

Learnflow Mining

Robin Bergenthum, Jörg Desel, Andreas Harrer, Sebastian Mauser
Katholische Universität Eichstätt-Ingolstadt

Abstract: Dieser Artikel überträgt Konzepte und Methoden aus dem Gebiet der Geschäftsprozessmodellierung und des Workflow Management auf Lehr- und Lernprozesse (Learnflows). Dabei werden Gemeinsamkeiten der Gebiete und Spezifika von Lernprozessen herausgearbeitet und eine Methode zur Synthese von Lernprozessmodellen aus realen Lernabläufen vorgeschlagen, die sich auf bestehende Verfahren des Workflow Minings abstützt. An Beispielen wird das Verfahren erläutert. Schließlich werden Folgerungen für die Entwicklung von Werkzeugen zur Erstellung von Lernprozessen durch Dozenten abgeleitet.

1 Einleitung - Modellierung von Lehr-/Lernprozessen

Eigenaktivität des Lerners steht in modernen Lehrkonzepten (wie z.B. konstruktivistischen und selbst-regulatorischen Ansätzen) meist im Vordergrund. Dennoch gibt es Erkenntnisse, dass bei völlig freiem Lernen produktive Aktivitäten, wie z.B. Reflektion und Elaboration, oft nicht ausgeführt werden (können) [Bar03]. Die Strukturierung von Lernaktivitäten durch Skripte [SA77] und Hilfen hat sich dagegen als lernfördernd herausgestellt. In den meisten lernunterstützenden Systemen (z.B. [WEFM05]) sind diese Hilfsmechanismen fest eingebaut, bspw. als Bestandteile der grafischen Oberfläche, und somit nicht wiederverwendbar oder transferierbar in andere Lernkontexte und Lernplattformen.

Die explizite Repräsentation der Lernprozessmodelle und Hilfsmechanismen ist eine Möglichkeit, die pädagogische Expertise wiederverwendbar zu machen und somit den Entwicklungsaufwand für Lehr-/Lernsysteme beträchtlich zu senken sowie die pädagogischen Designprinzipien deutlicher zu machen. Während die Bestrebungen, Lernprozessmodelle explizit zu machen, erst in den letzten Jahren unter den Begriffen *Educational Modelling* [RvK⁺02] und *Learning Design* [KT05] zu Bedeutung gelangt sind, haben sich im Bereich der *Geschäftsprozessmodellierung* (bzw. *Workflow Management*) bereits seit vielen Jahren eine rigorose Methodik und formale Techniken entwickelt, die für Abläufe in Betrieben eingesetzt werden [vdAvH02, Wes07]. Wir wollen im Folgenden die Möglichkeiten der Übertragung von Erkenntnissen und Methoden aus der Geschäftsprozessmodellierung auf das Gebiet der Lernprozesse diskutieren und einen speziellen Ansatz vorschlagen, der es ermöglicht, Lernprozessmodelle (teil-) automatisiert aus Beispielszenarien zu erzeugen.

2 Learnflows und Workflows – analoge Probleme und Verfahren?

In diesem Abschnitt sollen Ähnlichkeiten zwischen den seit den 1990ern intensiv untersuchten Problemen des Geschäftsprozessmanagements und den Herausforderungen in dem sich in der Anfangsphase befindlichen Educational Modelling untersucht werden.

2.1 Das Workflow Management Reference Model und Vergleich mit der Praxis im Learnflow Design

Betriebliche Informationssysteme unterstützen Geschäftsprozesse, die jeweils die Abfolge von Aufgaben definieren. Bei traditionellen Informationssystemen sind diese Prozesse in der Software integriert. Sie sind damit nicht deutlich sichtbar und können insbesondere nur mit größerem Aufwand verändert werden. Modernere ERP-Systeme wie SAP R3 legen die unterstützten Prozesse offen, allerdings erfordert die Änderung von Prozessen (das sog. Customizing) immer noch einen größeren Programmieraufwand. Flexibler sind Workflow Systeme, die auf Workflow Management Systemen basieren. Letztere sind Standardsoftware, bei der die jeweilige Prozesslogik beliebig und flexibel definiert werden kann und eine zusätzliche Eingabe des Systems darstellt. Prozesslogik und Anwendungsfunktionalität sind hier also konsequent getrennt. Neben der eigentlichen Workflow Engine, die die Ausführung eines Workflow Systems bewirkt, gehören zum Prozess- oder Workflow Management Techniken und Werkzeuge zur Prozessentwicklung, -definition und -analyse, zur Interaktion mit Anwendern und Applikationen, zur Administration, und zur Kooperation mehrerer Workflow-Systeme.

Die Workflow Management Coalition (WfMC) wurde 1993 gegründet und besteht heute aus über 250 Software-Entwicklern, Nutzern, Analysten und Forschungseinrichtungen aus dem Bereich des Workflow Management (s. www.wfmc.org). Sie ist um eine Standardisierung der Konzepte, Terminologie und Technologien bemüht.

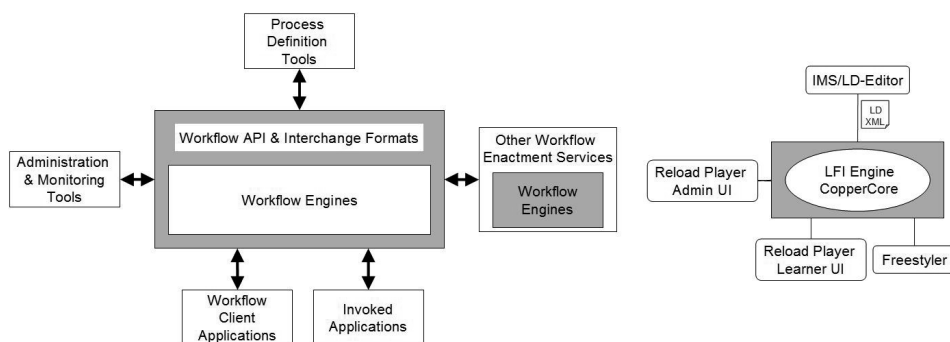


Abbildung 1: Vergleich der Workflow Referenzarchitektur (links) und existierenden Implementierungen von Learnflow Ansätzen (rechts).

