

Ein Autorenwerkzeug zur Konfiguration von eLearning-Argumentationsframeworks

Frank Loll
Niels Pinkwart

Veröffentlicht in:
Multikonferenz Wirtschaftsinformatik 2012
Tagungsband der MKWI 2012
Hrsg.: Dirk Christian Mattfeld; Susanne Robra-Bissantz



Braunschweig: Institut für Wirtschaftsinformatik, 2012

Ein Autorenwerkzeug zur Konfiguration von eLearning-Argumentationsframeworks

Frank Loll, Niels Pinkwart

Technische Universität Clausthal, Institut für Informatik, 38678 Clausthal-Zellerfeld,
E-Mail: {frank.loll|niels.pinkwart}@tu-clausthal.de

Abstract

Argumentationsfertigkeiten sind essentiell in vielen Bereichen des Lebens. Dies gilt sowohl auf privater als auch auf geschäftlicher Ebene. Daher spielt die Ausbildung von Argumentationsfertigkeiten eine zentrale Rolle. Eine Möglichkeit, diese Ausbildung zu unterstützen, sind computerbasierte Argumentationssysteme, die sich jedoch oftmals als zu unflexibel für den Einsatz über Domänengrenzen hinweg herausgestellt haben. Dieser Beitrag beschreibt das für verschiedene Domänen speziell konfigurierbare LASAD Argumentationsframework und zeigt, dass komplexe Konfigurationsmechanismen durch entsprechende Hilfsmittel beherrschbar bleiben können.

1 Einleitung

Argumentationsfertigkeiten sind essentiell in vielen Bereichen des Lebens, nicht zuletzt im geschäftlichen Umfeld, wo Entscheidungen weitreichende und kostenintensive Konsequenzen haben. Nichtsdestotrotz scheitern viele Menschen daran, begründete Argumente zu formulieren, Alternativen abzuwägen und geeignete Konsequenzen zu ziehen [6]. Dementsprechend wichtig ist die Ausbildung von Argumentationsfertigkeiten. Klassische, passive Lehrmethoden jedoch haben sich als wenig effektiv herausgestellt im Vergleich zu interaktiven Methoden wie beispielsweise Gruppendiskussionen und aktive Mitgestaltung des Unterrichts [2]. Dennoch birgt die Ausbildung mittels interaktiven Lehrmethoden auch Schattenseiten. Typischerweise ist diese Art der Ausbildung für größere Gruppen aufgrund von personellen und zeitlichen Beschränkungen kaum anwendbar. Einen Ansatz dazu, dieser Probleme Herr zu werden, bieten Argumentationssysteme [11] [4], d.h. eLearning-Werkzeuge, die Gruppen von Lernwilligen dabei unterstützen, sich aktiv an Diskussionen untereinander zu beteiligen und gleichzeitig das Lehrpersonal entlasten. Im Rahmen eines ausführlichen Reviews bestehender Ansätze in diesem Bereich [11] wurden offene Herausforderungen in zwei Richtungen deutlich. Einerseits ist ein flexibles Argumentationssystem notwendig, um systematisch einzelne Faktoren von computergestützter Argumentation, deren Auswirkungen auch heute noch zu großen Teilen unklar sind, empirisch zu untersuchen. Andererseits müssen Barrieren, die einen Einsatz derartiger Systeme

in der Praxis verhindern, abgebaut werden. Um die erste Herausforderung (ein flexibles Argumentationssystem) zu meistern, wurde das LASAD Argumentationsframework entwickelt [8] [9], welches im folgenden Abschnitt kurz beschrieben wird. In diesem Beitrag hingegen steht die zweite Herausforderung (der Abbau von Einsatzbarrieren, hier speziell der Konfigurationsaufwand) im Vordergrund. Aus diesem Grund wird im Rahmen eines Design Science Ansatzes [3] die Entwicklung und Evaluation eines Autorenwerkzeugs beschrieben, welches es zum Ziel hat, auch unerfahrene Nutzer von den Flexibilisierungsmechanismen im LASAD-System profitieren zu lassen.

2 Technisches Rahmensystem

2.1 Das LASAD Argumentationsframework

Bestehende Argumentationssysteme unterscheiden sich in einer Vielzahl von Kriterien, u.a. in der Repräsentation von Argumentationsstrukturen, dem Interaktionsdesign und den zugrundeliegenden Argumentationsmodellen [11]. Nichtsdestotrotz sind die konkreten Auswirkungen einzelner Faktoren weitestgehend unklar [11]. Um die Implikationen dieser Faktoren zu evaluieren, wurde das LASAD Argumentationsframework entwickelt [10] [9] [8]. LASAD bietet eine webbasierte, domänenunabhängige, hochflexible Plattform, die es auf Basis von XML-Konfigurationen ermöglicht, einzelne Faktoren zu manipulieren. Ein konkreter Anwendungsfall, der den Einfluss von unterschiedlichen Argumentationsmodellen und Kollaboration auf die Ergebnisse von computergestützter Argumentation untersucht, ist in [7] beschrieben. Eine Beispielkonfiguration des Frameworks ist in Bild 1 dargestellt.

