössten Lernerfolg verspricht. Der Lerner wird dabei in die Lage versetzt, zu analysieren, zu designen, zu kreieren und zu evaluieren. Nach Dales Erfahrungspyramide (Abb. 19) können sich Lerner an 90% ihrer aktiv analysierten und selbst kreierten Inhalte auch später noch erinnern. Hingegen können ca. 70% des mündlich Repetierten und Aufgeschriebenen mit einigem zeitlichen Abstand zur Lernphase wieder abgerufen werden. Fusst Lernen auf kombiniert auditiv-visuellen Inputs, beträgt die Erinnernsquote nur noch knapp 50% und schlussendlich können sich Lerner nur gerade an 20% der ausschliesslich gehörten und an 10% der gelesenen Informationen erinnern.

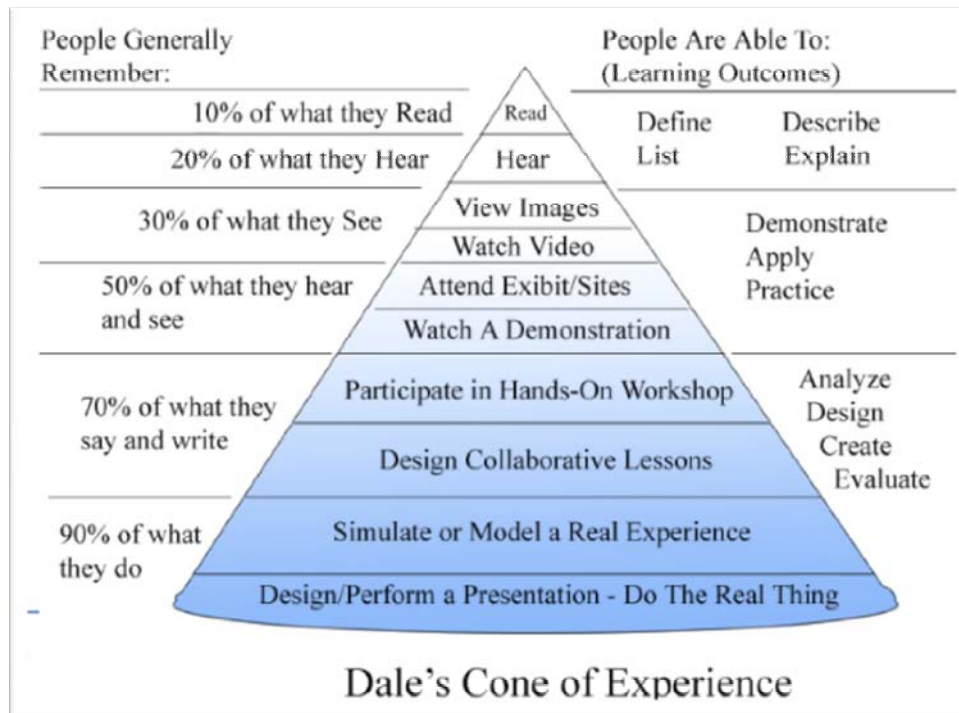


Abb. 19: Dales Erfahrungspyramide (1969), Darstellung von Dr. Göbel

Die erwähnten konstruktivistischen Ansätze lassen schlussfolgern, dass die inhärenten Merkmale eines digitalen Spiels, wie Interaktion oder multisensorische Ausgabe (= Multimedialität), den Wissenserwerb und die dauerhafte Verankerung nachhaltig unterstützen können.

2.5.2 Game Generation

Marc Prensky (2003) sowie Lynee C. Lancaster und David Stillman (2003) diskutieren die Veränderungen individueller Lernstile über die Jahrzehnte. Auffällig sind die Veränderungen bei der Generation der heute unter 30-Jährigen im Vergleich zu älteren Personen. Prensky führt die gefundenen Unterschiede auf die rasante technologische Entwicklung der letzten Jahre zurück: Die Generation der heute über 30-Jährigen ist noch weitgehend ohne Computer, digitale Spiele oder Mobiltelefone aufgewachsen. Diese Generation konnte die Entwicklung der Informationstechnologie, wie z.B. die Heimcomputer, Game Boys oder Spielkonsolen zwar miterleben, weiss aber aus eigener Erfahrung auch, wie das Leben ohne diese Geräte gestaltet werden kann. Demgegenüber betrachtet die Generation der heute unter 30-Jährigen

den aktuellen Stand und Status der Informationstechnologie als selbstverständlich und kann sich das Leben ohne diese Technologien gar nicht vorstellen (*Digital Natives*) (Prensky & Gee, 2006). Alan Kay (2006) beschreibt diesen Generationsunterschied wie folgt:

„*Technology is only technology if it was invented after your were born.*“

Unterstützt wird diese Aussage durch Studien zu unterschiedlichem Nutzungsverhalten in Bezug auf Technologien im Vergleich älterer (*Digital Immigrants*) und jüngerer Kohorten (*Game Generation*). Diese belegen, dass 13- bis 18-Jährige täglich im Schnitt ca. 3 Stunden Fernsehen, zwischen 10 und 60 Minuten auf dem Internet surfen und etwa 3,5 Stunden Computerspiele spielen. Diese intensive Technologie-nutzung prägt das Denk- und Lernverhalten der *Game Generation* entscheidend (Tallal, 2000; Diamond, 1988). Derartige Veränderungen beeinflussen den Lernprozess jedoch nur dann, wenn Lernende gezielt auf die sensorischen Eingaben und die zu bewältigenden Aufgaben fokussieren (Prensky, 2001). Die Voraussetzungen für die phylogenetische Reorganisation des Gehirns sind im Falle der *Game Generation* in Bezug auf Computerspiele und Fernsehen gegeben. Diese Generation beschäftigt sich täglich intensiv mit Medien und Technologien. Sichtbar wird diese Beeinflussung durch die Technologie ebenfalls im SMS-Schreibverhalten vieler Jugendlicher: Diese verwenden zunehmend HTTP-Status-Codes (vgl. RFC2616, 1999) als Abkürzungen in ihren Textnachrichten. So wird etwa der Status-Code 404¹⁶ benutzt, um anzumerken, dass man die Antwort auf eine gestellte Frage nicht kennt.

Patricia Marks Greenfield (1984) erforschte die Effekte von digitalen Computerspielen auf die Psyche der Spieler. Sie stellte fest, dass Menschen, die digitale Computerspiele spielen, signifikant häufiger die Fähigkeit entwickeln, Regeln und Muster durch interaktive *Trial-and-Error*-Prozesse, durch Beobachtungen oder durch Hypothesentests zu entdecken. Des Weiteren konnte gezeigt werden, dass die *Game Generation* besser in der Lage ist, ihre Aufmerksamkeit auf mehrere gleichzeitig zu bearbeitende Aufgaben zu fokussieren (*Multi-tasking*).

¹⁶ HTTP-Antwort auf die Anfrage eines Clients, falls keine Ressource unter der eingegebenen URL gefunden werden kann.

Basierend auf Greenfields Erkenntnissen wurden zehn kognitive Lernstilveränderungen postuliert (Prensky, 2001), welche nahelegen, dass die klassischen Lehrmethoden für die *Game Generation* angepasst oder neue Methoden entwickelt werden müssen, die dem veränderten Lernverhalten IT-affinerer Generationen konzeptuell stärker entsprechen (Tab. 10).

Tab. 10: Zehn kognitive Lernstilveränderungen nach Prensky (2001)

Merkmal	Beschreibung
Geschwindigkeit	Die <i>Game Generation</i> wird durch den häufigen Konsum digitaler Spiele, Musikvideos oder Fernsehwerbung trainiert, hochfrequente Bildabfolgen zu analysieren und die wichtigsten Informationen daraus zu extrahieren.
Paralleles Be- und Verarbeiten	Das Vorurteil, dass bei gleichzeitiger Bearbeitung von mehreren Aufgaben die Qualität des Resultats leide, kann nicht widerlegt werden. Jedoch wurde beobachtet, dass z.B. Programmierer Software entwickeln und gleichzeitig Musik hören oder Börsenhändler Aktienkurse verfolgen und gleichzeitig telefonieren können. Diese Fähigkeit wird den Spielern bei der Bewältigung von Herausforderungen der Computerspiele abverlangt. Der Spieler muss z.B. auf einem Radar die Positionierung gegnerischer Einheiten beobachten und sich gleichzeitig in einem unwegsamen Terrain bewegen.
Hypertext	Im Gegensatz zu klassischen Lehrbüchern, die Informationen kapitelweise darbieten, entstanden in den letzten Jahrzehnten neue Methoden des Informationsaufrufs und der Wissensgenerierung. Auf dem Internet können Inhalte bzw. Informationen via Hypertext mit sogenannten <i>Hyperlinks</i> miteinander verknüpft werden. Der Leser kann dadurch von einem Inhalt zum nächsten gelangen und sich relevante Informationen beschaffen. Er erfasst die subjektiv relevanten Inhalte in der von ihm didaktisch

	<p>bevorzugten Reihenfolge. Lerner bekommen somit einen Überblick über das gesamte zu erarbeitende Gebiet. Gegebenenfalls können einzelne Themen vertieft werden.</p>
<p>Visualisierungen</p>	<p>Die <i>Game Generation</i> ist mit multimedialen Informationen aufgewachsen. Visualisierungen haben für sie einen ganz anderen Stellenwert zur Informationsdarstellung als für ältere Generationen. Selbstverständlich schätzen letztere Visualisierung ebenfalls, jedoch scheint die ältere Generation mit Inhalten in reiner Textform besser umgehen zu können als die <i>Game Generation</i>.</p> <p>Das Sprichwort "Ein Bild sagt mehr als tausend Worte" beschreibt diesen Punkt treffend.</p>
<p>Aktiv</p>	<p>Die <i>Game Generation</i> hat sich durch die Nutzung digitaler Spiele daran gewöhnt, Entscheidungen interaktiv und kollaborativ zu treffen. Passiver, fremdbestimmter Konsum von Informationen entspricht in zunehmendem Masse nicht dem typischen Lernverhalten dieser Generation.</p>
<p>Spielen statt Arbeiten</p>	<p>Sowohl in analogen wie digitalen Spielen werden Spieler auf das problem-orientierte Lösen von Aufgaben sensibilisiert. Das könnte ein Vorteil in Bezug auf die rasch wechselnden Bedingungen in der Arbeitswelt sein, wenn es ihre Flexibilität und Anpassbarkeit fördert.</p>
<p>Geduld und Erfolg</p>	<p>In Computerspielen werden die Spieler oft auf die Geduldprobe gestellt, um knifflige Rätsel zu lösen. Nach Stunden der Lösungsfindung wird der Spieler dann mit dem Erreichen des nächsten Spielszenarios belohnt. Hat sich der Aufwand nicht gelohnt oder meint der Spieler, dass sich der Aufwand nicht lohnt, verliert er die Geduld und sucht den Weg des geringsten Widerstandes, um die Rätsel zu lösen.</p>

	Analog zum Verhalten in Computerspielen reagiert die <i>Game Generation</i> in der Realität: Kann ihr nicht klar gemacht werden, was der Nutzen der investierten Zeit ist, verliert sie schnell die Geduld und widmet sich anderen Aufgaben.
Kreativität und Phantasie	Durch das Web 2.0 ist die <i>Game Generation</i> mit der Rolle des Produzenten und somit der Erstellung von Inhalten vertraut. Insbesondere explorative Spiele und Simulationen eignen sich zur Förderung der Kreativität und Phantasie. Wie erwähnt, ist Kreativität u.a. für Mitchell Resnick ein wichtiger Schlüssel zum Erfolg im Berufsleben.
Sensibilisiert auf Technologien	Da die <i>Game Generation</i> mit den heutigen Technologien aufgewachsen ist, sind diese vollkommen in ihren Alltag integriert. Es entstand somit eine gewisse Abhängigkeit, sodass bei Nichtvorhandensein gewohnter Technologien die Effizienz der <i>Game Generation</i> stark beeinträchtigt würde. Ältere Generationen betrachten neue Technologien oft noch als Blackbox und trauen ihnen nicht, so dass sie kritische Aufgaben lieber nach altbekannten Methoden bearbeiten.
Haltung der Game Generation	Die <i>Game Generation</i> möchte ihr Befinden und ihre Meinung direkt und offen mitteilen.

Diese zehn kognitiven Veränderungen müssen nach Prensky für das Lernverhalten der *Game Generation* berücksichtigt werden, um optimale Unterstützungsmöglichkeiten für ihren Lernstil anzubieten. Das kann zum Teil durch Einbeziehen von Spielelementen in den Lerninhalt erfolgen. Ein anderer Teil muss jedoch speziell berücksichtigt werden, da er nicht inhärent in Spielen vorhanden ist. Beispielsweise sind nicht in jedem *Genre* a priori genügend Pausen vorgesehen, die dem Spieler Reflexion ermöglichen.

Aus den Untersuchungen bezüglich kognitiver Veränderungen der *Game Generation* und der Synergien zwischen Lernelementen und Spielmerkmalen lassen sich die

Anforderungen an digitale Lernspiele ableiten. Diese sollen den Lernenden motivieren und ihn durch die Verschmelzung von Spiel und Lerninhalt längerfristig an das Spiel binden (Tab. 11). Die genauen Anforderungen an digitale Lernspiele hängen wiederum vom Zielpublikum, Lerninhalt und Kontext ab (Prensky, 2001).

Tab. 11: Synergien zwischen Lernelementen und Spielmerkmalen

Lernelemente	Spielmerkmale	Motivierendes Lernerlebnis
Kontextabhängig	Thematische Einbettung	Thema
Klare Ziele	Klare Ziele	Ziele
Angemessene Herausforderung	Adaptierte Herausforderung	Herausforderung
Verankert	Relevanz der Aktion auf die Umgebung	Verlinkung zwischen Aktionen und Lerninhalt
Relevanz	Relevanz des Problems für den Spieler	Inhalt adaptiert auf die Zielgruppe
Explorativ	Aktionsvielfalt	Aktiv
aktiv Entscheidungen treffen	Direkte Manipulationen	Direkt anwendbar
Angemessenes Feedback	Feedback	Feedback
Aufmerksamkeit	Ereignisse und Informationen	Affekt

David Shaffer (2006) nutzt diese Synergien zwischen Lernelement und Spielmerkmal als Grundlage für seine *epistemic Games*, welche Studenten Fähigkeiten und Fertigkeiten des zukünftigen Berufsalltags vermitteln sollen. *Epistemic Games* stellen anwendbares Wissen über das bloße Erinnern theoretischer Grundlagen (Shaffer, 2006, Shaffer et al., 2005), in der Annahme, dass die Studenten auf diesem Weg besser auf ihr späteres Berufsleben vorbereitet würden. In den folgenden Abschnitten werden die resultierenden Lernerlebnisse aus Tabelle 11 kurz diskutiert.

Thema

Für die Bindung des Lernenden an das Spiel und das Eintauchen in die virtuelle Welt ist es notwendig, das Spiel in einen ansprechenden Kontext einzubetten. Dafür eignen sich bekannte Themen aus Literatur und Film. Des Weiteren verstärkt die Integration des Lerninhalts in eine Hintergrundgeschichte das Spielerlebnis des Lernenden (vgl. Gilleade et al., 2005). Sollten sich Hintergrundgeschichte und Kontext als inkompatibel erweisen, kann dies zu einer Minderung des Lernerlebnisses führen. Um derartige unerwünschte Effekte zu vermeiden, eignet sich der Einsatz eines Lernspiels basierend auf dem *Shooter-Genre* z.B. nicht im Anwendungskontext einer Menschenrechtsorganisation. Die Mitarbeiter dieser Organisation können sich

höchstwahrscheinlich nicht mit diesem *Genre* identifizieren und würden dadurch das Spiel und seine inhärenten Lerninhalte nicht akzeptieren.

Ziele

Die Ziele von Lernspielen sind mit deren Hintergrundgeschichte und Kontext verknüpft. Sie müssen unmissverständlich definiert und formuliert sein und den Lernenden auf die passenden Aktionen resp. Aufgaben führen. Die Zielerreichung soll zudem quantitativ festgehalten werden. Lernziele entfalten ihre optimale Wirkung nur dann, wenn sie den Fähig- und Fertigkeiten des Lerners entsprechen und diesen zwar heraus-, aber nicht überfordern.

Herausforderung

Deshalb sollten die Herausforderungen an den individuellen Voraussetzungen der Lernenden ausgerichtet werden (Charles et al., 2005), so dass dieser alles rund um das Lernspiel vergessen und im Idealfall in den Flow-Zustand versetzt werden kann. Angemessene Herausforderungen binden den Lernenden längerfristig an ein Spiel und sind ein zentraler Bestandteil positiver Lernerlebnisse. Der Lernende soll durch die zu bearbeitenden Aufgaben nicht über- oder unterfordert und somit frustriert werden, sondern sich mit Spass von einer Herausforderung zur nächsten vorarbeiten. Herausforderung und Fähigkeitsniveau des Lernenden sollen nahezu ausbalanciert sein und ihn dadurch zur Verfolgung des Lernziels motivieren (Engeser et al., 2003). Aktive Personalisierung der Herausforderung soll den Lernenden zudem erlauben, ihre Grenzen auszutesten und sich einzuschätzen lernen (Spring-Keller & Ito, 2007).

Verlinkung zwischen Aktionen und Lerninhalt

Der Spieldesigner steht vor der Aufgabe, den gewünschten Lerninhalt sinnvoll mit der Spielwelt zu verbinden. Für den Lernenden muss die Einbettung des Lerninhalts in die Hintergrundgeschichte nachvollziehbar sein, d.h. er muss verstehen, warum er für die Zielerreichung gewisse Aktionen ausführen muss. Ein Beispiel, das diese Verlinkung nicht beachtet, ist etwa ein Lernspiel mit Lerninhalt Mathematik, in welchem der Lernende durch das Lösen von Rechenaufgaben Munition zur Bekämpfung von Ausserirdischen bekommt. In diesem Beispiel steht die Hintergrundgeschichte („Die Befreiung der Menschen von Ausserirdischen“) in keinem plausiblen Zusammenhang mit dem Lerninhalt Mathematik, sodass die Vorteile von

Game-based Learning nicht zum Tragen kommen. Das Resultat ist stattdessen eine klassische E-Learning-Applikation mit Spielmerkmalen.

Es spricht jedoch nichts dagegen, Herausforderungen von bestehenden Problemen in einen anderen Kontext einzubetten und so von der Realität zu abstrahieren.

Inhalt adaptiert auf die Zielgruppe

Das Zielpublikum muss in der Konzeptionsphase eindeutig definiert werden, damit das digitale Lernspiel seinen Anforderungen und Bedürfnissen möglichst exakt angepasst werden kann (Charles et al., 2005). Zudem müssen die Hintergrundgeschichte, die Farbwelt, das Genre und die Interaktionsform auf die Lernenden und ihre Kultur abgestimmt sein, um deren Involvierung zu erhöhen (Quinn, 2005). Die *Personas* müssen sowohl hinsichtlich Alter, Geschlecht, Wettbewerbsfähigkeit wie auch Erfahrungen für das Spiel definiert werden.

Nicht jeder Lerninhalt kann mit jedem *Genre* gleich gut kombiniert werden. Prensky (2003, S. 156) definiert Lernaktivitäten und mögliche anwendbare Spielgenres für verschiedene Inhalte, wie z.B. das Erlernen von Fakten, Sprachen, Theorien oder Prozessen. So eignen sich Rollen- und Adventure-Spiele seiner Auffassung nach gut zum Erlernen von Interviewtechniken. Interaktives Lernen wird durch *Learning by doing*, Exploratives Lernen und Coaching gefördert (Roussou 2004). Lernaktivitäten im Bereich Interviewtechniken entsprechen demnach Imitation und Feedback (vgl. Adams, 1999). Es eignen sich allerdings nicht alle Lerntypen (vgl. Kolb, 1981; Kolb & Kolb 1976) für Lernspiele, sodass in gewissen Fällen eine Alternative zu Lernspielen bevorzugt werden sollte (Stangl, 2003). Z.B. benötigt der kontakt- oder personenorientierte Lerntyp eine Lehrperson, welche mit ihm die Aufgaben und Lösungen bespricht und klärt.

Aktiv

In einem Lernspiel entscheiden die Lernenden aktiv, welche Strategien sie zur Lösung von Problemen wählen. Sie können dadurch z.B. mit der Methode explorativen Lernens auf die Inhalte sensibilisiert werden. Die Lernenden wechseln zunehmend aus der Rolle passiver Konsumenten in die aktiver Entscheidungsträger. Sie werden gefordert, sich über gefällte und zu fällende Entscheidungen Gedanken zu machen,

um eventuelle Fehlentscheidungen nicht zu wiederholen. Die Handlungen in der Hintergrundgeschichte sollten für lehrreiches Feedback direkt aus den getroffenen Entscheidungen abgeleitet sein.

Direkt anwendbar

Der Lernende muss agieren, um Entscheidungen zu treffen, und die erforderliche Aktion verweist idealer Weise auf die Semantik der Auswahl (Quinn, 2005, S. 61). So kann beispielsweise die Unterscheidung von essbaren und giftigen Pilzen einerseits per textbasierter Beschreibung des Aussehens oder andererseits anhand einer graphischen Abbildung des Pilzes aus verschiedenen Blickwinkeln erfolgen. Die zweite Möglichkeit entspricht dem realen Anwendungsfall „im Wald Pilze suchen“ in viel stärkerem Masse als eine bloße schriftliche Beschreibung.

Feedback

Unmittelbares Feedback auf getroffene Entscheidungen und ausgeführte Aktionen stellt einen wichtigen Nutzen und grossen Vorteil von digitalen Lernspielen gegenüber anderen Lehrmethoden dar. Das System reagiert unmittelbar und verdeutlicht dem Lernenden direkt die Konsequenzen seines Handelns. Daher ist es zum Zwecke der Lernförderlichkeit unverzichtbar, dass die Aktionen des Spielers mit der Geschichte des Spiels gekoppelt sind. Externes Feedback (z.B. durch eine Lehrperson), welches ausgeführte Aktion oder getroffene Entscheidung bezüglich Zielerreichung kommentiert, sollte wenn möglich vermieden werden, um die Involvierung des Spielers in das Spiel nicht zu stören. Verhaltensrelevante Informationen können und sollen im Spiel an den entsprechenden Stellen integriert werden und dem Lernenden sinnvolle Hilfestellungen zur Bewältigung der Aufgaben leisten (Barsalou, 1999; Brown et al., 1989; Glenberg & Robertson, 1999).

Affekt

Ein Spiel sollte für den Lernenden nicht vollkommen vorhersagbar sein, ansonsten würde es an Spannung und dadurch an Anziehungskraft verlieren. Deshalb sollten, wenn möglich und wo angemessen, Ereignisse basierend auf Wahrscheinlichkeiten und Zufällen eintreten, sofern sie den Spieler nicht in eine Sackgasse des Spielablaufs manövrieren. Sind zufällige Ereignisse nicht umsetzbar, helfen z.B. angemessene Dialoge, Musikeffekte oder Situationskomik (Quinn, 2005).

Als potentielle *Showstopper* für die Entwicklung von Lernspielen wurden verfügbare Ressourcen und zu verwendende Technologie identifiziert. Diese Anforderungen sollten bereits während der Planungsphase evaluiert werden. Die verfügbaren Ressourcen müssen in der Planung des Lernspiels berücksichtigt werden, da die Herstellung von Spielen sehr aufwendig ist und je nach *Genre* und Technologie hoch spezialisierte Mitarbeiter benötigt. Im Gegensatz zu kommerziellen Unterhaltungsspielen stehen jedoch für die Entwicklung von Lernspielen nur kleine Budgets zur Verfügung (Bergeron, 2006).

Auch die einzusetzende Technologie ist abhängig von den verfügbaren Ressourcen. Von der kommerziellen Unterhaltungsspielbranche eingesetzte Technologien erfordern viel Wissen über Softwareentwicklung, Programmierung und Design. Der zu stämmende Aufwand erweist sich meist als so gross, dass Block-Buster-Projekte Dutzende von Mitarbeiter benötigen. Dieser Aufwand rechnet sich zurzeit noch nicht für die Entwicklung von Lernspielen. Es besteht jedoch die Möglichkeit, Lernspiele mit Hilfe von Autorenwerkzeuge und somit erheblich weniger aufwendig zu produzieren. Auf aufwendige 3-dimensionale Animationen kann, falls diese nicht zentraler Bestandteils des Inhalts sind, ohne Weiteres verzichtet werden, da die Spielstruktur für die Bindung und die Motivation der Spieler von wesentlich grösserer Bedeutung ist als realistische 3-dimensionale Welten (Rollings & Adams, 2003). Ausserdem belegt Scott McCloud (1994), dass sich Leser/Spieler mit abstrakten Figuren besser identifizieren können als mit sehr detailgetreuen.

Schlussfolgernd kann festgehalten werden, dass allgemeine Richtlinien für die Entwicklung von Lernspielen schwierig zu identifizieren sind, da das optimale Design eines digitalen Lernspiels (mit starker Bindung und grossem Lerneffekt) stark vom Lernkontext abhängt (Michael & Chen, 2005). Grundsätzlich sollten die strukturellen Merkmale von Spielen für die Motivation und Bindung des Spielers bzw. Lerners berücksichtigt werden und anhand des Lernkontexts auf das geeignete Spielgenre geschlossen werden (Prensky, 2001).

2.5.3 Zusammenfassung

Digitale Lernspiele sind in Unternehmen und an Schulen als Ergänzung zu klassischen Lehrmethoden noch nicht durchgängig akzeptiert und haben sich noch nicht vollständig durchsetzen können (Clark, 1983, 2007). Stattdessen werden sie in Europa meist noch abschätzig als Spiele ohne grossen Lerneffekt angesehen, weil die Synergien und genauen Wirkmechanismen zwischen Lernelementen und Spielmerkmalen noch nicht ausreichend erforscht und bekannt sind. Die *Game-based Learning*-Gemeinschaft wird hier noch mehr Forschungs- und Aufklärungsarbeit leisten müssen. Das Potential von digitalen Lernspielen wird aufgrund inhärenter Charakteristika von Spielen als enorm eingeschätzt, jedoch in den meisten Umsetzungen von Lernspielen noch bei weitem nicht optimal ausgenutzt. Ganz offensichtlich fehlt dazu noch das Wissen, motivierende Spielerlebnisse zu schaffen, und die Kosten für die Realisierung eines Lernspiels, das die in diesem Kapitel beschriebenen Anforderungen erfüllt, übersteigen die aktuell bereitgestellten Budgets der Hersteller deutlich.

Dieses Kapitel sollte aufzeigen, warum Massnahmen zur Adaption klassischer Lehrmethoden für die *Game Generation* getroffen werden müssen und wie diese umgesetzt werden können. Des Weiteren wurden die Anforderungen an Lernspiele sowie die Synergien zwischen Lernelementen und Spielmerkmalen verglichen und erläutert.

James Paul Gee (2003) formuliert dazu folgendes:

„Cognitive science has had a hard time defining motivation, though one definition is a learner's willingness to make an extended commitment to engage in a new area of learning [diSessa 2000]. [...]

Finally, we can state that when players play in massive multiplayer games, they often collaborate in teams, each using a different, but overlapping, set of skills, and share knowledge, skills, and values with others both inside the game and on various Internet sites. In the process, they create distributed and dispersed knowledge within a community in ways that would please any contemporary high-tech, cross-functional-team-centered workplace [...]

In this respect, games may be better sites for preparing workers for modern workplaces than traditional schools. However, in the end, the real importance of good computer and video games is that they allow people to re-create themselves in new worlds and achieve recreation and deep learning at one and the same time.“

Garris und Kollegen (2002) formulieren dazu – basierend auf Cordova & Lepper (1996) sowie Ricci und Kollegen (1996) – folgendes:

„A second reason underlying current professional interest in computer games is that some empirical evidence exists that games can be effective tools for enhancing learning and understanding of complex subject matter.

In brief, the potential of instructional games as platforms for training is appealing.

As Simons (1993) claimed, “If video games can be transformed so that their users learn, a great many people may come to understand and control dynamic systems” (p. 149).

Caillois (1961) has provided perhaps the most comprehensive analysis of games per se, describing a game as an activity that is voluntary and enjoyable, separate from the real world, uncertain, unproductive in that the activity does not produce any goods of external value, and governed by rules.“

Jenkins (2006) notierte sich folgendes zu *Game-based Learning*:

„Whitehall and McDonald argued that incorporating a variable payoff schedule into a simulation game led to increased risk taking among students, which resulted in greater persistence on the task and improved performance. Ricci et al. proposed that instruction that incorporated game features enhanced student motivation, which led to greater attention to training content and greater retention.

Druckman (1995) concluded that games seem to be effective in enhancing motivation and increasing student interest in subject matter, yet the extent to which this translates into more effective learning is less clear.“

3. Evolutionäres Lernspielkonzept

3.1 Einleitung

Die rasante Entwicklung der Kommunikationstechnologien bedingt immer grösser werdende Mengen an Informationen, die Studenten, Forscher und Mitarbeiter mit ständig wechselnden Anforderungen konfrontiert und kontinuierliche Wissenserweiterung resp. -aktualisierung von ihnen verlangt. Diese ständige Weiterbildung hat unter dem Schlagwort "Lebenslanges Lernen" bereits Einzug in den alltäglichen Sprachgebrauch gehalten und erfordert neben einer revidierten Einstellung zu selbstverantwortlichem Lernen auch neue Methoden, um die Lernenden zu begeistern und zu motivieren (Meier & Seufert, 2003). *Game-based Learning* gilt als vielversprechende Option, diesen Anforderungen gerecht werden zu können und kombiniert spielerische Elemente mit konstruktivistischen Lernmethoden (s. Kapitel 2.5.1). Folgend werden die im Kapitel 1 beschriebenen Grundlagen um Ausführungen zum *Evolutionären Lernspielkonzept* erweitert. Dieses stellt somit als Kombination mehrerer Ansätze eine weitere Methode der Wissensvermittlung dar.

Inspiziert von Mitchel Resnicks '*Lifelong Kindergarten Approach*' und Kerstin Reichs Forschung zu Interaktionistischem Konstruktivismus wurden beide Ansätze in einem neuen Konzept *Evolutionärer Lernspiele* kombiniert. Dieses vereinigt zudem die in den Abschnitten 2.5 und 2.1 vorgestellten lerntheoretischen Grundlagen *Game-based Learning* und aktuelle Trends des Internets, wie z.B. nutzergenerierte Inhalte, aktive Personalisierung, Feedback- und Bewertungssysteme.

Die Lernmethode des *Evolutionären Lernspielkonzepts* beruht, ähnlich dem *Game-based Learning*, auf dem konstruktivistischen Paradigma. Darauf übertragen, können die Theorien von Piaget (2003) und Papert (1998) wie folgt interpretiert werden:

Nutzer lernen nicht nur während des Spielens, sondern ebenfalls beim Erstellen von Spielinhalten. Durch das iterative Optimieren der Inhalte gleichen Lernende ihre konstruierte Welt der Umwelt/Realität an. Diese als „Lernen durch Synthese“ bekannte gewordene Methode fördert das Verständnis der internen Mechanismen des Lernspiels und stimuliert das kreative Denken zum Erstellen eigener Szenarien (Papert, 1993). Des Weiteren können während der kollaborativ Gestaltung neuer Inhalte Kommunikations- und Führungsfähigkeiten sowie weitere soziale Kompetenzen trainiert werden.

Das *Evolutionäre Lernspielkonzept* wird in dieser Arbeit auf die Vermittlung sozialer Kompetenzen und Techniken kreativen Denkens angewandt, denn soziale Kompetenzen gewinnen gegenüber fachlichen immer stärker an Einfluss auf schulische und berufliche Qualifikationen. Auch die Fähigkeit, kreativ zu denken hat sich als wichtige Anforderung an Studenten und Berufstätige etabliert, weil das Beherrschen entsprechender Techniken zur Lösungsfindung von komplexen, situativ-variablen Problemen beiträgt.

Es wird an dieser Stelle nochmals darauf hingewiesen, dass der Schwerpunkt dieser Arbeit auf der technologischen Umsetzung des *Evolutionären Lernspielkonzepts* liegt und deshalb nicht detaillierter auf didaktische und pädagogische Aspekte eingegangen wird. Untersuchungen über Lernerfolg und *E-Assessment* wurden von Robert Stoyan durchgeführt und sind Bestandteil seiner Dissertation.

Neben den genannten didaktischen und pädagogischen Effekten soll durch Einbeziehen der Nutzer in den Erstellungsprozess der kostenintensive Teil der Inhaltsgenerierung günstiger gestaltet werden. Wichtigster Faktor ist dabei das Erreichen einer kritischen Masse an aktiven Nutzern, welche bereit sind, Inhalte zu erstellen. Dazu muss die Entstehung einer virtuellen Gemeinschaft rund um das Lernspiel gefördert und genutzt werden. Dies bedeutet einerseits, dass die Lerngemeinschaft mit zweckmässiger und zeitgemässer Technologie bei der gemeinsamen Identitäts- und Wissensbildung unterstützt werden muss und andererseits, dass die so gewonnenen Erkenntnisse der Lerngemeinschaft auch in die Weiterentwicklung der Lernplattform einfließen müssen. Zur Förderung langfristiger Motivation der Mitglieder können ihnen mit steigender Erfahrung neue Aufgaben und Verantwortungen sowohl im

Spiel, als auch im Kontext der virtuellen Gemeinschaft der Lerner und Inhaltentwickler übertragen werden.

3.2 Konzept

Das *Evolutionäre Lernspielkonzept* orientiert sich – wie bereits einführend erwähnt – an Mitchell Resnicks „*Lifelong Kindergarten Approach*“, Kersten Reichs Modell *Interaktionistischem Konstruktivismus* und aktuellen Internettrends. Die vorgestellten Modelle beinhalten jeweils die drei Phasen Spielen, Kreieren und Zurverfügungstellen des Modells von Evans (2007). Diese bilden den Kern des *Evolutionären Lernspielkonzepts* und werden deshalb noch einmal überblicksmässig skizziert: In der Phase 'Spielen' macht sich der Lernende mit den Grundinteraktionsmustern des Spiels vertraut und kann explorativ das Szenario erkundschaften, d.h. er darf und soll Fehler machen und aus diesen lernen (engl. *Trial and Error*). Des Weiteren unterstützt die Phase ‚Spielen‘ den Lernenden dabei, die Mechanismen und den Aufbau der Szenarien zu verstehen. In der Phase 'Kreieren' benutzt der Lernende seine Erfahrungen mit bereits gespielten Szenarien sowie seine Phantasie, um eigene Szenarien individuell oder kollaborativ mit anderen Nutzern zu erstellen. Ziel dieser Phase ist es, die internen Mechanismen des Spiels und des Lerninhalts zu verstehen und aufgrund dessen eigene Inhalte zu generieren. Hierfür muss der Lernende den Lerninhalt verstanden haben und fähig sein, die Lerninhalte auf sein eigenes Szenario anzuwenden (Papert, 2003). Richtlinien und Applikationen sollen dem Lernenden dabei Hilfsmittel zur Verfügung stellen, die ihn bei der Generierung der Inhalte führen. In der letzte Phase des Zyklus' werden die erstellten Szenarien anderen Lernenden zur Verfügung gestellt. Diese können die Inhalte bewerten, vergleichen und den Autoren Feedback geben. Durch das Feedback wird die Qualität erhöht, denn wie in Webapplikationen soll das Publizieren der Szenarien eine Präsentation eigenen Könnens und eigener Ideen sein. Das bietet unter anderem eine Gelegenheit, hohe Reputation zu erlangen. Ergänzt wird das Grundmodell durch die drei Phasen Inhalte bewerten, imaginieren und reflektieren.

Zum besseren Verständnis des Lesers wird das Konzept der *Evolutionären Lernspiele* anhand des Beispiels Projektmanagement Game (*PM Game*) erläutert, welchem

Kapitel 4 eine ausführliche Diskussion widmet. Beim *PM Game* handelt es sich um eine Mischung aus Text- und Graphik-Adventure (s. Kapitel 2.4.3). Die Nutzer des *PM Game* übernehmen die Rolle eines Projektmanagers und interagieren zur Bearbeitung typischer Aufgaben aus dem Berufsalltag von Projektmanagern mit Teammitgliedern, Kunden, Vorgesetzten und anderen Interessensvertretern. Diese Interaktionspartner werden vom System simuliert und konfrontieren den Nutzer mit einem breiten Spektrum verschiedener Persönlichkeiten. Das Verhalten und die Emotionen der computergesteuerten Personen (Non-player Characters, *NPC*) können mit Hilfe eines Editors erweitert bzw. modifiziert werden. Der Nutzer interagiert mit den *NPC* mittels eines neuartigen Dialogsystems, welches auf so genannten Sprechakten (*Speech Acts*) basiert (Austin, 2005; Searle, 1969). In Abhängigkeit der Reaktion der Nutzer auf die unterschiedlichen Verhaltensweisen, Emotionen und Persönlichkeiten der *NPC* entwickelt sich der weitere Verlauf des Szenarios (vgl. Diller et al., 2004). Folgendes Anwendungsszenario wird prototypisch für ein *Evolutionäres Lernspiel* verwendet:

Ein Student hat eine Einladung für die Teilnahme am *PM Game* per E-Mail von einem Studienkollegen erhalten. Die Einladungen können direkt von der Spieloberfläche an potentielle Interessenten versendet werden. Der Student gelangt über den Link im E-Mail auf eine Registrierungsseite. Nachdem er sich mit Benutzernamen und E-Mailadresse registriert hat, kann er sich ins Spiel einloggen. Als neuer Nutzer steht dem Spieler ein Tutorial zur Verfügung, das ihm Schritt für Schritt die Interaktionsmöglichkeiten erklärt und ihn auf die dem *PM Game* zugrundeliegenden Handlungsmuster sensibilisiert. Nach dieser Einarbeitung kann der Nutzer nun Szenarien spielen, welche von erfahrenen Nutzern oder Experten aus dem Bereich Projektmanagement erstellt wurden. Zur Information, welches Szenario seinen Qualitätsansprüchen genügt und den gewünschten Lerninhalt zur Verfügung stellt, sieht der Nutzer eine Detailseite mit Bewertungen und qualitativem Feedback anderer Nutzer. Diese Informationen sollen als Entscheidungsgrundlage für die Auswahl der zu spielenden Szenarien dienen. Der Nutzer kann während und nach dem Spielen Feedback zum Szenario, zu allfälligen Fehlern in der Applikation oder zu Unklarheiten in den Lerninhalten geben. Des Weiteren kann er Verbesserungen des *PM Game* resp. des Szenarios vorschlagen. Diese werden vom Lernspielhersteller oder Verantwortlichen analysiert und gegebenenfalls realisiert. Mit diesen nutzergenerier-

ten Inhalten werden die Vielfalt des *PM Game* gesteigert und emergente Unterhaltung geschaffen. Je mehr Szenarien der Nutzer durchgespielt hat, desto besser wird sein Verständnis für die Regeln/Inhalte des *PM Game* und desto differenzierterer fällt sein Feedback an die Ersteller aus. Die so gewonnene Erfahrung versetzt ihn auch in die Lage, schlechte von guten Szenarien zu unterscheiden und die internen Mechanismen des Spiels immer besser zu verstehen. Ausserdem wird der Nutzer animiert, sich eigene Szenarien vorzustellen und zu implementieren zu versuchen. Hat der Nutzer die nötige Erfahrung und verfügt über entsprechende, von den Administratoren / Spielerstellern vergebene Berechtigungen, kann er andere Nutzer in eine von ihm erstellte virtuelle Gruppe einladen und mit ihnen kollaborativ Szenarien erstellen. Durch die Kollaboration kann u.U. eine Eigendynamik innerhalb der Gruppe erzeugt werden, welche die Gruppenmitglieder zu Kreativität und Engagement motiviert und neue Möglichkeiten der Implementierung beflügelt.