Abbildung 1 zeigt links die Workflow Referenzarchitektur der WfMC [vdAvH02, Wes07]. Ihre einzelnen Komponenten haben folgende Bedeutung:

Process Definition Tools: Werkzeuge zum Erstellen und Verändern von Prozessdefinitionen, aber auch für ihre Analyse. Die Prozesse werden in einer geeigneten Form über eine Schnittstelle von den Workflow Engines gelesen. Eine Prozessdefinition legt unter anderem Aufgaben, ihre Reihenfolge und notwendige Ressourcen (Benutzer, externe Programme, ...) fest. Aufgaben können nebenläufig oder auch alternativ ausführbar sein.

Workflow Engine: Der Workflow Enactment Service erzeugt für die eingehenden Fälle Prozessinstanzen, die von einer oder mehreren Workflow Engines ausgeführt bzw. gesteuert

werden. Für jeden Fall wird jede auszuführende Aufgabe Work Item genannt. Ihr werden geeignete Ressourcen gemäß der Prozessdefinition zugeordnet.

Workflow Client Applications: Interaktions-Schnittstelle zu den Anwendern, insbesondere über Worklists, die die für einen Anwender bearbeitbaren Work Items zeigen.

Invoked Applications: Interaktions-Schnittstelle zu externen Anwendungen, die interaktiv oder vollautomatisch Work Items bearbeiten können.

Workflow Interoperability: Für die Kooperation mehrerer Workflowsysteme werden in dieser Schnittstelle Standardfunktionen und Protokolle festgelegt.

Administration and Monitoring: Administratoren und Verantwortliche einzelner Prozessinstanzen können über diese Schnittstelle Prozessparameter setzen, Statusinformationen bekommen und allgemein Daten zur Analyse und Reorganisation von Prozessen erheben.

Im Bereich des *Educational Modelling* hat sich eine ähnliche Aufteilung in Komponenten etabliert. Die formalen, meist XML-basierten Darstellungen wie z.B. IMS/LD [IMS03], LDL [MV⁺06], PALO [Rod02] oder MoCoLADe [HH08] werden selten direkt auf der XML-Ebene editiert, sondern mit speziellen Editoren auf abstrahiertem Niveau erstellt. Die Beschreibungen werden durch sog. *Learning Design Engines* interpretiert und zur Ausführung gebracht. Die Engine steuert - je nach Ansatz - eine web-basierte Benutzerschnittstelle für den Lerner (z.B. beim WebPlayer von IMS/LD) oder externe Lernanwendungen, wie im Falle des *Remote Control* Ansatzes [HMW08], bei dem die existierenden kollaborativen Lernumgebungen FreeStyler und CoolModes verwendet wurden. Einige der Ansätze setzen auf eine vollständige Realisierung von Editor, Engine und Lernumgebung, während andere Ansätze, wie z.B. das Collage Werkzeug [Her07] oder die MoCoLADe Modellierungssprache ein Mapping der erstellten Modelle auf IMS/LD als Zielsprache vornehmen, im Prinzip also IMS/LD als *Educational Assembler* verwenden.

Zur Administration der Lernprozesse und zur Beobachtung können - wie im Workflow Management - Konfigurations- und Monitoring-Werkzeuge eingesetzt werden, wobei diese bisher nur sehr grundlegende Funktionalitäten bereitstellen und sicherlich in Zukunft weiteren Entwicklungsbedarf haben.

Forschungsgruppen im Educational Design, die auch im CSCW-Bereich forschen und entwickeln, schlagen zunehmend Service-orientierte Ansätze [VMN⁺06] für Architekturen und einzelne Komponenten vor. Z.B. integriert das GridCole-System [BHD⁺04] die frei verfügbare CopperCore-Engine (s. www.coppercore.org) für IMS/LD mit als Gridservices implementierten Lernwerkzeugen. Interoperabilitätsansätze für Workflows wie z.B. bpe4ws wurden deshalb auch bereits im Educational Design thematisiert [HM06].

Insgesamt soll Abbildung 1 einen ersten Schritt hin zu einem Referenzmodell für Educational Modelling illustrieren. In Analogie zum Workflow Referenz Modell soll die rechte Seite der Abbildung das konzeptuelle Zusammenspiel von Funktionalitäten prozessgestützter Lernsysteme anhand von existierender Software darstellen.

2.2 Spezifika von Lernprozessen

Trotz all der Gemeinsamkeiten, die sich konzeptionell zwischen Workflows und Learnflows ergeben und sich folglich in ähnlichen technischen Realisierungen niederschlagen, sind auch wesentliche prinzipielle Unterschiede festzustellen:

Beispielsweise ist die Nebenläufigkeit, die in Workflows gleichzeitige Bearbeitung von Teilprozessen an verschiedenen Stellen erlaubt, bei Lernprozessen differenziert zu betrachten. Beim Einzellernen werden aufgrund des eindeutigen Aufmerksamkeitsfokus des Lerner unabhängige Aufgaben meist einen beliebig sequenzialisierten Ablauf der Aufgaben ergeben, da die nebenläufige oder verzahnte Bearbeitung (interleaving) erhöhte kognitive Last beim Lerner erzeugen würde. Beim Gruppenlernen hingegen ist durch Arbeitsteilung eine nebenläufige Bearbeitung durchaus realistisch – falls die Bearbeitung einer Aufgabe jedoch die Aufmerksamkeit oder Kompetenzen aller Lerner erfordert, ist auch hier Nebenläufigkeit eher unwahrscheinlich.

