

# Mehrwert der rollenbasierten Umsetzung von kollaborativen Lernumgebungen

Hung Tran Duc

Technische Universität Dresden

Dresden, Deutschland

hung.tduc@yahoo.com

**Abstract**—Rollenbasierte Sprachen werden seit Jahrzehnten als Alternative zu objektorientierten Ansätzen untersucht. Die Natur der Rolle erlaubt es Objekten, sich dynamisch und verschiedene Anforderungen anzupassen. Mit zunehmend komplexer und offener Software, besteht eine wachsende Nachfrage nach adaptiven Systemen. Kollaborative Lernumgebungen fördern je nach Anwendung die Sozialkompetenz, Eigeninitiative oder Konzentrationsfähigkeit der Lernenden. In dieser Arbeit soll beleuchtet werden, welche Vorzüge eine Lernumgebung auf der Grundlage von Rollen aufweist.

**Stichwoerter**—kollaborative Lernumgebung, e-Learning, rollenbasiert

## I. EINLEITUNG

Objektorientierte Programmierung ist das verbreitetste, am häufigsten angewendete und ein weitläufig akzeptiertes Programmierparadigma. Dennoch kann es sich lohnen, Alternativen zu untersuchen. Sie könnten zu innovativen Konzepten führen, welche Mittel und Wege aufzeigen, die mit Objektorientierung nicht erkenntlich waren. Beispielalternativen sind funktionsorientierte Programmierung, welche sich auf die Modellierung von Prozessen fokusierte, oder aspektorientierte Programmierung. Letztere bietet die Möglichkeit zentral Verhalten, das mehrere unabhängige Klassen annehmen müssen, zu definieren.

Rollenbasierte Programmierung stellt eine weitere Alternative zur Objektorientierung dar. Zum ersten Mal wurden Rollen 1977 von Bachman et al. charakterisiert. Sie beschreiben Rollen als festgelegtes Verhalten, welches von Objekten verschiedener Klassen angenommen werden kann [13]. Seitdem wurde der Begriff der Rolle stets wieder aufgegriffen und in verschiedenen Bereichen thematisiert. Diese Bereiche umfassen Ontologien [7] [3], Datenmodellierung [4], Konzeptmodellierung [5] und Programmiersprachen [12]. In [11] hat Steimann aktuelle Untersuchungen zu Rollen zusammengefasst und bewertete den Einfluss der Rolle auf die moderne Datenmodellierung als gering. Diese Beobachtung und die Menge an Untersuchungen, sowohl vor als auch nach Steimann, weisen auf ein großes aber ungenutztes Potenzial der Rolle als Programmierparadigma hin. Eine Rolle beschreibt Attribute und Verhalten von Objekten, die in einem bestimmten Kontext miteinander kollaborieren. Ein Objekt kann also je nach Bedarf eine neue Rolle und somit auch neues Verhalten annehmen. Diese Anpassungsfähigkeit kann behilflich sein, da moderne Softwaresysteme zunehmend komplexer

und offener werden [8].

In den meisten Hochschulen werden mittlerweile Lernplattformen verwendet. Neben der Bereitstellung von Lehrmaterialien, dienen sie auch zur Organisation von Lernvorgängen. Als Lernumgebung erleichtern sie den Lernenden die Kommunikation untereinander oder mit den Vortragenden. Der Lernprozess kann somit zu beliebiger Zeit an einem beliebigen Ort mit einem mobilen Endgerät durchgeführt werden. Audience Response Systeme (ARS) werden während Lehrveranstaltungen eingesetzt. Sie ermöglichen es, von allen Zuhörern gleichzeitig Feedback bzw. Input einzuholen. Browserbasiert oder als mobile App verfügbar, benötigen die Systeme keine spezielle Hardware. Primärer Anwendungsfall ist das Stellen von Fachfragen um den Vorbereitungsgrad oder den Lernstand des Publikums zu ermitteln. Es konnte beobachtet, dass das ARS zu einem deutlich höherem Engagement in Vorlesungen führt. Zusätzlich konnten bei Studenten eine kontinuierliche Aufmerksamkeit und höhere Lernleistung festgestellt werden [10]. Eine andere Anwendung stellen kollaborative Lernumgebungen dar. Sie haben ein einfaches Grundprinzip: Lernende werden in Gruppen geteilt um gemeinsam Aufgaben zu lösen. Sie teilen die Aufgaben untereinander auf und tauschen ihre Kenntnisse aus. Auf diese Weise werden Sozialkompetenz und Organisationsfähigkeit gefördert. Auch die Motivation eines Studierenden, kann durch das Gefühl der Gruppenzugehörigkeit gesteigert werden. Der Fokus auf Kollaboration, sowie die verschiedenen Anwendungsfälle legen nahe, dass Lernumgebungen aus einer rollenbasierten Umsetzung einen Vorteil ziehen könnten.