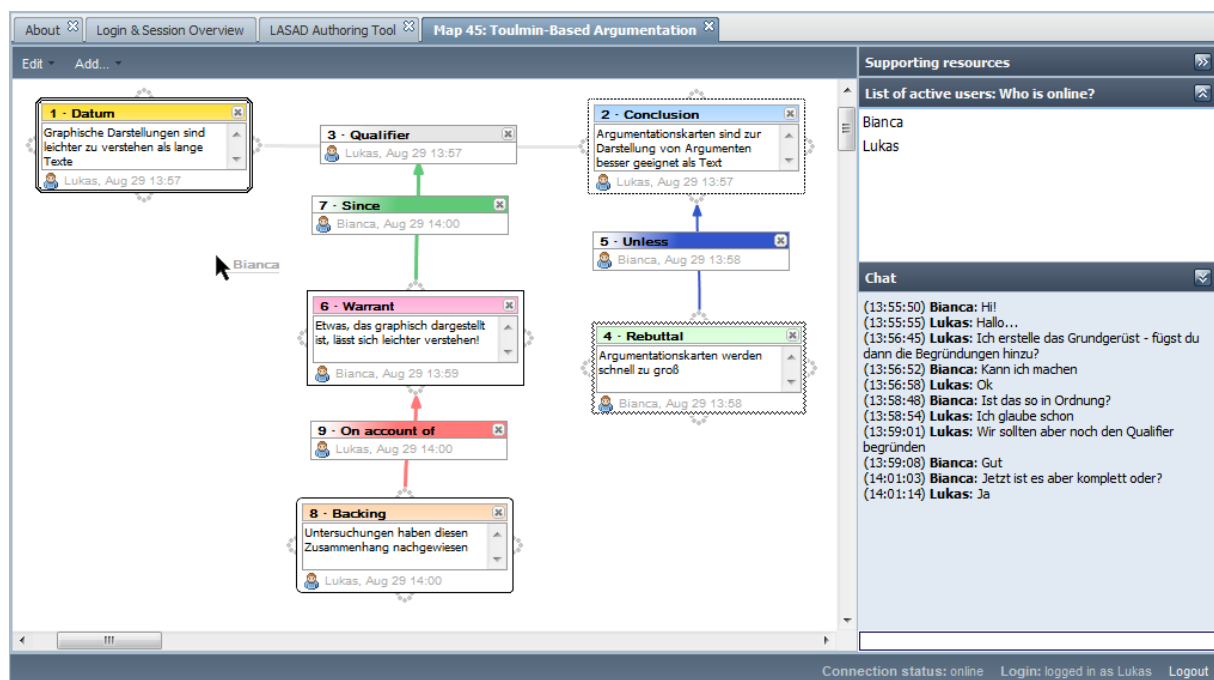


Bild 1: Beispielhafte Gruppenargumentation zur Darstellung eines Arguments nach Toulmin in LASAD [12]

2.2 Konfigurationsmöglichkeiten von LASAD

Eine *Konfiguration* des LASAD Argumentationsframeworks besteht aus vier Teilen. Der erste Teil ist die Definition von *Nutzern* in Verbindung mit einer Menge von *Rechten* (z.B. die Erlaubnis, neue Argumentationsteile zu erstellen oder alte zu löschen). Der zweite Teil ist die Definition des zugrundeliegenden Argumentationsmodells (*Ontology*) in Form von vordefinierten Argumentstrukturen (z.B. die Boxen und Beziehungen in Bild 1). Jede Argumentstruktur besteht wiederum aus einem oder mehreren *Elementen* (z.B. ein Textfeld, ein externer Weblink, eine Auswahlmöglichkeit, etc.). Aufbauend auf der Definition des Argumentationsmodells erfolgt eine Definition der verfügbaren Benutzerinterfaceelemente sowie des Argumentationskontexts (*Template*). Hierbei wird ausgewählt, welche Komponenten (z.B. ein gegebener Hintergrundtext zur internen Verlinkung oder ein Chat) verfügbar sind und was für eine Art von Argumentation unterstützt werden soll (z.B. individuelle Argumentation oder Argumentation in kleinen bzw. großen Gruppen). Abschließend wird eine konkrete Instanz, eine sog. *Session*, erzeugt, die die vorherigen Definitionen nutzt. Auf diesem Weg können beispielsweise mehrere Gruppen parallel und unabhängig voneinander dieselbe Konfiguration nutzen. Durch die Kombination mehrerer Konfigurationen ist es zudem möglich, geskriptete Ansätze zur Lernunterstützung [1] umzusetzen. Ein Überblick über den internen Zusammenhang der Konfigurationsteile ist in Bild 2 gegeben. Für eine detailliertere Beschreibung der Konfigurationsmechanismen mit ausführlichen Beispielen sei auf [8] und [9] verwiesen.

