

Virtuelle Mitglieder in virtuellen Teams – Kompensation defizitärer Rollen durch Simulation

Unterstützung virtueller Teams beim Lernen einer Programmiersprache*

Ralph Kölle

Universität Hildesheim
Informationswissenschaft
Marienburger Platz 22
31141 Hildesheim
koelle@uni-hildesheim.de

1 Einleitung

Dem kooperativen Lernen werden gegenüber individuellen Lernformen viele Vorteile bescheinigt. So fördere die Kooperation einen höheren Lernerfolg, Sozialkompetenz und erhöhte Lernmotivation. Auch die „*Entwicklung komplexer Softwaresysteme findet heutzutage meistens arbeitsteilig in – zunehmend räumlich verteilten – Teams statt*“ (Tietze & Schümmer 2001, S.264). Beim Programmieren lernen werden daher theoretische Inhalte durch tutoriell unterstützte Gruppenübungen und -projekte angereichert. Die technische Übertragung dieser Tutorien ins E-Learning erreicht man durch die Kombination kooperativer Editoren und Werkzeugen der computervermittelten Kommunikation wie z. B. Chat. Werden auch Teletutoren, also die Personen, die virtuelle Teams bei ihrer Arbeit unterstützen, durch elektronische Tutoren ersetzt oder ergänzt, so erreicht man für die virtuellen Teams eine absolute Unabhängigkeit von räumlichen und zeitlichen Gegebenheiten (Rautenstrauch 2001).

2 Modellbildung

Bei intelligenten tutoriellen Systemen sind solche virtuellen Tutoren wie ihre menschlichen Kollegen in der Lage, sich dem Wissensstand des Lernenden anzu-

* Veröffentlicht in: OSSWALD, Achim; STEMPFHUBER, Maximilian; WOLFF, Christian (Hrsg.) (2007). Open Innovation. Proc. 10. Internationales Symposium für Informationswissenschaft. Konstanz: UVK, 385-389.

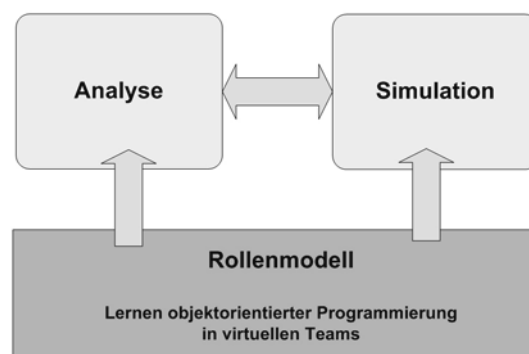
passen. Was beim individuellen Lernen relativ gut funktioniert, lässt sich jedoch nicht direkt auf das kooperative Lernen übertragen, da dort nicht individuelles Wissen, sondern Teamwissen modelliert werden muss.

Das an der Universität Hildesheim entwickelte VitaminL-System modelliert nicht unmittelbar das Teamwissen, sondern bildet sein Teammodell auf Basis eines Rollenmodells (Kölle & Langemeier 2005). Dieses Rollenmodell basiert auf dem Modell von Spencer & Pruss (1995) und besteht aus zehn nach technischen, integrierenden und sozialen Eigenschaften klassifizierten Rollen wie bspw. dem Informationsbeschaffer, dem Berater oder dem Schlichter. Im Gegensatz zu anderen Modellen, die in der Regel davon ausgehen, dass alle Rollen möglichst gleichmäßig besetzt sein sollten, werden die Rollen bei VitaminL zusätzlich nach ihrer Relevanz für den Anwendungsfall der kooperativen Programmierung bewertet und geordnet.

Während der Arbeit des Teams wird dieses auf Ausprägungen einzelner Rollen analysiert. Sollte eine wichtige Rolle unterrepräsentiert sein, übernimmt der virtuelle Tutor diese Rolle, indem er dem Team Hilfsmaßnahmen anbietet, die dieser Rolle zugeordnet sind. Hat das Team bspw. Probleme mit der Syntax oder der Semantik der Programmiersprache, sucht der Informationsbeschaffer geeignete Dateien aus einer Beispielsammlung heraus und präsentiert diese. Er gleicht somit Defizite im Team aus, indem er als virtuelles Mitglied die Funktionen der entsprechenden Rolle simuliert.

3 Systementwicklung

Das VitaminL-System ist grundsätzlich in eine Analyse- und eine Simulationskomponente geteilt. Beide Komponenten basieren auf dem Rollenmodell.



Die Schnittstelle zwischen Analyse und Simulation besteht aus einer Case-based-Reasoning-Komponente, die Problemfälle speichert. Die Fallbasis wird durch

Logfiles vergangener Benutzertests und virtueller Tutorien gespeist und kann durch neue Fälle erweitert werden.

Die modulare, objektorientierte Architektur des Systems, kombiniert mit der Aufspaltung des virtuellen Tutors in zehn Rollen, erleichtert die Weiterentwicklung und Evaluation, da die einzelnen Rollen getrennt voneinander parallel in verschiedenen Teilprojekten entwickelt werden können.