Nachdem der Nutzer ein Szenario erstellt hat und mit dem Resultat zufrieden ist, kann er es anderen zum Spielen zur Verfügung stellen, nachdem es von Experten begutachtet und freigegeben wurde. Falls keine gravierenden Mängel vorhanden sind, wird das Szenario für alle zugänglich gemacht. Der Zyklus beginnt von vorn: Der Nutzer erhält nun selbst Feedback von anderen Spielern, reflektiert und kann sein Szenario stetig verbessern oder neue entwerfen. Diese iterative Vorgehensweise zur Optimierung der Inhalte für das *Evolutionäre Lernspiel* wurde von agilen Softwareentwicklungsprozessen inspiriert (vgl. Beck, 2000). Die Herausforderung ist dabei die Qualitätssicherung der Inhalte, damit die Nutzer der Szenarien nicht durch Fehler oder Inkonsistenzen im Inhalt frustriert werden. Das Ziel dieses Vorgehens besteht darin, die Szenarien kontinuierlich zu optimieren. Dadurch erhalten die Ersteller die Möglichkeit immer besserer Bewertungen, was zur Steigerung ihrer Reputation innerhalb der virtuellen *PM Game*-Gemeinschaft führt.

Tabelle 12 fasst die 6 Phasen im Detail nochmals zusammen. Abbildung 20 visualisiert den gesamten Zyklus des *Evolutionären Lernspielkonzepts*, welcher die iterative Lernverankerung widerspiegelt.

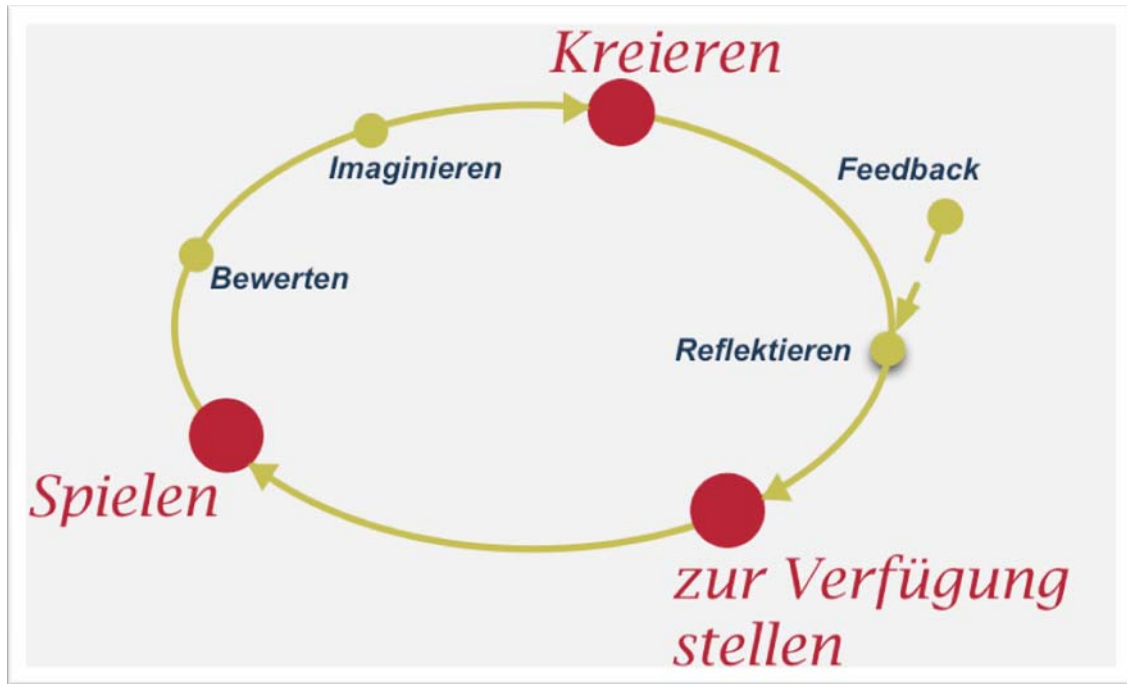


Abb. 20: Phasen des Evolutionären Lernspielkonzepts

Tab. 12: Übersicht über die Phasen des Evolutionären Lernspielkonzepts

Phase	Beschreibung
Spielen	<p>Der Nutzer spielt Szenarien, wird auf die internen Mechanismen des Spiels und Lerninhalte sensibilisiert und zur Erstellung eigener Szenarien inspiriert.</p> <p>Diese Phase ist der Einstiegspunkt für neu registrierte Nutzer, um sich mit dem System und den Prozessen bekannt zu machen.</p>
Bewerten	<p>Ein wichtiger Bestandteil des Modells sind die Bewertungsmechanismen im Spiel zur Optimierung der Inhalte. Die Bewertungen geben einerseits Auskunft über die Reputation des Erstellers und andererseits differenzieren sie gute von schlechten Inhalten, was eine 'natürliche' Selektion zur Folge hat.</p> <p>Des Weiteren sollen Feedbacks der Nutzer direkt vom System ausgewertet und gespeichert werden, um die Diversität durch individuelle Erfahrungen und</p>

	<p>branchenspezifische <i>Best Practices</i> zu erhöhen. Mit diesem Verfahren entsteht ein evolutionäres System.</p> <p>Diese Phase stellt eine der zentralen Massnahmen zur Qualitätssicherung im Konzept dar und erfüllt eines der wichtigsten Merkmale des Web 2.0 (s. Kapitel 2.1).</p>
Vorstellen/Imaginieren	<p>Durch Spielen und Bewerten der Szenarien wird der Spieler angeregt, eigene Inhalte zu entwickeln (Resnick, 2007), da er entweder Potential für eine Erweiterung eines bestehenden Inhalts sieht oder weil er den Lerninhalt in einen anderen Kontext transferieren möchte.</p>
Kreieren	<p>Der erfahrene Nutzer kann sich einer bestehenden Gruppe anschliessen und mithelfen, Inhalte zu entwickeln, oder eine eigene Gruppe gründen und kollaborativ neue Inhalte nach eigenen Ideen generieren.</p>
Feedback & Reflektieren	<p>Inhalte, welche anderen Nutzern zur Verfügung gestellt werden sollen, müssen zur Qualitätssicherung von Experten begutachtet werden, um Systemblockaden oder pornographische, hetzerische oder rassistische Szenarien zu verhindern. Das Feedback der Experten soll zudem den Nutzer oder die Gruppe zur Reflexion anregen: Die Ersteller können sich Gedanken zur Verbesserung der Inhalte machen und somit die Qualität ihrer Inhalte verbessern.</p>
Zur Verfügung stellen	<p>Experten machen die Inhalte nach erfolgter Prüfung für alle Nutzer zugänglich. Diese werden benachrichtigt, dass neue Inhalte aufgeschaltet wurden und können diese nun spielen, bewerten und Feedback geben.</p>

3.3 Komponenten *Evolutionärer Lernspiele*

Evolutionäre Lernspiele sind eine Kombination aus Lernplattform (s. Kapitel 2.3), Autorenwerkzeug (s. Kapitel 2.2) und digitalem Lernspiel (s. Kapitel 2.4) ergänzt um Web 2.0-Funktionalitäten (s. Kapitel 2.1). In diesem Kapitel sollen die zuvor erläuterten Grundlagen dieser Komponenten zusammengeführt und ihre Schnittstellen diskutiert werden, um Kombinationsmöglichkeiten aufzuzeigen und einen generischen Bauplan *Evolutionärer Lernspiele* abzuleiten.

Die Basis *Evolutionärer Lernspiele* bildet die Lernplattform. Die für diesen speziellen Anwendungskontext adaptierten Aufgaben einer Lernplattform werden im Folgenden erläutert:

Benutzerverwaltung

Zugangsdaten, Präferenzen sowie Profile der Nutzer werden in der Benutzerverwaltung editiert. Ebenfalls werden Lernfortschritt und Evaluation der Lernenden mit den deren individuellen Daten verknüpft. Die Benutzerverwaltung beinhaltet ein Berechtigungssystem, über welches die Rechte der Nutzer im *Evolutionären Lernspiel* festgelegt werden. So darf ein neuregistrierter Nutzer z.B. Lernszenarien spielen und diese bewerten, jedoch keine Veränderungen an den Szenarien vornehmen oder eigene erstellen. Die Zuteilung dieser Rechte erfolgt entweder durch einen Systemadministrator oder durch das System selbst, basierend auf der Erfahrung des Nutzers mit dem Lernspiel, die über dessen aufgewendete Gesamtspielzeit und erstellte Beiträge (z.B. Feedbacks oder Bewertungen) operationalisiert wird.

Tracking und Auswertung des Benutzerverhaltens und Lernfortschritts der Nutzer

Auswertungsfunktionen sind ein wichtiger Bestandteil *Evolutionärer Lernspiele*. Lehrende und Experten können Informationen über die Lernenden, wie Punktestand, gelöste Aufgaben und absolvierte Tests abfragen und miteinander vergleichen. Damit wird dem Lehrenden die Möglichkeit gegeben, individuell Feedback zu geben und auf eventuelle Probleme der Lernenden einzugehen. Des Weiteren können Evaluationen durchgeführt werden, um das System zu verbessern.

Kommunikationsschnittstelle Experten:Lernende bzw. Lernende:Lernende

Nutzer können andere Nutzer in Lerngruppen einladen, mit anderen Lernenden chatten, Empfehlung abgeben, mit Experten Kontakt aufnehmen oder Feedback zu Lernszenarien geben. Die Integrierung einer Kommunikationsschnittstelle soll die Hemmschwelle zur Kommunikation zwischen Lernenden und Lehrenden senken.

Verwaltung und Koordination der Lernszenarien (inklusive Qualitätssicherung)

Lehrende und/oder Experten begutachten die erstellten Inhalte, kategorisieren diese, geben den Autoren Feedback und schalten die Inhalte nach erfolgter Prüfung für alle Nutzer frei. Für diese Aufgaben steht den Experten/Lehrenden eine graphische Benutzerschnittstelle zur Verfügung. Der expertenbasierte Freischaltmechanismus stellt einen ersten Schritt zur Qualitätssicherung dar. Weitere Massnahmen werden im nächsten Kapitel 3.3 beschrieben.

Infrastruktur für kollaboratives Lernen

Nutzer können Lerngruppen bilden, um mit den Autorenwerkzeugen kollaborativ Lernszenarien zu entwickeln. Die Lernplattform stellt die für Gruppenbildung und -verwaltung nötige Infrastruktur zur Verfügung. So wird Gruppen ein asynchrones Erstellen von Inhalten ermöglicht.

Coachingsystem

Das digitale Lernspiel gibt dem Nutzer die Gelegenheit, sein erworbenes Wissen durch personalisiertes Coaching zu verankern. Die Regeln des Lernspiels steuern nicht nur die Reaktion des Spiels, sondern können ebenfalls Fehlentscheidungen erkennen und dem Lernenden gegebenenfalls Hinweise dazu geben. Diese Hilfestellungen werden in der Lernplattform verwaltet und mit Hilfe eines Autorenwerkzeugs erstellt. Auch die Auswertungen der gespielten Lernszenarien werden mit Hilfe der Lernplattform verwaltet und den Lehrenden zur individuellen Betreuung der Lernenden zur Verfügung gestellt.

Lernplattformen bieten meist keine integrierte Lösung zur Erstellung multimedialer Inhalte (eLearn jku, 2006). Hierfür muss meist auf eine weitere Applikation zur Verfügung zurückgegriffen und deren Benutzung von den Autoren erlernt werden. In

Evolutionären Lernspielen sollen die Autorenwerkzeuge in das System integriert sein, um die Prozesse zur Inhaltserstellung möglichst dem Interaktionsdesign des Spiels anzupassen und so den Einstieg in die Inhaltsgenerierung durch Wiedererkennungseffekte zu vereinfachen. Mit Hilfe von Autorenwerkzeugen können Inhalte *Evolutionärer Lernspiele* erstellt und/oder konfiguriert werden. Da Spiele im Allgemeinen komplexer aufgebaut sind als Multiple Choice-Aufgaben oder Simulationen werden für die Erstellung verschiedener Inhalte auch mehrere Autorenwerkzeuge benötigt.

Regeleditor

Spiele bestehen aus Regeln, welche die Spiellogik abbilden (Björk & Holopainen, 2005, Salen & Zimmerman, 2003). Mit einem Regeleditor wird es möglich die Spiellogik zur Laufzeit zu erstellen oder anzupassen. Es ist jedoch sehr aufwändig, die gesamte Spiellogik per Regeleditor konfigurierbar zu machen. Deshalb sind im Rahmen des in dieser Arbeit realisierten Prototyps nur ausgewählte Aspekte der Spiellogik veränderbar. Die grundlegende Spiellogik ist durch das Genre oder Subgenre vorgegeben und nicht veränderbar. Es ist somit nicht möglich, mit Hilfe des Regeleditors aus einem Brettspiel ein Adventure-Spiel zu kreieren.

Das Erstellen von Regeln erfordert viel Wissen über die internen Mechanismen des Spiels und den Aufbau der Lerninhalte weshalb dazu vornehmlich auf Experten oder erfahrene Nutzer zurückgegriffen wird. Fehlerhafte Regeln können zu Systemblockaden führen, was die Nutzer frustrieren und die Akzeptanz des *PM Game* schmälern kann. Deshalb spielen Qualitätssicherungsmaßnahmen für benutzer-generierte Regeln eine wichtige Rolle in *Evolutionären Lernspielen*.

Für die Implementierung der Regeln eines Spiels können grundsätzlich Varianten aus der Automatentheorie und der Prädikatenlogik verwendet werden. Für detaillierte Beschreibungen wird auf die Literatur von Homuth (1977) und Barwise & Etchemendy (2005) verwiesen.

Die Einarbeitungszeit in die Nutzung des Regeleditors ist – einhelligem Feedback zufolge – für Nutzer ohne Affinität zu Informationstechnologien sehr gross, da grundsätzliches Wissen über Prädikatenlogik, Algorithmen und Datenstrukturen

vorausgesetzt werden muss. Bei den Recherchen zu dieser Arbeit wurde kein Autorenwerkzeug resp. keine Methode gefunden, welche ein *end-user programming* ohne längere Einarbeitungszeiten ermöglichte.

Konfigurator zum Erstellen der Inhalte

Das Erstellen von Lernszenarien ist jedoch für jeden Nutzer ohne lange Einarbeitungszeiten möglich, wenn dazu keine Editierung von Regeln notwendig ist (s. Kapitel 5). Stattdessen erfordert es die Konfiguration der visuellen Elemente und der Grundeinstellungen der Lernszenarien sowie die Definition des Lerninhalts. Eine derartige Erstellung von Lernszenarien kann ohne technologisches Vorwissen voraussetzen kollaborativ und iterativ stattfinden.

Evolutionäre Lernspiele benötigen erstellte Inhalte, welche via Autorenwerkzeug ins System eingespeist wurden sowie von der Lernplattform verwaltete Angaben über die Lernenden. Im Gegensatz zu herkömmlichen Lernplattformen wird in dieser Arbeit die Applikation für die Lernenden (im Falle von *Evolutionären Lernspielen* das eigentliche Lernspiel) als dedizierter Dienst und nicht als integrierter Bestandteil der Lernplattform vorgestellt, da das Spiel auch ohne Lernplattform implementiert werden sollte. Das Spiel besteht aus einer Benutzerschnittstelle (Spieloberfläche) mit diversen Bewertungs- und Feedbackfunktionen. Die Lernszenarien und die per Regeleditor erstellte Spiellogik werden in das Spiel geladen und bestimmen die Interaktionsmöglichkeiten des Lernenden. Bewertungs- und Feedbackfunktionalitäten sind wichtiger Bestandteil des Spiels und dienen der Qualitätssicherung der Lernszenarien. Durch die Bewertungen wird eine nutzerbasierte Selektion der Inhalte realisiert: Unzureichend geeignete Lernszenarien werden von den Nutzern aussortiert, während gut bewertete Szenarien häufiger gespielt und als Vorbild für gute Lernszenarien herangezogen werden können. Ausserdem wird der Autor durch die Anerkennung seiner Leistungen motiviert, weitere Lernszenarien zu generieren, um seine Reputation zu stärken.

In Abbildung 21 werden die Komponenten *Evolutionärer Lernspiele* nochmals in Kombination dargestellt. Die Kommunikation der einzelnen Komponenten erfolgt über ihre definierten Schnittstellen (*API*). So werden z.B. die Rechte und Rollen eines Nutzers von der Lernplattform verwaltet und den anderen Komponenten über das *API* zur Verfügung gestellt. Der Regeleditor kann somit die Berechtigungen des Nutzers über die Schnittstelle der Lernplattform abfragen und für seine Zwecke weiterverwenden. Die Benutzerschnittstelle jeder Komponente wurde in Abbildung 21 mit dem

Akronym der englischen Bezeichnung „Graphical User Interface“ (GUI) gekennzeichnet.

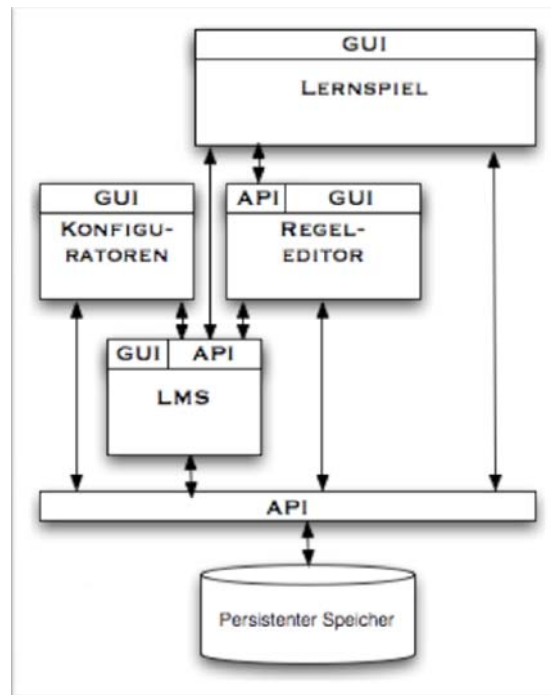


Abb. 21: Komposition der Komponenten

3.4 Softwarearchitektur *Evolutionärer Lernspiele*

Wegen der kontinuierlichen Änderungen und Erweiterungen, denen Webapplikationen heutzutage ausgesetzt sind, wird in dieser Arbeit eine modulare Softwarearchitektur vorgeschlagen, welche in der von Sun® entwickelten Programmiersprache / Technologie *Java™* (vgl. Sun, 2008) mittels *OSGi™* (vgl. OSGi Alliance, 2008) oder mittels der *Service Component Architecture*-Spezifikation (vgl. OSOA, 2008) implementiert werden könnte. *OSGi™* (Open Service Gateway Initiative) definiert ein dynamisches Komponentensystem für *Java™*, mit dem Ziel, Applikationen dynamisch aus verschiedenen, wieder verwendbaren Diensten aufzubauen. Dynamisch meint in diesem Fall das Hinzufügen von Modulen zur Laufzeit (vgl. Wilding, 2009). Die Spezifikation SCA (Service Component Architecture) definiert ebenfalls, wie Dienste aufgebaut und komponiert werden, ist jedoch nicht an eine Programmiersprache gebunden. Die Dienste können somit in verschiedenen Programmiersprachen in einer verteilten Umgebung komponiert werden (Thierwächter, 2008). In dieser Arbeit wird nicht genauer auf die Eigenschaften dieser Architekturen eingegangen. Der interessierte Leser wird auf Erl (2008) verwiesen.

Im ersten Teil dieses Abschnitts wird ein generischer Aufbau von *Evolutionären Lernspielen* und deren Komponenten diskutiert. Des Weiteren werden die Module einer Komponente erklärt, insbesondere deren Rollen, Verantwortlichkeiten und Kollaborationen. Der Schwerpunkt dieses Kapitel wird auf die Modularisierung, Skalierbarkeit und Wiederverwendbarkeit der Software gelegt. Die genannten Schlagworte bilden die fundamentalen Anforderungen an neue Softwarearchitekturen und spielen ebenfalls eine zentrale Rolle für *Evolutionäre Lernspiele*. In Kapitel 4 wird eine prototypische Implementierung anhand des Lernspiels *PM Game* besprochen, welches nur einen Teil sämtlicher Möglichkeiten ausschöpft.

Jede Komponente besteht aus den Modulen Steuerung, Repräsentation und Modell, basierend auf dem *MVC*-Architekturmuster (engl. *Model-View-Control*). Ziel dieses Musters ist die Trennung funktional verschiedener Einheiten der Software in

Benutzerschnittstelle, Datenhaltung und Steuerungsmechanismen, um dadurch eine für die Entwickler übersichtliche Struktur der Software zu schaffen (Freeman et al., 2004). Zudem bieten der Konfigurator, das Lernspiel, der Regeleditor und die Lernplattform Schnittstellen über das Entwurfsmuster *Facade* an (Gamma et al., 1995). Die Entwickler können gegen diese *Facaden* programmieren und somit auf die Module innerhalb der anderen Komponente zugreifen. Im Folgenden sollen nun die einzelnen Module generisch erklärt werden. Eine Implementation einer prototypischen Komponente wird in Kapitel 4.9 vorgestellt und diskutiert.

Die *Facade* ist die Kommunikationsschnittstelle jeder Komponente und assoziiert Steuerung, Präsentation und Modell mittels Komposition. Die *Facade*, die Steuerung, die Präsentation und das Modell werden nur einmal instanziiert.

Das Modell referenziert und ermöglicht Zugang zu Stellvertreter(n). Diese Stellvertreter speichern die Inhalte der Datenbank in Datenstrukturen oder halten Referenzen auf Datenobjekte. Falls die Datenbank oder Datenobjekte angepasst werden müssen, ist nur ein minimaler Anpassungsaufwand nötig, um die Schnittstellen zu aktualisieren (Freeman et al., 2004). Die Daten können entweder asynchron über Webservices und *Remote Procedure Calls* oder synchron von der Datenbank und Datenobjekten bezogen werden. Des Weiteren implementieren die Stellvertreter die Schnittstelle Notifikation, um über die *sendNotification*-Methode Nachrichten asynchron via *Facade* zu verschicken.

Mediatoren registrieren sich bei der Präsentation und müssen dieser eine Liste von Notifikationen übergeben, über die sie informiert werden möchten. Bei der Registrierung der Mediatoren wird ein Beobachterobjekt für jede gewünschte Notifikation durch die Präsentation instanziiert. Sobald nun eine – einen angemeldeten Mediator betreffende – Nachricht versendet wird, werden alle Beobachter informiert und die Methode *handleNotification* des betreffenden Mediators wird ausgeführt. Des Weiteren können Mediatoren auch selbst Nachrichten versenden, um andere Mediatoren zu informieren. Mediatoren sind damit das Bindeglied zwischen einzelnen Benutzerschnittstellen und den zu präsentierenden Daten. Die Benutzerschnittstellen können entweder über ein *Whiteboard* oder *Listeners* mit den Mediatoren kommunizieren (vgl. Kriens & Hargrave, 2004). Ziel dieser Abstraktion

ist es, die Repräsentation der Daten weitgehend von der Datenhaltung zu trennen und somit die Benutzerschnittstelle so einfach wie möglich austauschbar zu gestalten.

Die Steuerung wird durch die Referenzierung von Befehlen realisiert. Befehle interagieren mit den Stellvertretern und kommunizieren mit Mediatoren: Die Steuerung instanziiert zu jeder Notifikation einen Beobachter, registriert diesen bei der Präsentation und ordnet jeder Notifikation einen Befehl zu. Beim Versand einer Notifikation wird der zugeordnete Befehl durch die Steuerung instanziiert und anschliessend ausgeführt. Befehle können auch ineinander zu Makrobefehlen verschachtelt und so zu Befehlsfolgen orchestriert werden.

In Tabelle 13 werden die verwendeten Designmuster aufgelistet und ihr Zusammenspiel dokumentiert.

Tab. 13: Entwurfsmuster für *Evolutionäre Lernspiele*

Entwurfsmuster	Kommunikationsmöglichkeiten
Stellvertreter	versenden Nachrichten, können jedoch nicht auf solche reagieren
Mediatoren	können Nachrichten erhalten und versenden
Befehle	werden durch Nachrichten ausgelöst und können ebenfalls Nachrichten versenden

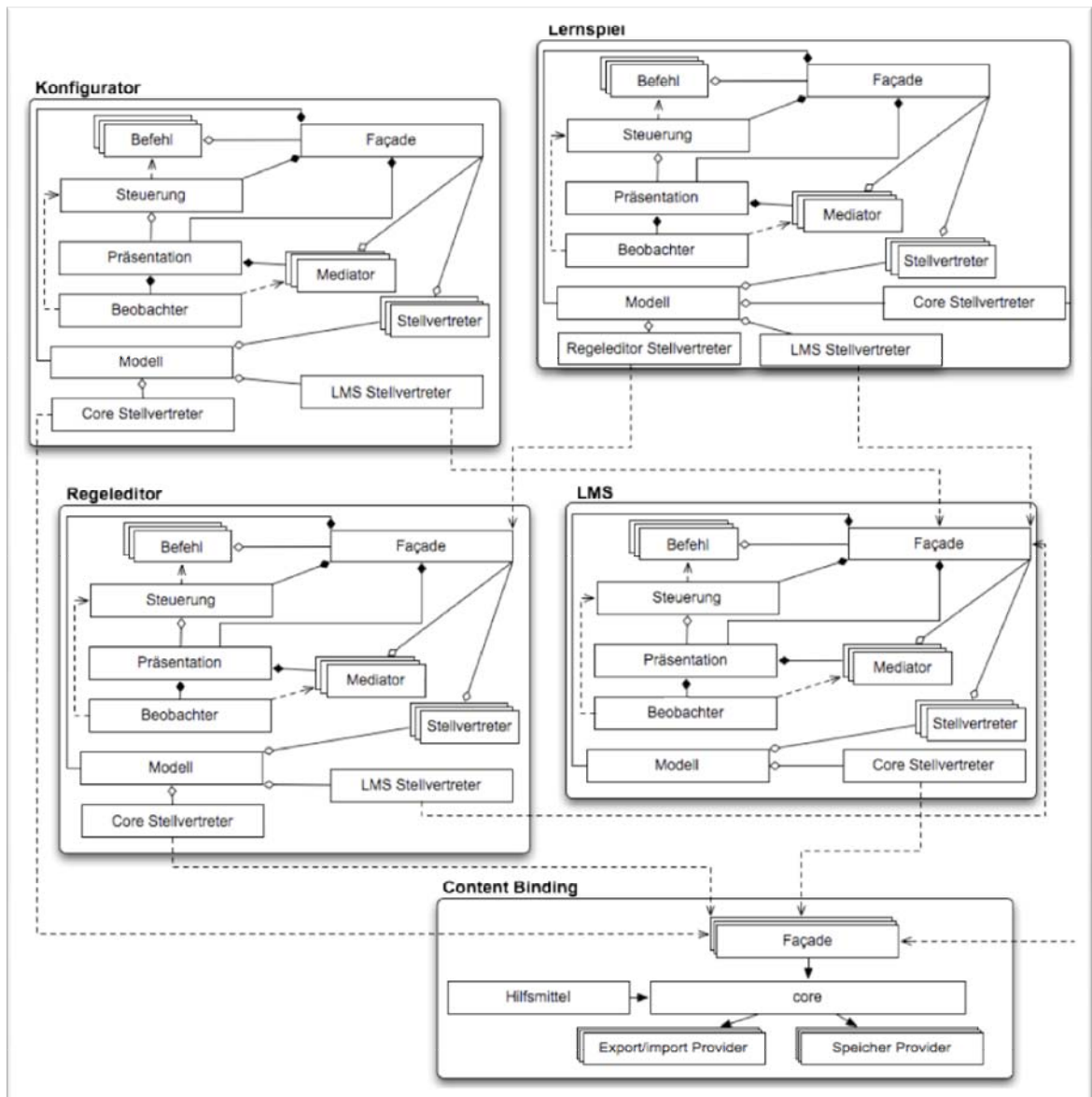


Abb. 22: Softwarearchitektur Evolutionärer Lernspiele

Die einzelnen Komponenten werden nach ihren Abhängigkeiten instanziiert: Zuerst das *Content Binding*, dann die Lernplattform, anschliessend der Regeleditor und der Konfigurator und abschliessend das Lernspiel. Jede Komponente wird über die Instanzierung der implementierten *Façade* initialisiert. Diese wiederum instanziiert die Präsentation, das Modell und die Steuerung. Nach der Initialisierung der Steuerung werden alle Befehle bei der Steuerung und die entsprechenden Beobachter bei der Präsentation registriert. Anschliessend wird eine Nachricht versendet, die den Startup-Befehl ausführt. Letzterer registriert alle Stellvertreter und Mediatoren. Bei der Registrierung der Mediatoren werden sämtliche Nachrichten, die für den Mediator von Interesse sind, aufgelistet und ein Beobachter instanziiert, d.h. der Mediator wird beim

Beobachter referenziert. Der Beobachter wird mit jeder gelisteten Nachricht registriert. Die Komponenten können z.B. via Webservices oder *Remote Procedure Calls* miteinander kommunizieren. Diese asynchrone Kommunikation erlaubt das Verteilen der Komponenten auf physisch getrennte Hardware (Server). Des Weiteren können andere Anwendungen über die *Facade* auf die freigegebenen Dienste der Komponente zugreifen und *Mashups* kreieren (s. Kapitel 2.1)

Zum besseren Verständnis des Zusammenspiels der Komponenten wird der Ablauf anhand des Login-Mechanismus' demonstriert:

Der Nutzer öffnet ein Browserfenster und gibt die URL der Lernplattform ein. Daraufhin erscheint ein Login-Fenster, in welches der Nutzer seine Zugangsdaten (Benutzernamen und Passwort) eingibt. Der Nutzer aktiviert die Schaltfläche "login" per Mausklick, worauf folgender Prozess gestartet wird: Der Mediator des Loginfensters versendet die Nachricht „Login“ mit den Zugangsdaten, worauf über die *Facade* die Beobachter informiert werden und der Login-Befehl über den Kontext Steuerung aufgerufen wird. Der Befehl ruft den *Core*-Stellvertreter auf, welcher die Zugangsdaten vom *Content Binding* holt und zwischenspeichert. Existiert kein Nutzer mit den versendeten Zugangsdaten, wird eine Nachricht vom *Core*-Stellvertreter mit der Fehlermeldung zurück geschickt, andernfalls wird der Nutzer authentifiziert und autorisiert.

Abbildung 23 bildet das beschriebene Szenario als Sequenzdiagramm ab.

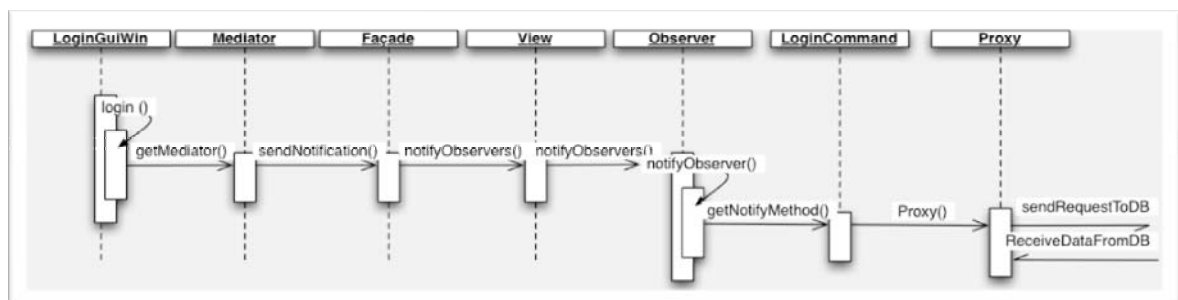


Abb. 23: Sequenzdiagramm des Login-Prozesses

Wird das *Evolutionäre Lernspiel* als Webanwendung realisiert, muss zusätzlich mit Latenzzeiten von der Benutzerschnittstelle des Clients zum Server gerechnet werden. Die Serveranfragen müssen bei jeder Aktion des Nutzers ausgelöst werden, können

jedoch durch asynchrone Kommunikationstechnologien, wie etwa *Asynchronous JavaScript and XML* (AJAX) optimiert werden.

3.5 Zusammenfassung

Das *Evolutionäre Lernspielkonzept* kombiniert aktuelle, unter dem Schlagwort Web 2.0 zusammenfassbare Trends des Internets mit der konstruktivistischen Lernmethode *Game-based Learning*. Letztere soll die *Game Generation* besser beim Lernen unterstützen als klassische Lehrmethoden und durch motivierende Spielmerkmale längerfristig an die Lernumgebung binden.

Ziel des *Evolutionären Lernspielkonzepts* ist einerseits die iterative Sensibilisierung der Lernenden auf den Lerninhalt durch aktives, selbständiges Erstellen eigener Inhalte (Lernen durch Synthese) sowie die Schaffung emergenter Unterhaltung durch die Kreativität der Lernenden andererseits. Des Weiteren dient die Kreativität als Basis für die Improvisationsfähigkeit der Lernenden im Berufsalltag. Das Modell wurde inspiriert von Mitchel Resnicks *„Lifelong Kindergarten Approach“*, welcher das Lernverhalten von Kindern im Vorschulalter zum Vorbild hat.

Das *Evolutionäre Lernspielkonzept* ist in 6 Phasen unterteilt: Spielen, Bewerten, Imaginieren, Kreieren, Feedback, Reflektieren und Zurverfügungstellen. Eine der grössten Herausforderungen bei der Implementierung des Konzepts liegt in der angemessenen, d.h kompetenzbasierten Zuordnung von Ausführrechten zu Nutzern. Diese soll die Frustration des Nutzers durch Über- oder Unterforderung vermeiden und die Qualität der Inhalte sicherstellen. In Kapitel 5 werden die Ergebnisse eines Akzeptanzuntersuchung beschrieben, welche einen Teil der offenen Fragen beantworten wird (s. Kapitel 1.2).

4. *PM Game*: Referenzimplementation des *Evolutionären Lernspielkonzepts*

4.1 Einleitung

Als Referenzimplementation des *Evolutionären Lernspielkonzepts* wurde das Projektmanagement Game (*PM Game*) entwickelt, das auf die Vermittlung von Lerninhalten in den Bereichen Projektkommunikation, Handlungskompetenzen und Führung von Kleingruppen ausgerichtet ist. Dieser Lerninhalt wurde aufgrund der am Educational Engineering-Lehrstuhl bestehenden Schwerpunkt-Arbeitsgruppe "Lernen und Beurteilen sozialer Kompetenzen im Projektmanagement" ausgewählt. Das Unterrichten sozialer Intelligenz und Handlungskompetenzen ist komplex, weil stark kulturabhängig (Marte et al., 2004; Schneider, 1996; Müller, 2004), und benötigt intensive Betreuung (Watzlawick et al., 2000). Darum verspricht das *Evolutionäre Lernspielkonzept*, eine gute Alternative zu personal- und kostenintensiven Rollenspielen und Kleingruppenunterricht zu sein. Trotz der besonderen Charakteristika digitaler Lernspiele, wie etwa Abstraktion und eingeschränkte Interaktionsmöglichkeiten, können durch ein adaptives System bereitgestellte Lerninhalte besser an die Bedürfnisse der Lernenden angepasst werden. Des Weiteren führt die Einbettung von Lerninhalten in ein digitales Spiel dazu, dass der Lernende durch die Herausforderungen des Spiels, die Verknüpfung des Lerninhalts mit einer Geschichte und die Möglichkeit der aktiven Beeinflussung des Spielgeschehens zur längerfristigen Auseinandersetzung mit den Lerninhalten motiviert wird (s. Kapitel 2.5.2).

Mit Hilfe des *PM Game* lernen die Nutzer typische Anforderungen an Fertigkeiten und Fähigkeiten eines Projektmanagers kennen und können diese spielerisch üben,

indem sie mit verschiedenen Stakeholdern interagieren. Dabei handelt es sich um vom System simulierte Figuren (*NPC*), die den Nutzer mit einem breiten Spektrum an verschiedenen Persönlichkeiten konfrontieren. In Abhängigkeit der Reaktion des Nutzers auf die verschiedenen Verhaltensweisen, Emotionen und Persönlichkeiten der *NPC* entwickelt sich ein anderer Verlauf des Szenarios.

Wie im Rahmen des *Evolutionären Lernspielkonzepts* vorgesehen, stellt das *PM Game* Autorenwerkzeuge zur Verfügung, mit denen kollaborativ Inhalte erstellt werden können. Das zugrunde liegende didaktische Konzept ist als „Lernen durch Synthese“ bekannt: Lernende sollen nicht nur im Spiel lernen, sondern auch während der aktiven Erstellung eigener Inhalte.

Das *PM Game* vermittelt neben theoretischem Hintergrundwissen zu Projektmanagement auch Beispiele für dessen Anwendung und gibt dem Spieler quantitatives Feedback, das auf Analysen seiner individuellen Aktionen während des Spielens basiert und auf Wunsch spezifische Coaching-Angebote während des Spiels oder danach auslösen kann. Dabei sollen Bezüge von konkreten Aktionen der Spieler zu wissenschaftlich abgesicherten und gleichzeitig praxisrelevanten Modellen der Führungs- und Kommunikationspsychologie sowie der Betriebswirtschaft hergestellt werden. Dabei wird u.a. auf Ohlsson (1994) Bezug genommen, der zeigen konnte, dass e-Coaching mit *constraint based*-Modelling die Effektivität menschlichen Einzelcoachings erreichen kann.

Das *PM Game* wurde 2008 an der Universität Zürich im Rahmen einer Lehrveranstaltung zu Projektmanagement eingesetzt, um einerseits die Benutzerakzeptanz (s. Kapitel 5) und andererseits den Praxistransfer zu untersuchen. Des Weiteren wurde das *PM Game* zum Assessmentwerkzeug weiterentwickelt (vgl. Stoyan, 2009). Im Assessment-Modus bietet das *PM Game* die Vorteile einer grundlegenden Innovation zur Ergänzung herkömmlicher Verfahren zur Prüfung von Sozialkompetenzen, z.B. Assessment Center (*AC*). *AC* dienen zur Beurteilung von Kompetenz und können aus methodischen Gründen der Normierbarkeit nur innerhalb eines dem Testsetting entsprechenden Rahmens interindividuell unterschiedliche Handlungsstränge abbilden. Vereinfacht ausgedrückt: Jeder Assessee durchläuft einen genau festgelegten Plan an Einzelsituationen. Damit entfällt die Möglichkeit, einen längerfristigen

Prozess bewerten zu können. Ein solcher Prozess erfordert, dass man für Konsequenzen seiner Handlungen einsteht und mit diesen umgeht. Denn in der Realität reicht es nicht, eine schulbuchmässige Lösung in einer schwerwiegenden *Trade-Off*-Situation zu finden, man muss auch mit den Konsequenzen der getroffenen Entscheidung weiterarbeiten können. Durch Computerspiele wird diese Form der Konsequenzverkoppelung ermöglicht. Das *PM Game* richtet sich v.a. an Studenten, die an alternativen Lernmethoden interessiert sind.

4.2 Komponenten des *Evolutionären Lernspiels „PM Game“*

Im diesem Kapitel werden die Zusammenhänge der einzelnen Komponenten des *PM Game* anhand seines Lern-/Spielzyklus⁷ erläutert.

Ablauf

Neue Nutzer registrieren sich über das Webportal des *PM Game*. Sie eröffnen ein Benutzerkonto, welches einerseits ihre Spielstände und andererseits Informationen über ihren Lernfortschritt speichert. Der neu registrierte Nutzer meldet sich zum Spiel an und macht sich mit Hilfe einer Einführungsgeschichte mit dem System und seiner Benutzung vertraut. Beobachtungen und qualitative Befragungen haben gezeigt, dass ein neuer Nutzer die Grundmechanismen des Spiels innerhalb von durchschnittlich 30 Minuten verstehen und in die Lage versetzt werden kann, mit NPC zu interagieren. Falls der Nutzer eine Handlungsalternative im Spiel dennoch nicht umsetzen kann, besteht die Möglichkeit, die gewünschte Handlung in Form eines Freitexts festzuhalten. Dieser Text wird später von Experten analysiert, um bestehende Interaktionsmuster im Spiel zu verbessern, und gegebenenfalls via Regeleditor ins System implementiert.