Während bei den Workflows die Durchführung des Prozesses im Vordergrund steht, ergibt sich in Lehr-/Lernszenarien ein anderer Fokus, nämlich der Lerner und sein Lernerfolg, weniger das Produkt des Prozesses. Die Durchführung des Prozesses (ein Fall) ist lediglich Vehikel für das individuelle und kollektive Lernen.

Sog. Rollen von Benutzern bestimmen bei Workflows welcher Benutzer welche Aufgaben durchführen kann. Eine Besonderheit bei Lernprozessen ist, dass Rollen nicht fest zugeordnet werden wie bei Workflows üblich, sondern typischerweise Lerner sich wandelnde Rollen innerhalb des Lernprozesses (s. etwa [WEFM05] mit rotierenden Rollen) einnehmen. Die Rolle hängt meist von den bereits erfüllten Lernaufgaben ab und kann sich dynamisch ändern. Bei der Modellierung von Lernflows sind also evtl. ausdrucksmächtigere Mittel notwendig, um Lernprozesse mit all ihren Facetten abzubilden.

2.3 Petrinetze als formales Modell für Lernprozesse

Sowohl Geschäftsprozesse als auch Lernprozesse werden meist mit Hilfe von Diagrammen modelliert und kommuniziert. Im Bereich des Workflow Management haben sich unter Anderem *Petrinetze* [Des05] und verwandte Sprachen wie Aktivitätsdiagramme oder ereignisgesteuerte Prozessketten durchgesetzt [vdAvH02, Wes07]. Petrinetze bieten neben einer einfachen Syntax und einer klaren Semantik bei großer Ausdruckskraft einen umfangreichen Fundus an Analyse-, Simulations- und Syntheseverfahren samt entsprechender Werkzeuge. Während elementare Petrinetze zur Modellierung des Kontrollflusses von Prozessen gut geeignet sind, bieten höhere Netze Sprachmittel, um auch komplexere Sachverhalte, Daten- und Zeitaspekte kompakt darzustellen. In der vorliegenden Arbeit beschränken wir uns aber zu Gunsten der Verständlichkeit auf elementare Petrinetze aus Stellen und Transitionen. Wir gehen davon aus, dass die graphische Darstellung eines Petrinetzes sowie die Schaltregel dem Leser vertraut ist [Rei82]. Im Bereich des Workflow Management werden häufig sog. Workflow-Netze [vdAvH02] zur Prozessmodellierung verwendet. Dies sind spezielle Stellen/Transitionen-Netze, die wir auch zur Modellierung von Lernprozessen verwenden werden.

Zur Modellierung von Ressourcen werden im Workflow Management den Transitionen, die die Aufgaben modellieren, Gruppen und Rollen zugeordnet. Die Problematik des Rollenkonzepts bei Lernprozessen, wurde bereits diskutiert. Um dieser gerecht zu werden führen wir, insbesondere für kollaborative Lernprozesse, ein flexibleres Rollenkonzept ein, das im folgenden Abschnitt an einem Beispiel erläutert wird.

3 Learnflow Mining - ein Verfahren zur Synthese von Lernprozessmodellen aus realen Abläufen

Nachdem wir die Analogien von Lernprozessen zum Bereich des Geschäftsprozessmanagement aufgezeigt haben, wollen wir nun ein spezielles Modellierungskonzept aus dem Bereich des Geschäftsprozessmanagement auf Lernprozesse übertragen. Die Definition eines Geschäftsprozesses kann automatisiert oder wenigstens unterstützt werden, wenn Beispielausführungen zu dem Prozess vorliegen. Dazu werden insbesondere Protokolldateien (sog. Event Logs) genutzt, in denen festgehalten wird, welche Aufgaben für welchen Fall (zu welcher Uhrzeit) in einem Informationssystem durchgeführt wurden. Entsprechende Ansätze sind seit einigen Jahren unter den Begriffen *Process Mining* und *Workflow Mining* [vdAvDH⁺03] bekannt (s. www.processmining.org). Diese Verfahren können völlig analog auch zur Erstellung von Lernprozessen eingesetzt werden. Ihre Anwendung in diesem Bereich wollen wir im Folgenden als *Learnflow Mining* bezeichnen.

Das explizite Modellieren von Lernprozessen ist für den Dozenten eine unvertraute Tätigkeit, die normalerweise auf eine sequenzialisierte / tabellarische Unterrichtsplanung beschränkt wird; komplexere Lernprozesse, wie z.B. Spiralansätze zur inkrementellen Vermittlung immer mehr verfeinerter Sachverhalte (bspw. Atommodelle), sind durch solche Ansätze jedoch gar nicht darstellbar. Weiterhin ist die explizite Definition von Lernabläufen häufig dem sog. *Expert Blindspot* [NKA01] unterworfen, d.h. der Dozent legt sich auf für ihn plausible Abläufe fest, und rechnet nicht mit unkonventionellen Lösungsansätzen, die für ihn im *toten Winkel* liegen. Beide Aspekte legen eine direkte Nutzung realer Lernabläufe zur Generierung von Learnflows nahe, um eine - zumindest als Grundform verwendbare - Lernprozessbeschreibung zu erhalten. Unserer Ansicht nach kann dies auch zur Verringerung des Erstellungsaufwands führen – analog zu den Erkenntnissen im Bereich der kognitiven Tutoren [AMSK06], bei denen die Erstellung allgemeiner Regelsätze durch beispielbasierte Repräsentationen ersetzt wurde –, wobei wir für eine empirische Untermauerung auf geplante Folgearbeiten verweisen müssen.