Da die Analysekomponente noch nicht fertig gestellt ist, wurde für die Evaluierung der Simulation zunächst die Rolle des Informationsbeschaffers als wichtigste herausgegriffen und prototypisch implementiert. Außerdem stellt diese Rolle durch die Präsentation von Beispielen kombiniert mit kommunikativen Elementen (im Chat) eine Art generalisierbaren Prototyp für andere Rollen dar. Die Aufspaltung der Hilfe nach fachlicher (mittels Präsentation von Dateien) und sozialer Hilfe (mittels Kommunikation im Chat) orientiert sich am *splitting role tutoring*, wie es Kerres (2005, S.173) vorschlägt.

4 Evaluierung

Die Hilfe des Informationsbeschaffers basiert auf der Präsentation von Programmierbeispielen, die von kommunikativen Phrasen im Chat begleitet werden. In einem zweistufigen Verfahren wurden jeweils Benutzertests durchgeführt, beide Phasen wurden durch Teilnehmer-Fragebögen abgeschlossen.

Alle Benutzertests wurden als Wizard-of-Oz (Woz)-Experiment („hidden operator“) durchgeführt, d.h. das Verhalten des virtuellen Tutors wurde exakt definiert und entsprechende Operationen manuell von einem Bediener nach einem festgelegten Verfahren ausgelöst. Für die Benutzer bleibt das Verfahren transparent, er glaubt mit einem technischen System zu interagieren (vgl. Rapp & Strube 2002, S.661). Da die Simulationskomponente selbst simuliert wurde, kann man auch von *Simulation der Simulation* sprechen.

Als typische Problemsituationen bei Anfängern wurden zunächst Syntaxfehler in den bearbeiteten Dateien gewählt, ein Compiler führte dazu zyklisch einen Syntax-Check durch.

Während in der ersten manuellen Phase die Beispiele intellektuell von einem geschulten Tutor ausgewählt und präsentiert wurden, übernahm diese Aufgabe in der zweiten halbautomatischen Phase eine softwaretechnische Klassifikationskomponente. Die Ergebnisse beider Phasen wurden schließlich miteinander verglichen um

festzustellen, ob sich die Auswahl relevanter Beispieldateien automatisieren ließe, ob also die Simulation des Informationsbeschaffers simulierbar sei.

Die Ergebnisse lassen auf die Tragfähigkeit des rollenbasierten, tutoriellen Konzepts schließen, konnten aber gleichzeitig großes Weiterentwicklungspotenzial aufdecken. Die Bewertungen der Güte der Hilfsmaßnahmen unterschieden sich beim manuellen und beim halbautomatischen Verfahren nur marginal. Dieses Ergebnis überrascht durchaus, offenbar macht die Geschwindigkeit der automatischen Auswahl die Vorteile der intellektuellen Auswahl wett. Immerhin musste ein Tutor aus etwa 70 Dateien die finden, die am besten passt. Allerdings wurden die kommunikativen Fähigkeiten des virtuellen Tutors stark kritisiert. Dieser beherrschte nur einige Phrasen, um die Präsentation der Beispiele zu begleiten. Insbesondere für soziale Rollen sind die kommunikativen Fähigkeiten des integrierten Chatbots zukünftig deutlich auszubauen.

5 Referenzen

- [Kerres 2005] Kerres, Michael: Didaktisches Design und eLearning. In: Miller, D. (Hrsg.): E-Learning. Eine multiperspektivische Standortbestimmung. Haupt, 2005, S. 156 – 182
- [Kölle & Langemeier 2005] Kölle, Ralph; Langemeier, Glenn: Rollen in virtuellen Teams – Analyse und Simulation – Arbeitsbericht vom 21.12.2005.
<http://www.vitaminl.de/downloads/arbeitsbericht2005-12-21.pdf>
- [Rapp & Strube 2002] Rapp, Stefan; Strube, Michael: An Iterative Data Collection Approach for Multimodal Dialogue Systems. In: Rodríguez, M. G.; Araujo, C.P.S. (Hrsg.): Proceedings of the 3rd International Conference on Language Resources and Evaluation, LREC 2002, Las Palmas de Gran Canaria, 29-31 May 2002. Paris: ELRA, 2002, S. 661 – 665
- [Rautenstrauch 2001] Rautenstrauch, Christina: Tele-Tutoren: Qualifizierungsmerkmale einer neu entstehenden Profession. Bertelsmann, 2001
- [Spencer & Pruss 1995] Spencer, John; Pruss, Adrian: Top Teams – Der Königsweg zu mehr Flexibilität, Effizienz und Erfolg im Betrieb. München: Knauer, 1995
- [Tietze & Schümmer 2001] Tietze, Daniel A.; Schümmer, Till: Kooperative Softwareentwicklung. In: Schwabe, G.; Streitz, N.; Unland, R. (Hrsg.): CSCW Kompendium. Springer, 2001, S. 264 – 275