PM Game-Nutzern stehen verschiedene Geschichten mit Lerninhalten zur Verfügung, welche von Experten und anderen, erfahrenen Nutzern erstellt worden sind. Der Nutzer kann frei entscheiden, welche Geschichte mit welchem Lerninhalt er in welcher Reihenfolge spielen möchte.

Hat der Nutzer einige Erfahrungen im Spiel gesammelt, erhält er die Berechtigung, eigene Inhalte zu generieren. Er muss dafür von einem Gruppenleiter in eine bestehende Gruppe eingeladen werden. Der Nutzer kann den Gruppenleiter auch selbst darauf aufmerksam machen, dass er seiner Gruppe beitreten möchte. Die Aufgaben eines Gruppenleiters sind die Zusammenstellung der Gruppe, die Aufgabenverteilung und die Koordination der einzelnen Gruppenmitglieder zur Erstellung von Szenarien. Die gesamte Administration und Koordination der Gruppe wird über eine rudimentäre Verwaltung innerhalb des Autorenwerkzeugs *PEdi* geregelt. Die Berechtigung zur Leitung von Gruppen wird in der aktuellen Version des *PM Game* noch manuell von den Superadministratoren vergeben. Vorgesehen ist, dass das System die zusätzlichen Berechtigungen zukünftig in Abhängigkeit der Erfahrung des Nutzers freischaltet, welche an so genannten Erfahrungspunkten gemessen wird. Erfahrungspunkte werden aufgrund von erstellten Geschichten und deren jeweiligen Bewertungen durch andere Nutzer und die Superadministratoren berechnet.

Weitere Berechtigungen erlauben das Editieren der so genannten Sprechakte sowie die Generierung von Regeln, welche relevant für das Lernspiel sind. Diese Berechtigungen werden von den Superadministratoren manuell vergeben, da zurzeit noch keine Methode entwickelt wurde, mit der die Qualifikation des Nutzers und die Qualität der von ihm generierten Inhalte automatisch festgestellt werden kann. Sprechakte werden im *PEdi* verwaltet und können mit Hilfe dieses Werkzeugs editiert und erweitert werden. Die Regeln des Lernspiels werden über den *RuEdi* eingespeist und verwaltet.

4.3 Das Spiel

Das Spiel gehört zum *Genre* der *Adventures*. Ein *Adventure* beinhaltet verschiedene, in eine Hintergrundgeschichte verpackte Rätsel, welche vom Spieler zur Zielerreichung gelöst werden müssen. Die im Vordergrund stehende Handlung wird vorangetrieben, indem der Spieler mit anderen Figuren spricht oder Gegenstände findet, welche ihm Aufschluss über die fiktive Welt und mögliche Wege zur Lösung der Rätsel geben.

Das *PM Game* ist eine Mischform aus den beiden *Subgenres* Text- und Graphik-Adventure. Text-Adventure präsentieren das Spielgeschehen in rein textueller Form und verwenden Graphikelemente, wenn überhaupt, ausschliesslich zur Dekoration. Die Kommunikation zwischen Spieler und Spiel findet über ein Texteingabefeld statt. Anweisungen werden in natürlicher Sprache über die Tastatur eingegeben oder aus vorhandenen Textbausteinen zusammengesetzt, der Text-Parser analysiert die Anweisungen und der Interpreter führt sie anschliessend aus. Die Spielfigur wird mit Kommandos wie "Rede mit Kunde" oder "Baue Haus" gesteuert. Graphik-Adventure nutzen, im Gegensatz zu Text-Adventure, graphische Elemente nicht nur zur Dekoration, sondern auch zur Interaktion und Visualisierung von Szenarien, welche implizit Aufschluss über die gestellten Rätsel geben. So muss z.B. das Szenario "Vase auf Tisch stellen" in einem Text-Adventure in Textform ausformuliert werden, wohingegen es in einem Graphik-Adventure visualisiert werden kann.

Zugunsten der Übersichtlichkeit und Bedienerfreundlichkeit werden die beiden vorgestellten Subgenres im *PM Game* kombiniert. Das Szenario wird visualisiert, und für die Interaktion mit den Figuren wird auf die bewährte und weit verbreitete *Drag 'n' Drop*-Methode gesetzt (vgl. e-teaching.org, 2008). Der Spieler interagiert mit den Figuren über so genannte Sprechakte. Darunter werden in der Computerlinguistik sprachliche Äusserungen verstanden, welche nicht nur einen Sachverhalt beschreiben, sondern auch Handlungen, wie z.B. einen Befehl oder ein Feedback (Searle, 1969). Im *PM Game* sind Sprechakte parametrisierte Sätze, die je nach Interaktion des Spielers automatisch vom System vervollständigt werden. Der Spieler klickt dafür mit

der Maus auf einen parametrisierten Satz und kann ihn anschliessend durch Klicken auf die entsprechenden graphischen Repräsentationen vervollständigen (s. Abb. 24).

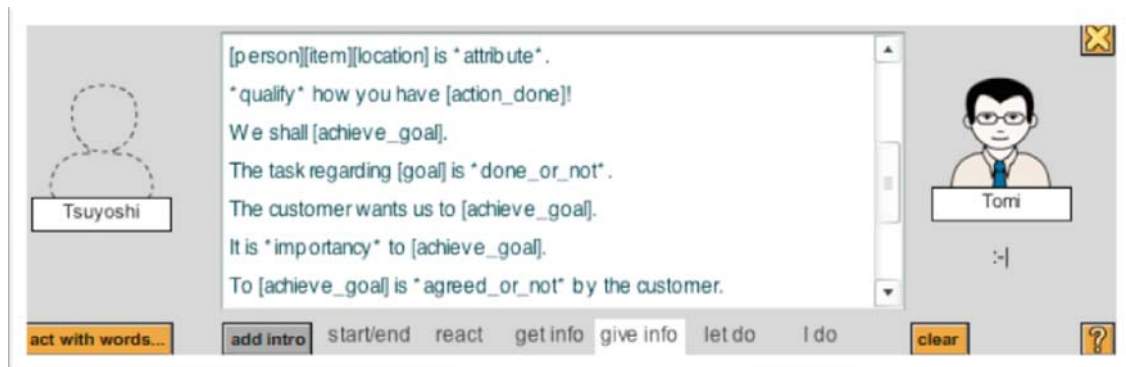


Abb. 24: *PM Game*-Sprechakte

Wie bereits in der Einleitung erwähnt wurde, ist es Ziel des *PM Game*, fachübergreifende Kompetenzen mit Schwerpunkt Projektmanagement zu fördern. D.h. mit Hilfe der Sprechakte werden Aspekte der Koordination und Führung kleiner Gruppen sowie der Kommunikation vermittelt, z.B. Kundengespräche. Es wird gezielt darauf hingewirkt, das Handlungsrepertoire des Lernenden zu erweitern, sein situatives Handeln zu verbessern sowie mögliche langfristige Auswirkungen eigener Handlungen abzuschätzen. Die Art, das Ausmass und der Praxistransfer der erzielten Verbesserungen werden im Rahmen eines Forschungsprojekts von Robert Stoyan untersucht.

Auf den ersten Blick erscheinen Sozialkompetenzen schwer per Computer trainierbar. Massnahmen, wie etwa eine natürlichsprachige Benutzerschnittstelle oder die weitgehende Abbildung sozialer Situationen im Spiel, können die Schwächen des Systems nur kaschieren. Deshalb bietet auch das *PM Game* weniger reichhaltige Interaktionsmöglichkeiten als sie die direkte zwischenmenschliche Kommunikation ermöglichen würde.

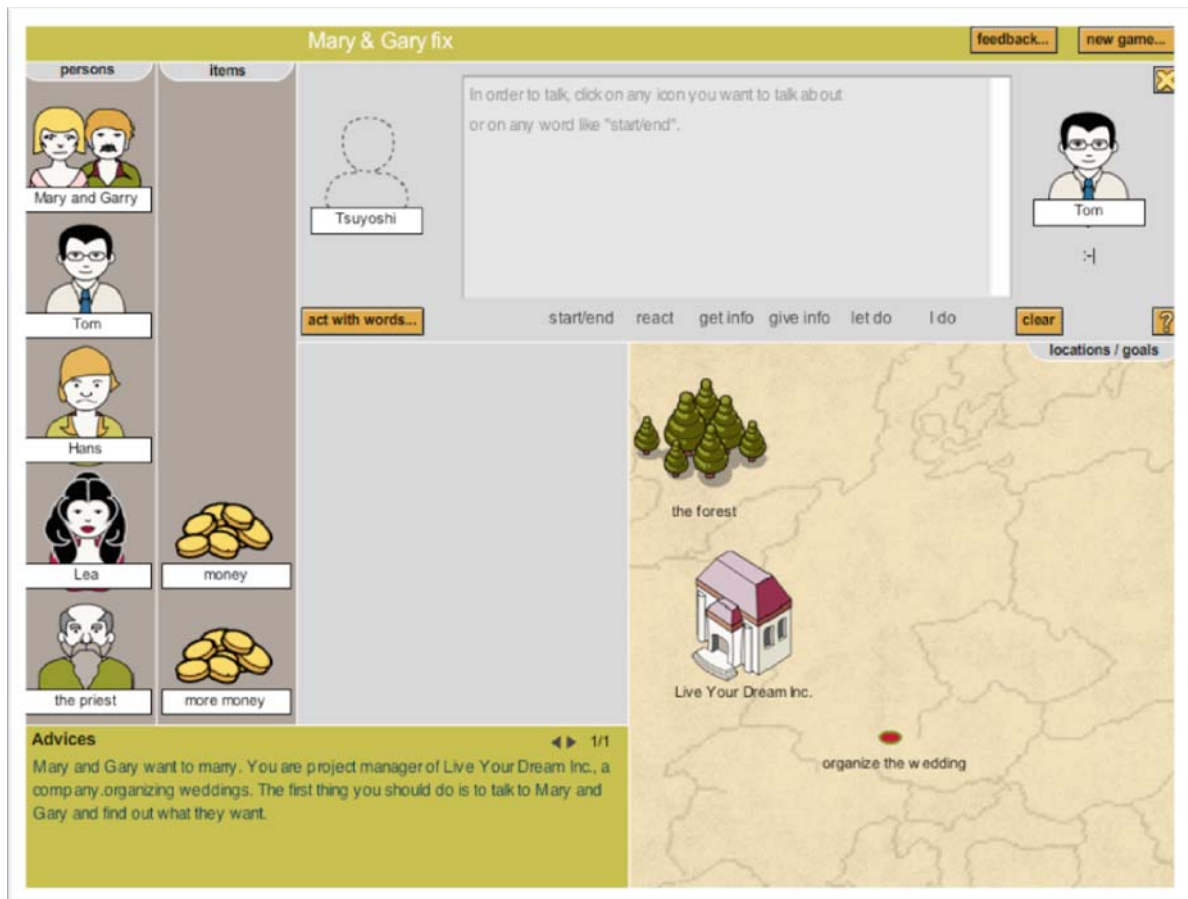


Abb. 25: PM Game-Spieloberfläche

Den wesentlichen Vorteil computerunterstützten Lernens von Sozialkompetenzen bieten Abläufe, welche gegenüber der Realität im Zeitraffer durchgespielt werden können. Lerninhalte, bei denen dieser Vorteil zum Tragen kommt, sind z.B. der Aufbau eines kleinen Teams oder die Gestaltung von Arbeits- und Kundenbeziehungen. Sicher kann man eine Lehrveranstaltung konzipieren, in der beispielsweise die fünf Phasen der Teamarbeit nach Tuckman vertieft oder die Theorie praktisch in Teamarbeit erlebt werden kann, jedoch dauert es jeweils ein Semester bis solch ein Prozess abläuft (Stoyan, 2009). Alternativ könnte der Teambildungsprozess auch in Einzelsituationen zerlegt und nachgestellt werden. Dabei wird jedoch unter anderem der Zusammenhang der Phasen nicht erlebt. In digitalen Lernspielen können derartige Prozesse jedoch ohne Weiteres mehrfach innerhalb eines einzigen Tages durchgespielt werden. So gibt das *PM Game* Lernenden die Möglichkeit, ihre theoretischen Kenntnisse binnen Stunden wiederholt einzusetzen bis der Lerninhalt soweit verinnerlicht ist, dass verschiedene Situationen mit unvorhergesehenen Ereignissen ausgereift gemeistert werden können.

In diesem Abschnitt soll nun ein für den Nutzer des *PM Game* typischer Anwendungsfall festgehalten werden. Die Interaktionsfolge stellt sich folgendermassen dar: Nachdem ein Szenario gewählt und das Ziel verdeutlicht wurde, wählt der Spieler einen *NPC* aus, mit dem er interagieren möchte. Anschliessend initiiert er eine Kommunikation mit dem gewählten *NPC*. Dafür stehen ihm sechs Kategorien verschiedener Sprechakte zur Verfügung, welche die Grundhandlungsmuster des gesamten *PM Game* repräsentieren:

- *start/end conversation*
- *react*
- *give information*
- *get information*
- *let do*
- *do yourself*

Hat der Spieler per *Drag-n-Drop* einen Sprechakt formuliert und ausgelöst, reagiert der *NPC*, gefolgt von eventuellen Änderungen des Szenarios. Die Reaktionen des *NPC* werden aufgezeichnet und können Auswirkungen auf den Fortgang des Szenarios sowie zukünftige Interaktionen haben. Der Spieler erhält – ebenfalls via Sprechakt – Informationen vom *NPC*, die ihm helfen, herauszufinden, wie die Ziele erreicht resp. die Rätsel gelöst werden können. Wird ein (Teil-)Ziel erreicht, führt das zur automatischen Aktualisierung der Elemente im Szenario, und neue Ziele resp. Aufgaben werden sichtbar.

Neben dem Lösen der Rätsel resp. Bearbeiten der Lerninhalte werden dem Nutzer diverse Feedbackmöglichkeiten angeboten, welche die Qualität der Geschichten, Lerninhalte und Benutzerfreundlichkeit verbessern. Diese Feedbackarten sind in der folgenden Liste aufgezählt:

1. Qualitatives Feedback zur Geschichte, Spiellogik und zu Projektmanagement-Handlungen
2. Berichterstattung zu Fehlern in der Software
3. Bewertung der Geschichte anhand einer fünfstufigen Skala
4. Möglichkeit, das beabsichtigte weitere Vorgehen in natürlicher Sprache festzuhalten, falls keine passende Handlungsalternative im Spiel gefunden wird

Die Bewertungen und Feedbacks werden gespeichert und den zuständigen Personen zugestellt. Diese Bewertungs- und Feedbackfunktionalitäten sind ein wichtiger Bestandteil des Reputationssystems und fördern die Motivation der Nutzer.

4.4 NPC-Editor

Zur Visualisierung der *NPC* für die benutzergenerierten Geschichten des *PM Game* wurde ein Konfigurator entwickelt. Dieser soll einerseits die Persönlichkeitstypen visualisieren und andererseits Aufschluss über den aktuellen emotionalen Zustand der *NPC* geben. Solche Personalisierungsmöglichkeiten gelten als ein wichtiger Bestandteil zur Förderung der Kreativität und Motivation der Spieler (Resnick, 2007; Koster & Wright, 2005).

Anordnung und Kategorisierung der graphischen Elemente lassen sich mit Hilfe des *NPC-Editors* je nach Bedarf anpassen (Abb. 26). So können anstatt graphischer menschlicher Körperteile und Accessoires Teile von Tieren hochgeladen und zusammengestellt werden. Bei seiner Entwicklung wurde darauf geachtet, dass der *NPC-Editor* universell auch für andere Spiele verwendbar ist.

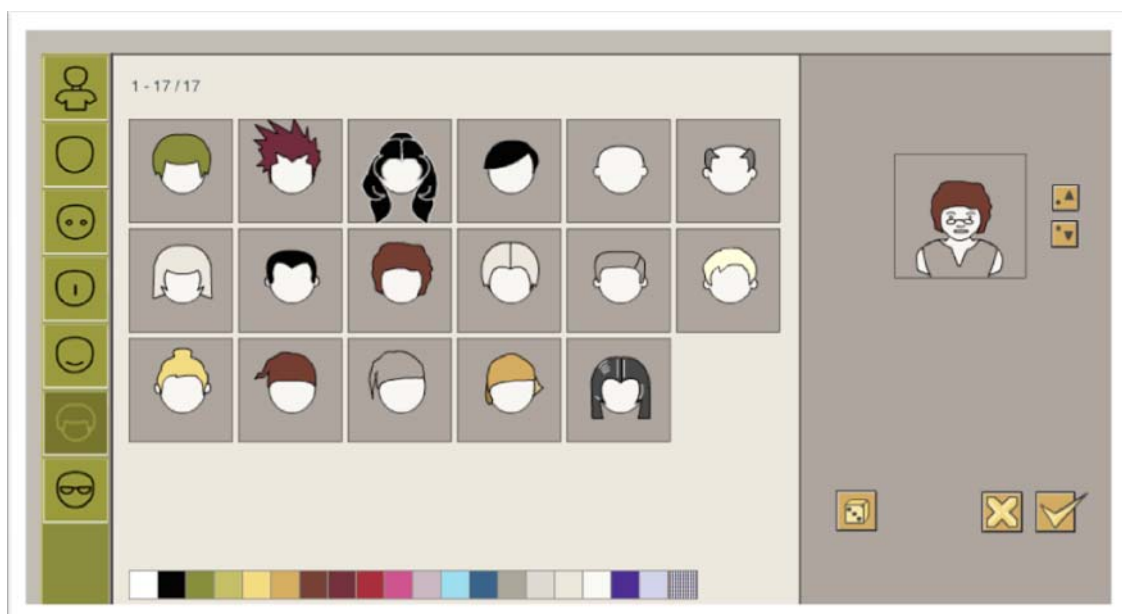


Abb. 26: *NPC-Editor*

4.5 Autorenwerkzeug *PEdi*

Das Autorenwerkzeug *PEdi* erlaubt es den Nutzern, kollaborativ Szenarien für das *PM Game* zu erstellen und sie anschliessend anderen Nutzern zur Verfügung zu stellen. *PEdi* ist neben dem Spiel der wichtigste Bestandteil für die Realisierung des *Evolutionären Lernspielkonzepts*. Durch das Einbeziehen der Nutzer in den Erstellungsprozess sollen kreatives Denken und Lernen durch Synthese unterstützt und so Kompetenzen, wie Koordinations-, Kommunikations- und Delegationsfähigkeit, gefördert werden. Zusätzlich ermöglicht das Konzept der benutzergenerierten Inhalte in Lernspielen niedrigere Entwicklungskosten, da die aufwendige Szenarienerstellung den Nutzern überlassen wird. Als Nebeneffekt dieses Konzepts entsteht durch die Kreativität der Nutzer emergente Unterhaltung. Des Weiteren soll die Qualität der Inhalte durch implizite und explizite Massnahmen einer wachsenden virtuellen Gemeinschaft gesichert und erhöht werden.

Zur Förderung der Inhaltsgenerierung wurde in das *PM Game* ein Reputationssystem integriert: Die Nutzer haben die Möglichkeit, sich sowohl im Kontext des Lernspiels (per Wiki, Forum etc.) wie auch im Spiel selbst über spezielle Bewertungsfunktionen gegenseitig Feedback zu geben, Wissen auszutauschen und auf qualitativ hochstehende Inhalte aufmerksam zu machen. Je besser ein Lernszenario bewertet wird, desto besser wird der Ruf des Autors und desto stärker steigt die Wahrscheinlichkeit, dass es von weiteren Nutzern gespielt wird. Dieses Reputationssystem basiert auf den gleichen Mechanismen wie z.B. *Wikipedia*, *Ebay*TM, Amazon, *StudiVZ* oder andere *social software*.

Die Inhaltsgenerierung durch die Nutzer ist jedoch mit dem Risiko qualitativ minderwertiger Inhalte verbunden (s. Kapitel 2.1.2). Deshalb wurden für das *PM Game* verschiedene Rollen und Rechte zur Qualitätssicherung der Inhalte eingeführt. Das Berechtigungssystem basiert auf so genannten Erfahrungspunkten: Über je mehr Erfahrung mit dem *PM Game* ein Nutzer verfügt, desto mehr Rechte werden ihm zugestanden. Mit diesem System soll eine Überforderung unerfahrener Nutzer infolge zu komplexer Aufgaben verhindert werden, weil diese dadurch wahrscheinlich frustriert würden. Stattdessen ist es Ziel des Rechtesystems, die Nutzer Schritt für

Schritt an die verschiedenen Aufgaben heranzuführen und ihnen so stets ideale Herausforderungen zu schaffen. Abbildung 27 zeigt die verschiedenen Rollen des *PM Game* mit den jeweiligen Berechtigungen, welche inkrementell vom Spieler zum Superadministrator erweitert werden.



Abb. 27: Rollen und Rechte im *PM Game*

Im Rahmen des *PM Game* werden grundsätzlich drei Stufen generierbarer Inhalte unterschieden:

1. Szenarien (z.B. Geschäftswelt, Gerichtssaal, Märchenwald usw.),
2. Sprechakte, welche im *PEdi* bearbeitet werden können und
3. Verhaltensregeln (s. Kapitel *RuEdi* 4.6).

Im Folgenden wird nun auf den Aufbau der Szenarien und Sprechakte eingegangen und diese anschliessend in Zusammenhang mit den Pflichten und Rechten der verschiedenen Rollen in Bezug zur Inhaltsgenerierung gebracht. Die Verhaltensregeln werden in Kapitel 4.6 diskutiert.

Szenarien bestehen aus den Spielobjekten *NPC*, Gegenstände, Standorte und Ziele. Diese wiederum bestehen aus editierbaren Eigenschaften (*Attributes*) und Aktionen. Letztere können den Spielobjekten über Sprechakte zur Ausführung delegiert werden, um andere Objekte zu verändern. Ziele (*Goals*) sind zudem um die Eigenschaft einer bedingten Aktion (*Conditional Action*) erweitert worden, welche bestimmt, ob ein Ziel erreicht wurde oder nicht. In Abbildung 28 wird der Zusammenhang zwischen den Objekten in einem Klassendiagramm dargestellt.

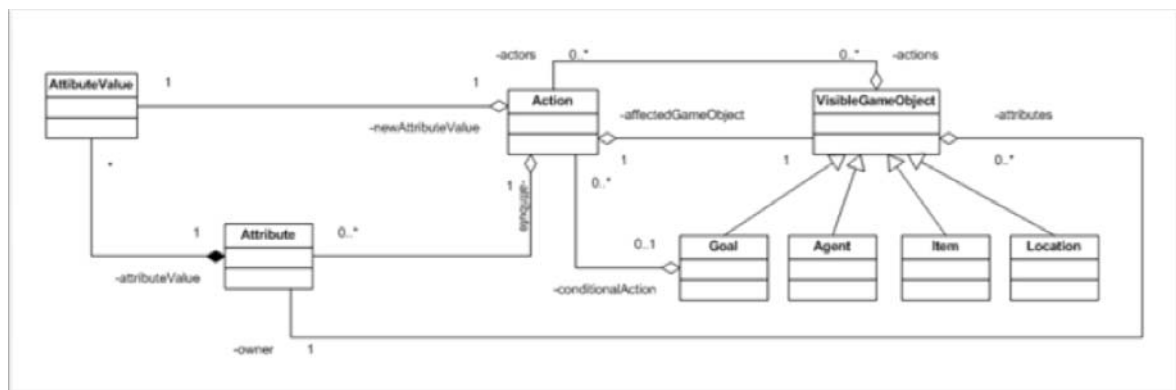


Abb. 28: Klassendiagramm der Spielobjekte

Alle Ereignisse werden in einem Log gespeichert und können spätere Aktionen und Handlungen beeinflussen. Somit besitzen die *NPC* eine Art Gedächtnis, das es ihnen ermöglicht, situativ und in Bezug auf vergangene Ereignisse reagieren zu können.

Des Weiteren gibt es so genannte Informationsobjekte (*InfoObject*), welche durch einen Auslöser (*Trigger*) in einem Sprechakt „gefeuert“ werden können. Sie beinhalten Informationen über die im Szenario vorhandenen Spielobjekte und können ebenfalls als Feedback auf getätigte Aktionen des Spielers eingesetzt werden (Abb. 29).

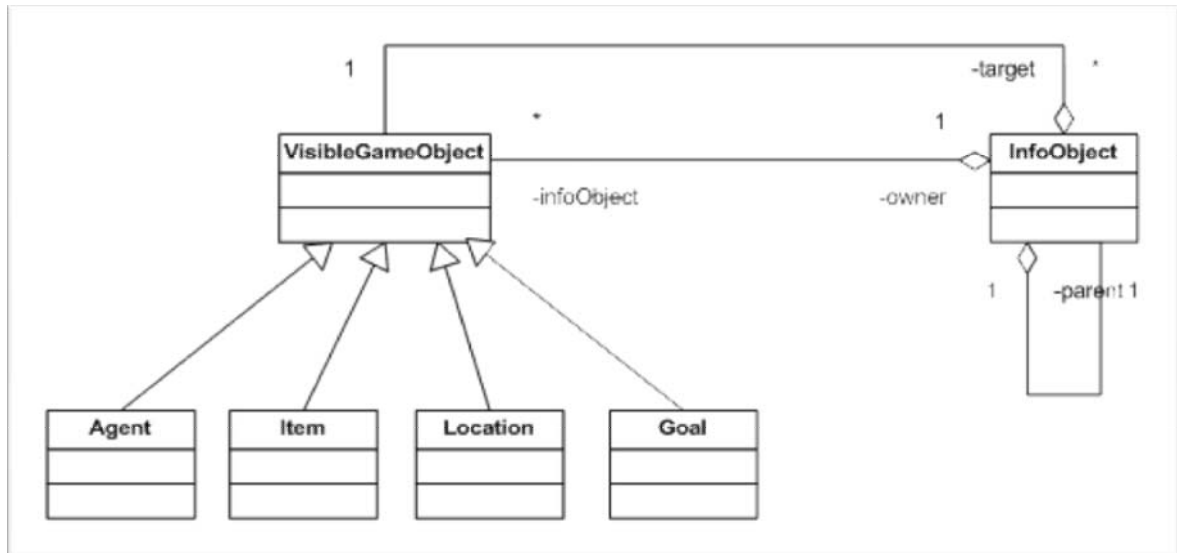


Abb. 29: Klassendiagramm Informationsobjekte

Die Spielobjekte und Informationsobjekte werden im *PEdi* bearbeitet. Tabelle 14 beschreibt die verschiedenen verfügbaren Objekte.

Tab. 14: Spielobjekte des *PM Game*

Objekt	Beschreibung
NPC (<i>Agent</i>)	<i>NPC</i> sind die Hauptinteraktionselemente im <i>PM Game</i> . Der Spieler befragt sie oder delegiert ihnen Aufgaben. <i>NPC</i> besitzen individuelle, veränderliche Eigenschaften, welche ihren Zustand definieren. Mit Hilfe dieser Attribute werden die Persönlichkeiten der <i>NPC</i> realisiert.
Gegenstand (<i>Item</i>)	Mit Hilfe der Gegenstände werden Rätsel gelöst. Sie bestehen wie die <i>NPC</i> aus Eigenschaften, welche der Nutzer im <i>PEdi</i> konfiguriert.
Standorte (<i>Location</i>)	Standorte sind Orte, an welchen <i>NPC</i> und Gegenstände platziert werden. Sie werden ebenso wie die Ziele auf einer Karte eingetragen. Die Standorte dienen dem Nutzer für einen besseren Überblick.
Ziele (<i>Goal</i>)	Ziele werden über Zustände von bestimmten Elementen definiert. Durch (Inter-)Aktionen des Spielers werden diese Zustände erreicht und die Ziele erfüllt. Ziele können miteinander verknüpft und so komplexe Aufgabenstellungen generiert werden.

Informationsobjekte (InfoObject)	Informationsobjekte sind Objekte, welche Erläuterungen zu bestimmten Elementen im Szenario bereitstellen und durch das Erreichen von Zielen oder Befragungen der <i>NPC</i> ausgelöst werden. Dadurch werden bisher verborgene, dem Spieler somit unbekannte Elemente sichtbar und können für weitere Abklärungen miteinbezogen werden. Informationsobjekte werden ebenfalls im <i>PEdi</i> generiert und konfiguriert.
-------------------------------------	---

Mit Hilfe des *PEdi* werden, neben der Erstellung der Szenarien, die Sprechakte editiert. Wie in Kapitel 3.1 erläutert wurde, sind Sprechakte Äusserungen der Spieler und der *NPC*, die sich aus freiem Text und Parametern zusammensetzen. Der Administrator der Sprechakte kann Textbausteine im *PEdi* vorfertigen und diese mit Parametern anreichern, welche der Spieler zur Laufzeit des Spiels ergänzt. Durch die Parametrisierung der Sprechakte können sich die Äusserungen an die jeweilige Situation des Szenarios anpassen.

Abbildung 30 zeigt die Pflichten und Rechte der einzelnen Rollen im Erstellungsprozess. Die Zuteilung komplementärer Rechte an die Gruppenmitglieder durch den Gruppenleiter erlaubt ein asynchrones Erstellen der Inhalte. Der Gruppenleiter kann z.B. einem Mitglied die Rechte zur Erstellung von Zielen und einem anderen die Rechte zur Erstellung der *NPC* und Gegenstände erteilen. In diesem Fall können die beiden Mitglieder ihre Aufgabe asynchron bearbeiten, ohne sich gegenseitig zu behindern oder die Arbeiten des anderen zu überschreiben resp. zu löschen.

Der Erstellungsprozess eines Szenarios kann in drei Phasen unterteilt werden: In der ersten Erstellungsphase ist das Lernszenario im Status "**Unter Bearbeitung**". Sobald der Gruppenleiter das Szenario zur Prüfung durch den Superadministrator (Super Admin/Experte) freigibt, wechselt es in den Status "**Prüfung**". Nach erfolgter Prüfung publiziert der Superadministrator das Szenario (Status "**Publiziert**") oder gibt es der Gruppe zur Überarbeitung zurück (Status "Unter Bearbeitung"). Ein publiziertes Szenario kann von den Spielern bewertet und kommentiert werden. Die Verbesserungsvorschläge kann der Gruppenleiter berücksichtigen, wozu er das Szenario wieder in den Status "Unter Bearbeitung" setzen muss.



Abb. 30: Prozessschema der Szenariengenerierung im Rahmen des PM Game

4.6 Regeleditor *RuEdi*

Im *PM Game* können Regeln editiert werden, die auf der grundlegenden Spiellogik aufbauen, d.h. die Reaktionen der Elemente einer simulierten Welt auf die Aktionen des Spielers können vom Regeladministrator ohne Programmierkenntnisse kreiert und modifiziert werden. In den meisten digitalen Spielen wird die Spiellogik von den Spieldesignern spezifiziert und anschliessend implementiert. Dies hat zur Folge, dass die Weiterentwicklung oder Anpassung der Spiellogik von Programmierern vollzogen werden muss, was ein Kompilieren resp. *Patchen* der Software zur Folge hat. Dies ist für die meisten digitalen Unterhaltungsspiele ausreichend, doch das *PM Game* braucht für die stetige Weiterentwicklung der Handlungsmuster der NPC eine Schnittstelle zur Erweiterung der Spiellogik, welche Projektmanager (Regeladministratoren) bedienen können und die deren Resultate dynamisch (ohne Kompilieren) in das *PM Game* einspeist. Dies wird etwa zur Umsetzung adäquater Reaktionen der NPC auf die Aktionen kulturell unterschiedlicher Spieler oder in multikulturellen Anwendungskontexten nötig. Ausserdem nutzt das *PM Game* den Regeleditor, um erfahrenen Nutzern die Gelegenheit zu geben, gelernte theoretische Grundlagen mit Hilfe des Autorenwerkzeugs zu implementieren.

Eine Regel besteht aus einer Bedingung, welche die Regel aktiviert, einer Aktion / Reaktion, falls eine gewisse Situation eingetroffen ist und einer Bedingung, welche die Regel deaktiviert. Folgendes Beispiel soll dies illustrieren:

Bedingung für die Aktivierung der Regel: Wenn ein Dialog zwischen Spieler und einem NPC initiiert wird, dann wird die Regel aktiviert. (1)

Aktion: Grüsst der Spieler den NPC, dann erwidert der NPC den Gruss. (2)

Bedingung für die Deaktivierung der Regel: Wenn der Spieler dem NPC eine Aufgabe delegiert, ohne ihn zuvor zu grüssen, dann wird die Regel deaktiviert, d.h. der NPC grüsst den Spieler nicht. (3)

Um das Zusammenspiel der Regeln und der simulierten Welt zu verstehen, wird zuerst auf den Spielmechanismus eingegangen. Im *PM Game* besitzen alle Elemente der simulierten Welt interne Zustände, welche durch die Interaktion des Spielers mit dem System verändert werden können (Zustandsautomatens). Diese Zustands-

änderungen werden durch Ereignisse im *PM Game* ausgelöst. Ein Ereignis ist z.B., dass der Spieler einen *NPC* grüsst oder ein *NPC* ein Haus baut. Alle diese Ereignisse werden gespeichert, sodass erstellte Regeln auf vergangene Ereignisse zugreifen können. Tritt nun im Spiel ein Ereignis ein, werden die entsprechenden Regeln abgearbeitet und das Ereignis dahingehend analysiert, ob eine Aktivierungsbedingung einer Regel erfüllt ist (1). Ist mindestens eine Bedingung erfüllt, wird die Regel aktiviert und gegebenenfalls ausgeführt (2), was im Erfolgsfall zu einer Zustandsänderung der simulierten Welt führt. Tritt jedoch zusätzlich ein Ereignis ein, welches in der Deaktivierungsbedingung dieser Regel festgehalten ist, dann wird die aktive Regel deaktiviert (3).

Im *PM Game* existieren drei verschiedene Arten von Regeln:

1. Situative Regeln sind Kontext affine Regeln, welche nur unter ganz bestimmten Bedingungen gefeuert werden.
2. Repertoire-Regeln sind allgemeine Handlungsmuster, welche in verschiedenen Szenarien Anwendung finden.
3. Sprechakt-Regeln sind allgemeine Antworten oder Fragen in einer Dialogsequenz.

Als erstes werden die situativen Regeln vom System prozessiert. Wird davon mindestens eine erfüllt und eine Reaktion ausgelöst, werden die Repertoire-Regeln und die Sprechakt-Regeln inhibiert. Bei Nichterfüllung mindestens einer situativen Regel werden die Repertoire-Regeln durchgegangen und allenfalls eine Reaktion ausgelöst. Wird auch keine Repertoire-Regel erfüllt und keine Reaktion ausgelöst, werden die Sprechakt-Regeln analog zum vorhergehenden Prozess durchgegangen. Dieses System orientiert sich an der *Subsumption*-Architektur von Rodney Brooks, welche für die Navigation von Braitenbergvehikeln benutzt wird (Brooks, 1986; Pfeifer & Scheier, 2001). In Abbildung 31 wird der beschriebene Prozess anhand einer situativen Regel, einer Repertoire-Regel und einer Sprechakt-Regel verdeutlicht.

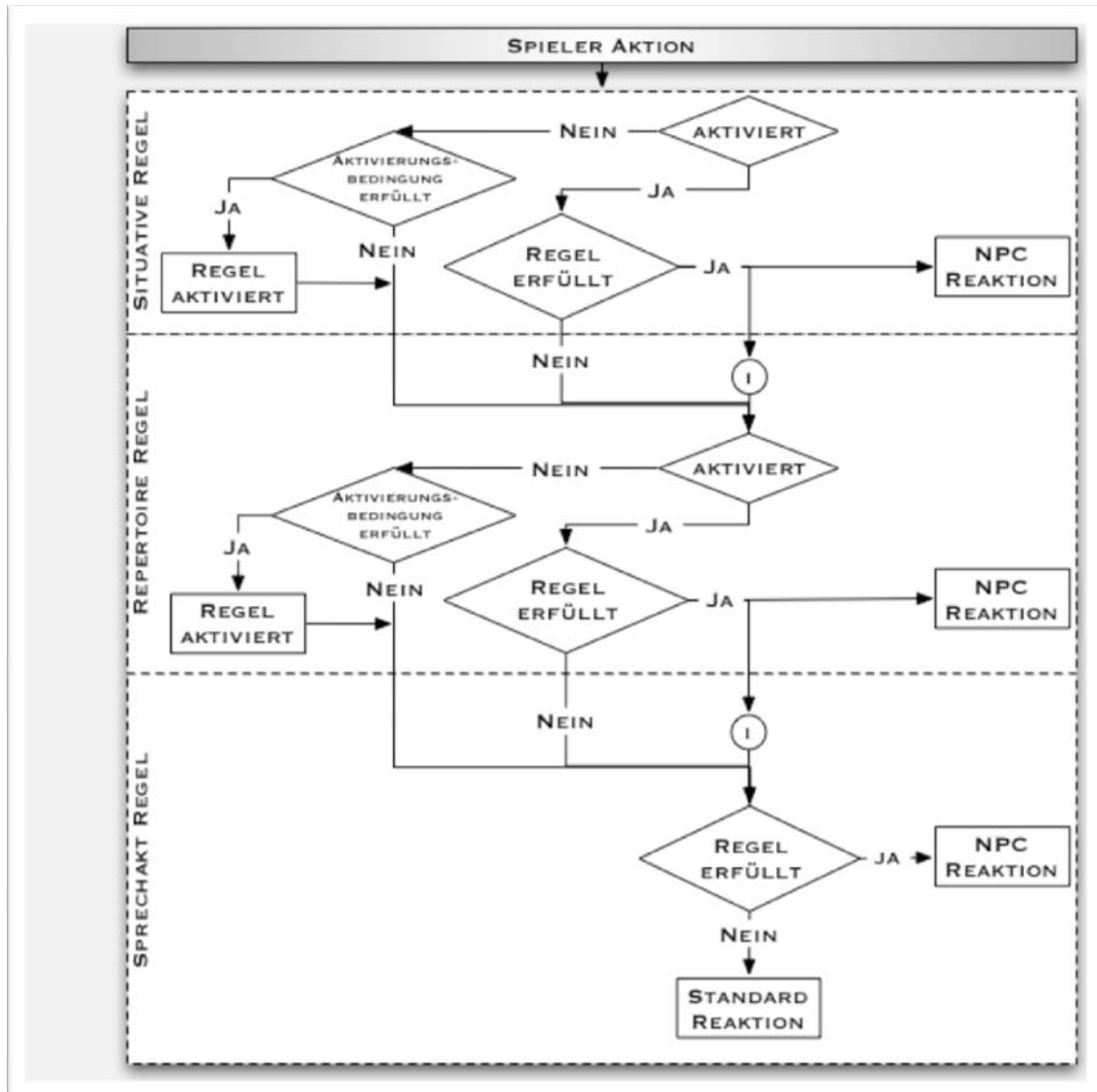


Abb. 31: Subsumption-Architektur der PM Game-Regeln

Die Entscheidung, ob eine situative Regel oder eine Repertoire-Regel erfüllt ist, hängt auch davon ab, ob eine Gelegenheit für eine solche gegeben ist oder schon verpasst wurde. D.h. eine aktive Regel wird auch dann deaktiviert, wenn eine Gelegenheit zur Erfüllung der Regel ungenutzt verstrichen ist (Burkhard, 2008).

In den folgenden Abschnitten wird auf die Struktur und den Aufbau der Regeln und deren Komposition eingegangen. Die Regeln, und somit der *RuEdi*, basieren auf der Prädikatenlogik erster Stufe und beschränken sich auf Allquantoren und Existenzquantoren (vgl. Barwise & Etchemendy, 2005). Im weiteren Verlauf dieses Abschnitts werden die Komponenten einer Regel aus technischer Sicht aufgelistet und erläutert.

