

Logik zur Simulation von Laborversuchen für eine Game Based Learning Anwendung

Felix Gers

Beuth Hochschule für Technik Berlin, Fachbereich VI - Informatik und Medien

Steffen Prowe

Beuth Hochschule für Technik Berlin, Fachbereich V - Life Sciences and Technology

Kurzfassung: Zur Vorbereitung von Laborversuchen in der Mikrobiologie setzen wir eine Game Based Learning (GBL) Anwendung ein. Studenten simulieren mittels der GBL-Anwendung die komplexen, nebenläufigen Prozesse der realen Experimente. Mittels der Logik der Anwendung sollen die Versuche realitätsnah simuliert werden. Dazu werden alle im Labor durchzuführenden Abläufe auf eine Netzwerkstruktur als Ablaufdiagramm abgebildet. Das Ablaufdiagramm kann im UML-Format zur Laufzeit in die GBL-Anwendung geladen werden.

Abstract: In preparation of laboratory experiments in microbiology we use a game-based learning (GBL) application. Students simulate by means of GBL the complex, concurrent processes of the real experiments. The logic of the application allows for a realistic simulation of the experiments. All processes that should later be carried out in the laboratory are mapped to a network structure as a flow chart. The flow chart in UML format can be loaded at runtime into the application.

Einleitung

Game Based Learning (GBL) verbindet spielerische Elemente mit einer Lernanwendung [Det 11] [Rig 11][Fuc 14]. GBL hält derzeit Einzug in unterschiedliche Bereiche von schulischer und universitärer Lehre und industrieller Ausbildung [Det 11a]. Studien gehen davon aus, dass der Bereich in den kommenden Jahren weiter an Relevanz gewinnen wird [Gar 12].

Wir nutzen GBL für die Vorbereitung von Laborübungen im Fach Mikrobiologie. Ab dem 2. Fachsemesters Biotechnologie erlernen die Studierenden in ihren Pflicht-Laborpraktika grundlegende und vertiefende Techniken der Mikrobiologie und erhalten parallel zu einer theoretischen Vorlesung praktische Bezüge. Die Laborversuche laufen entsprechend eines komplexen nebenläufigen Prozessplans ab. Es ist die Aufgabe der Studierenden den Ablauf der Prozesse selbständig zu planen und im vorgegebenen Zeitrahmen erfolgreich durchzuführen. Die zur Vorbereitung traditionell eingesetzten schriftlichen Anleitungen (Skripte) ermöglichen lediglich eine analoge und eindimensionale Vorbereitung und Anleitung zu den Themen der Laborübung. So sind nicht alle Studierenden nach einer "konventionellen" Vorbereitung in der Lage den Transfer in die praktische Laborübung zu vollziehen. Entsprechend erstreckt sich der Betreuungsaufwand neben der stets notwendigen aktiven Unterstützung während der Laborübungen auch auf grundlegende Erklärungen zum Sinn, der Theorie als auch auf Hinweise zur Abfolge der Arbeitsschritte.

Damit die Studierenden die Labore und die Betreuung in den Übungen optimal ausnutzen können, sollen die Vorbereitungen der Versuche mittels GBL effektiver gestaltet werden [Bon 14]. Die Studierenden sollen in einer GBL-Anwendung [BHT 13] mit beliebigen Wiederholungen und zeitlich unabhängig vom Kurstermin vorab versuchen den Prozessablauf möglichst realitätsnah und spielerisch zu erlernen und für sich zu optimieren. Solche Überprüfungen können auch als positives Assessment-Tool angesehen werden, was von Wang [Wan 13] als förderlich beschrieben wurde. Zudem wurde von Bowling [Bow 13] ein Online-Tool im Mikrobiologielabor als hilfreich beschrieben.

Die GBL-Anwendung soll eine möglichst realistische Vorbereitung von mikrobiologischen Laborübungen ermöglichen, dazu sollen die Studenten spielerisch in einem virtuellen Labor agieren können. Um das Versuchspensum im vorgegebenen Zeitrahmen zu absolvieren, gilt es die Abfolge von Schritten in nebenläufigen Teilprozessen strategisch so zu planen, dass diese optimal ineinander greifen. Um dies spielerisch umsetzen zu können, ergeben sich spezielle Anforderungen an die Logik der GBL-Anwendung. Diese und deren Umsetzung werden im Folgenden beschrieben.