Für die Erzeugung von Beispielabläufen schlagen wir zwei verschiedene Szenarien vor, die jeweils direkt auf ein Logging / Protokollieren der Lernsysteme aufsetzen:

- Der Dozent erstellt mehrere Abläufe durch Demonstration am Lernsystem (*Learnflow by Demonstration*) und kategorisiert die resultierenden Event Logs bezüglich des Korrektheitsgrads oder des Potentials für Lernerfolge.
- Reale Lernabläufe der Schüler mit dem Lernsystem werden geloggt (*Learnflow by Example*) und mit Hilfe einer Selbsteinschätzung (im Sinne eines *Collaborative Filtering* Ansatzes) oder durch anschließende Dozentenbewertung (des Ergebnisses und/oder Prozesses) klassifiziert.

In beiden Fällen erhalten wir attributierte / klassifizierte Event Logs, die als Eingabe für diverse Process Mining Algorithmen verwendet werden können. Dies wird im Folgenden am Beispiel konkreter computerunterstützter Lernszenarien erläutert.

3.1 Verfahren und Beispiele

Das in diesem Artikel vorgestellte Verfahren des Learnflow Mining ist direkt aus dem Verfahren des Process Mining abgeleitet. Abbildung 2 illustriert den Process Mining Ansatz [vdAvDH⁺03].

Jedes prozessorientierte Informationssystem unterstützt eine Menge von Aktionen des operationalen Prozesses. Werden diese Aktionen ausgeführt, entstehen Ereignisse und durch Aufzeichnung der Ereignisse Event Logs. Jede Aufzeichnung eines Ereignisses soll Informationen über den zugehörigen Prozess, den zugehörigen Fall, den Name der Aktion und den Zeitpunkt ihrer Ausführung enthalten. Weitere Informationen, wie z.B. eventuelle Daten / Attribute des Ereignisses oder den Benutzer, der die Aktionen ausführt, sind optional. Für Event Logs

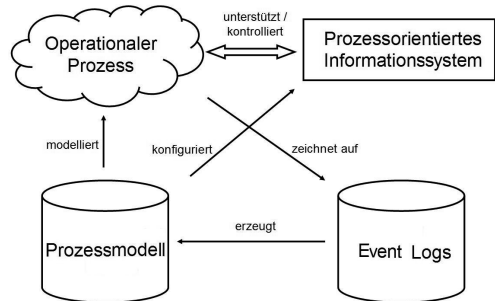


Abbildung 2: Process Mining.

existiert ein standardisiertes XML-Format [vdAea07], in welchem jeder Prozess seine Fälle und jeder Fall seine Ereignisse enthält. Die Ereignisse sind innerhalb der Fälle in der Reihenfolge ihres Ausführungszeitpunktes geordnet. Die in dem Event Log enthaltenen Sequenzen von Aktionen lassen sich als Ausgangspunkt für verschiedene Process Mining Algorithmen verwenden, um Prozessmodelle bzw. deren Kontrollfluss zu erzeugen. Da man nicht davon ausgehen kann, dass man das gesamte Verhalten des operationalen Prozesses beobachten kann, vervollständigen die meisten Process Mining Algorithmen das gegebene Verhalten. Das erzeugte Prozessmodell kann zur Verifikation, Analyse oder zur Steuerung des operationalen Prozesses durch ein Informationssystem benutzt werden.

Als Beispiel für Learnflow Mining betrachten wir im Folgenden eine Schulklasse, die unterstützt durch das Tool FreeStyler [HMW08] lernt, wie sich verschiedene Faktoren (z.B. Lichtverhältnisse, CO₂-Gehalt, ...) auf das Wachstum von Pflanzen auswirken (Lernprozess "Pflanzen"). FreeStyler stellt hierfür verschiedene Registerkarten zur Verfügung, auf denen Fragen formuliert, einfache Modelle gezeichnet oder Daten aus einem Simulationsprogramm importiert werden können (vgl. [HMW08]). Die Menge der Registerkarten ist somit in unserem Beispiel die Menge der unterstützten Aktionen. Wenn ein Schüler eine dieser Registerkarten bearbeitet, wird dies als Ereignis identifiziert. Es stehen Registerkarten mit folgenden Inhalten zur Verfügung, wobei in Klammern der Aktionsname steht, den wir der Registerkarte zuordnen wollen.

- Eine kurze schriftsprachliche Einführung (**Einführung**).
- Eine Zeichenfläche, auf der der Schüler ein grobes Modell, bestehend aus mit Faktoren beschrifteten Knoten, zeichnen kann. Aus diesem soll hervorgehen, wie sich die Faktoren seiner Meinung nach auf das Wachstum auswirken (**Planung**).
- Eine Zeichenfläche, auf der der Schüler sein Modell verfeinern kann. Zusätzlich stehen ihm jetzt Beschriftungen der Kanten zur Verfügung, mit denen er den Einfluss der Faktoren quantifizieren und danach sein Modell in einer Simulation testen kann (**Modell**).