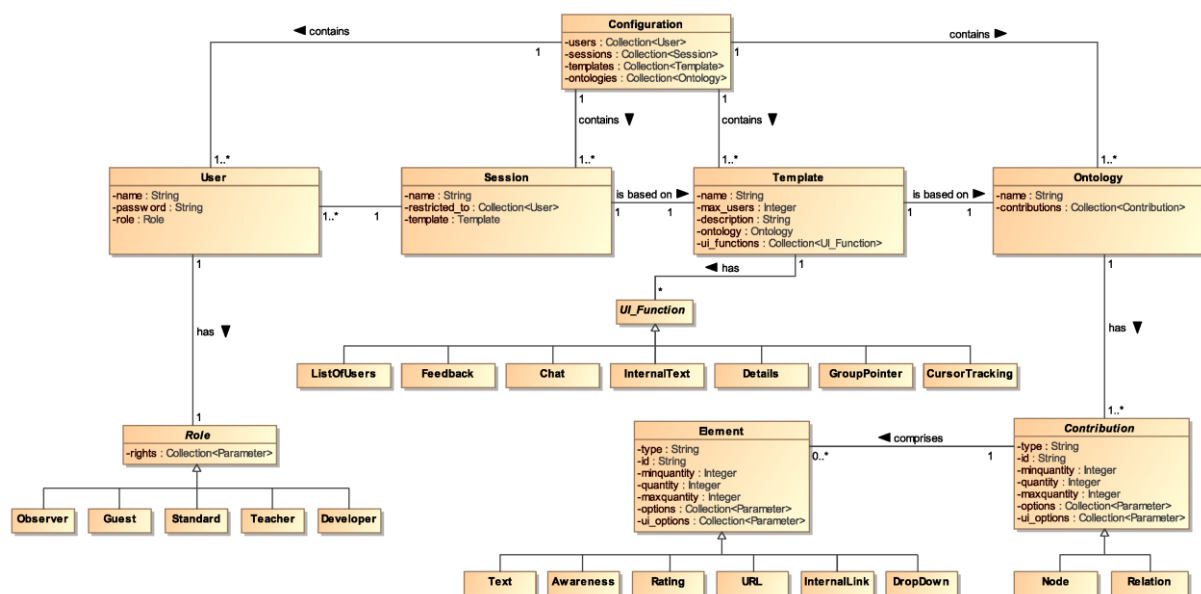


Bild 2: Überblick über Konfigurationsmöglichkeiten in LASAD

3 Autorenwerkzeug

Die in 2.2 beschriebenen Konfigurationsmöglichkeiten lassen sich nutzen, um das LASAD-System an die Anforderungen bestimmter Lernszenarien oder Domänen anzupassen. Um diese Anpassungen vorzunehmen, war bisher eine Konfiguration via XML notwendig. In Anbetracht der Tatsache, dass zu den Zielgruppen von LASAD neben Entwicklern vor allen Dingen Forscher und Lehrende mit nicht-informatikaffinem Hintergrund gehören, ist eine solche Konfigurationsmöglichkeit jedoch problematisch. Aus diesem Grund sollte ein Autorenwerkzeug in Form eines graphischen Editors entwickelt werden.