Die dazu benötigten Grundlagen wurden von Burkhard (2008) im Rahmen einer vom Autor betreuten Diplomarbeit erarbeitet und in den *RuEdi* implementiert.

Kürzel

```
<Token:<data>,<label>>
```

Ein Kürzel (*Token*) ist ein Beschriftungs-Daten-Paar und stellt die kleinste Einheit in einer Regel dar. Das Datenobjekt (*data*) ist der maschinenlesbare Name und repräsentiert eine Funktion oder ein Attribut eines Kontextobjekts. Des Weiteren kann der Inhalt des Datenobjekts einen statischen Wert beinhalten. Die Beschriftung (*label*) ist die menschenlesbare Beschreibung der Funktionalität.

Term

```
<Term:[<Token>,<Token>,<Token>]>
```

Ein Term ist eine geordnete Aufzählung von Kürzeln. Für jedes Kürzel wird die dazugehörige Funktion des Kontextobjekts gesucht, welche als Wert des vorangegangenen Kürzels zurückgegeben wird. Ein Term hat immer einen Rückgabewert, welcher identisch mit dem Rückgabewert des letzten Kürzels ist.

Teilbedingung

```
ConditionTerm: Operator ( <Term>, <Term> ):Boolean  
<ConditionTerm:  
  <Operator:is_equal>,  
  <Term:[<Token:npc>,<Token:getMood>,<Token:getValue>]>,  
  <Term:[<Token:good>]>>
```

Eine Teilbedingung (*ConditionTerm*) besteht aus zwei Termen und einem Operator. Das Ergebnis dieser Operation ist ein BOOLE'scher Wert.

Bedingung

```
<Condition:  
  <Boolean:AND>,  
  [<ConditionTerm>,<ConditionTerm>]>
```

Eine Bedingung (*Condition*) enthält eine oder mehrere Teilbedingungen, welche mit einer BOOLE'schen Operation verknüpft werden.

Aktion

```
<Action:  
  [<Term>,<Term>]>
```

Eine Aktion (*Action*) löst einen Zustandswechsel eines Elements der simulierten, virtuellen Welt aus und entspricht einer Reaktion des Systems. Ein allfälliger Rückgabewert wird ignoriert. Eine Aktion besteht aus einem oder mehreren Termen.

Block

```
<Block:  
  <Condition>  
  <Action>>
```

Ein Block (*Block*) setzt sich aus mindestens einer Bedingung und einer Aktion zusammen. Die Aktion wird ausgeführt, falls die Bedingung erfüllt ist.

Ablauf

<Workflow:
 [<Block >, <Block>]>

Mit Hilfe des Ablaufs (*Workflow*) können komplizierte Regeln definiert werden. Sie werden aus mehreren Blöcken gebildet.

Rule Scheme Definition

Die Terme, welche ein Nutzer zur Erstellung einer Regel verwenden darf, sind in einer gewurzelten Baumstruktur (*Out Tree*) gespeichert, um einen schnellen Zugriff auf die wichtigsten Elemente (Knoten) der simulierten Welt und deren Attribute zu gewährleisten (Burkhard, 2008). Des Weiteren wird anhand dieser Baumstruktur festgelegt, welche Funktionen und Attribute der Knoten vom System aufgerufen werden können und wie die Knoten resp. Elemente miteinander zusammenhängen. Neben den Elementen werden noch Parameter gespeichert, welche Argumente von Funktionen der jeweiligen Elemente definieren.

Ein Knoten hat die folgenden, in Tabelle 15 aufgeführten Attribute:

Tab. 15: Beschreibung der Attribute von Regeln

Name	Beschreibung
label	Beschriftung des Kürzels (Token)
data	Name der Funktion oder des Attributs
children_key	Identifikation des Knotens, von welchem die Daten geladen werden sollen
type	Typ des Inhalts (z.B. freier Text)

Die erstellten Regeln werden in XML-Format gespeichert und während der Initialisierung des Spiels geladen. Die Analyse der Regeln übernimmt der *Dispatcher*, welcher ebenfalls das Auffinden und Ausführen der Funktionen oder Attribute der jeweiligen Elemente (*context object*) übernimmt. Der Ablauf bis zur Ausführung einer Funktion kann wie folgt beschrieben werden: Der *Dispatcher* erkennt eine Funktion anhand des Kürzels und der Position des Kürzels im Term und ist dadurch in der

Lage, die Funktion zu identifizieren und auszuführen. Ein Kürzel T_i repräsentiert eine Funktion F_i eines Elements, welches seinerseits der Rückgabewert der Funktion $F(i-1)$ ist. Die Funktion $F(i-1)$ wiederum ist abgeleitet vom Kürzel $T(i-1)$, d.h. ein Objekt, welches als Rückgabewert einer Funktion zurückgegeben wurde, agiert als *context object* für das nächste Kürzel (Burkhard, 2008).

4.7 PM Game-Viewer

Im *PM Game* werden alle Interaktionen in Form von Ereignissen aufgezeichnet und persistent gespeichert. Dadurch können die Sitzungen der Spieler (absolvierte Szenarien) nachträglich abgespielt und analysiert werden. Das Abspielen der aufgezeichneten Sitzungen erfolgt über den *PM Game-Viewer*. Das ermöglicht es einerseits Experten, die Sitzungen zu evaluieren und andererseits dem Spieler, seine Entscheidungen nochmals anzuschauen und sein eigenes Verhalten zu reflektieren. Somit kann der *PM Game-Viewer* sowohl für autodidaktische Lerneinheiten wie auch im Assessment-Modus (summatives Feedback) eingesetzt werden.

4.8 Verwendete Technologien und Frameworks

Aufgrund der Anforderungen, das *PM Game* auf jedem Computer mit Internetzugang ohne die Installation zusätzlicher Software verwenden zu können, kamen nur die *Java*TM-Technologie, *Javascript*TM und *Adobe*[®] *Flash*TM in Frage. Da sich *Adobe*[®] *Flash*TM ausgezeichnet zur Präsentation multimedialer Inhalte eignet und die Verbreitung des *Flash Players*TM 8 in Europa über 93% beträgt (Stand: Dezember 2006), wurde für die Implementierung der graphischen Benutzerschnittstelle und der Spiellogik des *PM Game* *Adobe*[®] *Flash*TM verwendet. Ein weiterer Vorteil dieser Technologie ist die Delegation von rechenintensiven Aufgaben an die Clients und die Kommunikation mit dem Server via *Webservices*. Für die persistente Speicherung der Daten wurde das Relationale Datenbankmanagementsystem *MS SQL Server 2000* eingesetzt. Die Verbindung zwischen *Flash*TM und *MS SQL Server 2000* wird über *Web-*

services durch *Adobe® Coldfusion™* realisiert. Des Weiteren können zeitkritische, leistungsintensive Aufgaben an *Coldfusion™* und *Java™* übergeben werden.

4.9 Software-Architektur des *PM Game*

Das *PM Game* setzt auf das *pureMVC*-Framework (Abb. 32; vgl. *pureMVC*, 2008), welches sich auf die Entwurfsmuster der *Gang of Four* (vgl. Gamma, 1995) und auf das *MVC*-Architekturmuster (s. Kapitel 3.4) stützt und Hilfestellungen zur Entwicklung von leistungsstarken *Rich Internet*-Applikationen in Teams bietet. Dieses Framework setzt sich die konsequente Modularisierung der Software in Modell, Repräsentation und Steuerung zum Ziel und gibt somit dem Entwicklerteam die Möglichkeit, rasch und konsistent zu entwickeln. Ausserdem wird das parallele Entwickeln in Teams durch die Aufteilung funktional unterschiedlicher Einheiten vereinfacht oder sogar erst vollumfänglich ermöglicht.

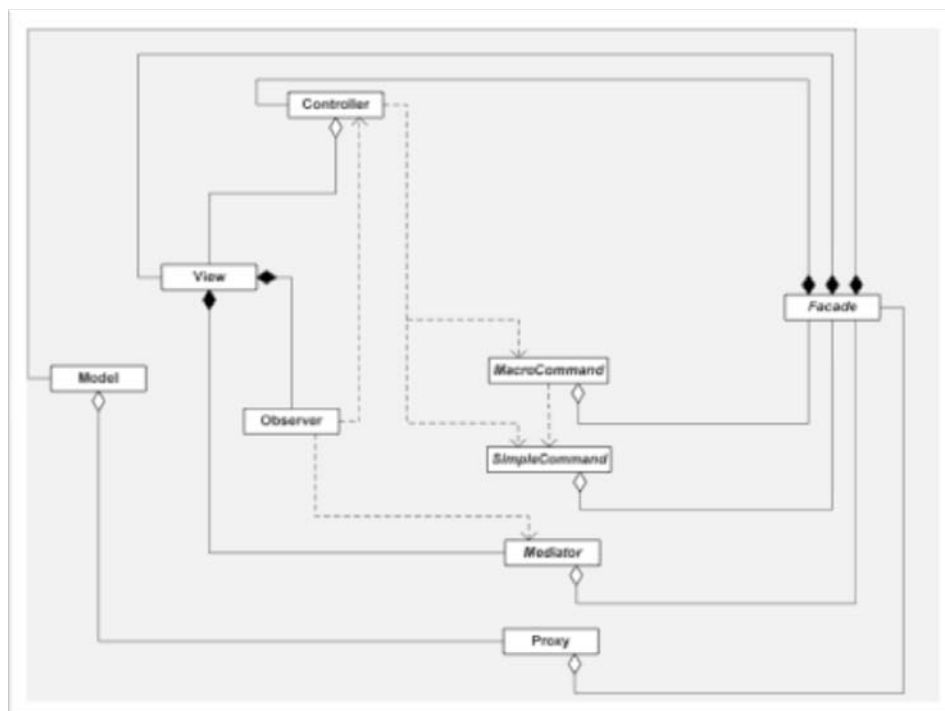


Abb. 32: *pureMVC*-Architekturübersicht

Realisiert wird das *MVC*-Architekturmuster durch die Implementation des *Singleton*-Entwurfsmusters für die Hauptakteure Modell, Repräsentation, Steuerung und die

Facade, sowie die Entwurfsmuster Mediator, Vermittler, Befehl und Beobachter (s. Kapitel 3.4). Die *Facade* ist eine Schnittstelle zur Vereinfachung der Kommunikation zwischen den Hauptakteuren. Sie initialisiert letztere und stellt alle öffentlichen Methoden der Hauptakteure zur Verfügung. Die Vermittler sind referenziert im Modell und speichern die Daten, welche via *Webservices* von der Datenbank abgefragt werden. Die Mediatoren sind das Bindeglied zwischen der Repräsentation und der graphischen Benutzerschnittstelle. Sie verwalten letztere, senden und empfangen Nachrichten für die graphischen Elemente und beeinflussen so deren Zustand. Kommandos werden von der Steuerung instanziiert, sind zustandslos, kommunizieren mit den Vermittlern und können Nachrichten an weitere Kommandos auslösen (pureMVC, 2008). Sie beinhalten die eigentliche Spiellogik. Zustandsänderungen des Systems werden asynchron durch Nachrichten ausgelöst und den jeweiligen graphischen Elementen von den Mediatoren via Ereignissen übergeben. Abbildung 33 gibt einen Überblick über das Zusammenspiel der einzelnen Module im *PM Game*. Ein Beispiel des Zusammenspiels dieser Muster wurde bereits in Kapitel 3.4 vorgestellt und diskutiert.

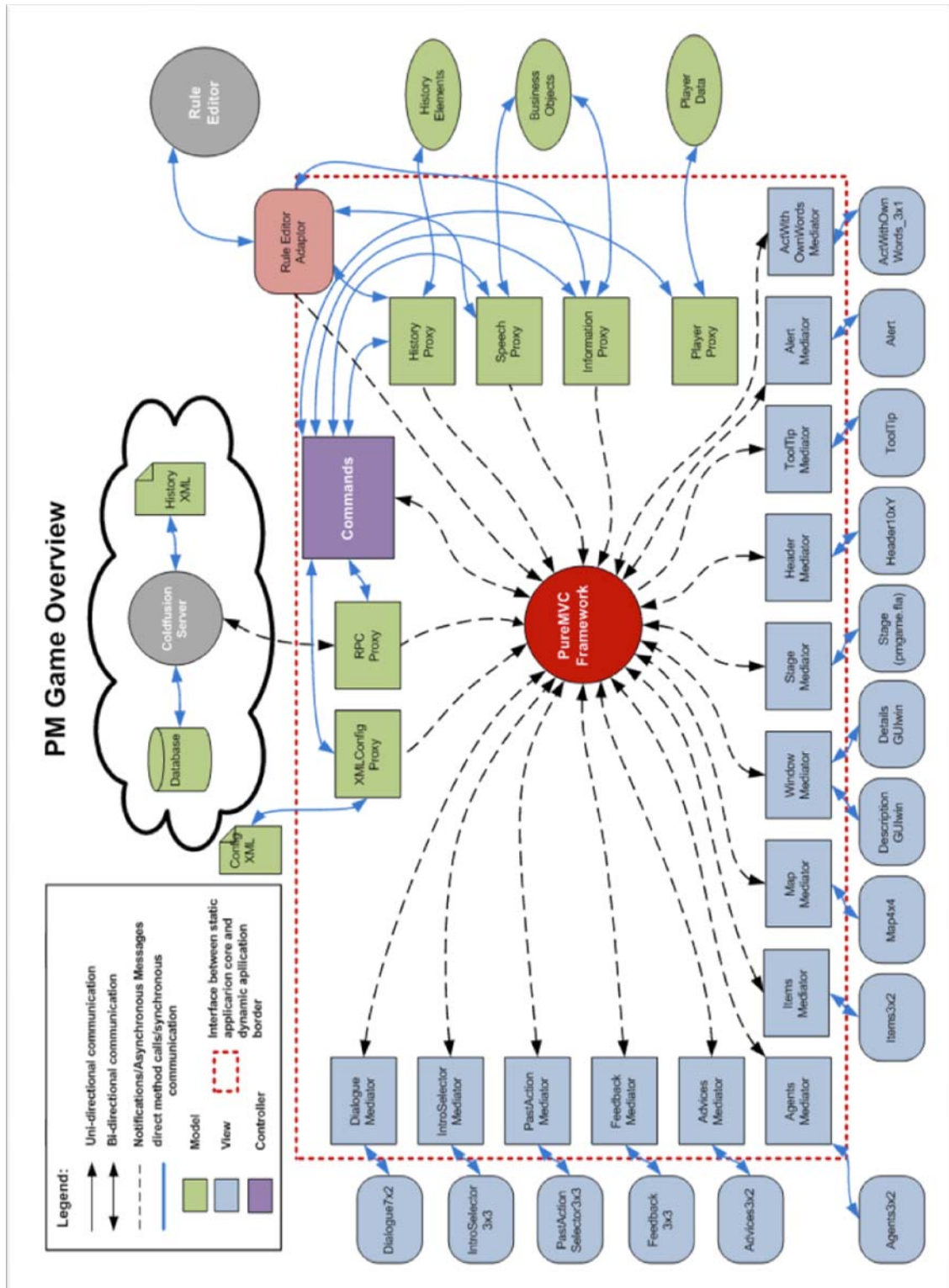


Abb. 33: PM Game-Architekturübersicht

Zur Performanzsteigerung werden rechenintensive Aufgaben beim *PM Game* dem Client delegiert. Dadurch werden Hardwarekosten für die Serverinfrastruktur gespart und die Anzahl der Serveranfragen reduziert, was zu einer Steigerung der Netzwerkperformanz führt. Dem Client werden alle Präsentationsaufgaben und Berechnungen überlassen, der Server stellt nur noch das Datenbankmanagementsystem und dessen Anbindung zur persistenten Speicherung der Daten zur Verfügung. Das Softwarearchitekturmuster wird bei *Sun*® als *Business Delegate* bezeichnet und löst das Problem der direkten Interaktion der Präsentation mit den Businessdiensten (Sun, 2008)¹⁷. Diese direkte Interaktion erfolgt durch die Benutzung der *API* der Businessdienste zur Präsentation und hat zur Folge, dass bei einer Aktualisierung der *API* des Businessdiensts auch die Präsentation aktualisiert werden muss (Entkopplung der verschiedenen Schichten).

Das Muster *Business Delegate* entkoppelt die clientseitige Präsentationsschicht von der Businesslogik durch Einführung einer Zwischenschicht. Dieses System macht die Businessdienste transparent und bietet einen *Lookup Service*, welcher den korrekten Zugang zur Businesslogik gewährleistet, ohne Details bekannt zu geben. Ausserdem können durch diese Delegation Informationen zwischengespeichert und müssen nicht wiederholt durch eine Serveranfrage geladen werden, was die Effizienz der Applikation steigert.

Abbildung 34 zeigt das Klassendiagramm des Musters *Business Delegate*. Der Client fragt den *Business Delegate* an, ob dieser Zugang zu einem bestimmten Dienst gewähren kann. Der *Lookup Service* sucht nach dem Dienst und gewährt Zugang zu diesem.

¹⁷ (<http://java.sun.com/blueprints/corej2eepatterns/Patterns/BusinessDelegate.html>)

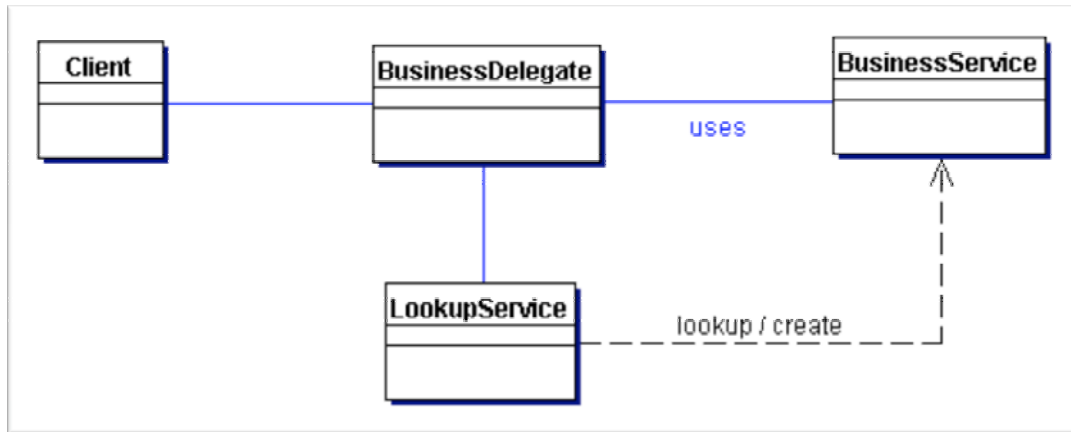


Abb. 34: Entwurfsmuster *Business Delegation* von Sun Microsystems® (2008)

4.10 Zusammenfassung

Das *PM Game* ist die prototypische Implementierung eines *Evolutionären Lernspielkonzepts* und wurde am Lehrstuhl für Educational Engineering in einem interdisziplinären Team von Forschern der Informatik, des Projektmanagements, der Psychologie sowie der Pädagogik entwickelt. Der Lerninhalt fokussiert die Vermittlung von Sozialkompetenzen, wie Kommunikation und Führung kleinerer Teams. Durch die geringere Veränderlichkeit von Sozialkompetenzen im Vergleich zu Technologiewissen ist das Problem der Aktualisierung der Ressourcen gemindert. Das Spiel baut auf einem konfigurierbaren, objektorientierten Regelwerk auf. Die technologische Basis bietet die Möglichkeit, Folgendes mit frei wählbaren, konkreten Inhalten zu modellieren: Personen mit Kompetenzen, Zuständen, Rollen, Beziehungen, Gesichtsausdruck, Gefühlen und parametrisierbaren Persönlichkeiten, Erinnerung, Wissen, Dialoge und Reaktionsmuster, Meetings mit mehreren Teilnehmern, Ziele, Gegenstände, Orte mit Eigenschaften und Zuständen.

Im Lernmodus des *PM Game* werden definierte Lektionen absolviert (Szenarien gespielt), das *PM Game* überprüft zielbezogen, ob die Szenarien bewältigt wurden und gibt Feedback. Nutzer können, neben dem Spielen der Szenarien, eigene Szenarien mit Hilfe spezieller Autorenwerkzeuge erstellen und ihr Wissen durch Synthese verankern. Die erstellten Szenarien werden von Experten (Lehrer, Assistenten

oder Dozenten) überprüft und freigeschaltet. Dies ermöglicht oberflächliche Qualitätskontrolle und Leistungsüberprüfung.

Die benutzergenerierten Inhalte sollen eine emergente Unterhaltung schaffen und dadurch die Neugierde der Benutzer wecken. Die Nutzer können die Szenarien bewerten und den Autoren Feedback geben, was zu einer unmittelbaren Rückkopplung führt, da die höchstbewerteten Szenarien den Nutzern prioritär angeboten werden. Beobachtungen abgebrochener oder häufig gespielter Szenarien, sowie Szenarien, deren Bewältigung eine hohe Trennschärfe hinsichtlich Nutzer-Qualifikation aufweisen, mit Hilfe des integrierten *PM Game*-Viewers erlauben weitere Rückschlüsse sowohl auf die Qualität der Szenarien als auch auf die Qualifikation der Nutzer.

5. Evaluation des *PM Game*

5.1 Einleitung

Im Rahmen der Lehrveranstaltung "IT-Projektmanagement" wurde 2008 ein Akzeptanztest durchgeführt, um die Nutzungseinstellung der Studenten bezüglich des Autorenwerkzeugs *PEdi* auf Benutzerfreundlichkeit zu untersuchen. Mit Hilfe dieses Autorenwerkzeugs werden die Projektmanagement-relevanten Lerninhalte für das *PM Game* editiert und konfiguriert (s. Kapitel 4.6). Didaktisches Ziel ist, dass Spieler nicht nur interaktiv im *PM Game* auf den Lerninhalt sensibilisiert werden, sondern dass auch durch Einbeziehen der Lernenden in die Inhaltsgenerierung Wissen bezüglich sozialer Kompetenzen und Kollaboration vermittelt wird.

5.2 Akzeptanz *Evolutionärer Lernspiele*

Im Folgenden werden grundlegende Begriffe der experimentellen Akzeptanzforschung sowie das Akzeptanzmodell von Venkatesh und Davis (2000) erläutert. Aus letzterem wird das Modell für den Akzeptanztest des *PEdi* hergeleitet. In dieser Arbeit setzt sich Akzeptanz aus einer Einstellungs- und Verhaltenskomponente zusammen (vgl. Ajzen & Fishbein, 2000; Aronson et al., 2004). Es geht somit bei der Einschätzung der Akzeptanz um die Frage, inwieweit eine positive Einstellung gegenüber dem *PM Game* zu dessen langfristiger, wiederholter Nutzung führt (Bürg, 2005).

5.2.1 Akzeptanzbegriff

Das Wort "akzeptieren" stammt vom lateinischen "accipere" ab und bedeutet „annehmen“, „billigen“ oder „gutheissen“. Dementsprechend kann Akzeptanz als die Bereitschaft, etwas zu akzeptieren, verstanden werden (Drosdowski, 1989). Eine andere Definition von Akzeptanz bietet Simon (2001):

Akzeptanz bezeichnet die positive Annahmeentscheidung einer Innovation durch die Anwender.

Müller-Bölling und Müller (1986) haben den Akzeptanzbegriff auf den Ansätzen von Simon und Kollmann basierend definiert und unterteilen ihn in Einstellungs- und Verhaltensakzeptanz gegenüber neuartigen Produkten oder Diensten bezüglich deren Nutzung. Des Weiteren unterscheiden sie eine affektive und eine kognitive Komponente der Einstellungsakzeptanz: Die affektive Komponente umfasst motivational-emotionale Faktoren, die kognitive Komponente hingegen rationale Abwägungen der Kosten/Nutzen einer Lösung unter Einbezug des persönlichen Kontexts (Kollman, 1998).

Einstellungs- und Verhaltensakzeptanz können jeweils eine positive oder negative Ausprägung annehmen. Folgende Matrix verdeutlicht dies (Tab. 16).

Tab. 16: Einstellungsakzeptanz und Verhaltensakzeptanz

		Verhaltensakzeptanz	
		Positiv	Negativ
Einstellungsakzeptanz	Positiv	Überzeugter Nutzer	Verhinderter Nutzer
	Negativ	Gezwungener Nutzer	Überzeugter Nicht-Nutzer

- Der **überzeugte Nutzer** ist vom System überzeugt und benutzt es zugleich auch.

- Der **verhinderte Nutzer** hat zwar eine positive Nutzungseinstellung zum System, nutzt es aber (aus unterschiedlichen, hier nicht näher diskutierten Gründen) nicht.
- Der **gezwungene Nutzer** hat zwar eine negative Einstellung gegenüber dem System, nutzt dieses aber (aus unterschiedlichen, hier nicht näher diskutierten Gründen) trotzdem.
- Der **überzeugte Nicht-Nutzer** ist jener, der das System bewusst nicht benutzt, da seine Einstellung dazu negativ ist.

Inwieweit oder ob ein Zusammenhang zwischen Einstellung und Verhalten besteht, ist eine viel diskutierte Fragestellung in der Psychologie (Rosch & Frey, 1997), der in dieser Arbeit nicht vertiefend nachgegangen werden soll. Eine ausführliche Übersicht findet der interessierte Leser in der Dissertation von Bürg (2005). Um den Zusammenhang zwischen Einstellung und Verhalten besser aufzudecken, haben sich in der empirischen Forschung so genannte Drittvariablen-Ansätze etabliert. Ein bekannter Ansatz ist diesbezüglich die "Theorie der überlegten Handlungen" (Ajzen & Fishbein, 1980), welche nicht das Verhalten selbst berücksichtigt, sondern die Verhaltensintention. Diese lässt sich aufgrund zweier Verhaltenskomplexe bestimmen:

1. Die Einstellung, welche als Einstellung zur Handlung definiert ist.
2. „Die soziale Norm, welche den Einfluss der Umgebung auf das Verhalten eines Individuums bestimmt. Die subjektive Norm beschreibt die obige Auffassung aus der Sichtweise des Individuums. Das Individuum macht sich dabei Gedanken, wie das eigene Verhalten von relevanten Drittpersonen gewertet wird. Daraus resultiert dann das entsprechende Verhalten [...].“ (Bürg, 2005)

5.2.2 Akzeptanzmodell: *Technology-Acceptance-Model*

Das vor allem im anglo-amerikanischen Raum weit verbreitete Akzeptanzmodell von Venkatesh und Davis (2000) basiert auf dem *Technology-Acceptance-Model*, welches die tatsächliche Nutzung abhängig von der Einstellungsakzeptanz macht: Hat eine Person eine positive Nutzungseinstellung, wird sie die Software nutzen. Die Nutzungseinstellung wird wiederum von den beiden Faktoren "wahrgenommener Nutzen" und "wahrgenommene einfache Bedienbarkeit" beeinflusst. Unter „wahrge-

nommenem Nutzen“ verstehen Davis et al., (1989; S. 985) „*the prospective user’s subjective probability that using a specific application system will increase his or her job performance within an organizational context*“, und unter "wahrgenommene einfache Bedienbarkeit", „*the degree to which the prospective user expects the target system to be free of effort*“. Daraus lässt sich schliessen: Je grösser der Nutzen und je einfacher die Bedienbarkeit einer Software eingeschätzt werden, desto grösser ist die Wahrscheinlichkeit, dass die Software benutzt wird.

Im *Technology-Acceptance-Model 2* (Abb. 35) werden ergänzend externe Einflussfaktoren definiert. Diese können in soziale Prozessvariablen und kognitiv-instrumentelle Variablen unterschieden werden. Zu ersteren gehören die subjektive Norm und das Image. Kognitiv-instrumentelle Variablen sind die Relevanz des Systems für die Ausübung des Jobs, Qualität des Outputs und Nachweisbarkeit der Ergebnisse. Unter subjektiver Norm wird der Einfluss anderer Personen auf das Individuum verstanden. Dieser kann etwa durch deren Einschätzung der Wichtigkeit, diese Software zu nutzen, ausgeübt werden. Entscheidend für die letztendliche Nutzung ist dem überarbeiteten Modell zufolge der persönliche Stellenwert der Person, die den Einfluss ausübt (Bürg, 2005). Der Faktor „Image“ wird von Moore und Benbasat (1991, S. 195) wie folgt definiert:

„the degree to which use of an innovation is perceived to enhance one’s status in one’s social system”.

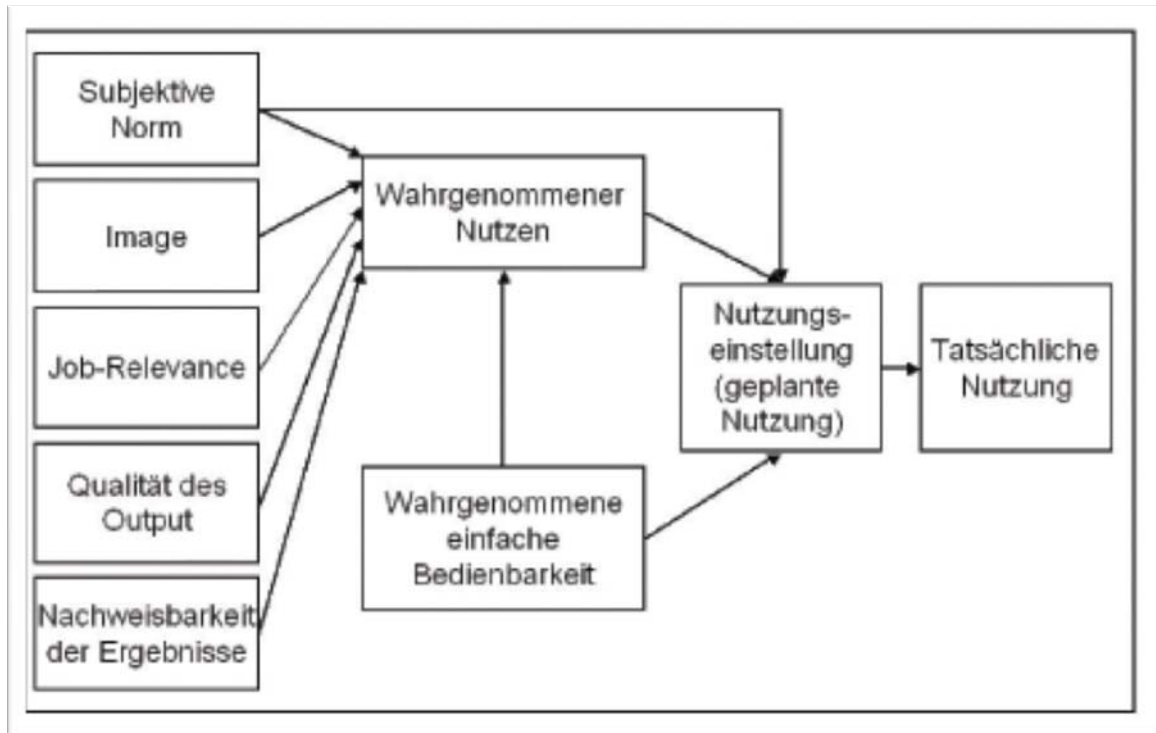


Abb. 35: Das *Technology-Acceptance-Modell 2* (Venkatesh & Davis, 2000)

5.3 *PEdi*-Evaluation

Ziel der Entwicklung und Durchführung des nachfolgend vorgestellten Akzeptanztests war einerseits, das Autorenwerkzeug *PEdi*, insbesondere seine Benutzerschnittstelle, das Tutorial und die Online-Hilfefunktion zu verbessern, und andererseits die Nutzungseinstellung der Studenten bezüglich *PEdi* zu untersuchen. Des Weiteren wurde getestet, ob und welche IT-Kenntnisse der Studenten vorausgesetzt werden müssen, damit diese in der Lage sind, adäquat mit dem *PEdi* zu arbeiten. Dazu wurde ein Benutzerakzeptanztest basierend auf dem beschriebenen *Technology-Acceptance-Modell 2* von Venkatesh und Davis (2000) entwickelt.

Die Evaluation des Autorenwerkzeugs war Teil der Lehrveranstaltung "IT-Projektmanagement" und somit Pflicht für die Studenten, um diesen erfolgreich zu bestehen. Die Teilnehmer absolvierten den Test also nicht völlig freiwillig. Darüber hinaus konnte das gewählte Test-Setting die tatsächliche Nutzung nicht exakt abbilden, da die Benutzung des *PEdi* einmalig in der Lehrveranstaltung angeboten wurde. Aufgrund dieser Einschränkungen wurde das *Technology-Acceptance-Modell 2* den

gegebenen Bedingungen angepasst und vereinfacht: Der "wahrgenommene Nutzen" und die "tatsächliche Nutzung" wurden eliminiert, da ihre Erfassung nicht sinnvoll oder nicht möglich war (Abb. 36).

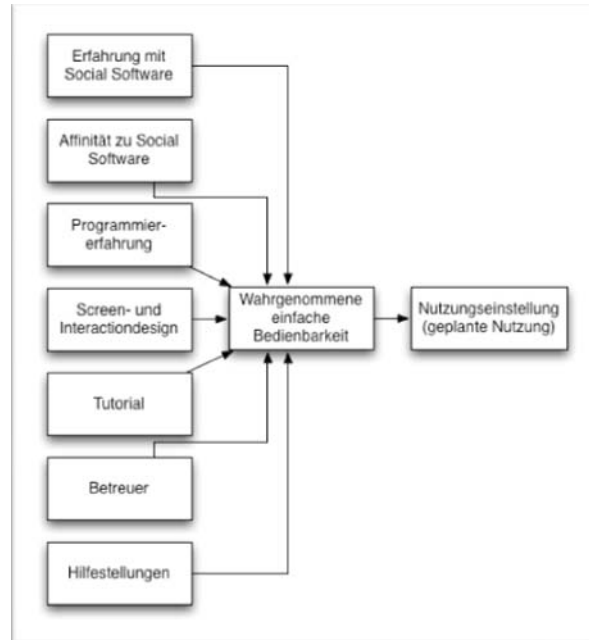


Abb. 36: Adaptiertes Modell für den Akzeptanztest des PEdi

Die Evaluation des Autorenwerkzeugs *PEdi* erfolgte in drei sequentiellen Schritten: Zunächst spielten die Teilnehmer ein Beispielszenario des *PM Game*, um erste Erfahrungen mit dem Spiel und seiner Bedienung zu sammeln, anschliessend arbeiteten sie ein Tutorial durch, welches die Erstellung und Konfiguration eines eigenen Szenarios beschreibt. Abschliessend wurden von Testleiter Dreiergruppen gebildet, welche kollaborativ ein komplexes Szenario anhand vorgegebener Anforderungen weiterentwickeln mussten.

Nach Beendigung der Gruppenarbeit nahmen die Teilnehmer Einschätzungen auf einem Online-Fragebogen vor, welcher Grundlage für die Evaluation war. Der Fragebogen war in folgende Bereiche gegliedert:

- Fragen zu demographischen Aspekten
- Fragen zu Affinität zur Nutzung von Informationstechnologie
- Fragen zur Motivation
- Fragen zur graphischen Benutzerschnittstelle

- Fragen zum Ablauf der Inhaltsgenerierung (Logik)
- Fragen zu Fehlern in der Software
- Fragen zum Zeitaufwand
- Fragen zum Tutorial
- Fragen zu den Hilfefunktionen
- Fragen zu den Betreuern
- Fragen zur Kooperation in der Gruppe
- Benotung der einzelnen Bereiche und
- Gesamtnote des *PEdi*

Der Fragebogen und die dazugehörigen vollständigen Auswertungen befinden sich im Anhang A und B. Der exakte Ablauf des Tests wird nachfolgend anhand der eingesetzten Instruktionen verdeutlicht.

5.4 Aufgabenstellungen

1. **Aufgabe:** Spiele das Szenario „Mary und Gary“ und protokolliere deine Aktionen und allfällige Unklarheiten.

Die Probanden bekamen die Aufgabe gestellt, ein Einführungsszenario mit Lerninhalt „Informationsbeschaffung“ zu spielen und sich so mit dem Spiel und seiner Bedienung vertraut zu machen. Ziel dieses Einführungsszenarios war es einerseits, die grundlegende Struktur, Handlungsabläufe und Interaktionsmuster des Spiels zu verstehen und andererseits allfällige Unklarheiten bezüglich Inhalt/Aufgabe, Benutzerschnittstelle oder Interaktionen zu protokollieren. Dieses Wissen resp. diese Informationen sind notwendig, um die Testpersonen in die Lage versetzen zu können, anschliessend eigene Szenarien zu konfigurieren.

Die weitere Aufgabenstellung lautete daher wie folgt: „Loggt euch mit euren Benutzerdaten, welche ihr per Mail erhalten habt, in das *PM Game* ein und spielt das Szenario "Mary und Gary" gemäss den Anweisungen und Hilfestellungen, welche im Spiel gegeben werden. Protokolliert während des Spielens, wo es

Unklarheiten bezüglich Inhalt (*Was muss ich als nächstes machen?*), Benutzerschnittstelle oder Interaktionen gibt.“

2. Aufgabe: Gehe das Tutorial Schritt für Schritt durch.

Im zweiten Schritt arbeiteten die Probanden das Tutorial durch, um anhand von Instruktionen und Screenshots ein Szenario zu konfigurieren. Nach jedem Abschnitt im Tutorial galt es, eine Aufgabe zu lösen, welche den Teilnehmern Zeit zum Reflektieren geben sollte. So sollte sichergestellt werden, dass die Informationen des Tutorials verarbeitet würden. Das Tutorial befindet sich in Anhang C dieser Arbeit.

3. Aufgabe: Erweitere das erstellte Szenario

Im letzten Schritt wurden Dreiergruppen gebildet, in denen das Szenario aus der vorherigen Teilaufgabe erweitert werden sollte. Ziel dieser Aufgabe war es, dass die Probanden die anspruchsvollen Erweiterungen kollaborativ in das Szenario einbauen. Dabei wurde vor allem auf den Austausch von Informationen (Wissens-transfer) innerhalb der Gruppen geachtet.

5.5 Organisation & Durchführung der Evaluation

5.5.1 Qualitätsnachweise

Validität

Validität eines Tests liegt dann vor, wenn dieser tatsächlich den Messgegenstand erfasst, den zu messen er vorgibt und seine Anwendung damit Gültigkeit erlangt (Schnell et al., 1999).

Reliabilität

Reliabilität gibt an, wie verlässlich die Messungen mit einem Verfahren sind, unabhängig davon, ob sie valide sind. Das Reliabilitätsmass liegt zwischen minus unendlich (vollkommen unzuverlässig) und 1 (vollkommen zuverlässig)

Die Reliabilität wurde in der Evaluation des *PM Game* mit Cronbach's Alpha berechnet. Reliabilitätsangaben können generell wie folgt abgestuft werden:

Cronbach's Alpha < 0.70 => ungenügend

Cronbach's Alpha zwischen 0.70 und 0.80 => befriedigend

Cronbach's Alpha zwischen 0.80 und 0.90 => gut

Cronbach's Alpha > 0.90 => hervorragend

5.5.2 Hypothesen & Stichprobe

Zur Bestimmung der Benutzerakzeptanz und zur Untersuchung, ob auch Benutzer ohne Affinität zu Online-Plattformen in der Lage sind, eigene Szenarien mit Hilfe des *PEdi* zu konfigurieren, wurden folgende Hypothesen aufgestellt und untersucht.

H1.0:	<i>PEdi</i> erfährt keine Akzeptanz durch den Nutzer (indiziert durch Gesamtnote < 4.0).
--------------	---

H1.1:	<i>PEdi</i> erfährt Akzeptanz durch den Nutzer (indiziert durch Gesamtnote ≥ 4.0).
H2.0:	Die untersuchten Bereiche des <i>PEdi</i> bilden dessen Gesamtnote nicht oder nur ungenügend ab (besitzen keinen oder nur geringen Vorhersagewert).
H2.1:	Die untersuchten Bereiche des <i>PEdi</i> bilden dessen Gesamtnote ab (besitzen aussagekräftigen Vorhersagewert).
H3.0:	Es gibt keinen Zusammenhang zwischen der Affinität der Probanden zur Nutzung von Online-Plattformen und ihrer Bewertung des <i>PEdi</i> (operationalisiert über Gesamtnote).
H3.1:	Je grösser die Affinität der Probanden zur Nutzung von Online-Plattformen ist, desto besser bewerten diese <i>PEdi</i> (operationalisiert über Gesamtnote).