- Textfelder, in denen der Schüler geleitet durch gezielte Fragen die Fragestellungen formuliert, die er im Experiment beantworten will (**Fragestellung**).
- Ein Koordinatensystem, in das der Schüler seine Hypothese über den Zusammenhang eines Faktors und dem Wachstum der Pflanzen als Funktion zeichnet (**Hypothese**).
- Einen Link zu einem Simulationswerkzeug (bspw. BioBLAST), in dem der Schüler sein Experiment durchführen kann (**Experiment**).
- Eine Tabelle mit vorgefertigten Ergebnissen aus verschiedenen Experimenten aus dem Simulationswerkzeug (**Daten**).
- Ein Koordinatensystem, in das man entweder die Daten aus seinem Experiment oder Daten aus der vorgefertigten Tabelle importieren kann. Die daraus entstehende Funktion kann mit der Hypothese verglichen werden (**Analyse**).
- Textfelder, in denen der Schüler geleitet durch gezielte Fragen (Explanation Prompts) seine Ergebnisse zusammenfasst (**Ergebnisse**).

Prozess	Fall	Aktion	Zeit
Pflanzen	Hans	Einführung	10:03:12
Pflanzen	Caro	Einführung	10:03:13
Pflanzen	Bert	Einführung	10:03:54
Pflanzen	Bert	Planung	10:06:43
Pflanzen	Hans	Fragestellung	10:07:33
Pflanzen	Bert	Modell	10:09:37
Pflanzen	Caro	Planung	10:09:59
...

Abbildung 3: Aktionen die von Schülern im FreeStyler ausgeführt werden.

Eine Auswahl dieser Registerkarten im FreeStyler ist in Abbildung 4 gezeigt. Jeder Schüler bearbeitet nun die von FreeStyler bereitgestellten Registerkarten in einer Reihenfolge, die ihm als sinnvoll erscheint. Jedes Bearbeiten einer Registerkarte wird als Ereignis erkannt und aufgezeichnet (vgl. Abbildung 3). Jeder Schüler erzeugt so einen Fall. Wie beschrieben können aus der Menge all dieser Fälle nun nach bestimmten Methoden erwünschte Fälle des Lernprozesses ausgewählt werden. Alternativ kann auch der

Lehrer erwünschte Lernabläufe (Fälle) vorgeben. Eine Menge von Fällen des Lernprozesses ergibt nun ein Event Log und wird als XML-Datei abgespeichert.

Selbst bei einem kleinen Beispiel ist die Darstellung aller möglichen Abfolgen der Aktionen eines Schülers in einem Modell nicht trivial. Ein Event Log eines Lernprozesses kann nun als Eingabe für Process Mining Algorithmen verwendet werden. Z.B. das Workflow Mining Tool ProM [vdAea07] oder das VipTool [BDLM08] stellen solche Algorithmen zur Verfügung. Durch diese Algorithmen erhalten wir automatisiert ein Prozessmodell des Lernprozesses. Ein durch Mining erzeugtes Prozessmodell gibt das aufgezeichnete (dem Dozenten evtl. unbekannt) Lernverhalten wieder. Aus einem Beispiel Event Log des Lernprozesses "Pflanzen" erzeugt eine in VipTool implementierte Process Mining Methode das in Abbildung 5 gezeigte Petrinetzmodell.

Man erkennt an diesem Model, dass alle Schüler mit *Einführung* beginnen. Danach führen sie *Planung* und *Fragestellung* in beliebiger Reihenfolge aus. Nach *Planung* führen manche Schüler eine zusätzliche Verfeinerung Ihres Modelles durch (*Modell*). Nach *Planung*, *Fragestellung* und evtl. *Modell* führen die Schüler *Hypothese* durch. Danach entscheiden sich manche für *Experiment*, einige verzichten darauf und sehen sich die vorgefertigten Daten in der Tabelle an (*Daten*). Nach *Analyse* wiederholen manche Schüler entweder *Experiment* oder *Daten*, wiederum gefolgt von *Analyse*. Schließlich führen alle Schüler als letzte Aktion *Ergebnisse* aus. Dieses Petrinetzmodell ist einerseits eine sehr kompakte,

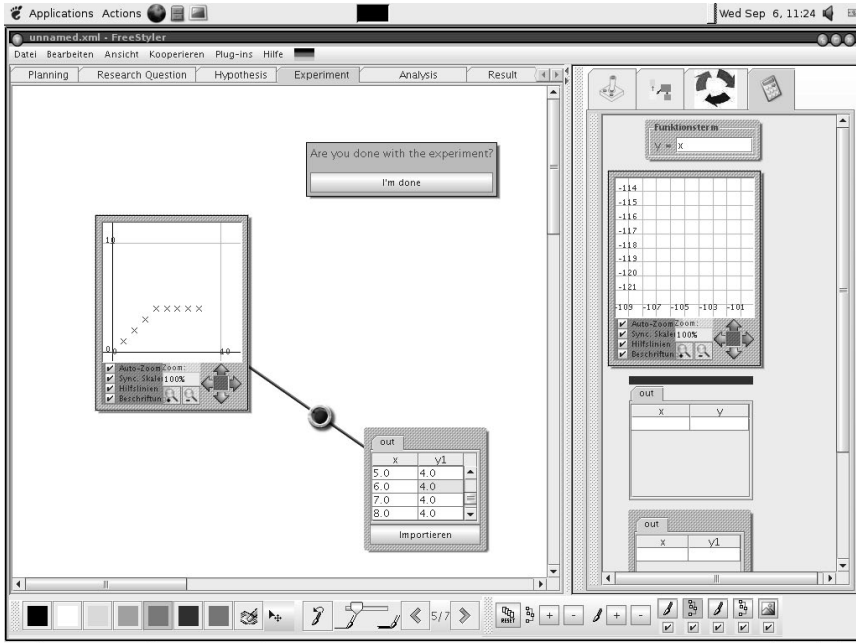


Abbildung 4: Registerkarten im FreeStyler.