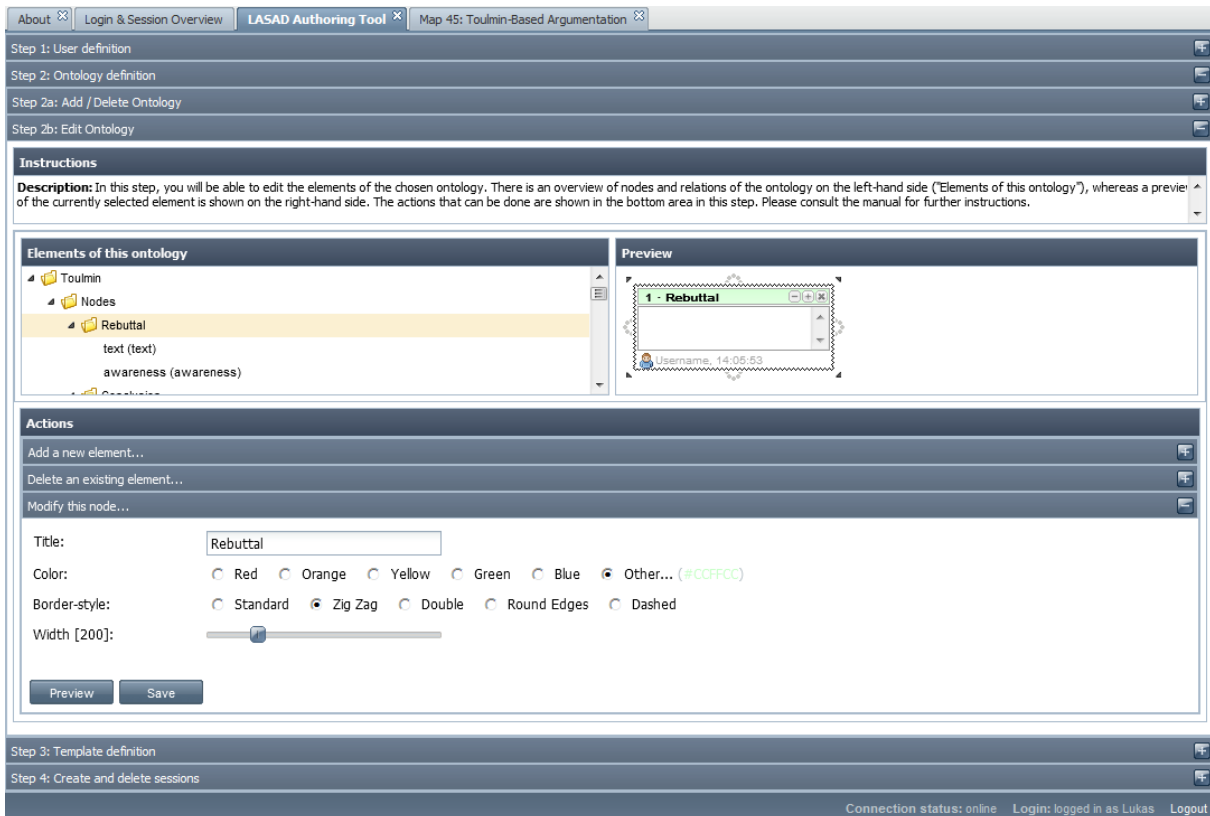


Bild 3: Screenshot des webbasierten Autorenwerkzeugs in LASAD

3.1 Anforderungsanalyse

Die zentralen Anforderungen an ein möglichst universell einsetzbares Autorenwerkzeug in LASAD, basierend auf Gesprächen mit ca. 100 Entwicklern, Lehrenden und Forschern unterschiedlicher Disziplinen (im Kontext von zwei Jahren Entwicklungszeit des LASAD-Systems), waren die folgenden:

- A1. *Konfiguration ohne Programmierkenntnisse:* Zu den wichtigsten Zielgruppen von LASAD gehören, neben den Entwicklern, Forscher und Lehrende verschiedenster Disziplinen. Hier können Programmierkenntnisse nicht vorausgesetzt werden. Aus diesem Grund muss eine Konfiguration unabhängig von der Code-Ebene durchführbar sein.
- A2. *Minimierung des Trainingsbedarfs:* Typische Einstiegsbarrieren, die eine aktive Nutzung von Software (insbesondere im Lernbereich) verhindern sind hohe Komplexität und damit verbunden ein hoher Trainingsbedarf. Letzteren gilt es daher zu minimieren.
- A3. *Nutzen vs. Komplexität:* Die Konfigurationsmöglichkeiten von LASAD sind umfangreich und müssen dies auch sein, um das System domänenunabhängig einsetzen zu können. Einige dieser Konfigurationsmöglichkeiten, beispielsweise die Erstellung von vordefinierten Argumentstrukturen, sind jedoch nur in speziellen Anwendungsfällen notwendig. Hier muss für das Autorensystem ein Kompromiss gefunden werden, der einerseits die wichtigsten Funktionen beinhaltet und andererseits die Komplexität so gering wie möglich hält.

3.2 Umsetzung

Basierend auf den genannten Anforderungen wurde ein erster Prototyp des Autorenwerkzeugs entwickelt. Um den Zugriff möglichst einfach zu gestalten, wurde das Autorenwerkzeug direkt in LASAD integriert (s. Tabs in Bild 1 & 3). Innerhalb des Autorenwerkzeugs ist der gesamte Konfigurationsprozess in Form von einzelnen Schritten gegliedert, die sich der Reihe nach bearbeiten lassen. Auf diesem Weg werden die Nutzer durch das System geleitet. Innerhalb der Schritte sind kurze Instruktionen präsentiert, die einen Überblick darüber geben, was in dem aktuellen Schritt konfiguriert werden kann. Die eigentliche Konfiguration erfolgt nach dem Baukastenprinzip mittels graphischer Hilfsmittel, die Programmierkenntnisse unnötig machen (→ A1). Hierbei werden einzelne Konfigurationsteile mittels webtypischer Formularstrukturen erstellt. Um die Auswirkungen der einzelnen Schritte zu verdeutlichen werden jeweils graphische Vorschauen gegeben (s. Bild 3 rechts). Zusammengenommen soll auf diesem Weg der notwendige Trainingsaufwand minimiert werden (→ A2). Zusätzlich dazu wurden die einzelnen Bausteine des Autorenwerkzeugs auf die Funktionen beschränkt, die entweder unverzichtbar für die Konfiguration sind oder bei einem Großteil der existierenden Systeme vorhanden sind (ein Überblick über die Funktionen ist gegeben in [11], → A3).