Die Variable „Affinität zur Nutzung von Online-Plattformen“ wurde aus folgenden erfassten Variablen nach diesem Berechnungsschema neu gebildet: Anzahl genutzter Online-Plattformen multipliziert mit der Häufigkeit der Benutzung dieser Plattformen.

Des Weiteren wurde untersucht, inwiefern Programmierkenntnisse die Gesamtnote des *PEdi* beeinflussen.

H4.0:	Es gibt keinen Zusammenhang zwischen der Anzahl beherrschter Programmiersprachen und der Bewertung des <i>PEdi</i> (operationalisiert über Gesamtnote).
H4.1:	Je mehr Programmiersprachen ein Proband beherrscht, desto besser bewertet er <i>PEdi</i> (operationalisiert über Gesamtnote).

Da ein Zusammenhang zwischen Programmierkenntnissen und der Gesamtnote des Bereichs „Logik“ vermutet wurde, sollen folgende Hypothesen gelten:

H5.0:	Es gibt keinen Zusammenhang zwischen der Anzahl beherrschter Programmiersprachen und der Gesamtnote im Bereich „Logik“.
H5.1:	Es gibt einen positiven Zusammenhang zwischen der Anzahl beherrschter Programmiersprachen und der Gesamtnote im Bereich „Logik“.

5.5.3 Weitere Hypothesen

Des Weiteren wurde untersucht, ob geschlechtsspezifische Unterschiede in Bezug auf die Affinität zur Nutzung von Online-Plattformen bestehen. Dazu wurden folgende Hypothesen aufgestellt:

H6.0:	Männer und Frauen unterscheiden sich nicht in ihrer Affinität zur Nutzung von Online-Plattformen.
H6.1:	Männer haben eine grössere Affinität zur Nutzung von Online-Plattformen als Frauen.

Ausserdem wurde untersucht, ob geschlechtsspezifische Unterschiede in Bezug auf die Benotung des Bereichs „Logik“ bestehen. Dazu wurden folgende Hypothesen aufgestellt und getestet:

H7.0:	Männer und Frauen unterscheiden sich nicht in ihrer Benotung des Bereichs „Logik“.
H7.1:	Männer bewerten den Bereich Logik überzufällig besser als Frauen.

Es war ebenso von Interesse, ob Unterschiede zwischen den Geschlechtern bezüglich der Gesamtbenotung des *PEdi* sowie der Anzahl beherrschter Programmiersprachen zu finden waren.

H8.0:	Männer und Frauen unterscheiden sich nicht in ihrer Gesamtbenotung des <i>PEdi</i> .
H8.1:	Männer geben dem <i>PEdi</i> eine bessere Gesamtnote als Frauen.
H9.0:	Männer und Frauen unterscheiden sich nicht in der Zahl beherrschter Programmiersprachen.
H9.1:	Männer beherrschen mehr Programmiersprachen als Frauen.

5.5.4 Analysestichprobe

Die Testpersonen rekrutierten sich aus der Grundgesamtheit aller Teilnehmer der Lehrveranstaltung "IT-Projektmanagement" im Jahr 2008. Für diese Lehrveranstal-

tung schrieben sich Studenten von verschiedenen Fakultäten der Universität Zürich sowie der ETH Zürich ein. Es bestand die Möglichkeit, sich neben den obligatorischen Lehreinheiten auch für zusätzliche Optionen (darunter auch die vorgestellte *PEdi*-Evaluation) anzumelden. Insgesamt mussten für eine erfolgreiche Kursteilnahme mindestens zwei Zusatzoptionen besucht werden. Von den 30 für diese Evaluation angemeldeten Personen mussten acht ausgeschlossen werden, weil sie bereits über zu grosse Vorkenntnisse des *PM Game* verfügten. Die finale Analysestichprobe zur Evaluation des *PEdi* bilden somit insgesamt 22 Personen, darunter zwölf Frauen. Die Teilnehmer studierten an der Universität Zürich und/oder der ETH Zürich unterschiedliche Fachrichtungen im Bachelor- oder Masterstudiengang. Vier Personen studierten Wirtschaftsinformatik, eine Informationstechnologie und zwei Informatik. Unter diesen sieben Studenten einer direkt mit dem Fachgebiet der Informatik in Verbindung stehenden Fachrichtung, waren drei Frauen.

Der Alterdurchschnitt betrug 25 Jahre ($SD = 3.25$), der älteste Teilnehmer war 32 Jahre und der Jüngste 22 Jahre alt.

5.6 Resultate

Nachfolgend werden die Resultate des Online-Fragebogens zur Akzeptanz des *PEdi* präsentiert und anhand der Hypothesen diskutiert. Details zum Akzeptanztest können dem vorherigen Kapitel entnommen werden. Zunächst werden die Ergebnisse deskriptiv dokumentiert und anschliessend entlang der Hypothesen und Fragestellungen diskutiert.

Deskriptives zum Akzeptanztest

Tabelle 17 stellt die Korrelationen der verschiedenen Bereiche des Akzeptanztests dar (N=22).

Tab. 17: Korrelationen der untersuchten Bereiche des Akzeptanztests

	1	2	3	4	5	6	7	8
1	–							
2	0.722**	–						
3	0.695**	0.647**	–					
4	0.265	0.155	0.188	–				
5	0.569**	0.489*	0.595**	0.158	–			
6	0.146	0.175	0.063	-0.143	0.425*	–		
7	0.265	-0.139	0.193	-0.051	0.433*	0.088	–	
8	0.203	0.311	0.026	-0.243	0.044	0.179	0.123	–

- 1 = Gesamtnote des *PEdi*
- 2 = Gesamtnote Benutzerschnittstelle
- 3 = Gesamtnote Logik
- 4 = Gesamtnote Fehler in der Software
- 5 = Gesamtnote Tutorial
- 6 = Gesamtnote Hilfesystem
- 7 = Gesamtnote Betreuer
- 8 = Gesamtnote Kollaboration

Wie aus Tabelle 17 ersichtlich, besteht ein signifikanter Zusammenhang zwischen der Benotung der Benutzerschnittstelle und Gesamtnote des *PEdi* ($r=.72$), d.h. das *Look-and-Feel* und das Interaktionsdesign des *PEdi* spielen für die Gesamtbenotung des *PEdi* eine statistisch bedeutsame Rolle. Es findet sich darüber hinaus auch signifikante Zusammenhänge der Gesamtnote des *PEdi* mit den Gesamtnoten der Bereiche

** Korrelation signifikant auf $\alpha < 0.01$ -Niveau

* Korrelation signifikant auf $\alpha < 0.05$ -Niveau

Logik ($r=.70$) und Tutorial ($r=.57$). Die Fehler in der Software, das Hilfesystem, die Betreuer sowie die Kollaborationsmöglichkeiten weisen hingegen keine signifikanten Zusammenhänge mit der Gesamtnote des *PEdi* auf. Diese Erkenntnisse wurden anschliessend auch im qualitativen Feedback belegt (z.B. haben nur drei Teilnehmer auf die Hilfestellungen im *PEdi* zurückgegriffen).

Tabelle 18 zeigt die Reliabilitäten der einzelnen Bereiche des Akzeptanztests.

Tab. 18: Reliabilitäten der einzelnen Bereiche des Akzeptanztests

Teilbereiche	Cronbach's Alpha	Anzahl Items
Benutzerschnittstelle	0.521	7
Logik	0.804	4
Fehler in der Software	0.822	3
Tutorial	0.810	10
Hilfesystem	0.476	5
Betreuer	0.890	5
Kollaboration	0.687	3

Tabelle 18 zeigt, dass die berechneten Reliabilitäten für fünf der getesteten sieben Bereiche mindestens „befriedigend“ (vier davon sogar „gut“) ausfallen (Cronbach's Alpha > 0.68). Unbefriedigend sind lediglich die Reliabilitäten der Teilbereiche Benutzerschnittstelle (Cronbach's Alpha = 0.521) und Hilfesystem (Cronbach's Alpha = 0.476). Letzteres kann durch die geringe Nutzung dieser Funktionalität (s.u.) begründet sein, die diesbezüglich verlässliche Einschätzungen nicht zulies. Schlussfolgernd kann festgehalten werden, dass die mit diesem Test erhobenen Daten als weitgehend zuverlässig betrachtet und somit weiteren Interpretationen unterzogen werden können.

Für die Beurteilung ihrer Akzeptanz wurde im Fragebogen nach der Benotung der einzelnen Teilbereiche sowie nach einer Gesamtnote des *PEdi* gefragt. Die Schulnotenskala umfasste die Abstufungen 1 bis 6, wobei 6 die beste Note widerspiegelt. Die Mittelwerte und Standardabweichungen der Bewertung dokumentiert Tabelle 19 (N=22).

Tab. 19: Mittelwerte und Standardabweichungen der einzelnen Bereiche

Testbereich	mittlere Benotung	Standardabweichung
Gesamtnote des <i>PEdi</i>	4.14	0.89
Gesamtnote Benutzerschnittstelle	3.64	0.58
Gesamtnote Logik	2.77	0.86
Gesamtnote Fehler in der Software	4.18	1.10
Gesamtnote Tutorial	3.33	0.68
Gesamtnote Hilfesystem	3.02	0.57
Gesamtnote Betreuer	4.44	0.91
Gesamtnote Kollaboration	4.06	0.83

Es fällt auf, dass nur die Hälfte aller erfassten Teilbereiche eine Bewertung > 4.0 erzielten und damit als genügend beurteilt wurden.

Die Testteilnehmer bewerteten den Bereich Logik mit einer Gesamtnote von 2.77 (SD=0.86) als besonders ungenügend. Insgesamt erzielt der *PEdi* eine Gesamtnote von 4.14 (SD=0.89) und wird somit als befriedigend eingeschätzt. Das Zustandekommen der Gesamtnote des Hilfesystems ist auf den ersten Blick nicht nachvollziehbar, da nur drei Studenten angaben, das Hilfesystem überhaupt benutzt zu haben und somit 85% der Befragten eigentlich gar nicht in der Lage waren, die Qualität des Hilfesystems zuverlässig zu bewerten. Somit lässt sich schlussfolgern, dass die Durchschnittsnote von 3.0 (SD=0.57) durch neutrales Bewerten dieses *Items* zustande gekommen ist, wofür auch die im Vergleich zu den anderen Benotungen kleinste Streuung der Bewertungen spricht. Wie sich im Zuge der qualitativen Befragung der Probanden herausstellte, war die Schaltfläche für den Aufruf des Hilfesystems unglücklich platziert, sodass das Hilfesystem nicht ohne Weiteres gefunden werden konnte.

Obwohl die Studenten in der Lage waren, das Tutorial selbständig durchzuarbeiten und das gewünschte Produkt zu erstellen, wurde der Bereich Tutorial mit 3.33 (SD=0.68) als ungenügend benotet. Es benötigt weiterer Abklärungen, um diese Note zu interpretieren. Auch die qualitative Befragung liess keine Rückschlüsse auf ihr Zustandekommen zu. Des Weiteren wurden die Teilnehmer zur Evaluation der Benutzerakzeptanz gebeten, folgende Aussagen einzuschätzen. Die Skala zur Bewertung der einzelnen Bereiche umfasste folgende fünf Abstufungen: 1 = lehne völlig ab, 2 = lehne eher ab, 3 = teils/teils, 4 = stimme eher zu, 5 = stimme völlig zu.

Tab. 20: Mittelwerte und Standardabweichungen zur Motivation der Studenten (n=22)

Aussage/Item	Mittelwert	Standardabweichung
Es hat mir Spass gemacht, mit dem Editor zu arbeiten.	3.41	1.22
Es war für mich lehrreich, mit dem Editor zu arbeiten.	3.27	1.28
Ich würde auch zukünftig mit dem Editor arbeiten.	2.73	1.20
Ich würde mich gern aktiv an der Weiterentwicklung des <i>PEdi</i> beteiligen.	2.23	1.31
Ich kann die Arbeit mit dem Editor weiterempfehlen.	3.00	1.11
Ich würde die gleiche Option ¹⁸ erneut wählen.	3.32	1.59

Wie aus Tabelle 20 ersichtlich, hatten die Teilnehmer Spass an der Arbeit mit dem *PEdi*. Das qualitative Feedback einzelner Teilnehmer bestätigte den Befund, dass die Probanden die Arbeit mit dem *PEdi* interessant und abwechslungsreich empfunden haben. Die Einschätzungen der Aussagen bezüglich zukünftiger *PEdi*-Nutzung lassen jedoch darauf schliessen, dass die Teilnehmer tendenziell unentschlossen darüber sind, ob sie *PEdi* auch ausserhalb der Testsituation nutzen wollen. Möglicherweise ist es nicht gelungen, den Nutzen des *PEdi* deutlich genug hervorzuheben. Dies könnte auf das eher triviale Szenario zurückzuführen sein, welches die Teilnehmer mit Hilfe des Tutorials erstellen mussten. Die tiefe Bewertung der Aussage 4 (M=2.23) zur Bereitschaft, sich auch an der Weiterentwicklung des *PEdi* zu beteiligen, ist nach Ansicht des Autors auf die geringe Anzahl an Teilnehmern mit Programmierkenntnissen zurückzuführen.

Der Mittelwert der Gesamtnote des *PEdi* liegt bei M=4.14 (SD=0.89) und somit über der geforderten Note von 4.0. Aufgrund dessen muss H1.0 verworfen werden.

H1.0: Der PM Game Editor erfährt keine Akzeptanz (indiziert durch Gesamtnote < 4.0) durch den Nutzer. **(verworfen)**

H1.1: Der PM Game Editor erfährt überdurchschnittliche Akzeptanz (indiziert durch Gesamtnote \geq 4.0) durch den Nutzer. **(angenommen)**

Da neben der Gesamtnote des *PEdi* noch die Benotungen der einzelnen Teilbereiche erhoben wurden, stellte sich die Frage, ob die Benotungen der Teilbereiche die Gesamtnote des *PEdi* vorhersagen und damit gezeigt werden kann, dass die

¹⁸ Gemeint ist die Kursoption "Evaluation des *PEdi*" der von den Probanden besuchten Lehrveranstaltung "IT-Projektmanagement"

„richtigen“ Teilbereiche zur Evaluation des *PEdi* herangezogen wurden. Zur Überprüfung wurden folgende Hypothesen aufgestellt:

H2.0: Die untersuchten Bereiche des *PEdi* bilden dessen Gesamtnote nicht (ausreichend) ab (sagen sie nicht vorher). (**verworfen**)

H2.1: Die untersuchten Bereiche des *PEdi* bilden dessen Gesamtnote (statistisch bedeutsam) ab (sagen sie vorher). (**angenommen**)

H2.0 muss verworfen werden, da die Teilbereiche (= Prädiktoren) per Regressionsanalyse jeweils signifikant und insgesamt 71% der Varianz in der Gesamtnote des *PEdi* (= Kriterium) aufklären (R^2 ; $\alpha < 0.01$, s. Anhang B).

Wie die Auswertung zeigt, findet sich kein signifikanter Zusammenhang zwischen der Affinität zur Nutzung von Online-Plattformen und der Gesamtnote des *PEdi* ($r=0.071$, $\alpha=?$). Dies ist in dem vom Autor gewünschten Sinn, da der *PEdi* für ein möglichst breites Spektrum unterschiedlicher Nutzer konzipiert wurde und nicht nur auf Nutzer mit grosser Affinität zu Online-Plattformen beschränkt bleiben soll.

H3.0: Es gibt keinen Zusammenhang zwischen der Affinität der Probanden zur Nutzung von Online-Plattformen und der Gesamtnote des *PEdi* (**angenommen**)

H3.1: Je grösser die Affinität der Testpersonen zur Nutzung von Online-Plattformen ist, desto besser fällt ihre Gesamtnote für den *PEdi* aus. (**verworfen**)

Von den 22 Teilnehmern dieser Studie beherrschten vier Teilnehmer eine oder mehrere Programmiersprachen, dies ergibt einen Durchschnitt von 0.5 Programmiersprachen pro Teilnehmer. Sieben Teilnehmer studierten ein Fach, welches im direkten Zusammenhang mit Informatik steht. Die Auswertung der Daten ergab einen signifikanten Zusammenhang zwischen Anzahl beherrschter Programmiersprachen und der Gesamtbenotung des *PEdi* ($r=0.362$, $\alpha < 0.05$). Somit kann **H4.0** nicht bestätigt werden.

H4.0: Es besteht kein signifikanter Zusammenhang zwischen der Anzahl beherrschter Programmiersprachen und der Gesamtnote des *PEdi*. (**verworfen**)

H4.1: Je mehr Programmiersprachen beherrscht werden, desto besser fällt die Gesamtnote für den *PEdi* aus. (**angenommen**)

Die Annahme, dass kein signifikanter Zusammenhang zwischen der Anzahl beherrschter Programmiersprachen und der Gesamtnote des Teilbereichs Logik bestünde, bestätigt sich nicht ($r=0.448$, $\alpha < 0.05$). Somit muss **H5.0** verworfen werden.

H5.0: Es besteht kein Zusammenhang zwischen der Anzahl beherrschter Programmiersprachen und der Gesamtnote des Teilbereichs Logik. (**verworfen**)

H5.1: Es gibt einen positiven Zusammenhang zwischen der Anzahl beherrschter Programmiersprachen und der Gesamtnote im Teilbereich Logik. (**angenommen**)

Die Auswertung der Hypothesen zeigt, dass Nutzer mit grösserer Affinität zur Nutzung von Online-Plattformen keine bessere Gesamtbenotung des *PEdi* vornehmen, obwohl dessen zur Verfügung gestellte Funktionen Ähnlichkeiten mit den gängiger Online-Plattformen aufweisen und die Interaktionsmöglichkeiten somit schon bekannt sein sollten. Jedoch hat sich gezeigt, dass Nutzer mit grösserer Programmiererfahrung (erfasst über die Anzahl beherrschter Programmiersprachen) *PEdi* eine bessere Gesamtnote geben, als solche mit geringerer Programmiererfahrung. Dies ist darauf zurückzuführen, dass die Konfiguration der Objekte für ein Szenario sehr stark an das *Objektorientierte Paradigma* angelehnt ist, das sich nicht intuitiv, sondern erst nach intensiverer Auseinandersetzung mit informationstechnologischer Modellierung verstehen lässt.

Mittels T-Tests wurden die Hypothesen bezüglich geschlechterspezifischer Unterschiede überprüft. Die Hypothese, Männer und Frauen würden sich nicht in ihrer Affinität zur Nutzung von Online-Plattformen unterscheiden (H6), konnte ebenso bestätigt werden wie die Behauptungen, beide Geschlechter unterschieden sich nicht in ihrer Benotung des Teilbereichs Logik (H7) und in ihrer Gesamtbewertung des *PEdi* (H9).

H6.0: Männer und Frauen unterscheiden sich nicht in ihrer Affinität zur Nutzung von Online-Plattformen. (**angenommen**)

H6.1: Männer weisen eine grössere Affinität zur Nutzung von Online-Plattformen auf als Frauen (**verworfen**)

H7.0: Männer und Frauen unterscheiden sich nicht in ihrer Benotung des Teilbereichs Logik. (**angenommen**)

H7.1: Männer beurteilen den Teilbereich Logik positiver als Frauen
(**verworfen**)

H8.0: Männer und Frauen unterscheiden sich nicht in ihrer Gesamtbenotung des
PEdi. (**angenommen**)

H8.1: Männer beurteilen *PEdi* insgesamt positiver als Frauen (**verworfen**)

Hypothesenkonform lassen sich keine geschlechterspezifischen Unterschiede bezüglich der Akzeptanz des *PEdi* mittels T-Test finden.

Die Annahme, Männer und Frauen würden sich nicht in der Anzahl beherrschter Programmiersprachen unterscheiden (H9), bestätigt sich hingegen nicht. Männer gaben signifikant mehr beherrschte Programmiersprachen an als Frauen (Mittelwert Männer = 1.0 gegenüber Mittelwert Frauen = 0.08, $t = 1.78$, $\alpha < 0.05$).

H9.0: Männer und Frauen unterscheiden sich nicht in der Zahl beherrschter Programmiersprachen. (**verworfen**)

H9.1: Männer beherrschen mehr Programmiersprachen als Frauen
(**angenommen**)

Zusammenfassend lässt sich festhalten, dass der Evaluationsfragebogen offensichtlich valide Teilbereiche zur Einschätzung der Akzeptanz erfasst und als weitgehend reliabel gelten kann. Die mit ihm erfassten Daten können daher analysiert und interpretiert werden.

Die Teilnehmer gaben mehrheitlich an, dass sie Spass an der Arbeit mit dem *PEdi* hatten und diese als interessant und abwechslungsreich empfunden haben. Der Mittelwert der Gesamtnote des *PEdi* liegt im Bereich „befriedigend“.

Die Einschätzungen der Aussagen bezüglich zukünftiger *PEdi*-Nutzung lassen jedoch darauf schliessen, dass die Teilnehmer tendenziell unentschlossen darüber sind, ob sie *PEdi* auch ausserhalb der Testsituation nutzen wollen. Möglicherweise ist es nicht gelungen, den Nutzen des *PEdi* deutlich genug hervorzuheben. Die tiefe Bewertung der Bereitschaft, sich auch aktiv an der Weiterentwicklung des *PEdi* zu beteiligen, ist nach Ansicht des Autors auf die geringe Anzahl an Teilnehmern mit Programmierkenntnissen zurückzuführen. Erfreulich ist dabei allerdings, dass *PEdi* von Online-Plattform-affinen und nicht-affinen Testteilnehmern gleichermaßen geschätzt wird.

Hingegen ist es kritisch zu sehen, dass die Gesamtnote für *PEdi* parallel zur Anzahl beherrschter Programmiersprachen steigt. Dies weist darauf hin, dass sich der volle Nutzen und das ganze Verständnis für die Logik der Applikation erst mit zunehmendem Verständnis informationstechnologischer Denkmuster und Modellierungsschemata erschliessen. Für ein intuitiveres Verständnis müssen daher noch Systemvereinfachungen erfolgen.

Die fehlenden geschlechtsspezifischen Unterschiede stimmen zuversichtlich, dass dies gelingen kann. *PEdi* scheint prinzipiell für Nutzer beiderlei Geschlechts und unabhängig deren Affinität zur Nutzung von Online-Plattformen geeignet. Anpassungen werden insbesondere in Bezug auf die Aspekte Benutzerschnittstelle und Tutorial gewünscht. Die tendenzielle Unzufriedenheit mit dem Hilfesystem kann eventuell bereits durch eine offensichtlichere Platzierung des entsprechenden *Buttons* behoben werden und erfordert nicht zwingend inhaltliche oder methodische Modifikationen.

5.7 Qualitatives Feedback

Ergänzend zum Akzeptanztest wurde ein qualitatives Interview mit den Studienteilnehmern durchgeführt. Ziel dieser Befragung war es, detaillierte Auskunft bezüglich der Emotionen und des Spielverständnis' für *PEdi* zu gewinnen und Verbesserungsvorschläge zu sammeln.

Die strukturierten Interviews beinhalteten folgende vier Fragen:

1. Was denkt ihr, war das Lernziel des gespielten Szenarios „Mary und Gary“?
2. Wie seid ihr zur Erreichung des Ziels vorgegangen?
3. Was hat euch gefallen? Was nicht?
4. Was würdet ihr am Spiel verbessern oder wo seht ihr die Schwächen des Spiels?

Im folgenden Abschnitt werden die Ergebnisse der Befragung zusammengefasst und analysiert.

Im Szenario "Mary und Gary" sollten die Probanden eine Hochzeit organisieren und durchführen. Lernziel dieses Szenarios waren das Verständnis sowie die Anwendbar-

keit der Informationsbeschaffung im *PM Game*. Lernziel und richtiger Lösungsweg wurden von den Studienteilnehmern durchgängig erkannt und verstanden. Probleme entstanden jedoch bei der Ausführung des Lösungswegs: Die Studenten empfanden es als mühsam, die gewünschten Sprechakte anhand des Menüsystems zu erzeugen. Die Kategorisierung der Sprechakte ist zwar flexibel in Bezug auf Erweiterungsmöglichkeiten, für den Nutzer jedoch nicht intuitiv nachvollziehbar. Somit bestand die Herausforderung für die meisten Teilnehmer nicht darin, den Lösungsweg zu finden, sondern die richtigen Sprechakte mit Hilfe des Menüsystems zu generieren. Zudem wurde kritisiert, dass Anweisungen und Hilfestellung nicht optimal platziert worden seien und dadurch nicht immer die nötige Aufmerksamkeit der Teilnehmer erhalten hätten.

Als Resultat der beschriebenen Kritikpunkte wurde das Menüsystem durch ein *Point&Click*-System auf graphische Repräsentationen der Objekte ersetzt. Durch diese Systemänderung können die Nutzer nun mit zwei bis drei Mausklicks einen gewünschten Sprechakt erstellen: Die parametrisierten Sprechakte sind in fünf Kategorien eingeteilt und können durch Anklicken ausgewählt werden. Die Parameter jedes Sprechakts können ebenfalls durch Klicken auf die entsprechenden Objekte ausgewählt werden. Die Platzhalter (Parameter) der verfügbaren Sprechakte werden daraufhin vom System ergänzt und somit unmittelbar für den Nutzer verständlich. Sprechakte, welche in Kombination mit dem ausgewählten Objekt keinen Sinn ergeben, werden von der Auswahl automatisch entfernt, so dass sich der Nutzer nur zwischen drei bis fünf Sprechakten pro Anfrage entscheiden muss.

Ein positives Erlebnis sowohl der Evaluation als auch der Feedbackrunde war die Kreativität der Teilnehmer bei der Szenarien-Erstellung, die zu völlig neuen Unterhaltungsformen führte, die vom Designer des Spiels ursprünglich gar nicht vorgesehen wurden. So wurden z.B. graphische Repräsentationen der Spielobjekte im *PEdi* übereinandergelegt und durch Sichtbar- resp. Unsichtbarmachen der einzelnen Graphiken zu Animationen verknüpft.

5.8 Zusammenfassung

Der Akzeptanztest hat gezeigt, dass die Studenten innerhalb von vier Stunden in der Lage waren, sich die Bedienung des *PEdi* mit Hilfe des bereitgestellten Tutorials zu erarbeiten und eigene Szenarien zu entwickeln. Als wichtige Faktoren für die Akzeptanz stellten sich die Benutzerschnittstelle, Arbeitsprozess/Logik und das Tutorial heraus. Die Arbeitsprozesse und die Logik wurden von den Probanden noch als ungenügend bewertet. Das Hilfesystem wurde kaum verwendet und spielte für die Gesamtnote des *PEdi* keine Rolle. Weitere Untersuchungen müssten durchgeführt werden, um festzustellen, ob dies an der Benutzerschnittstelle oder an anderen Faktoren lag.

Die Studenten waren motiviert, sich mit dem Lerninhalt „Projektmanagement“ mittels unkonventioneller Lernmethoden auseinanderzusetzen. Studenten, die über grössere Erfahrung mit Online-Plattformen verfügen, bewerten *PEdi* aber nicht automatisch besser als Studenten ohne derartige Erfahrung. Daraus lässt sich der zu begrüssende Befund ableiten, dass die Affinität zu Online-Plattformen für die Benutzung des *PEdi* offenbar keine Rolle spielt. Ebenfalls positiv sind die fehlenden geschlechterspezifischen Unterschiede bezüglich der Akzeptanz des *PEdi* zu bewerten. Allerdings liess sich ein signifikant positiver Zusammenhang zwischen der Benotung des *PEdi* und der Anzahl beherrschter Programmiersprachen finden. Dies ist vermutlich darauf zurückzuführen, dass die Szenarien im *PEdi* auf Objekten basieren und Wissen über das Objektorientierte Paradigma Vorteile beim Konfigurieren schafft.

6. Schlussbetrachtung

Ziel dieser Arbeit war es, aktuelle Internetgeschäftsmodelle zu analysieren, daraufhin passende Web 2.0-Merkmale/Prinzipien mit der Lernmethode *Game-based Learning* in einer Applikation zu kombinieren und daraus ein neues Lernspielkonzept – das *Evolutionäre Lernspielkonzept* – herzuleiten. Die Lernmethode *Game-based Learning* soll für die *Game Generation* adäquate Lernhilfen zur Verfügung stellen, sie intrinsisch motivieren und dadurch längerfristig an die Auseinandersetzung mit Lerninhalten binden (Holiday et al., 2007; Webster & Martocchio, 1993). Da die Herstellung digitaler Lernspiele jedoch äusserst aufwendig ist und die Budgets im Gegensatz zu kommerziellen Unterhaltungsspielen eher knapp kalkuliert sind, nutzen aktuelle digitale Lernspiele ihr Motivationspotential noch nicht einmal annähernd (Bergeron, 2006).

In den letzten Jahren entstanden neue Geschäftsmodelle und Webapplikationen für das Internet, welche helfen könnten, die Auswirkungen der Budgetknappheit für *Game-based Learning*-Anwendungen zu verringern: Die Nutzer werden zusehends häufiger als Produzenten eingesetzt und schaffen durch ihre Kreativität Unterhaltung für sich und andere. Da ein Grossteil der bisherigen Kosten für die Gestaltung und Kontrolle der Lerninhalte aufgewendet werden müssen, versucht das erarbeitete *Evolutionäre Lernspielkonzept*, den Ansatz benutzergenerierter Inhalte mit *Game-based Learning* zu kombinieren. Die grössten erfolgskritischen Herausforderungen bei der Realisierung dieser Verknüpfung betreffen dabei den Einschwingungsprozess, die Qualitätssicherung, die Benutzerakzeptanz sowie den Lerneffekt der Nutzer. In folgendem wird nun auf diese Herausforderungen, die daraus abgeleiteten Empfehlungen und deren Implikationen für die Forschung eingegangen.

6.1 Empfehlungen zum Betrieb eines *Evolutionären Lernspiels*

Einschwingungsprozess

Der Einschwingungsprozess für die Einführung des *PM Game* ist grundsätzlich nicht generalisierbar und somit nicht übertragbar auf andere Lernspiele, basierend auf dem gleichem Konzept, weil der Aufbau eines Spiels vom Kontext der Lernenden und dem Lerninhalt abhängt. Dennoch lassen sich Eckpunkte identifizieren, auf die bei der Einführung eines *Evolutionären Lernspiels* geachtet werden sollte:

Während der Einführung des Spiels ist weder die kritische Masse an Inhalten, noch an Nutzern vorhanden, so dass zunächst keine Eigendynamik in Bezug auf die „Evolution“ des Lernspiels erzeugt werden kann. Anfangs erscheint es unabdingbar, professionelle Autoren zu engagieren, welche eine qualitativ hochstehende Basis von Lernszenarien als *Best-Practice*-Beispiele für die Nutzer entwickeln, damit diese später eigene Inhalte anhand solcher Vorlagen generieren können. Wenn möglich, sollten diese Autoren schon vor der offiziellen Inbetriebnahme des *Evolutionären Lernspiels* beigezogen werden, damit allfällige Schwachstellen früh erkannt und Massnahmen zur Verbesserung eingeleitet werden können. Die Autoren müssen von Experten/Herstellern des Spiels geschult werden, um die Rollen und damit verknüpften Rechte kennenzulernen. In dieser Phase können darüber hinaus Anhaltspunkte dafür definiert werden, wann Nutzer in der Lage sind, die jeweiligen Rechte und Pflichten zu übernehmen. Das System sollte diesbezüglich möglichst flexibel gehalten sein, sodass Anpassungen der Schwellenwerte, Rollen und Rechte selbst zur Laufzeit des Systems noch aktualisiert werden können. Es hat sich als nicht sinnvoll erwiesen, die Rollen und Rechte während der Einschwingungsphase bereits definitiv zu bestimmen, denn häufig wird erst durch die Benutzung des Systems klar, wie die Einteilungen optimaler Weise erfolgen sollten. Dementsprechend sollte das Berechtigungssystem wie die Inhalte auch, evolutiven Prozessen unterliegen (*Perpetual Beta*).

Nachdem eine Basis von Inhalten erstellt worden ist, sollte eine virtuelle Gemeinschaft aufgebaut werden, welche spielt, bewertet und weitere Inhalte erstellt. Eine

Möglichkeit wäre, dass sich Studenten oder Schüler in einer Lehrveranstaltung mit dem *Evolutionären Lernspiel* vertraut machen und für die Gegenleistung eines Leistungsnachweises aktiv während eines Semesters mit dem Spiel auseinandersetzen. In dieser Einschwingungsphase könnten die Parameter des Systems aufgrund entsprechender Analysen und empirischer Evaluation der Erfahrungen der Nutzer adaptiert werden.

Ein Expertenteam sollte sowohl die Inhalte anhand aufgestellter Richtlinien kontrollieren und Feedback geben, wie auch als Supportteam auftreten, das (einführende) Schulungen anbietet und allfällige Fragen der Nutzer beantwortet.

Qualitätssicherung

Während des Einschwingungsprozesses erscheint es ratsam, die Qualitätskontrolle von Experten und professionellen Autoren durchführen zu lassen, damit später ein *Benchmark* für die Beurteilung der nutzergenerierten Inhalte zur Verfügung steht. Wenn die virtuelle Gemeinschaft die kritische Grösse erreicht hat, können die Inhalte von einem kleinen Experten-/Supportteam überprüft und freigeschaltet werden. Die Nutzer des *Evolutionären Lernspiels* bewerten die Inhalte zuvor, geben wenn möglich Feedback und führen so eine (Vor-)Selektion durch. Zu bemerken ist allerdings, dass neu erstellte Inhalte besonders gekennzeichnet werden sollten, damit sie in der Masse und aufgrund fehlender Bewertungen nicht untergehen. Die Nennung der Gruppe, welche die Inhalte erstellt hat, kann bei der Qualitätssicherung eine wichtige Rolle spielen, denn die Gruppen werden mit der Qualität der Szenarien assoziiert (Surowiecki, 2004). Als Nebeneffekt steigern die Gruppen oder deren Mitglieder infolge qualitativ hochstehender Inhalte ihre Reputation innerhalb der virtuellen Gemeinschaft.

Benutzerakzeptanz

In Bezug auf die Benutzerakzeptanz wurde im Rahmen dieser Arbeit ein Benutzerakzeptanztest des Autorenwerkzeugs *PEdi* entwickelt und mit Testpersonen verschiedener Studienrichtungen von der Universität Zürich und ETH Zürich durchgeführt. *PEdi* ermöglicht die Erstellung und Konfiguration von Lernszenarien für das *PM Game*. Der Benutzerakzeptanztest zeigte, dass die Nutzer mit dem Autorenwerkzeug arbeiten konnten, auch wenn keine Affinität zu Online-Plattformen bestand. Zudem

liessen sich keine geschlechtsspezifischen Unterschiede bezüglich der Akzeptanz des *PEdi* finden. Die Abwechslung gegenüber klassischem Frontalunterricht wurde von den Teilnehmern besonders geschätzt.

Offensichtlich ist für die Benutzerakzeptanz entscheidend, ob die Prozesse/Logik, Schnittstellen und Lernmethode den Bedürfnissen des Zielpublikums abgepasst sind, d.h. die graphische Oberfläche, das Spieldesign und Interaktionsmuster müssen den Anforderungen des Zielpublikums entsprechen, damit dieses die Applikation langfristig, wiederholt nutzt.

Lerneffekt

Ein Lernspiel muss auch auf seine Validität hin überprüft werden. Denn falls die Nutzer trotz Einsatz des Spiels nicht das lernen, was das Spiel zu vermitteln beabsichtigt, wird im besten Fall nur noch ein Unterhaltungs-, aber kein Lerneffekt erzielt. Deshalb sollte die Effektivität eines Lernspiels systematisch in Bezug auf seine Lerninhalte und -methodik evaluiert werden.

6.2 Implikationen für die Forschung

Das Forschungsgebiet *Game-based Learning* ist sehr jung, interdisziplinär und fordert noch sehr viel Grundlagenforschung. Wichtig erscheint dabei die Sensibilisierung anderer Forschungsdisziplinen und der Industrie auf Spiele als Lernwerkzeug. Insbesondere an europäischen Forschungs- und Lehrinstituten wird *Game-based Learning* noch als "Kinderkram" belächelt (vgl. Clark, 1983, 2007; Mayer, 2007), obwohl Studien bereits zeigen konnten, dass die Methode eine sinnvolle Alternative zu den klassischen Lernmodellen sein kann (Squire, 2003, 2005). Diese Arbeit stellt die bestehenden Bestrebungen im Gebiet *Game-based Learning* zusammen und evaluiert die Möglichkeiten der Integration von Web 2.0-Konzepten in *Game-based Learning* sowohl theoretisch durch Analyse bestehender Lernmodelle wie auch praktisch mit der Umsetzung des *Evolutionären Lernspielkonzepts* im *PM Game*. Weiterführende Untersuchungen bezüglich der Assessmentmöglichkeiten und des Lerneffekts der Applikation sind Bestandteil der Dissertation von Robert Stoyan.

Aufgrund der engen Vernetzung und unvermeidlichen Überschneidungen des Konzepts *Evolutionärer Lernspiele* mit anderen Fachdisziplinen, wie etwa Pädagogik, Computerlinguistik, Psychologie, Informationstechnologie und Projektmanagement sind derzeit zahlreiche Kooperationsstudien, z.B. bezüglich interkultureller Unterschiede, des Einfluss' diverser Persönlichkeitsmerkmale auf den Lerneffekt, der Anwendbarkeit des *PM Game* für -/ Selbst- und Fremdeinschätzungen sozialer Kompetenz sowie Aspekten der Benutzerfreundlichkeit, im Rahmen eines eingereichten EU-Forschungsprojekts geplant. Dieses soll die prinzipielle Eignung des *PM Game* als E-Assessmentwerkzeug, basierend auf dem Evolutionären Lernspielkonzept, überprüfen. Dabei sollen die Grenzen des Konzepts qualitativ und quantitativ bestimmt werden und gegebenenfalls nach neuen Möglichkeiten zur Erweiterung geforscht werden.

Das *Evolutionäre Lernspielkonzept* wurde 2007 von Werner Winkelmann zur Bildung virtueller Gemeinschaften und Evaluation der dazu nötigen Werkzeuge aufgegriffen sowie ein diesbezüglicher Massnahmenkatalog erstellt. Die so gewonnenen Informationen sollen den Einschwingungsprozess *Evolutionärer Lernspiele* vereinfachen und beschleunigen helfen.

6.3 Infrastruktur für *Evolutionäre Lernspiele*

Das Softwaredesign *Evolutionärer Lernspiele* sollte dem dienst-orientierten Architekturansatz folgen, um Dienste von Drittanbietern integrieren zu können. Dies verbessert die Flexibilität des Spiels, welches so nach den Bedürfnissen der Nutzer erweitert und mit etablierten Web2.0-Applikationen kombiniert werden kann. Die Benutzerinformationen und Lerngruppen von Facebook (s. Kapitel 2.1.2) könnten beispielsweise so verwaltet und über ihr *API* synchronisiert werden, d.h. die Schnittstellen zu den wichtigsten Diensten eines *Evolutionären Lernspiels* sollten offen sein, damit *Mashups* kreiert und Synergien genutzt werden können.

Ein weiterer Vorteil *Evolutionärer Lernspiele* als Webapplikationen gegenüber *Desktop-* oder *Standalone-*Applikationen liegt darin, dass clientseitig kein Installationsaufwand entsteht und die Nutzungshemmschwelle dadurch minimiert werden

kann. Ausserdem entfällt die Notwendigkeit von Softwareupdates durch den Nutzer, da diese serverseitig erfolgen können. Dies unterstützt die stetige Erweiterung und Verbesserung der Software.