In dieser Arbeit wird diskutiert, welche Vorzüge eine rollenbasierte Lernumgebung gegenüber einer klassisch objektorientierten Umsetzung haben könnte. Dazu werden in Kapitel 2 zuerst Problem und Anforderungen, sowie verschiedene Szenarien in Kollaborativen Lernumgebungen beschrieben. Nachfolgend wird auf die Herausforderungen und Grenzen der objektorientierten Programmierung eingegangen. Kapitel 3 beschäftigt sich mit dem Konzept der Rolle und verschiedenen rollenbasierten Systemen. Kapitel 4 thematisiert die möglichen Vorteile von rollenbasierten Modellierungs- und Programmiersprachen, welche einer Kollaborativen Lernumgebung von Nutzen sein könnten. Die eventuell entstehenden Herausforderungen und Probleme rollenbasierter Ansätze werden in Kapitel 5 erläutert.

## II. HERAUSFORDERUNGEN STATE OF THE ART

In diesem Kapitel wird eine beispielhafte kollaborative Lernumgebung für eine Hochschule beschrieben. Es werden Use-Cases genannt und deren relevante Vorgänge beschrieben. Ziel ist es, eine Menge von Anforderungen zu bilden, auf die das Konzept eines rollenbasierten Systems angewandt werden kann.

### A. Herausforderungen Kollab e-Learning

Alle Studierende einer Hochschule besitzen einen Login für die Lernumgebung und sind als „Studierende“ gekennzeichnet. Professoren, wissenschaftliche Mitarbeiter und andere Angestellte der Hochschule sind unter „Mitarbeiter“ zusammengefasst und haben ebenfalls eigene Logindaten.

a) *Audience Response System(ARS)*: Das bereits genannte ARS ist ein System um in Vorlesungen simultan von allen Hörern Input einzuholen, beispielsweise um Fachfragen zu beantworten. Diese Fachfragen und die dazugehörigen Antwortmöglichkeiten müssen vorher vom Vortragenden formuliert und im System gespeichert worden sein. Um diese Fragen jederzeit schnell aufrufen zu können, ist eine Zuordnung zur jeweiligen Lehrveranstaltung nötig, möglicherweise in Form eines Katalogs. Es besteht die Möglichkeit, dass ein Mitarbeiter mehrere Kurse leitet, in anderen Kursen als Vortragender erscheint oder die Rolle des Hörers annimmt um selbst teilzunehmen. Ein Kurs kann mehrere wechselnde Vortragende haben. Das Recht, diese Rolle zu übertragen liegt beim Kursleiter, dem Mitarbeiter, der für diesen Kurs verantwortlich ist.