4 Ansatz zur Evaluation des Autorenwerkzeugs

4.1 Forschungsfragen

Im Rahmen der Evaluation des Nutzens des Autorenwerkzeugs standen drei Faktoren im Vordergrund, die direkt die Forschungsfragen motivieren:

- F1. *Benutzerfreundlichkeit*: Ist das Autorenwerkzeug selbsterklärend genug um ein zeitintensives Training zu vermeiden?
- F2. *Effektivität*: Können auch unerfahrene Nutzer LASAD ihren Wünschen anzupassen?
- F3. *Effizienz*:
 - a. Wie lange dauert es, das LASAD Argumentationsframework mit Hilfe des Autorenwerkzeugs zu konfigurieren?
 - b. Ist eine Zeitersparnis gegenüber einer direkten XML-Konfiguration vorhanden?

4.2 Forschungsmethodik

Da es sich um eine Erstentwicklung handelte und keine in der Funktionalität vergleichbaren Systeme existieren, waren keine direkten Vergleiche mit vorherigen Versionen oder Konkurrenzprodukten anhand typischer Kennziffern wie Zeit, Fehlerrate, etc. möglich. Aus diesem Grund erfolgte die Beantwortung der Fragen in einem ;Mehrphasen-Ansatz. Jede Phase wurde durch einen iterativen Entwicklungsschritt abgeschlossen, d.h. der Prototyp wurde anhand der Resultate entsprechend angepasst oder erweitert.

Die erste Phase (Pilottest) hierbei bestand aus einem Test mit erfahrenen LASAD Nutzern. Hierbei wurde das Autorensystem an drei Nutzer, die mit den Fähigkeiten und der Konfiguration von LASAD vertraut waren, gegeben. Diese bekamen die Aufgabe, eine bestehende Konfiguration nachzubauen. Als einziges Hilfsmittel wurde ein Handbuch an die Nutzer ausgegeben, in dem die Bedienung des Autorenwerkzeugs beschrieben war. Im Rahmen eines offenen

Interviews im Anschluss wurden potentielle Schwachstellen diskutiert und Verbesserungsvorschläge erarbeitet. Die erste Phase hatte folglich zum Ziel zu überprüfen, ob die Möglichkeiten, die der Prototyp des Autorenwerkzeugs bot, für eine praktische Nutzung ausreichen.

Die zweite Phase (Laborstudie) diente zum Transfer der Resultate aus Phase eins von erfahrenen LASAD Nutzern zu unerfahrenen. Hierzu wurde eine kontrollierte Laborstudie durchgeführt. Das freiwillige Teilnehmerfeld bestand aus zehn Studenten der Fächer Mathematik und Informatik, die weder das LASAD-System noch ein anderes Argumentationssystem zuvor benutzt hatten. Die Studie bestand aus drei Teilen. Den ersten Teil bildete die Trainingsphase. Hier bekamen alle Teilnehmer vom gleichen Experimentator eine kurze fünfminütige Einführung in LASAD. Anschließend wurde den Teilnehmern ein einführendes Tutorial ausgehändigt, welches ein Beispiel einer Konfiguration in Form einer Schritt-für-Schritt-Anleitung enthielt. Hierfür hatten die Studenten 15 Minuten Zeit. Sofern Fragen auftraten, wurden diese vom Experimentator beantwortet. Der zweite Teil der Studie bestand aus zwei konkreten Konfigurationsaufgaben. Diese unterschieden sich in der Art der Aufgabenstellung. Während die erste Aufgabe (T_1) analog zu den Schritten im Autorenwerkzeug strukturiert war (s. Anhang 8.2), bestand die zweite Aufgabenstellung (T_2) aus Fließtext (s. Anhang 8.3), aus dem die notwendigen Konzepte selbstständig zu extrahieren waren. Die Komplexität der Aufgabenstellung wie Lösung war für beide Aufgaben ähnlich, d.h. die Fehlerraten sind vergleichbar. Für beide Aufgaben wurde die Zeit gestoppt; es gab kein Zeitlimit. Um potentielle Trainingseffekte zu vermeiden, wurde die Reihenfolge der Aufgaben zwischen den Teilnehmern variiert. Während der Aufgabenbearbeitung wurden keinerlei Fragen beantwortet. Stattdessen erhielten die Teilnehmer das Handbuch ausgehändigt, um gegebenenfalls nachschlagen zu können. Alle Sessions wurden mit Hilfe von Screen-Capturing Verfahren aufgezeichnet, um potentielle Schwachpunkte mit Blick auf die Nutzbarkeit auch im Nachhinein erkennen zu können. Nach erfolgreicher Bearbeitung der Aufgaben wurden die Nutzer auf ihre Fehler aufmerksam gemacht und es wurde die Möglichkeit gegeben, diese zu beheben. Abschließend fand ein offenes Interview statt, in dem Probleme und Verbesserungsvorschläge gesammelt wurden, die in den nächsten Entwicklungsschritt einfließen.