Um den genannten Anforderungen gerecht zu werden, benötigen *Evolutionäre Lernspiele* eine flexible, wieder verwendbare, modulare Infrastruktur. Eine solche Infrastruktur, die zudem als Basis für Wissensmanagementsysteme und Lernplattformen genutzt werden kann, wird zurzeit am Lehrstuhl für Educational Engineering in Kooperation mit den Firmen getunik AG und trialox AG entwickelt. Im abschliessenden Kapitel Ausblick wird auf diese in Entwicklung befindliche Infrastruktur eingegangen.

7. Ausblick

Wie im Vorgängerkapitel 5 bereits erwähnt, wird derzeit eine Infrastruktur entwickelt – genannt *Semantic Content Infrastructure* (SCI) –, welche als Basis für *Evolutionäre Lernspiele* dienen soll. Dazu wurde das *Resource Description Framework* (RDF) des W3C verwendet. RDF ermöglicht die Beschreibung und eindeutige Identifizierung von Ressourcen (Inhalten). Dadurch können *verteilte Inhalte* kombiniert und in semantische Beziehung gesetzt werden. RDF eignet sich als Grundlage zur dezentralen Wissensbildung und zum einfacheren -austausch und unterstützt die Nutzer auf diesem Weg in ihrem Lernprozess durch adäquateres Auffinden relevanter Informationen.

Mit der Implementierung von SCI wird auf den entstehenden Trend Web 3.0 gesetzt. Web 3.0 erweitert das Web 2.0 um die Möglichkeit, Ressourcen miteinander zu verknüpfen. Dadurch werden diese maschinenverständlich, und das System kann die Nützlichkeit der Informationen für den Nutzer automatisch beurteilen (Reinsch, 2008). Maschinenverständlichkeit bedeutet nicht, dass eine künstliche Intelligenz die Maschinen befähigt, den Menschen zu verstehen (Berners-Lee, 1998). Der Begriff bezieht sich lediglich darauf, dass Maschinen ein definiertes Problem mit definierten Operationen in einer definierten Domäne lösen oder daraus Schlussfolgerungen ziehen können (vgl. Reinsch, 2008).

Die Firma trialox AG entwickelt zu diesem Zweck aktuell ein *open-source*-Framework namens *Smart Content Binding* (SCB), welches eine *Java*TM-Implementation des auf der W3C-RDF-Spezifikation basierenden Graph-Datenmodells zur Verfügung stellt. SCB separiert Applikationen von den darunterliegenden *RDF Stores* (auch *Triple Stores* genannt) und bietet Applikationen eine technologie-unabhängige Dienstschnittstelle für Zugriffe auf RDF-Graphen. Für Operationen auf RDF-Graphen kann SCB unterschiedliche Anbieter (*Providers*) einbinden, welche die Graphen in einer technologiespezifischen Art verwalten. Ein *Sesame Provider*-Modul (vgl. Openrdf.org, 2008) ist bereits implementiert und die Umsetzung weiterer Provider-

Module, z.B. Jena (vgl. Jena.sourceforge.net, 2008) geplant. Ferner stellt SCB einen Jena-Adapter zur Verfügung, welcher existierenden Applikationen basierend auf Jena erlaubt, die Jena-Schnittstelle für Zugriffe auf Graphen zu verwenden. Andere technologie-spezifische Adapter werden je nach Bedarf implementiert. Mit diesem Entwurfsmuster müssen existierende Applikationen mehr nicht angepasst werden, um SCB zu verwenden.

SCI erweitert SCB um die Möglichkeit des sicheren Teilens verteilter semantischer Inhalte. Die Dezentralisierung der Inhaltsverwaltung führt zu einer skalierbaren Architektur und erhöhten Zuverlässigkeit (kein „*single point of failure*“). Um verteilte Ressourcen austauschen zu können, braucht es eine Kommunikationsinfrastruktur mit einer standardisierten Schnittstelle. SCI wird Webdienste einsetzen, um die Vorteile der dienstorientierten Architektur zu nutzen. Eine diesbezüglich sehr wichtige Anforderung ist die Performanz. Durch die Verteilung und Dezentralisierung kann ein *Kommunikationsoverhead* die Performanz einerseits beeinträchtigen, andererseits ermöglicht die verteilte Verarbeitung gewisser Abfragen jedoch auch Performanzgewinn. Deshalb wird SCI in dieser operationellen Umgebung innovative Mechanismen zur Verfügung stellen, um die Performanz des gesamten Systems zu optimieren.

Jedes SCI verwaltet einen oder mehrere interne *Triple Stores* und kann auf die Ressourcen externer *Triple Stores* via anderer SCI zugreifen. Ein SCI stellt eine Reihe allgemeiner Funktionen zur Verfügung, welche von der grossen Mehrheit der Applikationen (wenn nicht sogar von allen) benötigt werden. Diese Funktionen können als Grundlagen von Lernplattformen angesehen werden:

User Manager: verwaltet die Nutzer und deren Profil

- Authentication and Authorization: zuständig für Benutzerauthentifizierung und Zugriffskontrolle
- Digital Asset Manager: verwaltet die Ressourcen
- Smart Content Binding: stellt eine generische Zugangsschicht zu den *Triple Stores* zur Verfügung
- Triple Stores: speichern Ressourcen und ihre Verknüpfungen (in Form von RDF-Graphen)
- Content Subscription: ermöglicht das Abonnieren von Ereignissen über Änderungen in internen Graphen

- External Content Access: ermöglicht den Zugang zu externen *Triple Stores* via SCI
- RESTful Web Service Framework (RWSF): stellt den RESTful Web-basierten Kommunikationsdienst zur Verfügung, mit dem Webanfragen empfangen und die Antworten zurückgesendet werden können
- Application Specific Module: implementiert Applikationslogik, deren Dienste via Web aufgerufen werden können

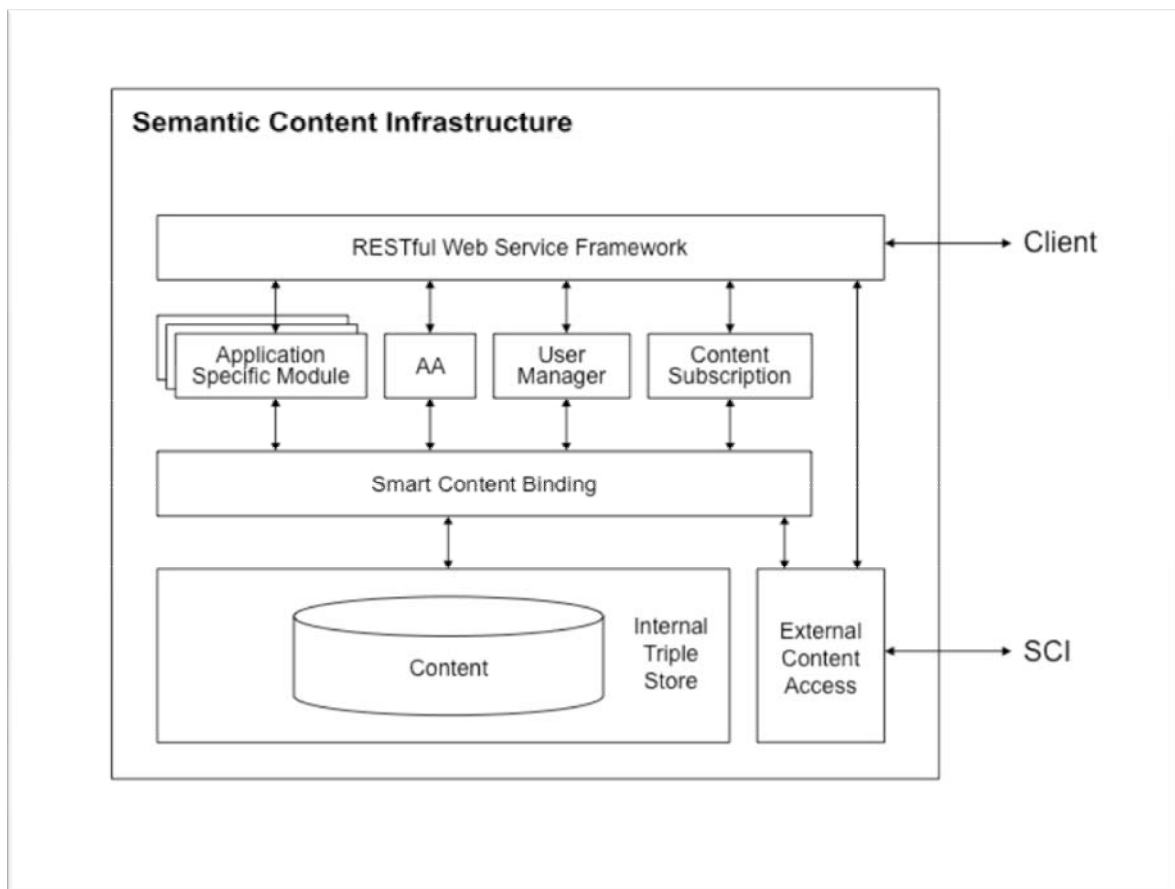


Abb. 37: Semantic Content Infrastructure

Die einzelnen Module des *Evolutionären Lernspiels* werden als *Application Specific Modules* implementiert.

Eine Webanfrage kann durch einen Webclient (*Browser*) oder einen SCI erfolgen. Die Anfrage wird von RWSF an das als dafür zuständig definierte Modul weitergeleitet. Dieses verarbeitet die Anfrage und liefert die Antwort via RWSF (s. Abb. 37).

Mit dieser Infrastruktur soll eine standardisierte Basis für *Evolutionäre Lernspiele* entwickelt werden, welche Funktionen des Web 2.0 und Web 3.0 zur Verfügung stellt und somit benutzergenerierte Inhalte, *Folksonomie*, Taxonomie sowie die Beschreibung und Verknüpfung von Inhalten kombiniert.

Prototyp *PM Game*

Das *PM Game* soll im Rahmen eines im Programm *Future and Emerging Technologies* beantragten EU-Forschungsprojekts zu einem E-Assessment-Werkzeug erweitert werden. Zudem soll der Prototyp zu einem Produkt weiterentwickelt werden, welches als kombiniertes E-Assessment- und Lernspiel an Hochschulen und in anderen Organisationen eingesetzt werden kann. Zur Möglichkeit der internationalen Nutzung soll das *PM Game* mehrsprachig werden und kulturelle Unterschiede berücksichtigen. Dafür werden im geplanten EU-Forschungsprojekt Studien durchgeführt, welche das *Evolutionäre Lernspiel* hinsichtlich seiner Parametrisierungsmöglichkeiten auf kulturelle und geschlechtsspezifische Unterschiede untersuchen sollen. Weitere Forschungsarbeiten betreffen die Ausweitung der Lerninhalte des *PM Game* auf andere Kompetenzgebiete. In Frage kommen hier u.a. Jura (Verhandlungen im Gerichtssaal) und Didaktik (Handlungen in einem Klassenraum)

Da das *PM Game* im Rahmen einer Lehrveranstaltung eingesetzt wurde, fiel die Bereitschaft der Nutzer, längerfristig mit dem Lernspiel zu arbeiten, eher gering aus. Da so keine Bindung zwischen Nutzer und Spiel erzeugt werden konnte, welche die Nutzer zur Inhaltsgenerierung oder aktiven Partizipation in der virtuellen Gemeinschaft hätte motivieren können, muss die virtuelle Gemeinschaft im Rahmen des geplanten Forschungsprojektes ausgebaut und unterstützt werden, damit das Konzept des *Evolutionären Lernspiels* seine volle lernförderliche Wirkung entfalten kann. Dies soll erreicht werden, indem nicht nur Studierende der Universität Zürich, sondern auch Personen anderer Hochschulen Zugang zum *PM Game* erhalten und mit dem Spiel im Rahmen ihres Studiums über mehrere Semester spielen können. Als Nebeneffekt der so vermuteten, längerfristigen Bindung können die Profile der Nutzer analysiert und als Grundlage für Empfehlungen an andere Nutzer ähnlichen Profils genutzt werden.

Bezüglich seiner Benutzerfreundlichkeit soll das *PM Game* zukünftig eine weitere Schnittstelle erhalten, welche die natürlichsprachige Interaktion der Nutzer mit dem System erlaubt. Für definierte Themengebiete sollen die Nutzer dem System Anfragen oder Aussagen über Mikrofon oder per Texteingaben übermitteln.

Die Universität Darmstadt arbeitet an Methoden, um Inhalte automatisiert vom System erstellen zu lassen. Basierend auf Regeln und bestehenden Inhalten könnten dadurch Szenarien für das *PM Game* automatisch kreiert und ihre Diversifikation erhöht werden.

Literaturverzeichnis

- [Adams, 1999] Adams, C. (1999). The use of computer games as an educational tool: identification of appropriate game types and game elements. *British Journal of Educational Technology*, 30(4), 311–321.
- [Adobe.com, 2009] adobe.com (2009). *Adobe Flash Version Penetration*. Retrieved February 14, 2009, from adobe.com: http://www.adobe.com/products/player_census/flashplayer/version_penetration.html
- [Ads.youtube.com 2008] ads.youtube.com (2008). *YouTube Sponsored Videos*. Retrieved February 14, 2009, from ads.youtube.com: <https://ads.youtube.com>
- [Adelsberger et al., 2000] Adelsberger, H., Bick, M., & Pawlowski, J. (2000). Design principles for teaching simulation with explorative learning environments. In: *Proceedings of Winter Simulation Conference*, pages 63–70.
- [Ajzen & Fishbein, 2000] Ajzen, I. & Fishbein, M. (2000). Attitudes and the attitude-behavior relation: Reasoned and automatic processes. *European Review of Social Psychology*, 10, 1–33.
- [Alby, 2008a] Alby, T. (2008). *Web 2.0*. München: Hanser Verlag.
- [Alby, 2008b] Alby, T. (2008). *Web 2.0. Konzepte, Anwendungen, Technologien*. München: Hanser Verlag.
- [Anderson, 2004] Anderson, C. (2004). *The Long Tail*. Retrieved September 2, 2006, from wired.com: <http://www.wired.com/wired/archive/12.10/tail.html>
- [Angermeier, 2005] Angermeier, M. (2005). *TagCloud Web 2.0*. Retrieved January 11, 2009, from kosmar.de: <http://kosmar.de/archives/2005/11/11/the-huge-cloud-lens-bubble-map-web20/>
- [Anthony et al., 2005] Anthony, D., Smith, S. W. & Williamson, T. (2005). *Explaining quality in internet collective goods: Zealots and good samaritans in the explaining quality in internet collective goods: Zealots and good samaritans in the case of wikipedia*. Technical report, Department of Computer Science, Hanover: Dartmouth College.
- [Aronson, 2004] Aronson, E., Wilson, T. & Akert, R. (2004). *Sozialpsychologie* (4. Aufl.). München: Pearson Studium.
- [Austin, 2005] Austin, J. L. (2005). *How to Do Things With Words*. Harvard University Press.
- [B.A., 2002] B.A., ANONYM (2002). IT-Know How, IT-Businessinformationen für die IT-Berufe IT-, ITS-Kaufmann /-frau. iPaper.
- [Barsalou, 1999] Barsalou, L. W. (1999). Language comprehension: Archival memory or preparation for situated action. *Discourse Process*, 28, 61–80.
- [Barwise and Etchemendy, 2005] Barwise, J. & Etchemendy, J. (2005). *Band 1: Aussagen- und Prädikatenlogik*. Mentis.
- [Bateman & Boon, 2005] Bateman, C. & Boon, R. (2005). *21st Century Game Design. Game Development Series*. Charles River Media.
- [Baumgartner, 2004] Baumgartner, P. (2004). *Content Management Systeme in e-education*. Innsbruck: StudienVerlag.
- [Baumgartner et al., 2002] Baumgartner, P., Häfele, H. & Meyer-Häfele, K. (2002). *E-*

Learning Praxishandbuch. Auswahl von Lernplattformen. Innsbruck: Studien Verlag.

- [Baur & Kolo, 2002] Baur, T. & Kolo, C. (2002). Feldforschung in multiplayer-onlinespielen. *Institution: Ludwig-Maximilians Universität München Article.*
- [Beck, 2000] Beck, K. (2000). *Extreme Programming. Das Manifest.* Addison-Wesley.
- [Bergeron, 2006] Bergeron, B. (2006). *Developing Serious Games.* Charles River Media.
- [Berners-Lee, 1998] Berners-Lee, T. (1998), *What the Semantic Web can represent.* Retrieved September 12, 2007, w3.org: <http://www.w3.org/DesignIssues/RDFnot.html>.
- [Berners-Lee, 1999] Berners-Lee, T. (1999). *Weaving the Web.* San Francisco: Harper.
- [Björk & Holopainen, 2005] Björk, S. & Holopainen, J. (2005). *Patterns in Game Design.* Thomson Learning.
- [Bopp, 2006] Bopp, M. (2006). *Didactic Analysis of Digital Games and Game-Based Learning.* In: M. Pivec (ed.): Affective and Emotional Aspects of Human-Computer Interaction. Game-Based and Innovative Learning Approaches. Amsterdam: IOS Press.
- [Boles, 1998] Boles, D. (1998). Kompositionsphase. Retrieved October 30, 2008, from [www-is.informatik.uni-oldenburg.de](http://www-is.informatik.uni-oldenburg.de/~dibo/teaching/pgmpig/endbericht-b/node50.html): <http://www-is.informatik.uni-oldenburg.de/~dibo/teaching/pgmpig/endbericht-b/node50.html>
- [Bremer, 2006] Bremer, C. (2006). Online lehren leicht gemacht. Leitfaden für die Planung und Gestaltung von virtuellen Hochschulveranstaltungen. Retrieved February 13, 2006 from [bremer.cx](http://www.bremer.cx/paper13/artikelraabe_bremer03.pdf): http://www.bremer.cx/paper13/artikelraabe_bremer03.pdf
- [Brin & Page, 1999] Brin, S. & Page, L. (1999). *The anatomy of a large-scale hypertextual web search engine.* Technical report, Computer Science Department, Stanford University, Stanford, CA 94305, USA Computer Science Department, Stanford University, Stanford.
- [Brooks, 1986] Brooks, R. (1986). A robust layered control system for a mobile robot. *EEE Journal of Robotics and Automation*, 236–243.
- [Brown et al. 1989] Brown, A. L., Collins, A. & Duguid (1989). Situated cognition and the culture of learning. *Educational Res.* 18, 32–42.
- [Bürg, 2005] Bürg, O. (2005). *Akzeptanz von E-Learning in Unternehmen.* PhD thesis, Ludwig-Maximilians-Universität München.
- [Bürg & Mandl, 2005] Bürg, O. & Mandl, H. (2005). Akzeptanz von e-learning in unternehmen. *Zeitschrift für Personalpsychologie*, 4 (2), 75–85. Göttingen: Hogrefe.
- [Burkhard, 2008] Burkhard, S. (2008). User-contributed business logic. Master's thesis, Departement of Informatics, University of Zurich.
- [Businessshaker.com, 2007] businessshaker.com (2007). *10,000 Facebook Applications, How Many Do You Use?* Retrieved June 11, 2008, from [businessshakers.com](http://businessshakers.com/internet-marketing-articles/10000-facebook-applications-how-many-do-you-use/): <http://businessshakers.com/internet-marketing-articles/10000-facebook-applications-how-many-do-you-use/>
- [Čadež, 2006] Čadež, B. (2006). *Line Rider.* Retrieved February 11, 2007, from linerider.com: <http://linerider.com/>
- [Carl et al., 2008] Carl, D., Clausen, J., Hassler, M. & Zund, A. (2008). *Mashups programmieren.* O'Reilly Media.
- [Charles et al., 2005] Charles, D., McNeill, M., McAlister, M., Black, M., Moore, A., Stringer,

- K., Kücklich, J. & Kerr, A. (2005). Player-centred game design: Player modelling and adaptive digital games. In *Changing Views – Worlds in Play*. Proceedings of DiGRA Conference.
- [Clark, 1983] Clark, R.E. (1983). Reconsidering Research on Learning from Media. *Review of Educational Research*, 53 (4), 445–459.
- [Clark, 2007] Clark, R.E. (2007). Learning from Serious Games? – Arguments, Evidence and Research Suggestions, Point of View. *Educational Technology – The Magazine for Managers of Change in Education*, 47 (3), 56–59.
- [Cordova & Lepper, 1996] Cordova, D. I. & Lepper, M. R. (1996). Intrinsic motivation and the process of learning: Beneficial effects of contextualization, personalization, and choice. *Journal of Educational Psychology*, 88, 715–730.
- [Crawford, 1984] Crawford, C. (1984). *The Art of Computer Game Design*. McGraw-Hill Osborne Media.
- [Csikszentmihalyi & Csikszentmihalyi, 1988] Csikszentmihalyi, M. & Csikszentmihalyi, I. (1988). *Optimal experience*. Cambridge: University Press.
- [Csikszentmihalyi et al., 1993] Csikszentmihalyi, M., Rathund, K. & Whalen, S. (1993). *Talented teenagers*. Cambridge: University Press.
- [Dale, 1969] Dale, E. (1969). *Audio-Visual Methods in Teaching*. International Thomson Publishing.
- [DaRin, 2005] DaRin, D. (2005). E-learning aus Mitarbeitersicht. Autonomie und Verlust. *Das Magazin für integrierte Managementsysteme*. Business Excellence.
- [Davis et al., 1989] Davis, F. D., Bagozzi R. P. & Warshaw, P. R. (1989). User acceptance of computer technology: A comparison of two theoretical models, *Management Science*, 35, 982-1003.
- [Deutsches Bundesministerium für Bildung und Forschung, 2007] Deutsches Bundesministerium für Bildung und Forschung. (2007). *Presse*. Retrieved October 16, 2008, from Deutsches Bundesministerium für Bildung und Forschung: <http://www.bmbf.de/press/1959.php>
- [Developers.facebook.com, 2009] developers.facebook.com (2009). *Build social applications on Facebook Platform*. Retrieved February 22, 2009, from developers.facebook.com: <http://developers.facebook.com/>
- [Diamond, 1988] Diamond, M. (1988). *Enriching Heredity. The Impact of the Environment on the Anatomy of the Brain*. The Free Press.
- [Dobusch & Forsterleitner, 2009] Dobusch, L. & Forsterleitner, C. (2009). *Freie Netze. Freies Wissen. iPaper*.
- [Diller et al., 2004] Diller, D. E., Ferguson, W., Leung, A. M., Benyo, B. & Foley, D. (2004). Behavior modeling in commercial games. BBN Technologies.
- [Drosdowski, 1989] Drosdowski, G. (1989): *Duden Etymologie : Herkunftswörterbuch der deutschen Sprache*. Mannheim: Dudenverlag.
- [eLearn jku, 2006] elearn.jku.at (2006) *Gestalten und Evaluieren von eLearning Szenarien/Autorenwerkzeuge zur Erstellung digitaler Inhalte für e-learning*. Retrieved September 11, 2008, from elearn.jku.at: http://elearn.jku.at/wiki/index.php/Gestalten_und_Evaluieren_von_eLearning_Szenarien/Autorenwerkzeuge_zur_Erstellung_digitaler_Inhalte_für_e-learning
- [Ellison et al., 2007] Ellison, N. B., Steinfield, C. & Lampe, C. (2007). The benefits of facebook

- "friends:" social capital and college students' use of online social network sites. *Journal of Computer-Mediated Communication*, 12.
- [Engeser et al., 2003] Engeser, S., Rheinberg, F., Vollmeyer, R. & Bischoff, J. (2003). Motivation, flow-experience, and performance in learning settings at universities. *German Journal of Educational Psychology*, pages 159–172.
- [Encyclopedia Britannica, 2003] Encyclopedia Britannica, E. B. (2003). *Encyclopedia Britannica*. Akademischer Lexikadienst.
- [Erl, 2008] Erl, T. (2008). *SOA: Entwurfsprinzipien für service-orientierte Architektur*. München: Addison-Wesley.
- [e-teaching.org, 2008] e-teaching.org (2008). *Drag & Drop*. Retrieved December 11, 2008, from e-teaching.org: <http://www.e-teaching.org/glossar/drag-drop>
- [Evans, 2007] Evans, A. (2007). *Little Big Planet*. E3 2007. Retrieved February 22, 2009, from golem.de <http://www.golem.de/0707/53494.html>
- [Facebook.com 2009] facebook (2009). *Press Room*. Retrieved February 22, 2009, from facebook.com: <http://www.facebook.com/press/info.php?statistics>
- [Faz.net, 2007] faz.net (2007). Retrieved January 23, 2009, from faz.net: http://www.faz.net/s/RubE2C6E0BCC2F04DD787CDC274993E94C1/Doc~EED8B834A359F4072A82E56C570D02C46~ATpl~Ecommon~Scontent.html?rss_googlefeed
- [Fikentscher & Hlava-Landeck, 2005] Fikentscher, K. & Hlava-Landeck, K. (2005). *trendbook e-learning 2005/06*. Haufe Fachmedia GmbH & Co. KG.
- [Foerster, 2004] Foerster, K. (2004). *Personalisiertes E-Learning in Unternehmen: Anforderungen an die Ausgestaltung Web-basierter Lerneinheiten im Hinblick auf die Wirksamkeit und die Effizienz des Lernprozesses*. Cuvillier Verlag.
- [Fulghum, 1986] Fulghum, R. (1986). *All I Really Need to Know I Learned in Kindergarten*. Ivy Books.
- [Funderstanding.com, 2008] funderstanding.com (2008). Constructivism. Retrieved June 21, 2008, from funderstanding.com: <http://www.funderstanding.com/content/constructivism>
- [Freeman et al., 2004] Freeman, E., Freeman, E., Bates, B. & Sierra, K. (2004). *Head First Design Patterns*. O'Reilly Media.
- [Froidevaux & Täube, 2006] Froidevaux, Y. & Täube, V. G. (2006). Internetnutzung in den Haushalten der Schweiz, Ergebnisse der Erhebung 2004 und Indikatoren. Technical report, Bundesamt für Statistik.
- [Gamma et al., 1995] Gamma, E., Helm, R. & Johnsons, R. E. (1995). *Design Patterns. Elements of Reusable Object-Oriented Software*. Amsterdam: Addison-Wesley Longman.
- [Garris et al., 2002] Garris, R., Ahlers, R. & Driskell, J. E. (2002). Games, motivation, and learning: A research and practice model. *Simulation & Gaming*.
- [Gee, 2003] Gee, J. (2003). *What Video Games Have To Teach Us About Learning And Literacy*. New York: Palgrave Macmillan.
- [Gilleade et al., 2005] Gilleade, K., Dix, A. & Allanson, J. (2005). Changing views – worlds in play. In *Affective Videogames and Modes of Affective Gaming: Assist Me, Challenge Me, Emote*. Proceedings of DiGRA 2005 Conference.
- [Glenberg & Robertson, 1999] Glenberg, A. M. & Robertson, D. A. (1999). Indexical understanding of instructions. *Discourse Process*, 28, 1–26.

- [Google Press Center, 2009] Google Press Center (2009). *Zeitgeist Archive*. Retrieved February 22, 2009, from google.com: <http://www.google.com/press/zeitgeist/weeks-oct06.html>
- [Gosh & Prakash, 2000] Ghosh, R. & Prakash, V. V. (2000). *The Orbiteen free software survey*. Retrieved from First Monday 5 (7): http://firstmonday.org/issues/issue5_7/ghosh/.
- [Grace, 2005] Grace, L. (2005). *Game Type and Game Genre*. Retrieved February 22, 2009, from http://aii.lgrace.com/documents/Game_types_and_genres.pdf
- [Greenfield, 1984] Greenfield, P. M. (1984). *Mind and Media: The Effects of Television, Video Games, and Computers*. Cambridge: Harvard University Press.
- [Haefele & Maier-Haefele, 2003] Haefele, H., & Maier-Haefele, K. (2003). *Studie Autorenwerkzeuge für Learning Content*. Retrieved November 12, 2005, from [bildungsserver.de: http://www.bildungsserver.de/db/mlesen.html?Id=19767](http://www.bildungsserver.de/db/mlesen.html?Id=19767)
- [Holiday et al., 2007] Holiday, G., Statler, M., & Flanders, M. (2007). Developing practically wise leaders through serious play. *Consulting Psychology Journal: Practice and Research*, 59, 126–134.
- [Homuth, 1977] Homuth, H. H. (1977). *Einführung in die Automatentheorie für Studenten der Mathematik, Informatik, Natur- und Ingenieurwissenschaften*. Braunschweig: Vieweg.
- [Ito, 2005] Ito, T. (2005). *Integration einer benutzerschnittstelle in GoaCore*. Master's thesis, Universität Zürich.
- [Jena.sourceforge.net, 2008] Jena.sourceforge.net (2008). *Jena*. Retrieved June 11, 2008, from [jena.sourceforge.net: http://jena.sourceforge.net](http://jena.sourceforge.net)
- [Jenkins, 2006] Jenkins, H. (2006). *Convergence Culture: Where Old and New Media Collide. Fluency at After-School Centers in Economically- Convergence Culture*. New York: University Press.
- [Kay, 2006] Kay, A. (2006). Retrieved February 22, 2009, from www.uwsp.edu/Education/kbuchana/presentations/taming_tempest_2007_11_09/index.htm
- [Keller, 1983] Keller, J. M. (1983). *Motivational design of instruction*. In: C. M. Reigeluth (Ed.). Instructional-design theories and models: An overview of their current status (pp. 383–434). Hillsdale, NJ: Lawrence Erlbaum.
- [Kirkpatrick, 2006] Kirkpatrick, M. (2006). *Youtube serves 100m videos each day*. Retrieved January 13, 2009, from [techcrunch.com: http://www.techcrunch.com/2006/07/17/youtube-serves-100m-videos-each-day/](http://www.techcrunch.com/2006/07/17/youtube-serves-100m-videos-each-day/)
- [Kirkpatrick, 2007] Kirkpatrick, M. (2007). *Facebook Apps Hit 10k in Number: Are We Still Excited?* Retrieved January 11, 2009, from [readwriteweb.com: http://www.readwriteweb.com/archives/facebook_apps_hit_10k_in_numbe.php](http://www.readwriteweb.com/archives/facebook_apps_hit_10k_in_numbe.php)
- [Koch & Richter, 2007] Koch, M. & Richter, A. (2007). *Planung, Einführung und erfolgreicher Einsatz von Social Software in Unternehmen*. Oldenbourg.
- [Kolb, 1981] Kolb, D. A. (1981). *Learning Styles and Disciplinary Differences*. In: A.W. Chickering (Hrsg.), The Modern American College. San Francisco: Jossey-Bass.
- [Kolb & Kolb, 1976] Kolb, A. Y. & Kolb, D. A. (1976). *The Learning Style Inventory: Technical Manual*. Case Western Reserve University.

- [Kollmann, 1998] Kollmann, T (1998). *Akzeptanz innovativer Nutzungsgüter und -systeme. Konsequenzen für die Einführung von Telekommunikations- und Multimediasystemen*. Wiesbaden: Gabler.
- [Koster & Wright, 2005] Koster, R. & Wright, W. (2005). *A Theory of Fun for Game Design*. Paraglyph Inc.,US.
- [Kriens & Hargrave, 2004] Kriens, P. & Hargrave, B. J. (2004). Whiteboard. Technical report, OSGi Alliance.
- [Küster, 2008] Küster, B. (2008). "Little Big Planet" macht den Spieler zum Spielmacher. Retrieved January 22, 2009, from tagesspiegel.de: <http://www.tagesspiegel.de/medien-news/digital/Little-Big-Planet;art303,2655849>
- [k-videogames.de] k-videogames.de (2008). *Sack and Pack*. Retrieved December 12, 2008, from k-videogames.de: http://www.k-ideogames.de/web/Little_Big_Planet/preview/id,9881/s,c
- [Lakhani & von Hippel] Lakhani, K. & von Hippel, E. (2002). How open source software works: "free" user-to-user assistance. *Research Policy*, 32, 923–43.
- [Lakhani & Wolf, 2005] Lakhani, K. R. & Wolf, R. G. (2005). *Why hackers do what they do: Understanding motivation and effort in free/open source software projects*. In Perspectives on free and open source software. Cambridge, MA: MIT Press.
- [Lancaster & Stillman, 2003] Lancaster, L. C. & Stillman, D. (2003). *When Generations Collide: Who They Are, why They Clash, how to Solve the Generational Puzzle at Work*. HarperCollins World.
- [Lanier, 2006] Lanier, J. (2006). Das so genannte Web 2.0; Digitaler Maoismus; Kollektivismus im Internet, Weisheit der Massen, Fortschritt der Communities? Alles Trugschlüsse. sueddeutsche.de.
- [Lerner & Tirole, 2002] Lerner, J. & Tirole, J. (2002). Some simple economics of open source. *Journal of Industrial Economics*, 2, 197–234.
- [Lih, 2004] Lih, A. (2004). Wikipedia as participatory journalism: Reliable sources? In *5th International Symposium on Online Journalism*, University of Texas, Austin.
- [Machinima.org, 2002] machinima.org. (2002). Academy of Machinima arts and sciences. Retrieved February 21, 2009 from machinima.org: <http://www.machinima.org>
- [Malone, 1980] Malone, T. W. (1980). What makes things fun to learn? heuristics for designing instructional computer games. In *SIGSMALL '80: Proceedings of the 3rd ACM SIGSMALL symposium and the first SIGPC symposium on Small systems*, pp. 162–169, New York: ACM Press.
- [Maloney et al., 2003] Maloney, J., Burd, L., Kafai, Y., Rusk, N., Silverman, B., & Resnick, M. (2003) Scratch: A sneak preview.
- [Mandl & Krause, 2001] Mandl, H. & Krause, U.-M. (2001). *Making learning fun: A taxonomy of intrinsic motivations for learning*. Technical report, Ludwig-Maximilians-Universität, Lehrstuhl für Empirische Pädagogik und Pädagogische Psychologie.
- [Marte et al., 2004] Marte, A., Schneider, U. & Schauer, H. (2004): Transmediales Kommunikationswissen: Eine Anleitung zum Unglücklichsein in Zehn Geboten. *Lernende Organisation*, 17, 28–35.

- [Mathieu et al. 1992] Mathieu, J. E., Tannenbaum, S. I. & Salas, E. (1992). Influences of individual and situational characteristics on measures of training effectiveness. *Academy of Management Journal*, 35, 828–847.
- [Matthias, 2008] Matthias, A. (2008). <http://de.wikipedia.org/wiki/Lernmethode>. Retrieved December 12, 2008, from Wikipedia: <http://de.wikipedia.org/wiki/Lernmethode>
- [Mayer, 2007] Mayer, R. (2007). *Problem-Solving Assessment in Game Environments*. Proceedings of the 2007 AERA Annual Meeting and Exhibition: The World of Educational Quality, Chicago.
- [McCloud, 1994] McCloud, S. (1994). *Understanding Comics: The Invisible Art*. Harpercollins.
- [Meier & Seufert, 2003] Meier, C. & Seufert, S. (2003). *Game-based learning: Erfahrungen mit und Perspektiven für digitale Lernspiele in der betrieblichen Bildung*. In: E-Learning. Deutscher Wirtschaftsdienst.
- [Michael & Chen, 2005] Michael, D. & Chen, S. (2005). *Serious Games: Games That Educate, Train, And Inform*. Course Technology.
- [Mmog-welt.de, 2008] mmog-welt.de (2008). *World of Warcraft – 11,5 Millionen Spieler weltweit*. Retrieved January 11, 2009, from mmog-welt.de: <http://www.mmog-welt.de/onlinegames/world-of-warcraft-115-millionen-spieler-weltweit>
- [Mödritscher et al., 2004] Mödritscher, F., Garcia-Barrios, V.M. & Gütl, C. (2004). The past, the present and the future of adaptive e-learning – an approach within the scope of the research project adele. In *In Proceedings of ICL, Austria, 2004*.
- [Moore & Benbasat, 1991] Moore, C. G. & Benbasat, I. (1991). Development of an instrument to measure the perceptions of adopting an information technology innovation. *Information Systems Research*, 2, 192-222.
- [Mprem.com, 2008] mprem.com. (2008). Celebrating Machinima. Retrieved February 21, 2009, from mprem.com: <http://www.mprem.com>
- [Müller, 2004] Müller, Ch. (2004). Going international – working internationally. LIMAK News.
- [Müller-Böling & Müller, 1986] Müller-Böling, D. & Müller, M. (1986). *Akzeptanzfaktoren der Bürokommunikation*. Oldenbourg.
- [Neff & Stark 2003] Neff, G. & Stark, D. (2003). *Permanently Beta: Responsive Organization in the Internet Era*. In: Howard and S. Jones (eds., 2002) P.E.N., The Internet and American Life. Thousands Oaks: Sage.
- [Nievergelt, 1999] Nievergelt, J. (1999). *Algorithmen und Datenstrukturen*. In P. Rechenberger & G. Pomberger (Hrsg.), Informatik-Handbuch, (pp. 425–467). München: Hanser.
- [Ohlsson, 1994] Ohlsson, S. (1994). *Constraint-based student modeling*. In: J.E. Greer & G. McCalla (eds), Student modeling: the key to individualized knowledge-based instruction (167–189).
- [OLAT, 2008] OLAT (2008). OLAT. Retrieved December 8, 2008, from www.elc.uzh.ch: <http://www.elc.uzh.ch/toolfinder/olat.html>
- [O'Mahony, 2003] O'Mahony, S. (2003). Guarding the Commons: How Community-managed Software Projects Protect Their Work, Research Policy, forthcoming.
- [Openrdf.org, 2008] openrdf.org (2008), *Aduna Open Source*. Retrieved February 12, 2009, from openrdf.org: <http://openrdf.org/>.