vervollständigte Representation aller beobachteten Abläufe eines Lernflows, kann andererseits aber auch als Ausgangspunkt für Analyse- oder Verifikations-Algorithmen dienen. Da Petrinetzmodelle per se ausführbar sind, kann man auch den Lernprozess "Pflanzen" für die Schüler mit dem Petrinetzmodell strukturieren. So könnte z.B. der FreeStyler für jeden Schüler, der mit dem Lernprozess "Pflanzen" beginnt, dieses Petrinetz als Prozessinstanz erzeugen und nur Aktionen anbieten, die für den Schüler in seiner Prozessinstanz gerade ausführbar sind. Eine derartige Lernflow Engine bietet dem Schüler Aktionen in einer durch das Modell beschriebenen Reihenfolge an und führt dadurch zu einem nach der Prozessspezifikation empfohlenen Lernprozess.

Im Folgenden wollen wir den Lernprozess "Pflanzen" beispielhaft zu einem kollaborativen

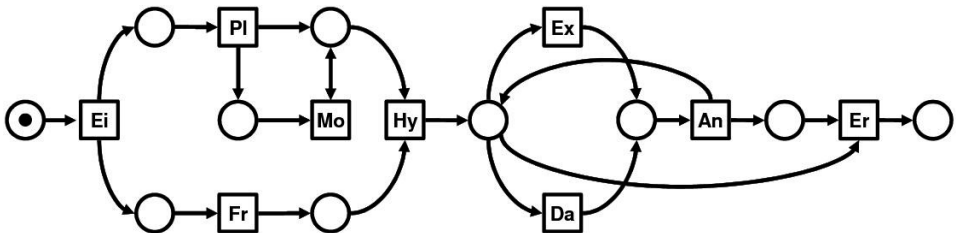


Abbildung 5: Petrinetzmodell des Lernprozesses "Pflanzen".

Lernprozess erweitern. Wir ändern das Petrinetzmodell aus Abbildung 5 derart, dass wir einen kollaborativen Lernprozess von jeweils einer Gruppe von Schülern steuern können. Neben den Aktionen *Experiment* und *Daten* führen wir noch ein weiteres Experiment in den neuen Lernprozess ein. Diese drei Aktionen sollen jetzt parallel zueinander durchgeführt werden, und wir lassen keine Wiederholungen der Experimente mehr zu. Außerdem teilen wir *Ergebnis* in zwei Aktionen *Präsentation1* und *Präsentation2* auf. Um die Rollenanforderungen an die Lerner zu repräsentieren, erweitern wir jede Transition um eine Beschriftung, in der wir spezifizieren, welche Rolle ein Lernender haben muss, um diese Aktion ausführen zu können. Andererseits spezifizieren wir, welche neue Rolle er einnimmt, wenn er diese Aktion durchführte. Das entstehende Modell zeigt Abbildung 6.

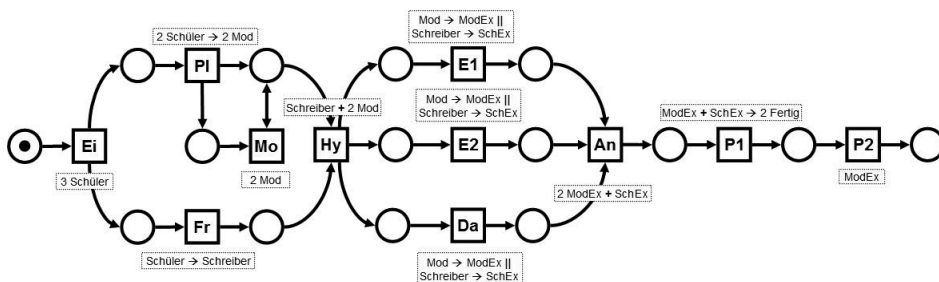


Abbildung 6: Petrinetz Modell des kollaborativen Lernprozesses "Pflanzen" für 3 Schüler.

Der modellierte kollaborative Lernprozess wird von drei Lernenden ausgeführt, die zu Beginn alle die Rolle Schüler haben. In der Startmarkierung ist nur *Einführung* aktiviert. Die Beschriftung von *Einführung* zeigt, dass drei Schüler diese Aktion zusammen ausführen müssen. Haben alle drei die Einführung gelesen, sind *Planung* und *Fragestellung* aktiviert. *Fragestellung* muss einem Lernenden mit der Rolle Schüler zugewiesen werden. Die drei Schüler können hier aussuchen, welcher von ihnen diese Aktion übernimmt. Der entsprechende Schüler ändert dadurch seine Rolle und wird zu einem Schreiber. Parallel dazu erfordert *Planung* zwei Schüler, die gemeinsam die Aktion durchführen. Sie wechseln dadurch ihre Rolle und werden zu Modellierern (Mod). Man beachte, dass kein Lernender an *Planung* und *Fragestellung* teilnehmen kann, da er nach der Ausführung einer dieser Aktionen nicht mehr die benötigte Rolle Schüler besitzt. Nach *Planung* ist optional *Modell* aktiviert. Falls diese Aktion ausgeführt wird, muss sie von den zwei Modellierern bearbeitet werden. *Hypothese* bearbeiten wieder alle drei Lernenden zusammen. *Experiment1*, *Experiment2* und *Daten* sind entweder von einem Schreiber oder von einem Modellierer zu bearbeiten. In beiden Fällen wechselt der Bearbeitende wieder seine Rolle. Die drei am Prozess beteiligten Lernenden teilen sich damit diese Aktionen auf. Sind die Aufgaben ausgeführt, ist *Analyse* aktiviert. Nach *Analyse* kann *Präsentation1* ausgeführt werden. Die geforderten Rollen bei *Präsentation1* sind so gewählt, dass der Schüler, der zu Beginn des Prozesses die Aufgabe *Fragestellung* bearbeitet hat, jetzt an *Präsentation1* beteiligt sein muss. Die zwei anderen Schüler teilen sich auf *Präsentation1* und *Präsentation2* auf.