Die dritte Phase (Expertenbefragung) bildete einen weiteren Schritt zur Verallgemeinerung der Ergebnisse. Während die studentischen Teilnehmer an der Laborstudie typischerweise sehr gute allgemeine IT-Kenntnisse haben, trifft dies nicht auf alle Zielgruppen des Autorenwerkzeugs zu – insbesondere nicht auf alle Wissenschaftler und Lehrenden. Aus diesem Grund wurde ein webbasierter Test (s. Anhang 8.1) mit Argumentationsexperten verschiedener Domänen durchgeführt. Die Auswahl der Experten erfolgte auf Basis eines vorher durchgeführten System-Reviews [11], d.h. es wurde eine Liste mit Experten zusammengestellt, die direkt in Verbindung mit bestehenden Systemen standen. Anschließend wurde die Liste erweitert, indem die Autorenlisten von Konferenzen und Journals, die sich mit computergestützter Argumentation beschäftigen, analysiert wurden. Insgesamt enthielt die Liste 153 Experten. Von diesen 153 Experten wurden in diesem Fall lediglich jene 100 Experten befragt, die in einer anderen vorher durchgeführten Studie [9] bereit gewesen waren, Antworten zu geben. Die Teilnahme war freiwillig und konnte anonym erfolgen. Als Motivation wurde ein Apple iPod Nano unter allen Teilnehmern verlost. Der Test bestand aus zwei Teilen: einer selbstständigen Arbeit mit LASAD und dem Autorenwerkzeug sowie einer webbasierten Umfrage (s. Anhang 8.1). Alle Experten wurden persönlich per E-Mail zur Teilnahme eingeladen und bekamen ein ausgearbeitetes Handbuch mit einem illustrierenden Schritt-für-Schritt-Beispiel als Hilfsmittel.

5 Forschungsergebnisse

5.1 Ergebnisse des Pilottests (Phase 1)

Im Rahmen des offenen Interviews stimmten alle Nutzer darin überein, dass das Autorenwerkzeug ein geeignetes Hilfsmittel sei, den Konfigurationsprozess von LASAD zu unterstützen. Verbesserungsvorschläge wurden sowohl für das Handbuch als auch für das Autorenwerkzeug vorgebracht. Im darauffolgenden nächsten iterativen Entwicklungsschritt wurde ersteres um Screenshots aller zur Verfügung stehender Elemente ergänzt, während letzteres um die Möglichkeit ergänzt wurde, die Argumentationsmodelle (Ontologies) zu klonen, um die Modifikation kleinerer Teile zu erleichtern. In nachfolgenden Meetings mit den Nutzern wurde deutlich, dass das Autorenwerkzeug die klassische Konfiguration via XML verdrängte, da die Konfiguration darüber weniger fehlerträchtig und daher effizienter war.

5.2 Ergebnisse der Laborstudie (Phase 2)

Alle Teilnehmer der Laborstudie waren in der Lage, die gegebenen Aufgaben ohne Hilfe des Tutors zu lösen. Um die Effizienz des Ansatzes zu überprüfen, wurden die benötigte Zeit sowie die Fehlerrate evaluiert. Als Fehler wurden alle Abweichungen von der Aufgabenstellung gezählt. Im Durchschnitt brauchten die Teilnehmer etwa 10,5 Minuten ($m=638,8$; $sd=182,82$ in Sekunden) für Aufgabe T₁ (strukturiert), wohingegen sie etwa 16 Minuten ($m=951,6$; $sd=154,47$ in Sekunden) für Aufgabe T₂ (unstrukturiert) benötigten. Beide Zeiträume sind angemessen, insbesondere da eine direkte XML-Konfiguration erheblich länger gedauert hätte. Eine vertiefende Analyse zeigte zudem, dass die jeweils zweite Konfigurationsaufgabe (gemäß variierender Reihenfolge) tendenziell weniger Bearbeitungszeit erforderte, wie in Bild 4 dargestellt. Dies legt den Schluss nahe, dass zusätzliches Training die durchschnittliche Bearbeitungszeit weiter gesenkt hätte.