Anforderungen an die GBL-Anwendung

Die GBL-Anwendung enthält eine game-basierte Simulation der dann später im Labor durchzuführenden Vorgänge. Hierbei wird die Planung der experimentellen Abläufe nicht als Lernziel vorgegeben, sondern wird den Studierenden überlassen. Vor allem müssen parallel verlaufende Versuchsabläufe zeitlich gut getaktet werden, so dass geschickt zwischen einzelnen Versuchsteilen gewechselt wird. Nur so kann im vorgegebenen Zeitrahmen das Versuchspensum sinnvoll absolviert werden. Hieraus ergeben sich die folgenden Anforderungen an die Logik der Anwendung:

- Alle Arbeitsschritte aus dem realen Labor können ausgeführt und nachvollzogen werden.
- Fehlerhafte Arbeitsschritte werden sofort aufgedeckt und gewertet.
- Zu jedem notwendigen Arbeitsschritt gibt es im Spiel eine kontextsensitive Hilfe.
- Die Anforderung alle Schritte richtig durchzuführen ist notwendig, aber nicht hinreichend, um eine Versuch optimal zu absolvieren.
- Ein begonnener virtueller Versuch kann, wenn auch nicht in der optimalen Zeit, stets erfolgreich abgeschlossen werden.
- Gewertet werden die korrekten und falsch durchgeführten Arbeitsschritte, die für einen Versuch benötigte Zeit und eine Geschicklichkeitswertung für die spielerische Interaktion in Labor.
- Die Anwendung soll ohne neue Software entwickeln zu müssen, über ein rein graphisches Interface verfügen und um neue Laborübungen in spielbarer Form erweiterbar sein.

Logik der GBL-Anwendung

In der, der GBL-Anwendung zu Grunde liegenden Logik, wird die Komplexität der Abläufe der Laborversuche auf eine Netzstruktur abgebildet. Diese Netzstruktur, das Ablaufdiagramm, verknüpft Stränge von Arbeitsabläufen, die korrekt abgearbeitet und zeitlich richtig getaktet werden müssen. Als Basismodell für die Logik der Prozesse wird ein modifiziertes, cyklenfreies Petrinetz-Modell eingesetzt. Die Knoten stellen Zwischenschritte bzw. Zwischenprodukte eines Prozesses dar. Die Anzahl der zu einem Zwischenprodukt zu kombinierenden und der daraus zur erzeugenden neuen Produkte ist beliebig. Zwei Knoten sind jeweils über eine Transition miteinander verbunden. Über die interaktive Nutzung von Werkzeugen und die Kombination von Knoten gelangt man zu einem weiteren Zwischenprodukt. In einem Netz sind in der Regel mehrere Knoten aus unterschiedlichen Arbeitssträngen aktiv. Diese sind parallel und zeitlich richtig aufeinander abgestimmt zu bearbeiten. In Abbildung 1 ist für den Versuch der Calciumbestimmung von Milch ein Ausschnitt aus dem zugehörigen Ablaufdiagramm abgebildet.