In Seminaren oder ähnlichen Lehrveranstaltung, muss es möglich sein, die Rolle des Vortragenden für einen begrenzten Zeitraum zu übertragen. In dieser Zeit kann der neue Vortragende ebenfalls fachbezogene Fragen stellen und sich Feedback vom Publikum einholen. Die Annahme der Rolle geschieht nur während der Lehrveranstaltung. Die Formulierung der Fragen samt Antworten findet während der Vorbereitung des Vortrags statt.

b) *Arbeitsgruppen*: Im Rahmen einer Lehrveranstaltung wurden Studierende in Gruppen eingeteilt. In diesen Gruppen sollen sie nun gemeinsam verschiedene Aufgaben lösen. Neben einer zufälligen Zuteilung, kann die Gruppenbildung auf der Lernplattform stattfinden, wo Studierende Gruppen eröffnen oder sich einer vorhandenen anschließen können. In [2] befragte Biasutti Studierende zu ihren Eindrücken, nachdem diese in einer kollaborativen Lernumgebung asynchron eine Aufgabe bearbeiteten. Als Kritikpunkte wurde, neben ungleichmäßiger Beteiligung, die Organisation innerhalb der Gruppe genannt. Eine klare Aufgabenverteilung wurde als wünschenswert betrachtet. In [9] und [1] wird betont wie wichtig es sei, dass einerseits ein Ansprechpartner zur Verfügung steht und dass andererseits die Aktivitäten der Gruppe zum Teil überwacht werden. Casamayor et al. stellen den Ansatz vor, relevante Statistiken einer Gruppe aufzuzeichnen und bei bestimmten Ereignissen den Kursleiter zu benachrichtigen [1].

c) *Offenes Forum*: In OPAL<sup>1</sup> können zu Lehrveranstaltungen offene Foren eröffnet werden. Der Kursleiter kann durch einen Eintrag alle Studierenden über neue Informationen in Kenntnis setzen. Studierende haben die Gelegenheit Fragen zur Veranstaltung oder zu Aufgaben zu stellen, welche von anderen Studierenden oder vom Lehrer beantwortet werden kann.

d) *Reviews*: Enthält eine Lehrveranstaltung eine schriftliche Ausarbeitung, kann jedem Studierenden die Arbeit eines anderen zur Bewertung zugewiesen werden. Selbstverständlich können auch Kursleiter die Ausarbeitungen bewerten.

### B. Probleme der Umsetzung in objektorientierter Programmierung

So etabliert und verbreitet objektorientierte Programmierung auch sein mag, so hat sie dennoch ihre Grenzen. Laut Reenskaug et al. entstand objektorientiertes Design mit dem Ziel, dass der Programmcode möglichst genau dem mentalem Modell des Endnutzers entspricht [?]. Ein Nutzer soll beim Bedienen der Nutzerschnittstelle eine konkrete Vorstellung damit haben, wie er mit den Objekten des Programms interagiert. Je genauer ein Programm die Veränderungen an inneren Zuständen durch den Nutzer wiedergibt, desto intuitiver die Handhabung. Er kann nicht die tatsächlichen Vorgänge beobachten. Er braucht lediglich eine Repräsentation des Programmzustands um es zu bedienen. Diese Abbildung der Vorstellung des Users auf den Code gelingt nicht überall. Klassische objektorientierte Programmierung bietet keine Möglichkeit, die Kollaboration zwischen Objekten zu beschreiben. Algorithmen, die durch Kollaborationen erfüllt werden, und Relationen haben wie Objekte eine Struktur, die sich im Quellcode nur umständlich repräsentieren lässt. Steimann hat dieses Problem anschaulich dargestellt [11]. Seine Veranschaulichung soll im folgenden auf eine kollaborative Lernumgebung mit den oben genannten Merkmalen übertragen werden.

a) *Unter der Annahme, eine Rolle bildet eine Unterklasse*: Ein *Studierender* kann gleichzeitig Hörer in einem als auch Vortragender im anderen Kurs sein. Dies würde bedeuten, dass *Hörer* und *Vortragender* Unterklassen von *Studierender* sind. Dasselbe Verhältnis besteht bei *Mitarbeiter*, die ebenfalls Hörer und Vortragende sein können. Bei einer Unterklasse von sowohl *Studierende* als auch *Mitarbeiter* beschränkt sich das Verhalten auf die Schnittmenge der beiden Oberklassen und fällt damit klein oder leer aus.