- [O'Reilly, 2007] O'Reilly, T. (2007). *What is Web 2.0: Design Patterns and Business Models for the Next Generation of Software*. O'Reilly Media.
- [OSGi Alliance, 2008] OSGi Alliance (2008). *OSGi*. Retrieved October 22, 2008, from osgi.org: <http://www.osgi.org/>
- [OSOA, 2008] OSOA (2008). *Open Service Oriented Architecture*. Retrieved December 12, 2008, from osoa.org: <http://www.osoa.org>
- [Papert, 1993] Papert, S. (1993). *The Children's Machine: Rethinking School in the Age of the Computer*. Basic Books.
- [Papert, 1998] Papert, S. (1998). *Does Easy Do It? Children, Games, and Learning*. Retrieved February 20, 2009, from papert.org: <http://www.papert.org/articles/Doeseasydoit.html>
- [Paras & Bizzocchi, 2005] Paras, B. & Bizzocchi, J. (2005). Game, motivation, and effective learning: An integrated model for educational game design. In *Changing Views – Worlds in Play*. DiGRA.
- [Pfeifer & Scheier, 2001] Pfeifer, R. & Scheier, C. (2001). *Understanding Intelligence*. MIT Press.
- [Pfeil, 2007] Pfeil, C. (2007). *The Long Tail*. Retrieved December 2, 2008, from www.christianpfeil.com: <http://www.christianpfeil.com/the-long-tail/>
- [Piaget, 1973] Piaget, J. (1973). *To Understand Is To Invent*. New York: The Viking Press, Inc.
- [Piaget, 2003] Piaget, J. (2003). *Nachahmung, Spiel und Traum*. Stuttgart: Klett-Cotta.
- [Pohl, 2006] Pohl, G. (2006). *The Long Tail: Das dünne Ende der Ladenhüter*. Retrieved December 7, 2008, from [spiegel.de](http://www.spiegel.de): <http://www.spiegel.de/netzwelt/web/0,1518,447490,00.html>
- [Preece et al., 2007] Preece, J., Rogers, Y., & Helen, S. (2007). *Interaction Design: Beyond Human-computer Interaction*. Wiley & Sons.
- [Prensky, 2001] Prensky, M. (2001). *Digital Game-Based Learning*. McGraw-Hill.
- [Prensky, 2003] Prensky, M. (2003). Digital game-based learning. *Comput. Entertain.*, 1 (1), 21.
- [Prensky & Gee, 2006] Prensky, M. & Gee, J. (2006). *Don't Bother Me, Mom – I'm Learning!* Paragon House Publishers.
- [Prestipino, 2008] Prestipino, M. (2008). *Die virtuelle Gemeinschaft als Informationssystem Informationsqualität nutzergenerierter Inhalte in der Domäne*. PhD thesis, Department of Informatics, University of Zurich.
- [PriceWaterHouseCoopers, 2006] PriceWaterhouseCoopers. (2006). *Medienbranche mit guten Perspektiven bis 2010: Digitalisierung schafft robustes Wachstum*. Retrieved October 11, 2008, from [presseportal.de](http://www.presseportal.de): <http://www.presseportal.de/meldung/838325/>
- [pureMVC, 2008] pureMVC (2008). *pureMVC*. Retrieved October 22, 2008, from puremvc.org: <http://puremvc.org>
- [Quinn, 2005] Quinn, C. N. (2005). *Engaging Learning: Designing E-learning Simulation Games*. Pfeiffer & Company.
- [Reich et al., 2005] Reich, K., Sehnbruch, L. & Wil, R. (2005). *Medien und Konstruktivismus. Eine Einführung in die Simulation als Kommunikation*. Münster: Waxmann.
- [Reich, 2002] Reich, K. (2002). *Konstruktivistische Didaktik*. Neuwied: Luchterhand.
- [Reinsch, 2008] Reinsch, H. (2008). *Semantic Web*. Retrieved December 11, 2008, from www.helge.reinsch.de: <http://www.helge-reinsch.de/ki2/semanticweb.htm>

- [Resnick, 2007] Resnick, M. (2007). *All I Really Need to Know (About Creative Thinking) I Learned (By Studying How Children Learn) in Kindergarten*, Proceedings of the 6th ACM SIGCHI conference on Creativity & Cognition, Wahsington, DC, pp. 1–6.
- [Resnick et al., 2003] Resnick, M., Kafai, Y., Maeda, J., Rusk, N. & Maloney, J. (2003). A networked, media-rich programming environment to enhance technological fuency at after-school centers in economically-disadvantaged communities. Proposal to the National Science Foundation.
- [Resnick, 1998] Resnick, M. (1998). Technologies for lifelong kindergarten. *Educational Technology Research and Development*.
- [RFC2616, 1999] RFC2616. (1999). The Hypertext Transfer Protocol. Retrieved October 14, 2008, from w3.org: <http://www.w3.org/Protocols/rfc2616/rfc2616.html>
- [Ricci et al., 1996] Ricci, K., Salas, E., & Cannon-Bowers, J. A. (1996). Do computer-based games facilitate knowledge acquisition and retention? *Military Psychology*, 8(4), 295–307.
- [Rosch & Frey, 1997] Rosch, M. & Frey, D. (1997). Soziale Einstellungen. In: D. Frey & S. Greif (Hrsg.), *Sozialpsychologie. Ein Handbuch in Schlüsselbegriffen*. (S. 296–305). Weinheim: Beltz.
- [Roussou, 2004] Roussou, M. (2004). Learning by doing and learning through play: an exploration of interactivity in virtual environments for children. *Comput. Entertain.*, 2 (1), 10.
- [Rheingold, 1993] Rheingold, H. (1993). *The Virtual Community - Homesteading on the Electronic Frontier*. Jackson : Perseus Books.
- [Rollings and Adams, 2003] Rollings, A. & Adams, E. (2003). *Andrew Rollings and Ernest Adams on Game Design*. New Riders.
- [Salen & Zimmerman, 2003] Salen, K. & Zimmerman, E. (2003). *Rules of Play: Game Design Fundamentals*. MIT Press.
- [Sawyer, 2006] Sawyer, R.K. (2006). *Education for Innovation. Thinking Skills and Creativity*. Department of Education, Washington University in St. Louis
- [Schafer et al., 1999] Schafer, J. B., Konstan, J. & Riedl, J. (1999). Recommender systems in e-commerce. In *Electronic Commerce*. ACM.
- [Schneider, 1996] Schneider, U. (1996): *Multikulturelles Management*. Manuskript am Institut für Internationales Management, Karl-Franzens-Universität.
- [Schnell et al., 1999] Schnell, R., Hill, P. & Esser, E. (1999). *Methoden der empirischen Sozialforschung*. Oldenbourg.
- [SCORM, 2006] SCORM (2006). *SCORM - Shareable Content Object Reference Model*. Retrieved June 5, 2007, from www.scormsoft.com: <http://www.scormsoft.com/scorm>
- [Searle, 1969] Searle, J. (1969). *Speech Acts*. Cambridge: University Press.
- [Segaran, 2007] Segaran, T. (2007). *Programming Collective Intelligence: Building Smart Web 2.0 Applications*. O'Reilly Media.
- [Shaffer, 2006] Shaffer, D. (2006). *How Computer Games Help Children Learn*. New York: Palgrave Macmillan.
- [Shaffer et al., 2005] Shaffer, D., Squire, K., Halverson, R. & Gee, J. (2005). Video games and the future of learning. Working Paper No. 2005–4.

- [Simon, 2001] Simon, B. (2001). *E-Learning an Hochschulen. Gestaltungsräume und Erfolgsfaktoren von Wissensmedien*. Köln: Josef Eul Verlag.
- [Spring-Keller, 2006] Spring-Keller, F. (2006). Adaptive game-based learning. In *Proceedings of the IFIP Workgroup 3.1, 3.3, 3.5 Joint Conference 2006*.
- [Spring-Keller & Ito, 2007] Spring-Keller, F. & Ito, T. (2007). Combining Personalisation and Adaptation in Game-Based Learning Systems, The IASTED European Conference on Internet and Multimedia Sytems and Applications, EuroIMSA 2007, Chamonix, France.
- [Squire, 2002] Squire, K. (2002). Cultural framing of computer/video games. *Game Studies, the International Journal of Computer Game Research*. 2 (1).
- [Squire, 2003] Squire, K. (2003). Video games in education. *International Journal of Intelligent Simulations and Gaming* 2 (1).
- [Squire, 2005] Squire, K. (2005a). Changing the game: What happens when videogames enter the classroom. *Innovate* 1(6).
- [Stangl, 2003] Stangl, W. (2003). *Lernstile – was ist dran?* Retrieved February 12, 2006, from stangl-taller.at: <http://www.stangl-taller.at/ARBEITSBLAETTER/PUBLIKATIONEN/Lernstile.shtml>
- [Steinmetz, 1993] Steinmetz, R. (1993). *Multimedia - Technologie: Einführung und Grundlagen*. Berlin: Springer Verlag.
- [Stoeckl et al., 2007] Stoeckl, R., Rohrmeier, P. & Hess, T. (2007). Motivations to produce user generated content: differences between webbloggers and videobloggers. In *Merging and Emerging Technologies, Processes, and Institutions*. eConference.
- [Stoyan, 2009] Stoyan, R. (in prep.). *WORKING TITLE*. PhD thesis, Departement of Informatics, University of Zurich.
- [Stöcker, 2006] Stöcker, C. (2006). Retrieved July 14, 2008, from [www.spiegel.de: http://www.spiegel.de/netzwelt/web/0,1518,421021,00.html](http://www.spiegel.de/netzwelt/web/0,1518,421021,00.html)
- [Sun, 2008] Sun (2008). *Core J2EE Patterns - Business Delegate*. Retrieved August 23, 2008, from [java.sun.com: http://java.sun.com/blueprints/corej2eepatterns/Patterns/BusinessDelegate.html](http://java.sun.com/blueprints/corej2eepatterns/Patterns/BusinessDelegate.html)
- [Surowiecki, 2004] Surowiecki, J. (2004). *The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies, and Nations*. Doubleday.
- [Tallal, 2000] Tallal, P. (2000). *Newsweek*, January 1
- [Thierwächter, 2008] Thierwächter, S. (2008). Service-orientierte Architekturen: OSGi vs. SCA. iPaper.
- [Thissen, 2003] Thissen, F. (2003). Neue Wege im e-Learning durch den Einsatz dramaturgischer Elemente. In: T. Schleiken (Hrsg.), *Wenn der Wind des Wandels weht...*, Kooperative Selbstqualifikation im organisationalen Kontext. Festschrift für Clemens Heidack anlässlich seines 65. Geburtstages. München – Mering: Hampp.
- [Venkatesh & Davis, 2000] Venkatesh, V. & Davis, F.D. (2000). A Theoretical Extension of the Technology Acceptance Model: Four Longitudinal Field Studies. *Management Science*, 46, 186–204.
- [Wache, 2003] Wache, M. (2003). *E-Learning – Bildung im digitalen Zeitalter*. Retrieved November 10, 2005, [bildungserver.de: http://www.bildungserver.de/db/mlesen.html?Id=23408](http://www.bildungserver.de/db/mlesen.html?Id=23408)

- [Watzlawick et al., 2000] Watzlawick, P., Beavin, J. & Jackson, D. (2000): Menschliche Kommunikation.
- [Webster & Martocchio, 1993] Webster, J. & Martocchio, J. J. (1993). Turning work into play: Implications for microcomputer software training. *Journal of Management*, 19, 127–146.
- [Wechselberger, 2005] Wechselberger, U. (2005). *Digital game-based learning - "Lernerfolg" ex machina ludi? Eine theoriegeleitete Reflexion zum Wissens- und Lerntransfer bei computerbasierten Lernspielen*. Master's thesis, Universität Koblenz-Landau.
- [Weinberg, 2005] Weinberg, D. (2005). *Wikipedia's Long Tail*. Retrieved November 23, 2008, from hyperorg.com: http://www.hyperorg.com/blogger/mtarchive/wikipedias_long_tail.html
- [Wellman & Gulia, 1999] Wellman, B. & Gulia, M. (1999). *Net surfers don't ride alone: Virtual communities as communities*. In: M. Smith & P. Kollock (Eds.): *Communities in Cyberspace* (pp. 167–194). London : Routledge.
- [Winkelmann, 2007] Winkelmann, D. (2007). *Entwicklung und Konzeptionisierung eines evolutionären Lernspieles – eine evaluative Studie*. Master's thesis, Departement of Informatics, University of Zurich
- [Wiegand, 2007] Wiegand, D. (2007). Entdeckungsreise: Digitale Enzyklopädien erklären die Welt. *c't*, 6/2007, 5. März 2007, p. 136–145 Vergleich zwischen Bertelsmann Enzyklopädie 2007, Brockhaus multimedial premium 2007, Encarta 2007 Enzyklopädie und der Wikipedia.
- [Wilding, 2009] Wilding, C. (2009). *Security in osgi*. Master's thesis, Departement of Informatics, University of Zurich.
- [Wikipedia, 2008] Wikipedia (2008). *Wikipedia*. Retrieved February 22, 2009, from [de.wikipedia.org: http://de.wikipedia.org/wiki/Wikipedia](http://de.wikipedia.org/wiki/Wikipedia)
- [Wonderlandblog, 2007] wonderlandblog (2007). *Phil Harrison's Keynote*. Retrieved March 8, 2007, from wonderlandblog: http://www.wonderlandblog.com/wonderland/2007/03/phil_harrisons_.html
- [Wright, 2000] Wright, W. (2000). *The Sims*. Maxis/EA.
- [5inline.de, 2009] 5inline.de. (2009). *Das Ende des Massenmarktes, der Long Tail ist die Zukunft*. Retrieved February 5, 2009, from 5inline: <http://5inline.de/2009/01/20/nischen/das-ende-des-massenmarktes-der-long-tail-ist-die-zukunft/#more-184>

Anhang A: Fragebogen

Herzlich willkommen bei der Evaluation der Benutzeroberfläche des PM Game Editors. Zurzeit wird das Lernspiel PM Game am Educational Engineering Lab entwickelt. Zu diesem Spiel gehört ein Editor, mit welchem die Benutzer eigene Inhalte erstellen und dadurch das Spiel erweitern können. Der Editor soll eine einfache Benutzerschnittstelle bieten, womit das Generieren von Spielinhalten intuitiv und kollaborativ möglich wird.

Wir möchten Sie bitten, den Fragebogen vollständig auszufüllen, da wir ihn sonst nicht für die Auswertung verwenden können. Bei der Mehrheit der Fragen sollten Sie die passendste Antwort auswählen. Selbstverständlich verwenden wir Ihre Angaben nur für Forschungszwecke (Dissertation) und darüber hinaus findet keine Speicherung und Verwendung Ihrer Daten statt. Die Umfrage dauert ca. 15min.
Bitte klicken Sie auf "weiter", um die Befragung zu starten!

Bitte geben Sie Ihr Geschlecht an.

- männlich
- weiblich

Bitte geben Sie Ihr Alter in Jahren an.

Nutzen Sie eine oder mehrere der folgenden Online-Plattformen?

- leider keine
- Wiki
- Facebook
- MySpace
- studiVZ
- Flickr
- sonstige

Wie häufig nutzen Sie Online-Plattformen?

- täglich
- wöchentlich
- monatlich
- seltener als monatlich

Beherrschen Sie eine oder mehrere der folgenden Programmier-/Skriptsprachen?

- keine
- Java
- Ruby

<input type="checkbox"/> C++
<input type="checkbox"/> Python
<input type="checkbox"/> PHP
<input type="checkbox"/> sonstige <input type="text"/>

**In welchem Rahmen nutzen Sie die eben angeführte(n) Programmiersprache(n)?
Bitte beantworten Sie die Frage, falls Sie eine der aufgeführten Programmiersprachen beherrschen.**

<input type="checkbox"/> beruflich (z.B. (Neben-)Job, Freelancer-Tätigkeit, eigene Firma etc.)
<input type="checkbox"/> privat
<input type="checkbox"/> Ausbildung

Bitte geben Sie an welche der folgenden Aussagen zutreffen.

<input type="checkbox"/> Es macht mir Spass, neue Software (inkl. Online-Tools) auszuprobieren.
<input type="checkbox"/> Es macht mir Spass, Stories zu erfinden.
<input type="checkbox"/> Es macht mir Spass, Spiele zu spielen.
<input type="checkbox"/> Es macht mir Spass, kollaborativ zu arbeiten.
<input type="checkbox"/> Es macht mir Spass, etwas zu entwickeln, das auch anderen nutzt.
<input type="checkbox"/> sonstiges <input type="text"/>

Bitte geben Sie an, in welchem Ausmass Sie den folgenden Aussagen zustimmen.

	lehne völlig ab	lehne eher ab	teils/teils	stimme eher zu	stimme völlig zu
Die Buttons sind übersichtlich angeordnet.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Es gibt unnötige Buttons.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Es gibt fehlende Buttons.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Die Icon-Symbole unterstützen das Verständnis der Button-Funktion.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Reine Textbuttons wären hilfreicher.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Die Help-Buttons sind leicht zu finden.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Die Benutzeroberfläche hat meine Motivation zur Arbeit mit dem Editor verringert.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Ich habe folgende Verbesserungsvorschläge:

Bitte geben Sie an, in welchem Ausmass Sie den folgenden Aussagen zustimmen.

	lehne völlig ab	lehne eher ab	teils/teils	stimme eher zu	stimme völlig zu
Mir war zu jeder Zeit klar, auf welchem Detail-Screen ich mich befinde.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mir war zu jeder Zeit klar, was ich als nächstes tun muss, um fortzufahren.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mir war zu jeder Zeit klar, wie ich meine Ideen technisch umsetzen kann.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Die Unverständlichkeit der Systemlogik (Erstellungsprozess) hat meine Motivation zur Arbeit mit dem Editor verringert.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Ich habe folgende Verbesserungsvorschläge:

Bitte geben Sie an, in welchem Ausmass Sie den folgenden Aussagen zustimmen.

	lehne völlig ab	lehne eher ab	teils/teils	stimme eher zu	stimme völlig zu
Aufgrund von Fehlern konnten manche Funktionen nicht genutzt werden.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Aufgrund von Fehlern konnte die Story nicht erstellt werden.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Aufgrund von Fehlern ist meine Motivation zur Arbeit mit dem Editor gesunken.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Mir sind folgende Fehler aufgefallen:

Ich habe folgende Verbesserungsvorschläge:

Der Zeitaufwand zur Erstellung einer Story ist

- angemessen
- zu gross

Ich habe folgende Verbesserungsvorschläge:

Die Response-Zeit zum Speichern oder Abrufen von Daten ist
Response-Zeit = Dauer bis zur Antwort einer Anfrage, z.B. Zeitdauer bis eine
Suchanfrage auf einer Webseite die Resultate anzeigt.

- angemessen
- zu lang

Bitte geben Sie an, in welchem Ausmass Sie folgender Aussage zustimmen.

	lehne völlig ab	lehne eher ab	teils/teils	stimme eher zu	stimme völlig zu
Aufgrund zu langer Response-Zeiten ist meine Motivation zur Arbeit mit dem Editor gesunken.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Bitte geben Sie an, in welchem Ausmass Sie den folgenden Aussagen zustimmen.

	lehne völlig ab	lehne eher ab	teils/teils	stimme eher zu	stimme völlig zu
Das Tutorial ist hilfreich.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Das Tutorial ist verständlich geschrieben.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Das Tutorial ist nachvollziehbar.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ich wusste jederzeit, warum ich welche Aktion ausführen musste.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Das Tutorial ist übersichtlich strukturiert.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Es wäre hilfreich, das Tutorial in den Editor zu integrieren.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Es wäre hilfreich, wenn das Tutorial um ein Glossar ergänzt würde.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Aufgrund der Instruktionen des Tutorials ist meine Motivation zur Arbeit mit dem Editor stark gesunken.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ich nutze Tutorials generell nicht gerne	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Das Tutorial ist vollständig.

- lehne völlig ab
- lehne eher ab
- teils/teils
- stimme eher zu
- stimme völlig zu

Welche zusätzliche Information hätten Sie sich vom Tutorial gewünscht?

Der Umfang des Tutorials ist

- angemessen
- zu gross

Ich habe folgende Verbesserungsvorschläge:

Bitte geben Sie an, in welchem Ausmass Sie den folgenden Aussagen zustimmen. Hinweis zu Tooltip resp. Roll-Over: Fährt man mit der Maus über eine Schaltfläche (Button), sollte, solange sich der Mauszeiger auf der Schaltfläche befindet, eine Hilfetext erscheinen. Dieser Text gibt Auskunft über die Aktion, welche ausgelöst wird, wenn die Schaltfläche gedrückt wird.

	lehne völlig ab	lehne eher ab	teils/teils	stimme eher zu	stimme völlig zu
Die Hilfe-Funktion ist tatsächlich hilfreich.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Es wäre sinnvoll, die Hilfetexte direkt im Editor anzuzeigen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Es wäre hilfreich, Tool-Tipps als Roll-Over anzubieten.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mithilfe der Hilfe-Funktion konnte ich alle meine Fragen/Probleme klären.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Aufgrund der Instruktionen der Hilfe-Funktion ist meine Motivation zur Arbeit mit dem Editor stark gesunken.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Mir hat folgende Hilfestellung gefehlt:

Ich habe folgende Verbesserungsvorschläge:

Bitte geben Sie an, in welchem Ausmass Sie den folgenden Aussagen zustimmen.

	lehne völlig ab	lehne eher ab	teils/teils	stimme eher zu	stimme völlig zu
Die persönliche Betreuung war hilfreich.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Wenn ich Fragen hatte, konnte ich mich jederzeit an den Betreuer wenden.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Der Betreuer konnte meine Fragen stets vollständig beantworten.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Die persönliche Betreuung war trotz Hilfe-Funktionen der Software bzw. des Tutorials wichtig für mich.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Aufgrund der persönlichen Betreuung ist meine Motivation zur Arbeit mit dem Editor stark gesunken.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Ich habe folgende Verbesserungsvorschläge:

Bitte geben Sie an, in welchem Ausmass Sie den folgenden Aussagen zustimmen.

	lehne völlig ab	lehne eher ab	teils/teils	stimme eher zu	stimme völlig zu
Der Austausch mit den anderen Gruppenmitgliedern war wichtig für die Erstellung der Story.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Die Unterstützung durch die anderen Gruppenmitglieder war trotz Hilfe-Funktionen der Software bzw. des Tutorials sowie der Betreuer hilfreich für mich.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Aufgrund des kollaborativen Arbeitens ist meine Motivation zur Arbeit mit dem Editor stark gesunken.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Ich habe folgende Verbesserungsvorschläge:

Bitte beurteilen Sie die folgenden Aspekte nach Schulnoten-Skala. 1 = schlechteste Note; 6 = beste Note						
	1	2	3	4	5	6
graphische Benutzeroberfläche	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Bedienbarkeit	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tutorial	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Help-Funktion	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
persönliche Betreuung	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Gruppenarbeit	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Bitte geben Sie dem Editor eine Gesamtnote nach Schulnoten-Skala. 1 = schlechteste Note; 6 = beste Note	
<input type="radio"/>	1
<input type="radio"/>	2
<input type="radio"/>	3
<input type="radio"/>	4
<input type="radio"/>	5
<input type="radio"/>	6

Bitte geben Sie an, in welchem Ausmass Sie den folgenden Aussagen zustimmen.					
	lehne völlig ab	lehne eher ab	teils/teils	stimme eher zu	stimme völlig zu
Es hat mir Spass gemacht, mit dem Editor zu arbeiten.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Es war für mich lehrreich, mit dem Editor zu arbeiten.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ich würde auch zukünftig mit dem Editor arbeiten.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ich würde mich gern aktiv an der Weiterentwicklung des Editors beteiligen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ich kann die Arbeit mit dem Editor weiterempfehlen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ich würde die gleiche Option erneut wählen.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Vielen Dank für Ihre Teilnahme!
 Um zum Fragebogen zurückzukehren, klicken Sie bitte auf "*zurück*".
 Um die Befragung endgültig abzuschließen, klicken Sie bitte auf "*Abschließen, Antworten speichern*". Ihre Antworten werden dann in der Datenbank gespeichert.

Anhang B: Statistische Daten des Fragebogens

Correlations

		Bitte geben Sie dem Editor eine Gesamtnote nach Schulnoten-Skala.	Zahl beherrscheter Programmiersprachen	Gesamtwert_GUI-Einschätzung	Gesamtwert_Logik-Einschätzung
Bitte geben Sie dem Editor eine Gesamtnote nach Schulnoten-Skala.	Pearson Correlation	1.000	.403	.722**	.695**
	Sig. (2-tailed)		.063	.000	.000
	N	22.000	22	22	22
Zahl beherrschter Programmiersprachen	Pearson Correlation	.403	1.000	.232	.448*
	Sig. (2-tailed)	.063		.298	.036
	N	22	22.000	22	22
Gesamtwert_GUI-Einschätzung	Pearson Correlation	.722**	.232	1.000	.647**
	Sig. (2-tailed)	.000	.298		.001
	N	22	22	22.000	22
Gesamtwert_Logik-Einschätzung	Pearson Correlation	.695**	.448*	.647**	1.000
	Sig. (2-tailed)				
	N	22	22	22	22

	Sig. (2-tailed)	.000	.036	.001	
	N	22	22	22	22.000
Gesamtwert_Bug-Einschätzung	Pearson Correlation	.265	.171	.155	.188
	Sig. (2-tailed)	.233	.446	.492	.403
	N	22	22	22	22
Gesamtwert_Tutorial-Einschätzung	Pearson Correlation	.569**	.247	.489*	.595**
	Sig. (2-tailed)	.006	.267	.021	.003
	N	22	22	22	22
Gesamtwert_Hilfe-Einschätzung	Pearson Correlation	.146	.427*	.175	.063
	Sig. (2-tailed)	.515	.048	.437	.782
	N	22	22	22	22
Gesamtwert_Betreuer-Einschätzung	Pearson Correlation	.265	.058	-.139	.196
	Sig. (2-tailed)	.234	.797	.537	.381
	N	22	22	22	22
Gesamtwert_Kollaborations-Einschätzung	Pearson Correlation	.203	.226	.311	.026
	Sig. (2-tailed)	.366	.311	.159	.910
	N	22	22	22	22

** . Correlation is significant at the 0.01 level (2-tailed).

* . Correlation is significant at the 0.05 level (2-tailed).

Correlations

		Gesamtwert_Bug-Einschätzung	Gesamtwert_Tutorial-Einschätzung	Gesamtwert_Hilfe-Einschätzung
Bitte geben Sie dem Editor eine Gesamtnote nach Schulnoten-Skala.	Pearson Correlation	.265	.569**	.146
	Sig. (2-tailed)	.233	.006	.515
	N	22	22	22
Zahl beherrschter Programmiersprachen	Pearson Correlation	.171	.247	.427*
	Sig. (2-tailed)	.446	.267	.048
	N	22	22	22
Gesamtwert_GUI-Einschätzung	Pearson Correlation	.155	.489*	.175
	Sig. (2-tailed)	.492	.021	.437
	N	22	22	22
Gesamtwert_Logik-Einschätzung	Pearson Correlation	.188	.595**	.063
	Sig. (2-tailed)	.403	.003	.782

	N	22	22	22
Gesamtwert_Bug-Einschätzung	Pearson Correlation	1.000	.158	-.143
	Sig. (2-tailed)		.484	.525
	N	22.000	22	22
Gesamtwert_Tutorial-Einschätzung	Pearson Correlation	.158	1.000	.425*
	Sig. (2-tailed)	.484		.048
	N	22	22.000	22
Gesamtwert_Hilfe-Einschätzung	Pearson Correlation	-.143	.425*	1.000
	Sig. (2-tailed)	.525	.048	
	N	22	22	22.000
Gesamtwert_Betreuer-Einschätzung	Pearson Correlation	-.051	.443*	.088
	Sig. (2-tailed)	.821	.039	.698
	N	22	22	22
Gesamtwert_Kollaborations-Einschätzung	Pearson Correlation	-.243	.044	.179
	Sig. (2-tailed)	.276	.845	.424
	N	22	22	22

** . Correlation is significant at the 0.01 level (2-tailed).

* . Correlation is significant at the 0.05 level (2-tailed).

Correlations

		Gesamtwert_Betreuer-Einschätzung	Gesamtwert_Kollaborations-Einschätzung
Bitte geben Sie dem Editor eine Gesamtnote nach Schulnoten-Skala.	Pearson Correlation	.265	.203
	Sig. (2-tailed)	.234	.366
	N	22	22
Zahl beherrschter Programmiersprachen	Pearson Correlation	.058	.226
	Sig. (2-tailed)	.797	.311
	N	22	22
Gesamtwert_GUI-Einschätzung	Pearson Correlation	-.139	.311
	Sig. (2-tailed)	.537	.159
	N	22	22
Gesamtwert_Logik-Einschätzung	Pearson Correlation	.196	.026
	Sig. (2-tailed)	.381	.910
	N	22	22
Gesamtwert_Bug-Einschätzung	Pearson Correlation	-.051	-.243
	Sig. (2-tailed)	.821	.276
	N	22	22

Gesamtwert_Tutorial-Einschätzung	Pearson Correlation	.443*	.044
	Sig. (2-tailed)	.039	.845
	N	22	22
Gesamtwert_Hilfe-Einschätzung	Pearson Correlation	.088	.179
	Sig. (2-tailed)	.698	.424
	N	22	22
Gesamtwert_Betreuer-Einschätzung	Pearson Correlation	1.000	.123
	Sig. (2-tailed)		.586
	N	22.000	22
Gesamtwert_Kollaborations-Einschätzung	Pearson Correlation	.123	1.000
	Sig. (2-tailed)	.586	
	N	22	22.000

*. Correlation is significant at the 0.05 level (2-tailed).

ANOVA Einschätzung der Teilaspekte

Descriptives

		N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean	
						Lower Bound	Upper Bound
Gesamtwert_GUI-Einschätzung	maennlich	10	3.6714	.55960	.17696	3.2711	4.0717
	weiblich	12	3.6071	.61859	.17857	3.2141	4.0002
	Total	22	3.6364	.57941	.12353	3.3795	3.8933
Gesamtwert_Logik-Einschätzung	maennlich	10	3.0750	.92082	.29119	2.4163	3.7337
	weiblich	12	2.5208	.75723	.21859	2.0397	3.0020
	Total	22	2.7727	.86227	.18384	2.3904	3.1550
Gesamtwert_Bug-Einschätzung	maennlich	10	4.3333	1.22726	.38809	3.4554	5.2113
	weiblich	12	4.0556	1.02330	.29540	3.4054	4.7057
	Total	22	4.1818	1.10184	.23491	3.6933	4.6703
Gesamtwert_Tutorial-Einschätzung	maennlich	10	3.2800	.54732	.17308	2.8885	3.6715
	weiblich	12	3.3750	.79330	.22900	2.8710	3.8790
	Total	22	3.3318	.67850	.14466	3.0310	3.6327
Gesamtwert_Hilfe-Einschätzung	maennlich	10	3.0400	.57194	.18086	2.6309	3.4491
	weiblich	12	3.0000	.58465	.16877	2.6285	3.3715

	Total	22	3.0182	.56538	.12054	2.7675	3.2689
Gesamtwert_Betreuer- Einschätzung	maennlich	10	4.3200	.92472	.29242	3.6585	4.9815
	weiblich	12	4.5333	.92376	.26667	3.9464	5.1203
	Total	22	4.4364	.90845	.19368	4.0336	4.8391
Gesamtwert_Kollaborations- Einschätzung	maennlich	10	4.1000	.70361	.22250	3.5967	4.6033
	weiblich	12	4.0278	.95831	.27664	3.4189	4.6367
	Total	22	4.0606	.83341	.17768	3.6911	4.4301
Bitte geben Sie dem Editor eine Gesamtnote nach Schulnoten-Skala.	maennlich	10	4.30	.949	.300	3.62	4.98
	weiblich	12	4.00	.853	.246	3.46	4.54
	Total	22	4.14	.889	.190	3.74	4.53

ANOVA

		Sum of Squares	df	Mean Square	F	Sig.
Gesamtwert_GUI- Einschätzung	Between Groups	.023	1	.023	.064	.803
	Within Groups	7.028	20	.351		
	Total	7.050	21			
Gesamtwert_Logik- Einschätzung	Between Groups	1.675	1	1.675	2.404	.137
	Within Groups	13.939	20	.697		
	Total	15.614	21			

Gesamtwert_Bug-Einschätzung	Between Groups	.421	1	.421	.336	.569
	Within Groups	25.074	20	1.254		
	Total	25.495	21			
Gesamtwert_Tutorial-Einschätzung	Between Groups	.049	1	.049	.102	.752
	Within Groups	9.618	20	.481		
	Total	9.668	21			
Gesamtwert_Hilfe-Einschätzung	Between Groups	.009	1	.009	.026	.873
	Within Groups	6.704	20	.335		
	Total	6.713	21			
Gesamtwert_Betreuer-Einschätzung	Between Groups	.248	1	.248	.291	.596
	Within Groups	17.083	20	.854		
	Total	17.331	21			
Gesamtwert_Kollaborations-Einschätzung	Between Groups	.028	1	.028	.039	.845
	Within Groups	14.557	20	.728		
	Total	14.586	21			
Bitte geben Sie dem Editor eine Gesamtnote nach Schulnoten-Skala.	Between Groups	.491	1	.491	.610	.444
	Within Groups	16.100	20	.805		
	Total	16.591	21			

ANOVA Geschlechtsunterschiede bezüglich genutzter Onlineplattformen

Descriptives

Zahl genutzter Onlineplattformen

	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
					Lower Bound	Upper Bound		
maennlich	10	1.80	.919	.291	1.14	2.46	1	4
weiblich	12	1.67	.888	.256	1.10	2.23	0	3
Total	22	1.73	.883	.188	1.34	2.12	0	4

ANOVA

Zahl genutzter Onlineplattformen

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	.097	1	.097	.119	.733
Within Groups	16.267	20	.813		
Total	16.364	21			

Correlations

Descriptive Statistics

	Mean	Std. Deviation	N
Bitte geben Sie dem Editor eine Gesamtnote nach Schulnoten-Skala.	4.14	.889	22
Affinität zur Nutzung von Informationstechnologie	6.1364	4.22347	22

Correlations

		Bitte geben Sie dem Editor eine Gesamtnote nach Schulnoten-Skala.	Affinität zur Nutzung von Informationstechnologie
Bitte geben Sie dem Editor eine Gesamtnote nach Schulnoten-Skala.	Pearson Correlation	1.000	.185
	Sig. (2-tailed)		.410
	N	22.000	22
Affinität zur Nutzung von Informationstechnologie	Pearson Correlation	.185	1.000
	Sig. (2-tailed)	.410	
	N	22	22.000

Correlations

Correlations

		Zahl beherrschter Programmiersprachen	Affinität zur Nutzung von Informationstechnologie	Produkt besuchter Plattformen x Besuchshäufigkeit
Zahl beherrschter Programmiersprachen	Pearson Correlation	1.000	.451*	.168
	Sig. (2-tailed)		.035	.456
	N	22.000	22	22
Affinität zur Nutzung von Informationstechnologie	Pearson Correlation	.451*	1.000	.956**
	Sig. (2-tailed)	.035		.000
	N	22	22.000	22
Produkt besuchter Plattformen x Besuchshäufigkeit	Pearson Correlation	.168	.956**	1.000
	Sig. (2-tailed)	.456	.000	
	N	22.000	22	22.000

*. Correlation is significant at the 0.05 level (2-tailed).

**. Correlation is significant at the 0.01 level (2-tailed).

Correlations

Correlations

		Produkt besuchter Plattformen x Besuchshäufigkeit	Bitte geben Sie dem Editor eine Gesamtnote nach Schulnoten-Skala.
Produkt besuchter Plattformen x Besuchshäufigkeit	Pearson Correlation	1.000	.071
	Sig. (2-tailed)		.752
	N	22.000	22
Bitte geben Sie dem Editor eine Gesamtnote nach Schulnoten- Skala.	Pearson Correlation	.071	1.000
	Sig. (2-tailed)	.752	
	N	22	22.000

Correlations

Correlations

		Gesamtwert_Logik-Einschätzung	Zahl beherrschter Programmiersprachen
Gesamtwert_Logik-Einschätzung	Pearson Correlation	1.000	.448*
	Sig. (2-tailed)		.036
	N	22.000	22
Zahl beherrschter Programmiersprachen	Pearson Correlation	.448*	1.000
	Sig. (2-tailed)	.036	
	N	22	22.000

*. Correlation is significant at the 0.05 level (2-tailed).

Regression

Model Summary

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.842 ^a	.709	.564	.587

a. Predictors: (Constant), Gesamtwert_Kollaborations-Einschätzung, Gesamtwert_Logik-Einschätzung, Gesamtwert_Hilfe-Einschätzung, Gesamtwert_Betreuer-Einschätzung, Gesamtwert_Bug-Einschätzung, Gesamtwert_Tutorial-Einschätzung, Gesamtwert_GUI-Einschätzung

ANOVA^b

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	11.769	7	1.681	4.881	.006 ^a
	Residual	4.822	14	.344		
	Total	16.591	21			

a. Predictors: (Constant), Gesamtwert_Kollaborations-Einschätzung, Gesamtwert_Logik-Einschätzung, Gesamtwert_Hilfe-Einschätzung, Gesamtwert_Betreuer-Einschätzung, Gesamtwert_Bug-Einschätzung, Gesamtwert_Tutorial-Einschätzung, Gesamtwert_GUI-Einschätzung

b. Dependent Variable: Bitte geben Sie dem Editor eine Gesamtnote nach Schulnoten-Skala.

Coefficients^a

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	-1.904	1.447		-1.316	.209
	Gesamtwert_GUI-Einschätzung	.949	.390	.619	2.431	.029
	Gesamtwert_Logik-Einschätzung	.244	.232	.237	1.052	.311
	Gesamtwert_Bug-Einschätzung	.129	.126	.159	1.018	.326
	Gesamtwert_Tutorial-Einschätzung	-.092	.331	-.070	-.278	.785
	Gesamtwert_Hilfe-Einschätzung	.074	.274	.047	.271	.791
	Gesamtwert_Betreuer-Einschätzung	.333	.197	.340	1.689	.113
	Gesamtwert_Kollaborations-Einschätzung	-.004	.189	-.004	-.023	.982

a. Dependent Variable: Bitte geben Sie dem Editor eine Gesamtnote nach Schulnoten-Skala.

T-Test

Group Statistics

Bitte geben Sie Ihr Geschlecht an.		N	Mean	Std. Deviation	Std. Error Mean
Produkt besuchter Plattformen x Besuchshäufigkeit	maennlich	10	5.2000	4.04969	1.28062
	weiblich	12	6.0000	3.76588	1.08711
Zahl beherrschr Programmiersprachen	maennlich	10	1.00	1.764	.558
	weiblich	12	.08	.289	.083
Bitte geben Sie dem Editor eine Gesamtnote nach Schulnoten-Skala.	maennlich	10	4.30	.949	.300
	weiblich	12	4.00	.853	.246
Gesamtwert_GUI-Einschätzung	maennlich	10	3.6714	.55960	.17696
	weiblich	12	3.6071	.61859	.17857
Gesamtwert_Logik-Einschätzung	maennlich	10	3.0750	.92082	.29119
	weiblich	12	2.5208	.75723	.21859
Gesamtwert_Bug-Einschätzung	maennlich	10	4.3333	1.22726	.38809
	weiblich	12	4.0556	1.02330	.29540
Gesamtwert_Tutorial-Einschätzung	maennlich	10	3.2800	.54732	.17308
	weiblich	12	3.3750	.79330	.22900

Gesamtwert_Hilfe-Einschätzung	maennlich	10	3.0400	.57194	.18086
	weiblich	12	3.0000	.58465	.16877
Gesamtwert_Betreuer-Einschätzung	maennlich	10	4.3200	.92472	.29242
	weiblich	12	4.5333	.92376	.26667

Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means		
		F	Sig.	t	df	Sig. (2-tailed)
Produkt besuchter Plattformen x Besuchshäufigkeit	Equal variances assumed	.220	.644	-.480	20	.637
	Equal variances not assumed			-.476	18.700	.639
Zahl beherrschr Programmiersprachen	Equal variances assumed	18.779	.000	1.780	20	.090
	Equal variances not assumed			1.625	9.402	.137
Bitte geben Sie dem Editor eine Gesamtnote nach Schulnoten-Skala.	Equal variances assumed	.532	.474	.781	20	.444
	Equal variances not assumed			.773	18.382	.449
Gesamtwert_GUI-	Equal variances assumed	.348	.562	.253	20	.803

	Equal variances not assumed			.256	19.834	.801
Gesamtwert_Logik-Einschätzung	Equal variances assumed	.509	.484	1.550	20	.137
	Equal variances not assumed			1.522	17.464	.146
Gesamtwert_Bug-Einschätzung	Equal variances assumed	.017	.897	.579	20	.569
	Equal variances not assumed			.570	17.612	.576
Gesamtwert_Tutorial-Einschätzung	Equal variances assumed	1.580	.223	-.320	20	.752
	Equal variances not assumed			-.331	19.414	.744
Gesamtwert_Hilfe-Einschätzung	Equal variances assumed	.025	.876	.161	20	.873
	Equal variances not assumed			.162	19.439	.873
Gesamtwert_Betreuer-Einschätzung	Equal variances assumed	.007	.933	-.539	20	.596
	Equal variances not assumed			-.539	19.282	.596

Independent Samples Test

		t-test for Equality of Means			
		Mean Difference	Std. Error Difference	95% Confidence Interval of the Difference	
				Lower	Upper
Produkt besucher Plattformen x Besuchshäufigkeit	Equal variances assumed	-.80000	1.66823	-4.27987	2.67987
	Equal variances not assumed	-.80000	1.67983	-4.31974	2.71974
Zahl beherrschr Programmiersprachen	Equal variances assumed	.917	.515	-.157	1.991
	Equal variances not assumed	.917	.564	-.351	2.184
Bitte geben Sie dem Editor eine Gesamtnote nach Schulnoten-Skala.	Equal variances assumed	.300	.384	-.501	1.101
	Equal variances not assumed	.300	.388	-.514	1.114
Gesamtwert_GUI- Einschätzung	Equal variances assumed	.06429	.25381	-.46515	.59372
	Equal variances not assumed	.06429	.25140	-.46041	.58898
Gesamtwert_Logik- Einschätzung	Equal variances assumed	.55417	.35745	-.19146	1.29979
	Equal variances not assumed	.55417	.36411	-.21248	1.32081
Gesamtwert_Bug-	Equal variances assumed	.27778	.47942	-.72228	1.27784

	Equal variances not assumed	.27778	.48773	-.74852	1.30408
Gesamtwert_Tutorial-Einschätzung	Equal variances assumed	-.09500	.29693	-.71439	.52439
	Equal variances not assumed	-.09500	.28705	-.69494	.50494
Gesamtwert_Hilfe-Einschätzung	Equal variances assumed	.04000	.24790	-.47711	.55711
	Equal variances not assumed	.04000	.24738	-.47698	.55698
Gesamtwert_Betreuer-Einschätzung	Equal variances assumed	-.21333	.39572	-1.03878	.61212
	Equal variances not assumed	-.21333	.39576	-1.04084	.61417

Anhang C: Tutorial zum *PM Game*

Beginner's tutorial

Table of Content

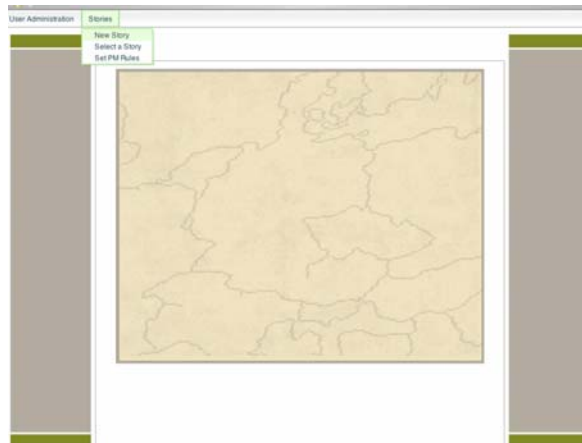
1. **Creating your first story**
 - Open a new story
 - Our story outline
 - The detail screen of an *item*
 - The agent's detail screen and *actions*
 - Adding two *goals*
 - The first Goal: Get the key
 - Second goal: Hans opens the Tower
 - Test your story
2. **Extending your Story**
 - Cascading info objects
 - Hiding objects
3. **Building story lines**
 - Clues
 - The solution list
 - Adding starting locations
 - Repetition
 - Further information
 - Publishing a finished story
 - Further information and help

Version 1.4

1. Creating your first story

In PM Game, story authoring is basically done according the bottom-up principle: We first set up our *agents* and *items* on the stage then add *properties* to them. Finally connect all created stuff with *goals* to bring the drama alive. We call this bottom-up principle. Keep in mind you need to create your story in PM Game starting with the smallest parts.