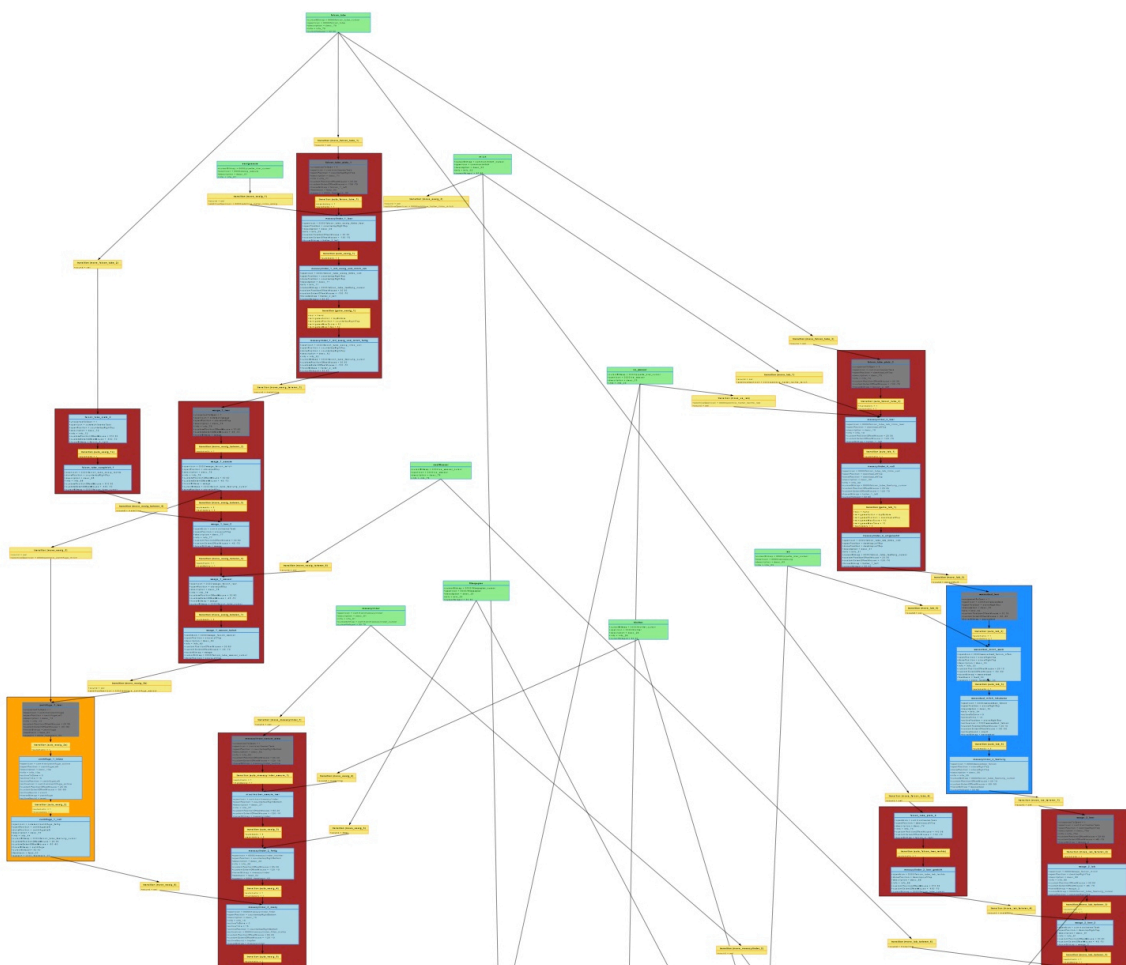


Abbildung 1: Ausschnitt aus dem Ablaufdiagramm für die Calciumbestimmung.

Die Knoten sind so angeordnet, dass der Ablauf von oben nach unten erfolgt. Bei den Knoten kann man Ausgangs- und Zwischenprodukte unterscheiden. Knoten für Ausgangsprodukte haben keine von oben kommenden, eingehenden Verbindungen. Die dunkel umrahmten, vertikalen Blöcke fassen jeweils eine Reihe von zusammengehörigen, nacheinander abzuarbeitenden Knoten und Transitionen zusammen. Sie entsprechen

seriellen Arbeitsabläufen die zeitlich optimal aufeinander abgestimmt werden müssen, um die minimale Zeit zum Absolvieren des gesamten Versuches zu erreichen.

Implementierung der Versuche

Die Anwendung soll ein graphisches Nutzerinterface zum Bearbeiten und Einpflegen der Logiken und medialen Inhalte (Grafiken und Geräusche) neuer Versuche enthalten.

Jeder Knoten des Ablaufdiagramms für einen Versuch enthält die Texte (dazu gehören Hinweise und kontextsensitive Hilfe), Grafiken und Geräusche, die im Spiel für das zugehörige Zwischenprodukt darzustellen beziehungsweise abzuspielen sind. In den Transitionen wird spezifiziert, welche Zwischenprodukte, eventuell unter Verwendung eines Werkzeugs, miteinander kombiniert werden können. Durch die Transitionen werden also die im Ablauf des Spiels möglichen Interaktionen bestimmt.

Ablaufdiagramme werden in der Unified Modelling Language (UML) grafisch erstellt. UML wird sonst üblicherweise zur Modellierung, Strukturierung und Darstellung von Software eingesetzt. Als Werkzeug zur Erstellung der Diagramme wird das freie und quelloffene Programm DIA genutzt [Gno 11]. DIA erlaubt es in der Programmiersprache Python eigene Exporter zu definieren und so ein Diagramm in einem selbst definierbaren Format auslesen und speichern zu können. Ein solcher in DIA eingebrachter Exporter erzeugt aus dem UML-Ablaufdiagramm für einen Versuch ein Script, das von der GBL-Anwendung eingelesen und interpretiert wird. Das Script steuert den Ablauf und die Darstellung des Versuches, welche mit der Torque Game Engine [Gag 09] realisiert wurde. Es kann zu Laufzeit der GBL-Anwendung jederzeit eingelesen werden. Beim Entwurf der Logik und Darstellung für einen Versuch ist es daher nicht notwendig zum Testen bei jeder Änderung die GBL-Anwendung neu zu starten. Es wird lediglich das zugehörige UML-Diagramm neu aus DIA exportiert und in die GBL-Anwendung eingelesen. Diese Eigenschaft ist für eine effiziente Erstellung der Versuche unerlässlich.