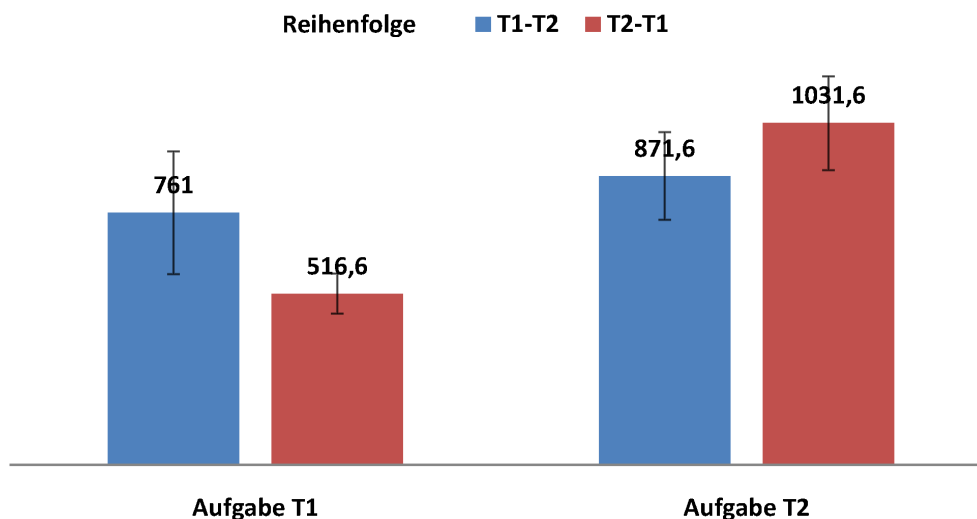


Bild 4: Durchschnittlich benötigte Zeit (in Sekunden) für die Bearbeitung der Konfigurationsaufgaben in Abhängigkeit der Reihenfolge

Die Fehlerrate war äußerst gering ($m(T_1)=0,4$; $sd(T_1)=0,7$; $m(T_2)=1,2$; $sd(T_2)=0,7$) und alle Teilnehmer waren in der Lage, Fehler selbstständig zu beheben nachdem sie darauf hingewiesen wurden. Als Fehlerursache wurde von den Teilnehmern ausnahmslos schlichtes Überlesen von Anweisungen (insbesondere im unstrukturierten Fall) angegeben.

Basierend auf den Screen-Capturings und dem abschließenden offenen Interview wurde ein weiterer iterativer Entwicklungsschritt durchgeführt. Zu den wichtigsten Änderungen hierbei gehörte das Hinzufügen einer nachträglichen Modifikationsmöglichkeit für das zugrundeliegende Argumentationsmodell (z. B. Ändern einer Argumentationsstruktur).

5.3 Ergebnisse der Expertenbefragung (Phase 3)

Insgesamt beteiligten sich 19 Argumentationsexperten (4 davon anonym) an der Umfrage. Eine Analyse der Logdateien sowie der Antworten zeigte jedoch, dass nicht alle Experten das Autorenwerkzeug genutzt haben. Stattdessen nutzten einige ausschließlich die Argumentationskomponente von LASAD. Insbesondere unter den 4 anonymen Antworten ist die Unterscheidung zwischen den Antwortenden, die das Autorenwerkzeug nutzten und jenen, die es nicht taten, nicht eindeutig, da sich eine Nutzung nicht aus dem Log nachprüfen lässt. Aus diesem Grund wurde die Analyse auf die offen gestellten Fragen beschränkt.

An Hand der Logfiles konnte zunächst festgestellt werden, dass keiner der Experten das einleitende Beispiel durchgelesen hat – jegliche Systemnutzung erfolgt also ohne Training bzw. Schulung.

Insgesamt identifizierten die Experten zwei Hauptprobleme. Das erste Problem betraf die graphische Oberfläche des Autorenwerkzeugs. Hier bemerkten einige Experten, dass aus Platz- bzw. Auflösungsgründen nicht alle Elemente zu jeder Zeit lesbar waren. Dies traf insbesondere auf die Instruktionen zu, die in jedem Schritt innerhalb des Tools präsentiert waren (s. Bild 3). Das zweite Problem ist verbunden mit dem mangelnden Training und bezog sich auf die verwandte Terminologie. Wenngleich die Konzepte von Ontology, Template und Session im Handbuch erklärt wurden, hatten viele Experten Probleme damit. *„It's rare that folks will read manuals“* brachte es ein Experte hierbei umgangssprachlich auf den Punkt.

Zusätzlich zu den berichteten Problemen gab es funktionelle Verbesserungsvorschläge. Diese beinhalteten Modifikationen an dem zugrundeliegenden Argumentationsmodell sowie Umbenennungsfunktionen für Konfigurationsteile (z. B. Sessionnamen).

Nichtsdestotrotz hatte rund ein Drittel der Befragten keinerlei Probleme. Zusätzlich fragten einige Experten in den Freitextkommentaren explizit danach, LASAD und das Autorenwerkzeug für ihre Zwecke einzusetzen, da die Konfigurationsmöglichkeiten begrüßt wurden.