Durch dieses Prinzip veränderlicher Rollen wird den Lernenden ein Rahmen gegeben, in dem sie die Aufgaben untereinander aufteilen können. Es kann der gewünschte Freiheits-

grad der eigenständigen Aufgabenaufteilung durch die Schüler genau spezifiziert werden. Ganz ohne Rollen wäre es möglich, dass ein Lernender alle Aufgaben alleine erledigt, was oft nicht dem gewünschten Lernprozess entspricht. Das neue Rollenkonzept erlaubt es insbesondere zu spezifizieren, wann nicht alle Personen einer Lerngruppe für die jeweiligen Aufgaben benötigt werden, so dass im Petrinetz Modell parallel aktivierte Aufgaben auch parallel bearbeitet werden können, wenn die benötigten Lernenden zur Verfügung stehen. Die Struktur des Petrinetzes modelliert wieder die dem kollaborativen Lernprozess zugrundeliegende Abfolge der Aufgaben (Kontrollfluss) und verkompliziert sich im Vergleich zu Lernprozessen für einzelne Lernende nicht. Das hat den Vorteil, dass auch dieses Petrinetzmodell, ohne Rollenbeschriftungen, leicht aus entsprechenden Event Logs mithilfe von Process Mining Algorithmen berechnet werden kann.

3.2 Diskussion und Anforderungen an (semi-)automatisierte Generierung

Für die Entwicklung von Learnflow Mining Tools ist die - ohnehin im Feld des Educational Modelling diskutierte - Vereinheitlichung von Log Formaten ein erstrebenswertes Ziel. Ähnlich wie im Bereich des Workflow Minings [vdAea07] müssen Learnflow Mining Tools neben den eigentlichen Mining Algorithmen auch noch Vorverarbeitung der Event Logs (sog. Preprocessing) und Nachbearbeitung der erzeugten Prozessmodelle (sog. Postprocessing) unterstützen.

Beispielsweise wird ein Event Log, das alle Mausclicks protokolliert, zu feingranular sein, um beim Mining direkt ein sinnvolles Modell zu erzeugen. Eine Kombination aus Filtertechniken, Aggregation von Ereignissen zu übergeordneten Aufgaben und Abstraktionen ist in diesem Fall nötig, um die Event Logs für den Mining Prozess geeignet vorzubereiten.

Da die durch Mining erzeugten Modelle im Allgemeinen nicht bereits alle wesentlichen Aspekte enthalten, gehen wir davon aus, dass im Rahmen des Postprocessing insbesondere eine anschließende Validierung und / oder Korrektur durch den Experten / Dozenten unerlässlich ist. Diese ist besonders wertvoll, falls bei der Erzeugung des Modells aus konkreten Schüler-Logfiles häufige Fehler mit in das Modell integriert wurden: zum einen bietet sich für den Dozenten die Möglichkeit, eine gezielte Hilfestellung für den Fehler anzubieten, zum anderen ist es auch möglich diesen modellierten Ablauf gezielt zu entfernen, also dem Schüler die Fehlermöglichkeit gar nicht anzubieten. Auch hier sehen wir Parallelen zur Erzeugung von kognitiven Tutoren [AMSK06] aus Beispielabläufen.

Für diese Formen des Postprocessing ist darauf zu achten, dass die Modelle eine weiterverarbeitbare Form und darauf zugeschnittene Editoren haben sollten; in unserem Vorschlag wären also Petrinetzeditoren mit speziellen Hilfen für den Learnflow Designer anzubieten, um ein direktes Editieren auf XML-Repräsentationen zu vermeiden.

Zur technischen Umsetzung einer Learnflow Engine sehen wir zwei Möglichkeiten: Entweder wird ein entsprechender Learnflow Interpreter direkt in die Lernumgebung eingebaut, oder die Lernumgebung wird von einer externen und wiederverwendbaren Learnflow Engine angesteuert, d.h. eine gewisse Schnittstelle wird angeboten, die eine Konfiguration der Lernumgebung durch die lose gekoppelte Engine ermöglicht. Ein Beispiel für einen solchen Ansatz findet sich in unserem *remote control* Architekturvorschlag [HMW08],

dort unter Verwendung einer IMS/LD-basierten Engine und Learnflow Beschreibung. Für einen Petrinetz-basierten Ansatz ist eine analoge Lösung naheliegend.

4 Zusammenfassung

In diesem Artikel haben wir untersucht, welche Methoden und Techniken aus dem Workflow Management auf die Modellierung von Lernszenarien (Learnflows) übertragen werden können. Zunächst arbeiteten wir die Gemeinsamkeiten und Unterschiede der beiden Formen von Prozessen heraus und verglichen resultierende informatische Lösungen in den jeweiligen Bereichen. Als besonders vielversprechend für einen Methodentransfer sehen wir die Erzeugung von Prozessbeschreibungen aus realen Abläufen, das *Process Mining*, und schlagen die Nutzung im e-Learning als sog. *Learnflow Mining* vor. Durch die direkte Nutzung realer Lernabläufe und Generierung von - zu diesen Abläufen kompatiblen - allgemeineren Modellen ist es möglich, den Dozenten beim Design der Lernszenarien zu entlasten und die konkreten Lernerfahrungen unmittelbar zu nutzen (z.B. zur Unterstützung der Schüler im Lernszenarium). Diesen Ansatz illustrierten wir an zwei Fallbeispielen von Lehr-/Lernprozessen, einem individuellen und einem kollaborativen Lernszenarium.