## III. ROLE CONCEPT AND FEATURES

- 26 Features von Rollen vorstellen (von steimann und kühn)
- Erweiterung zu OOM: Objekte wechseln Rollen zur Laufzeit
- wenige State of the Art-ansätze vorstellen - 3 Seiten von Kühn(Behavioral, Relational, Contextual) - 3 Repräsentationen von Steiman - Klassen von Rollen - RoSI vorstellen - CROM (Compartment Role Objekt Model) vorstellen -Tool: FRAMEDA

<sup>1</sup><https://bildungsportal.sachsen.de/opal/>

#### IV. VORTEILE ROLLENBASSIERTER ANSÄTZE/ANWENDUNG

- wo könnten die 26 features helfen? - trennung von aufgaben/problemen - Trennung von dynamischem und statischem - dynamische veränderungen des systems - ANPASSBARKEIT ZUR LAUFZEIT - gruppen in lernumgebungen nicht vorhersehbar - Langlebigkeit - berücksichtigen Kontextabhängiges und kollaboratives Verhalten von Objekten [6]
- Verhalten kann unabhängig vom Objekt sein - ermöglicht adaptives verhalten
- stärker je mehr contextwechsel -je mehr kontexte, desto mehr kontextwechsel -je mehr tasks desto mehr kontextwechsel - Kontexte: - Vorlesung - Gruppenarbeit - Offenes Forum - Rer
- Grafische Darstellung von Use Cases

#### V. PROBLEME VON ROLLEN

- wenig support - uneinigkeit über den begriff - produktion von vielen daten - langsam [13]

#### VI. DISKUSSION UND FAZIT LITERATURVERWEISE

- [1] Agustin Casamayor, Analia Amandi, and Marcelo Campo. Intelligent assistance for teachers in collaborative e-learning environments. *Computers & Education*, 53(4):1147–1154, 2009.
- [2] Silvia Dewiyanti, Saskia Brand-Gruwel, Wim Jochems, and Nick J Broers. Students’ experiences with collaborative learning in asynchronous computer-supported collaborative learning environments. *Computers in Human Behavior*, 23(1):496–514, 2007.
- [3] Nicola Guarino and Christopher A Welty. An overview of ontoclean. In *Handbook on ontologies*, pages 201–220. Springer, 2009.
- [4] Terry Halpin. Orm 2. In *OTM Confederated International Conferences” On the Move to Meaningful Internet Systems”*, pages 676–687. Springer, 2005.
- [5] Rolf Hennicker, Annabelle Klarl, and Martin Wirsing. Model-checking helena ensembles with spin. In *Logic, Rewriting, and Concurrency*, pages 331–360. Springer, 2015.
- [6] Thomas Kühn, Max Leuthäuser, Sebastian Götz, Christoph Seidl, and Uwe Alßmann. A metamodel family for role-based modeling and programming languages. pages 141–160, 2014.
- [7] Frank Loebe. Abstract versus social roles—a refined top-level ontological analysis. 2005.
- [8] Stephan Murer, Carl Worms, and Frank J Furrer. Managed evolution. *Informatik-Spektrum*, 31(6):537–547, 2008.
- [9] Manuela Paechter, Brigitte Maier, and Daniel Macher. Students’ expectations of, and experiences in e-learning: Their relation to learning achievements and course satisfaction. *Computers & education*, 54(1):222–229, 2010.
- [10] HFD Winter School. Hochschule im digitalen zeitalter gestalten, 2 2018.
- [11] Friedrich Steimann. On the representation of roles in object-oriented and conceptual modelling. *Data & Knowledge Engineering*, 35(1):83–106, 2000.
- [12] Naoyasu Ubayashi and Tetsuo Tamai. Roleep: role based evolutionary programming for cooperative mobile agent applications. In *Principles of Software Evolution, 2000. Proceedings. International Symposium on*, pages 232–240. IEEE, 2000.
- [13] Charles W. Bachman and Manilal Daya. The role concept in data models., 01 1977.