Zusammenfassung

Zur Vorbereitung von Laborversuchen für Studenten wurde eine GBL-Anwendung entwickelt. Die Logik für die dazu notwendige game-basierte Simulation der Arbeitsvorgänge im Mikrobiologielabor wurde auf der Basis einer Netzstruktur, als Ablaufdiagramm, umgesetzt. Dieses wird angelehnt an die Verarbeitung eines Petrinetzes ausgeführt. Parallel ablaufende Teilprozesse sind dabei zeitgleich aktiv und müssen zur Optimierung der Laborarbeitszeit von den Studenten aufeinander abgestimmt werden. Die Ablaufdiagramme der Laborversuche können im UML-Format mit der DIA-Software grafisch eingepflegt werden. Dabei ist jeder Knoten entweder einem Arbeitsschritt oder einem Zwischenprodukt des Versuchs zugeordnet. Er enthält die im Spiel damit assoziierten Texte, Grafiken und Geräusche. Das Ablaufdiagramm kann zur Laufzeit der GBL-Anwendung modifiziert und neu geladen werden. Diese Eigenschaft ist zur effizienten Erstellung der Versuchsabläufe und Inhalte unerlässlich. Die grafische Darstellung wird mittels einer Game-Engine realisiert.

Danksagung

Dieses Projekt wurde gefördert vom Stifterverband Deutsche Wissenschaft in Zusammenarbeit mit der Joachim Herz Stiftung im Rahmen der Initiative "Exzellenz in der Lehre".

Literatur

- [BHT 13] BHT (2013). GBL-Anwendung: <http://public.beuth-hochschule.de/~gers/gbl>
- [Bon 14] Bonde M., Makransky G, Wandall, Larsen M., Morsing M., Jarmer H, Sommer M. (2014). Improving biotech education through gamified laboratory simulations. *Nature Biotechnology* 2014 Jul, 32, 694–697.
- [Bow 13] Bowling, K. G., Klish, Y., Wang, S. & Beier, M. Examining an Online Microbiology Game as an Effective Tool for Teaching the Scientific Process. *J MICROBIOL BIOL EDU* 14, (2013).
- [Det 11] Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From game design elements to gamefulness: defining "gamification". In *MindTrek 2011 Proceedings, MindTrek '11* (S. 9–15). New York, NY, USA: ACM.
- [Det 11a] Deterding, S., Sicart, M., Nacke, L., O'Hara, K., and Dixon, (2011). Gamification: Using game-design elements in non-gaming contexts. *Proc. CHI EA '11*, ACM Press (2011), 2425-2428.
- [Fuc 14] Fuchs M., Sonia Fizek S., Ruffino P., Schrape N. (2014). *Rethinking Gamification*. ISBN: 978-3-95796-000-9, menson press, Lüneburg.
- [Gag 09] Garage Games (2009). Torque Game Engine (TGE), <http://www.garagegames.com/>
- [Gar 12] Gartner (2012). Gamification: engagement strategies for business and IT. Report G00245563.
- [Gno 11] Gnome (2011). DIA-Software, [http://de.wikipedia.org/wiki/Dia_\(Software\)](http://de.wikipedia.org/wiki/Dia_(Software)) und <https://wiki.gnome.org/Apps/Dia>.
- [Rig 11] Rigby, S. & Ryan, R. M. (2011). *Glued To Games – How Video Games Draw Us In and Hold Us Spellbound*. Santa Barbara, California: ABC-CLIO, LLC.
- [Wan 13] Wang, J. T. H., Schembri, M. A. & Hall, R. A. How Much Is Too Much Assessment? Insight into Assessment-Driven Student Learning Gains in Large-Scale Undergraduate Microbiology Courses. *J MICROBIOL BIOL EDU* 14, (2013).

Abbildungsverzeichnis

Abb. 1: Ausschnitt aus dem Ablaufdiagramm für den Versuch der Calciumbestimmung.

Kontakt

Felix Gers

E-Mail: gers@beuth-hochschule.de

Internet: <http://prof.beuth-hochschule.de/gers/>

Steffen Prowe

E-Mail: Steffen.Prowe@beuth-hochschule.de

Internet: <http://prof.beuth-hochschule.de/prowe/>

Beuth Hochschule für Technik Berlin, Luxemburger Straße 10, 13353 Berlin