Literatur

- [AMSK06] V. Alevan, B. M. McLaren, J. Sewell und K. R. Koedinger. The Cognitive Authoring Tools (CTAT): Preliminary Evaluation of Efficiency Gains. In *Proc. of ITS 2006, LNCS 4053*, Berlin, 2006. Springer.
- [Bar03] B. Barron. When Smart Groups Fail. *Journal of the Learning Sciences*, 12(3):307–359, 2003.
- [BDLM08] R. Bergenthum, J. Desel, R. Lorenz und S. Mauser. Synthesis of Petri Nets from Scenarios with VipTool. In *Angenommen bei ICATPN*, 2008.
- [BHD⁺04] M. L. Bote Lorenzo, D. Hernández Leo, Y. Dimitriadis, J. I. Asensio Pérez, E. Gómez Sánchez, G. Vega Gorgojo und L. M. Vaquero González. Towards Reusability and Tailorability in Collaborative Learning Systems Using IMS-LD and Grid Services. *Advanced Technology for Learning*, 1(3):129–138, 2004.
- [Des05] J. Desel. Process Modeling Using Petri Nets. In *Process-Aware Information Systems*, Seiten 147–177. Wiley, 2005.
- [Her07] D. Hernandez-Leo. *A pattern-based design process for the creation of CSCL macroscripts computationally represented with IMS LD*. Dissertation, University of Valladolid, 2007. European PhD Thesis.
- [HH08] A. Harrer und U. Hoppe. Visual Modelling of Collaborative Learning Processes – Uses, Desired Properties, and Approaches. In L. Botturi und S. T. Stubbs, Hrsg., *Handbook of Visual Languages for Instructional Design: Theory and Practices*, Seiten 281–298. Information Science Reference, Hershey, PA., 2008.

- [HM06] A. Harrer und N. Malzahn. Bridging the Gap - Towards a Graphical Modelling Language for Learning Designs and Collaboration Scripts of Various Granularities. In *ICALT 2006*, Seiten 296–300, Los Alamitos, CA, 2006. IEEE Computer Society.
- [HMW08] A. Harrer, N. Malzahn und A. Wichmann. The Remote Control Approach - An Architecture for Adaptive Scripting across Collaborative Learning Environments. *Journal of Universal Computer Science*, 14(1):148–173, 2008.
- [IMS03] IMS Global Consortium. IMS Learning Design XML Binding, 2003. 1.0 Specification.
- [KT05] R. Koper und C. Tattersall, Hrsg. *Learning Design - Modelling and implementing network-based education & training*. Springer, Berlin, 2005.
- [MVF⁺06] C. Martel, L. Vignollet, C. Ferraris, J. David und A. Lejeune. Modelling collaborative learning activities in e-learning platforms. In *ICALT 2006*, Seiten 707–709, Los Alamitos, CA, 2006. IEEE Computer Society.
- [NKA01] M. Nathan, K. R. Koedinger und M. Alibali. Expert blind spot: When content knowledge eclipses pedagogical content knowledge. AERA, Seattle, 2001.
- [Rei82] W. Reisig. *Petrinetze, Eine Einführung*. Springer, 1982.
- [Rod02] M. Rodriguez Artacho. PALO Language Overview. Bericht, Universidad Nacional de Educacion a Distancia, 2002.
- [RvK⁺02] A. Rawlings, P. van Rosmalen, R. Koper, M. Rodriguez Artacho und P. Lefrere. Survey of Educational Modelling Languages (EMLs), 2002. CEN/ISSS Learning Technology Workshop.
- [SA77] R. C. Schank und R. P. Abelson. *Scripts, Plans, Goals and Understanding: an Inquiry into Human Knowledge Structures*. L. Erlbaum, Hillsdale, NJ, 1977.
- [vdAea07] W. M. P. van der Aalst et al. ProM 4.0: Comprehensive Support for *Real* Process Analysis. In *ICATPN, LNCS 4546*, Seiten 484–494, 2007.
- [vdAvDH⁺03] W. M. P. van der Aalst, B. F. van Dongen, J. Herbst, L. Maruster, G. Schimm und A. J. M. M. Weijters. Workflow Mining: A Survey of Issues and Approaches. *Data Knowl. Eng.*, 47(2):237–267, 2003.
- [vdAvH02] W.M.P. van der Aalst und K. van Hee. *Workflow Management: Models, Methods, and Systems*. MIT Press, 2002.
- [VMN⁺06] H. Vogten, H. Martens, R. Nadolski, C. Tattersall, P. van Rosmalen und R. Koper. CopperCore Service Integration - Integrating IMS Learning Design and IMS Question and Test Interoperability. In *ICALT 2006*, Seiten 378–382, Los Alamitos, CA, 2006. IEEE Computer Society.
- [WEFM05] A. Weinberger, B. Ertl, F. Fischer und H. Mandl. Epistemic and social scripts in computer-supported collaborative learning. *Instructional Science*, 33(1):1–30, 2005.
- [Wes07] M. Weske. *Business Process Management – Concepts, Languages and Architectures*. Springer, 2007.