Open a new story...



Start screen

O.K. let's start the tutorial:

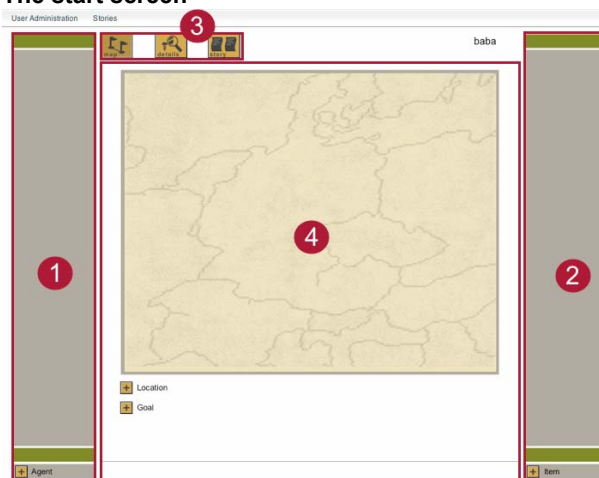
Login at <http://pm-game.ch/> then click the button "pm game editor" and type in your login data. If you don't have an account, request one from ito@ifi.uzh.ch. Open the menu "Stories" and select "New Story".

Give your story a name (We assume, you have the rights to open a new story, so your role is "contribution group leader" or higher. More info about this topic is available in the help¹⁹).

Our story outline

Let's build a basic story: Our agent Hans has to enter a Tower. For he needs a golden key.

The start screen



start screen

The editor presents you an empty story.

The interface consists of:

- (1): the "left bar" with your *agents*.
- (2): the "right bar" with your *items*.
- (3) 3 buttons "map", "details" and "story"

(4) the "center area" shows details corresponding to the 3 buttons:
"map" shows you the main map
"details" shows a detail screen of selected *game object*
"story" shows the story description (visible for the player) and a notices field for your group work support.

¹⁹ <http://www.pm-game.ch/wiki/index.php?title=RoleConcept>

Let's populate the story!

First, we need to fill our story with *agents*, *items* and *locations*:

Click on the "map" button first. The editor displays the game scenario overview. It's empty now. Press the "+" button next to the label "location" for adding a location. This "+" button exists for all *game objects*. Add an *agent* and an *item* by clicking the corresponding "+" button.



Map view with "+" buttons

You'll see now a new *agent* in the left bar, a new *location* icon at the bottom of the map and a new *item* icon in the right bar. Once you need to manage more than five objects per bar, the bars can be scrolled through. Now let's dig deeper: click on the *item* icon on the right bar for the detail screen of this *item*.

The detail screen of an item

The center area shows the details for this new *item object*. At the moment it's empty. Let's insert some details, we add a picture, name and description



open the **picture** gallery with the gallery button: select *items* in the dropdown menu and click on the picture of a key.

type "the golden key" in the **text field**.

Important: If you don't use names like "Hans" or "Peter" as names you have to put an article in front of it. E.g. "a key" or "the key" => because in the game the name of the object will be used to create a sentence in the dialogues!

type "a shiny bright key. Might unlock doors." into **description** field.

4. Press the save button before switching to another detail screen!

There is no global save function in the current release. You need to save every game object. Click the “Save” button at the lower right corner of the detail screen. You see an actualized *item* symbol and - name in the right bar now.

The location detail screen and attributes

Head back to the map by clicking the “map” toggle. It shows the map view. We define the Tower now:

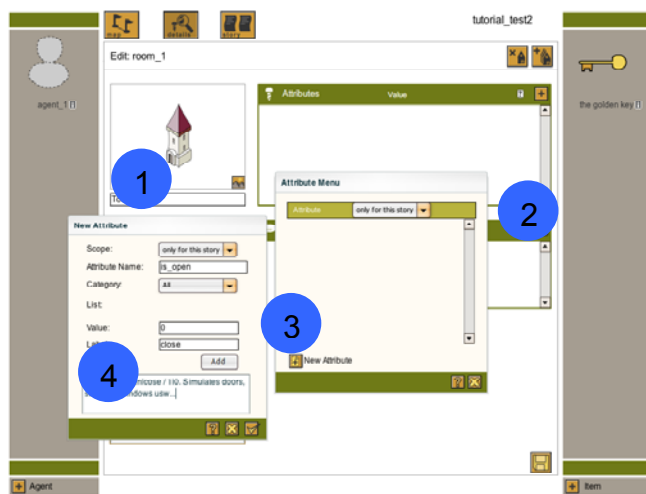
We click on the symbol of the location for his details. It shows the detail screen. We add a nice tower picture from the picture gallery and type in “Tower” as name. It looks nice, but it has no effect in the story so far. Let’s simulate a door by using an attribute with two states “open” and “closed”:

Push the “+” button (see screenshot “detail screen location”, sign No.2)

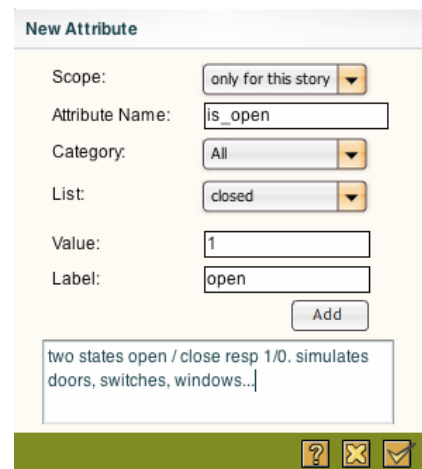
The *attributes template list* opens. All available *attribute* templates are shown here.

We create a *new attribute template* by clicking “New Attribute”. A new window opens.

Type in data as shown in screenshot “new attribute template” (right).



detail screen location



new attribute template

The *Scope* defines the visibility of this attribute to other editor roles. Default value should always be “only for this story” for all attributes in this tutorial.

Attribute Name: We use a generalistic name, so the attribute could be reused for other locations and items, something like “is_open”.

Category: let it set to “All”.

List with values and labels :

The attribute needs two states: one for “open” and one for “closed”. A state consists of a value and a label. The label will be used in the dialogue as “The Tower is *closed*” or “The Tower is *open*”. The values are required for manipulation by the system, but are not visible anymore to you as editor, once you created them.

List:

Value:

Label:


Step 1:
For the first state, we use as label “closed” and as value 0.
Then press “Add”. The list contains the first state now

List:

Value:

Label:

Step2:
For the second state, we use as label “open” and as value 1.
Then press “Add”. The list contains now both states. For this story, the values have no further purpose, but we introduced them as they will be of importance when using PM Rules...

The *description* field is not visible to the players but to other editors. It is highly desired to describe your *attributes* properly. Close the window by clicking (=“OK”) 
Now, our *attribute* “is_open” is shown in the *attribute template list* and can be added to our Tower by clicking the corresponding grey “+” sign.

It appears now in the detail screen of our Tower. Let’s set the start value of “is_open” to “closed”- our Tower can indicate now he’s locked and our agent will need to do something to open the Tower.

Save this screen by clicking the “Save” button at the right bottom corner.
Head back to the map view and drag the Tower symbol somewhere in the map and save then again.
Locations and goals can be dragged and placed in the map.

The agent’s detail screen and actions

We have a Tower and a golden key, but our agent is still generic.
Click on the *agent* symbol on the left bar. The detail screen for this *agent* appears.

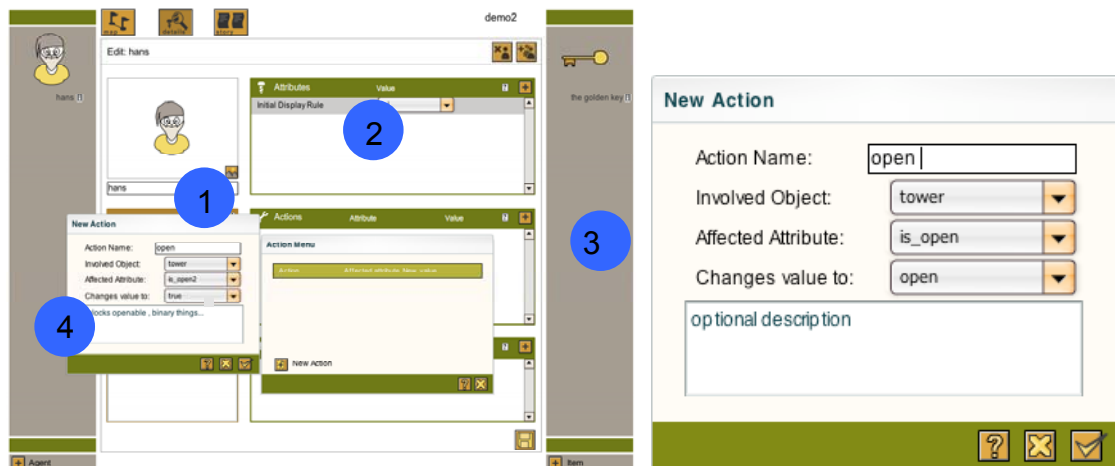
Let’s add a picture and “Hans” as name for this agent.

[Optional]

There is an *attribute* called “display_rule” (only for agents available): Open the Attribute Menu and choose the Entry “global (unvisible)” from the dropdown list, add the attribute “display_rule” to the agent. With this attribute you can set the visible starting mood of the character. Change the value in the list and watch the face of the agent in the left bar! Change it to “☹” (An agent with a bad mood doesn’t execute anything).

Agents own *attributes* and they can execute so called *actions*.

What does an *action*? An action sets a **single value** of a **single attribute somewhere** in the game. We give now Hans the ability to unlock the Tower, by adding an *action* “open”: “open” will change the attribute “is_open”.



detail screen agent

new action template window

We click on the “+” button in the *action list*. The *action template list* opens. It’s empty. Click on the “New Action” button, then set parameters as shown in screenshot “new action template window” (above, right) and save and close the window with “OK”
 O.K! little stop here. A few things about new *action templates*:
 Naming is crucial! The action name must be an infinitive verb like “open”, “do”, “ send”.
 It will be shown in the game and is used to build sentences like: “Could you please (*Action here*) the Tower?”

Actions are executed via dialogues.
Actions can be reused for other objects.

We have now a new action template in the action template list.
 We click on the corresponding grey “+” sign to add an *instance* of this action to Hans.
 It appears now in the detail screen, and we close the action template list.
 In contradiction to attributes, actions can’t change a value in the detail screen.
 Now Hans has the skill to “open” the Tower. *Any* agent or goal owning the action “open” can change the Tower attribute.

You may have remarked, defining a new property always follows the workflow:
 1. Define a template,
 2. Add it from the template list to detail screen. These two steps are necessary for adding all kinds of property types: attributes, actions, and the others.

We need a second *attribute* for later use. Click the “+” button for adding an attribute. In the list, select in the dropdown menu “for all stories” and click the grey “+” for “iscustomer”. Then close the list, and set the instance of “iscustomer” to “customer”.

We do the same again with the attribute “isteamember”. Click the “+” button, select “for all stories” in the dropdown menu and add “isteamember” to Hans’ attributes. Then set the instance to “team member”.

Save this screen. Our first agent is ready to act in the gameworld.

Adding two goals

Now, we have prepared: 1 agent, 1 item, 1 location, 1 attribute and 1 action executable by our agent. Our stage is ready now, but still no “drama” is available for our players. For this we use *goals*. Goals are **single tasks** presented to the player in order to **accomplish the game** and in their nature a bit more demanding than other game objects or properties. Remember our outline: “Our agent Hans needs to enter a Tower and for this he needs a golden key.”

How do we do that now?

We build two *goals*, (= our “project task plan”) :

Hans has to get the golden key

Then he has to open the Tower.

There is a dependency between these two *goals*: Hans can only fulfill the second goal, when he has finished the first goal.

We add two *goals* in the map view by clicking the “+” button for goals. They appear at the bottom of the map and are draggable. We place the goals near the Tower location.

We click then on the red dot representing the first goal and the detail screen appears.

The first Goal: Get the golden



Goals, in contradiction to other game objects, have two parts: an input part and an output part.

Inputs are “conditions” and “attributes” (1).

The output is named “Effects” (2) and can trigger info objects, actions and transfers, once the conditions are fulfilled.

If the goal condition is fulfilled, it will be marked in the map view with a flag and the effects will be triggered and execute their stuff.

key

Goal details

For this goal, we need a special action type: the *transfer*. A *transfer* moves objects in the game and can be executed by *agents* or *goals*. A transfer works a bit different than an action. The player will have to ask Hans to *transfer* the key to Hans himself.



New Transfer

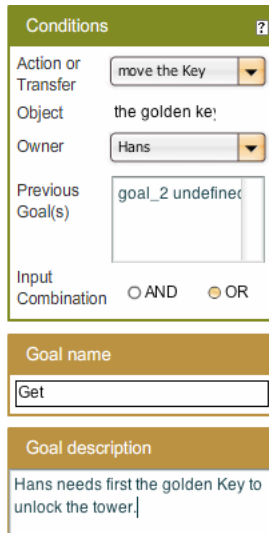
Name:

What:

Destination:

? X ✓

New transfer template window



Conditions ?

Action or Transfer:

Object: the golden key

Owner:

Previous Goal(s):

Input Combination AND OR

Goal name

Goal description

Hans needs first the golden Key to unlock the tower.

Goal "get the golden key" details

- To add a new transfer, press the "transfer" button (3)
- Type in the data as shown below.
- Set the parameter in "Conditions" as shown above. We explain them now:
 - Action or Transfer:** What action or transfer needs to be done in the game to fulfill this goal. "Conditions" *checks* parameters, it doesn't *execute* them in contradiction to "Effects" list.
 - Object:** On what object the action has to be executed. This value is set for Transfers.
 - Owner:** We explain this later. Use "Hans" for the moment.
 - Goal name:** Naming: Just use "get" as name. As soon as you have chosen an object, its name will be added to the goal name automatically. Always use verbs as goal names, because they will be presented to the player as composite names like [verb]+[item/object] in the map.
 - Goal description:** This field is important for the player, it gives him a briefing about the task. Don't select a goal in **Previous Goals** or deselect it again then with "Ctrl" (Control)+Click in PC or "Apple"+Click in Mac.
- Save the goal and head back to the map. Our first goal is ready.

Second goal: Hans opens the Tower

Conditions ?	
Action or Transfer	open
Object	tower
Owner	Hans
Previous Goal(s)	give bottle of wine Get the golden key
Input Combination	<input type="radio"/> AND <input checked="" type="radio"/> OR

Goal name
Enter

Goal description
Hans needs to open the locked tower. The golden key is required.

Goal "Enter the tower" details

Now, the player will ask Hans to *transfer* the key to himself to fulfill the first criteria. Then Hans is ready for the second goal:

Click on the second *goal* in the map view.

The detail screen appears.

Fill in parameters as shown left. There are two things different than before:

Action or Transfer: we use the action "open" and select the target game object of this action as fulfillment criteria. If there are more than one game object around with our attribute "is_open", they all can be used here.

We select a goal in **Previous Goal(s)** this time: This will be a necessary precondition for this goal. We can use up to four goals as preconditions and define with "AND", resp "OR" if all goals need to be fulfilled before or only one of them.

This way we can nest goals and build chains. Dependencies will be shown in the map view as darts.

Save it.

Test your story

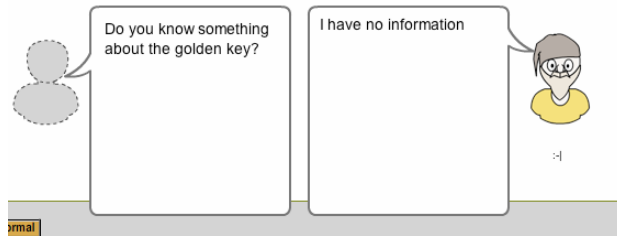
Your story is now complete and can be tested. Click on "Story" Button and then "Save".

For this, open the URL <http://pm-game.ch> in a new browser window and press "PM Game". Then login, select your story and play it.

Keep in mind: As long as a story is in "draft" state it can only be played by yourself and your contribution group. Now, congratulations! You built your first story in PM Game !

2. Extending your Story

Have you played your story? A bit dry, isn't it? Hans looked a bit stupid. He didn't know much, not about goals, nor about items or locations. While PM Game does the basic work of building dialogues, you still need to add some sense to make your story alive.



Dialogue screenshot

Let's open the editor with your story, we will do some refinement:
Hans needs to know more about the *things* in the story
Hans needs to have an idea about at least the second *goal*.

For this we need *info objects*. They allow:

- Giving hints to the player
- Personalization of dialogues
- Triggering other *info objects* (allowing cascades of information)
- revealing invisible game objects

O.K. We click on Hans for his details. At the bottom of the screen you see the info object list. Let's add some hints for the player.

1. Click on "+" of the info object list.
2. Click on "New Info" to set a new Info object template and fill in data as shown:

A screenshot of the "New Info Object" editor window. The window has a title bar "New Info Object". It contains several fields: "Name:" with the value "goldenKeyHint"; "Text:" with the value "I need the golden Key in my HANDS, if I shall enter the Tower!"; "Type:" with radio buttons for "Idea" (selected) and "Question"; "Trigger(s):" with a list box containing "Hans (Agent)", "the golden key (Item)" (highlighted), and "itam_1 (Item)"; and "Target(s):" with a list box containing "Hans (Agent)", "the golden key (Item)", and "itam_1 (Item)". At the bottom right, there are three icons: a question mark, a close button (X), and an OK button.

New info object template
"goldenKeyHint"

Name: Naming isn't an issue as it is only visible to you and your coeditors.

Text: What the agent shall speak

Type: we explain that later. Select idea.

Trigger: what the player has to ask about to trigger this info object.

Target: which objects are getting visible.

At least one trigger or target has to be defined.

3. Click "OK" to close this window and add "goldenKeyHint" to Hans' *info objects list*.

New Info Object

Name:

Text:

Type: Idea Question

Trigger(s):

Target(s):

New info object template
"TowerInfo"

4. Repeat Step 1.-3 with a new *info object* and fill in details as shown left.
5. Then add it to Hans' info object list. This will be presented to player, once he asks about the Tower (not as a goal, simply as location).

O.K. Hans "knows" two things about his game world, one about the golden key and one about the tower. He may response now to "about object or person/know something about?" in the dialogue.

6. Let's do the same with goals. Repeat Step 1 - 3: Fill in details as shown below and add it to Hans' info object list.

New Info Object

Name:

Text:

Type: Idea Question

Trigger(s):

Target(s):

New Info Object

Name:

Text:

Type: Idea Question

Trigger(s):

Target(s):

Two new info object templates about goals

Info object	Parent		
InfoGetKey	-	<input type="button" value="Info"/>	<input type="button" value="Close"/>
infoEnterTower	-	<input type="button" value="Info"/>	<input type="button" value="Close"/>
TowerInfo	-	<input type="button" value="Info"/>	<input type="button" value="Close"/>
goldenKeyHint	-	<input type="button" value="Info"/>	<input type="button" value="Close"/>

info objects in detail screen

These two info objects will give the player some hints about the *goals*. If you have added all four info objects as proposed to Hans' detail screen, it should look like this:

7. Save it.

We have now two hints about *goals* and two hints about the *item* and the *location* in the game, spoken by Hans. It's important not to mix the *info object trigger* and the *agent* at whom you attached an info object. The agent you attach the info object will be the speaker. The trigger is the topic the player has to ask for.

! Hint: Maybe you made a typing mistake. This is correctable without deleting and redoing the object. Info objects are the only editable object templates. Open the info object template list and click on the corresponding "i" sign so you can edit the text. You can also edit triggers and targets.

! Hint Multiselection: You can use multiple triggers and targets ! You may select more than one target with "Apple"-Key on Mac resp. "CTRL"-Key on PC, and therefore trigger all of them with this single info object!

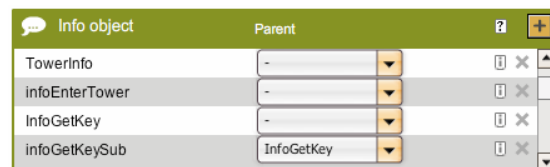
Cascading info objects

Sometimes, the player doesn't get the hint in the first run, and it may be reasonable, to give him a more concrete hint the second time he asks the same. For this case we can **cascade** info objects. A **cascaded** info object will only appear in the dialogue when its *parent* object already has been triggered. Let's try that!



New info object template "InfoGetKeySub"

1. Head back to detail screen of Hans and open the info object template list by using the corresponding "+" button.
2. Press "New info object" and fill in the text as shown below and close the window
3. Add the new info object to Hans' detail screen (2). In the detail screen, go to the info object list, scroll down to our new info object and select "InfoGetKey" as parent in the dropdown menu. Now, you have established a hierarchy of triggers! You can set the parent property in the detail screen only.



info objects in the detail screen (2)

Let's test that! Save the detail screen, open the game and select your story! Keep in mind, once you changed the story, you need to restart the game, as it does not necessarily actualize automatically.



dialogue menu screen

In the game, open a dialogue with Hans and ask then: "/about goal:customer/requirements /get the key/" Do this twice. You'll see, Hans gives two different answers the first and second time you ask!

Hiding and revealing game objects

We demonstrate now another very powerful feature of info objects: Hiding objects from the player and revealing them once he asks for them. This lets you guide the player more closely to your storyline and forces the player to use questions in the dialogues before commanding things. "Reveal" always means revealing a game object in the map as well as revealing it in the dialogue!

At the start of the game, the golden key and the Goal "Get the golden key" will be hidden. Once the player asks Hans how to fulfill the goal "enter the Tower" we show first "Get the golden key". If he asks for this goal, we reveal him the item "the golden key" in a second step. That's our plan.

1. Go to details of the golden key and open the attribute template list with the "+" button. Then select "global" in the dropdown menu, select the attribute "isvisible", and add it to the golden key, then close the window. Now we set "isvisible" to "false" and save it. The golden key will be replaced in the story with a question mark and no info objects are given in the dialogue for this item.
2. We open the details of goal "Get the golden key" and set it invisible like we did with the golden key.
3. If we use the predefined attribute "isvisible" to hide an object in the story, we need an info object to reveal it.



revealing the first hidden game object

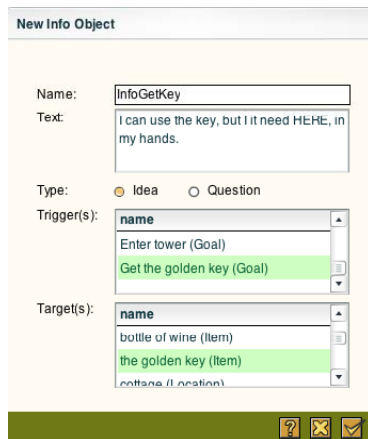
4. We open the *info object template list* with the "+" button.
5. The first accessible info object for the player in the story shall be "infoEnterTower", so we start here: click on the "i" sign to edit it.
6. Our *trigger* is "Enter the Tower (Goal)". This is the topic, the player has to ask for.
7. Then we add "Get the golden key (Goal)" as *target*. It will be revealed once infoEnterTower will be presented.



On the left you see a visualization of the flow we just built.

O.K. the goal can be revealed, but the golden key stays invisible.

In a second step, the item "the golden key" Needs also to be revealed with the info object "infoGetKey". Stop reading here and try to edit it yourself

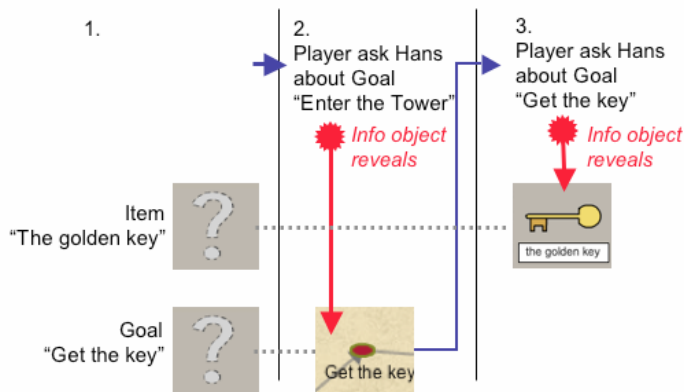


Changed infoGetKey reveals the key.

Revealing the second game object

1. We open the *info object template list* again and click on the “i” sign corresponding to “InfoGetKey”.
2. We add “the golden key(item)” as *target* and close the window with “OK”.
3. We attach the revised info object to Hans details again and save it.
4. We open a new game with your story and play it. Properly done, you’ll see how your two objects will be revealed: First the goal “Get the golden key” and then “The golden key” as well.

Using the combination of hidden game objects and revealing them with info object targets you have a powerful mechanism as gamedesigner to direct the story and player’s behaviour! You may have remarked your story is now easier to play, as less options are available to the player (and more hints are given). This is a main tool in controlling the difficulty of your story.



Schema of flow 2nd un hiding.

3. Building story lines

Our third and last part will show you how we can use all the elements to build more advanced story lines. Based on the second tutorial, we extend our story.

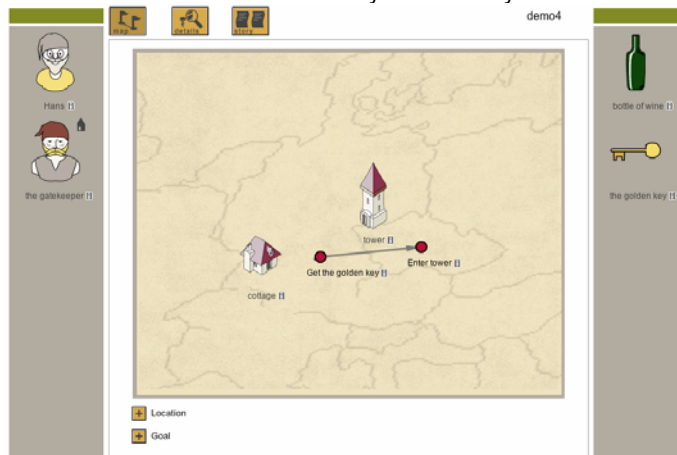
Our new story outline goes like this:

Hans has to give a bottle of wine to the gatekeeper to reveal the golden key.

Hans has to get the golden key

Hans opens the Tower.

This time, it is your charge to construct the story. But don't mind, we give you enough hints: Let's start with the visible "inventory" of our story:

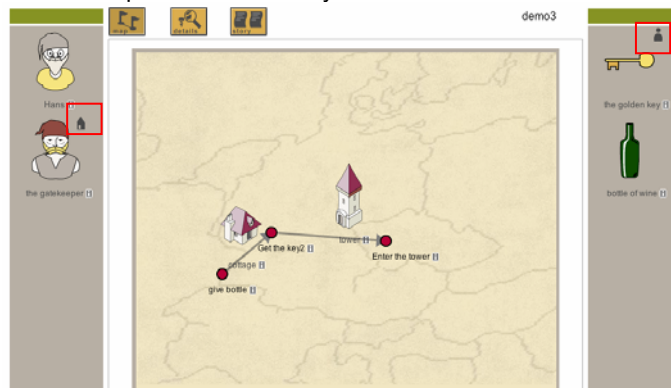


agent "Hans"
 agent "the gatekeeper"
 item "the golden key"
 item "bottle of wine"
 location "cottage"
 location "Tower"

Extended story with all game objects

Further topics: Adding starting locations

In PM Game, logistics is of second-rank. But sometimes it's necessary to have defined starting locations or ownership. This can be easy done:

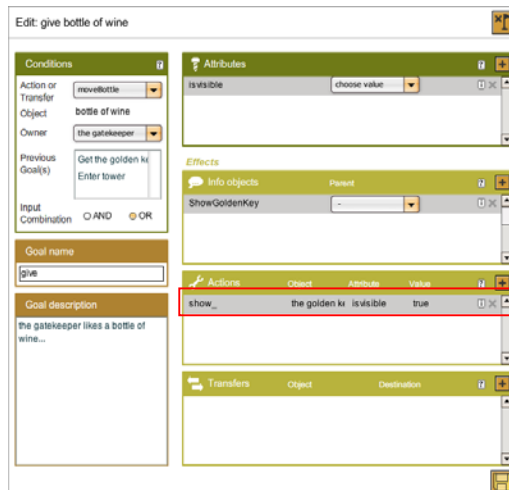


map view with starting locations

1. We set the gatekeeper into the cottage by dragging his symbol from the left bar to the cottage. A little housing icon above his symbol indicates his new location. This will not be visible in the game.
2. We then drag the golden key symbol to the gatekeeper symbol. Another small icon shows the location/ownership of the key. Every agent can still transfer any game object without restriction.

Analyze the story plot

- Then we analyze the plot for combining the goals:
 1. give bottle → 2. get the key → 3. open the TowerThat's easy. It's a single, sequential path through the story. That means, the 1st goal is the prerequisite of the 2nd and the 2nd for the 3rd goal.
- Let's analyze for further dependencies and effects:
 1. bring the bottle to the gatekeeper
 - ⇒ the golden key gets visible,
 - ⇒ 2nd goal gets visible
 2. bring the golden key to Hans ⇒ 3rd goal gets visible
 3. Hans opens the Tower ⇒ won.



Hints

Once the bottle is transferred to the gatekeeper, the golden key gets visible.

For this we need a new combination:

The goal "give bottle of wine" needs to unhide the golden key, when the condition is fulfilled.

For the un hiding we may use an **action** or an **info object**.

To trigger them, we need to attach them to the goal, but not the condition part, **it needs to be attached to the effects part of the goal**.

All game objects in the effects part will be executed, once the goal condition is fulfilled.

We can use goals as a kind of "macros" executing some stuff, when a condition is fulfilled.

- You are required to give an "owner" to every goal. Use always Hans as owner and set an attribute "iscustomer" = customer for him.
- Keep in mind: your story probably is created "top-down" (from rough idea to details) but building in the PM Game editor follows "bottom-up" scheme:
 1. Game objects → 2. Attributes → 3. Actions → 4. Goals, info objects.
- Make a scribble of this dependencies and effects for your understanding and discuss in your team how you want to implement this.
- On the **next page**, you'll find a few **hints and clues** for this task list, if you need help.
- On the **page after**, a complete **solutionslist** is ready.

This is a good moment to stop reading and try to model it yourself ☺

Clues

Do you need further help? We give a couple of clues:

I need more overview. The GUI confuses me. Use:

<http://pm-game.ch/wiki/index.php?title=Components>

<http://pm-game.ch/wiki/index.php?title=DetailScreen>

http://pm-game.ch/wiki/index.php?title=Templates_Instances

<http://pm-game.ch/wiki/index.php?title=Help>

Your first goal uses an action instead of a transfer?

That's a valid way of doing it. The only difference instead of using a transfer to model "give a bottle to the gatekeeper" is the way the player has to ask for it. Try both ways and compare!

Transfers, Ownership, and moving things in the game are completely confusing you:

Go to http://pm-game.ch/wiki/index.php?title=Properties_Locations_Transfers

You have problems with revealing game objects:

Read the 2nd chapter again or go to

<http://pm-game.ch/wiki/index.php?title=NewInfoObject>

Agents give no hint to the player (tell "No chance to do that!" or "Please ask the customer...")

Set Hans as owner of all goals and set an attribute "iscustomer"=true for him.

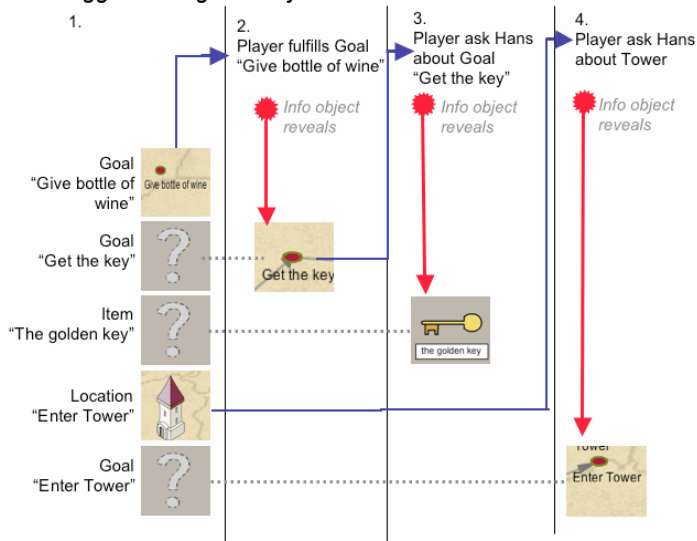
More infos: http://pm-game.ch/wiki/index.php?title=Customer_Clients_Stakeholders

Maybe you have forgotten to solve an previous goal earlier.

You have problems with the template list stuff:

Go to http://pm-game.ch/wiki/index.php?title=Templates_Instances

Flow of triggers and game objects



Flow of un hiding game objects and revealing info objects.

The solution list

Properties	Attribute	
Agent Hans:		
Attribute	display_rule	Value:☺
Attribute	iscustomer	Value:customer
Attribute	lsteammember	Value:team member
Action	open	Attribute:is_open, value:true Object: Tower
Info object	InfoGetKey	Trigger: Get the Key (goal) Target: The golden Key (item)
Info object	infoEnterTower	Trigger: Tower (item) Target: Enter The Tower (Goal)
Agent "the gatekeeper"		
Attribute	display_rule	Value:☺
Attribute	iscustomer	Value:customer
Attribute	lsteammember	Value:team member
Item "the golden Key"		
Attribute	Isvisible	Value:false
Item "bottle of wine"		
Location "The cottage"		
Location "The Tower"		
Attribute	Is_open	Value:closed
Goal "Give bottle of wine"		
Conditions:		
Action or Transfer	Move bottle of wine	Object: bottle, target: the gatekeeper
Object	bottle of wine	
Owner	the gatekeeper	
Previous goal	none	
Effects:		
Info objects	gotBottle*	Target: Get the golden Key(goal)
Goal "Get the golden key"		
Conditions:		
Action or Transfer	move the key	Object:golden key, Target: Hans
Object	the golden key	
Owner	Hans	
Previous goal	Give bottle of wine	
Attribute:	isvisible	Value:false
Goal "Enter the Tower"		
Conditions		
Action or Transfer	open	
Object	The tower	
Owner	Hans	
Previous goal	Get the golden key	
Attribute	isvisible	Value:false
Effects:		
Info Object	reachFinal**	Trigger: Hans (Agent)

The screenshot shows a web form titled "New Info Object". The "Name" field contains "gotBottle". The "Text" field contains "Thanks! Hicup! Great! You may proceed!". The "Type" section has "Idea" selected with a radio button. The "Trigger(s)" dropdown menu is open, showing "name" at the top, followed by "the gatekeeper (Agent)", "Hans (Agent)", and "bottle of wine (Item)". The "Target(s)" dropdown menu is also open, showing "name" at the top, followed by "Enter Tower (Goal)" and "Get the golden key (Goal)", which is highlighted in green. At the bottom of the form are three icons: a question mark, a close button (X), and a checkmark.

*Info Object "gotBottle"

The screenshot shows a web form titled "New Info Object". The "Name" field contains "reachFinal". The "Text" field contains "You finished the mission! Gratulation". The "Type" section has "Idea" selected with a radio button. The "Trigger(s)" dropdown menu is open, showing "name" at the top, followed by "the gatekeeper (Agent)", "Hans (Agent)", and "bottle of wine (Item)". The "Target(s)" dropdown menu is also open, showing "name" at the top, followed by "the gatekeeper (Agent)", "Hans (Agent)", and "bottle of wine (Item)". At the bottom of the form are three icons: a question mark, a close button (X), and a checkmark.

**Info Object "reachFinal"

Remember, this is just a story scaffolding. Your narratives in the info objects will spice it up seriously. This is of relevance for the game play experience (but not necessarily for project management).

Repetition

We used most basic concepts in this tutorial. If you don't fully understand, you can glean them in the help in "Concepts of PM Game": <http://www.pm-game.ch/wiki/index.php?title=Help>

Publishing a finished story

Once your story is ready for publishing, you can sign your story as "ready for publishing".

For this go to menu "User Administration" and select "Stories" / "My Stories".

Drafts are not visible outside your contribution group.

Click on the "i"/info sign and click "ready for publishing" or "release".

Now the superadmin gets an info and will have a final look on your story before publishing it to the community.

Further information and help

Your starting point for all basic concepts and reference is the **help**:

<http://www.pm-game.ch/wiki/index.php?title=Help>

A basic introduction to story building / game design patterns will come soon in our wiki.

A second source is the pm game forum <http://pm-game.ch/forum/>

In "HowTo create a Story" you can address story building issues.

Anhang D: Lebenslauf

Ausbildung

1998 - 2000	Grundstudium Bewegungswissenschaften, ETH Zürich
1998 - 2005	Studium Wirtschaftsinformatik mit Schwerpunkt Künstliche Intelligenz an der Universität Zürich
2005	Diplom in Wirtschaftsinformatik Universität Zürich (MSc in Computer Science)
Seit 2006	Doktorand am Lehrstuhl Educational Engineering (Institut für Informatik, Universität Zürich)
2006	Zertifikat zum Linux Systemadministrator

Berufserfahrung

2001 - 2002	Freelancer als Software Entwickler und Multimedia Producer bei der Skandia AG (IKEA), Gelterkinden
2002 - 2005	Freelancer als Multimedia Producer bei der getunik AG, Zürich
2005 - 2007	Softwareentwickler bei der getunik AG
Seit 2005	Technischer Projektleiter am Lehrstuhl Educational Engineering
Seit 2008	Geschäftsleiter trialox AG