5.4 Diskussion & Limitierungen

Im Rahmen der Evaluation wurden drei Hauptfaktoren evaluiert: Benutzerfreundlichkeit, Effektivität und Effizienz.

Bezogen auf die Benutzerfreundlichkeit (→ F1) zeigten sich gemischte Ergebnisse. Einerseits hatten die Studenten im Rahmen der Laborstudie keinerlei Probleme, LASAD mit Hilfe des Autorenwerkzeugs zu konfigurieren. Andererseits fiel es den Experten schwer. Die Ursache hierfür kann im fehlenden Training liegen. Während die Studenten allesamt eine kurze fünfminütige Einführung von einem Tutor erhielten und ein Schritt-für-Schritt Tutorial absolvierten, traf das auf keinen der Experten zu, obwohl diese Zugriff auf das Handbuch, welches beides enthielt, hatten. Wenngleich es bekannt ist, dass das Lesen von Handbüchern allgemein unbeliebt ist, ist es dennoch bemerkenswert, dass selbst Experten den Aufwand scheuen, sich in ein spezielles Werkzeug einzuarbeiten, obwohl dieses speziell dazu entworfen wurde, ihnen zu helfen.

In Bezug auf die anderen beiden Faktoren, Effektivität (→ F2) und Effizienz (→ F3), zeigte sich ein positives Bild. Die Pilottester bevorzugten nach kurzer Zeit das Autorenwerkzeug gegenüber der manuellen XML-Konfiguration, was darauf schließen lässt, dass die Konfiguration mittels des Autorenwerkzeugs effektiv und für den täglichen Einsatz geeignet ist. Dies wurde auch durch die Laborstudie bestätigt. Hier waren alle Teilnehmer in der Lage, die gegebenen Konfigurationsaufgaben annähernd fehlerfrei durchzuführen. Mit Blick auf die Effizienz waren die notwendigen Zeiträume (10,5 bzw. 16 Minuten, → F3a) einer manuellen Konfiguration (die typischerweise zw. 30 und 60 Minuten – abhängig von der Erfahrung des Entwicklers und dem Grad der Wiederverwendung bestehender Konfigurationen – benötigt) um Längen überlegen (→ F3b), sodass das Autorenwerkzeug auch hierbei überzeugte.

Insgesamt zeigte das Autorenwerkzeug damit seine Eignung zur vereinfachten Konfiguration von LASAD. Nichtsdestotrotz existieren einige Einschränkungen, die bei der Auswertung der Ergebnisse berücksichtigt werden sollten. Die Anzahl der Teilnehmer, insbesondere in der Laborstudie, war verhältnismäßig gering ($n=10$). Relativierend sollte man hierbei jedoch festhalten, dass es keineswegs um einen Fehlertest ging, für den typischerweise größerer Aufwand getrieben werden muss, sondern um das Auffinden von Usability Schwächen. Der Literatur [5] zufolge reichen bereits 3 bis 4 Teilnehmer, um hier die größten Schwächen zu identifizieren. Die Teilnehmer der Laborstudie waren allesamt technikaffin, was die Ergebnisse positiv beeinflussen haben könnte. Die Experten waren dies nicht und repräsentieren unterschiedliche Domänen – jedoch waren auch hier einige Domänen (z.B. Jura) stärker als andere (z.B. Ethik) vertreten.

6 Zusammenfassung und Ausblick

In diesem Beitrag wurden (methodisch im Rahmen eines Design Science Ansatzes) der Entwurf sowie die Evaluierung eines Prototyps eines Autorenwerkzeugs gezeigt. Klassische Evaluationsmethoden, welche auf direkten Vergleichen zu Vorgängerversionen oder manuellen Konfigurationen beruhen, ließen sich nicht anwenden. Stattdessen wurde ein mehrstufiger Evaluationsansatz präsentiert und durchgeführt.

Mit Blick auf das entstandene Softwareartefakt wurden die meisten der Entwurfsziele erreicht. Insgesamt zeigte das Autorenwerkzeug seine Eignung als Konfigurationshilfe für LASAD. Ein Ziel, welches nicht gänzlich erreicht wurde, besteht in der Eliminierung von notwendigem Training. Um die notwendige Zeit weiter zu verkürzen, sehen zukünftige Entwicklungen vor, das textuelle Handbuch durch eine kurze Videoeinführung zu ersetzen und komplizierte Begrifflichkeiten wie Ontology und Template durch intuitivere auszutauschen. In Verbindung mit einer kontextsensitiven Hilfsfunktion soll so die Usability erhöht werden. Die zukünftige Weiterentwicklung des Autorenwerkzeugs sieht eine Erweiterung um eine Skriptengine vor. Im Zuge dieser Erweiterung ist es notwendig, das Autorenwerkzeug um die Möglichkeit der Definition von Skripten zu erweitern. Zusätzlich dazu sollen weitere bestehende Möglichkeiten des LASAD-Argumentationsframeworks in das Autorenwerkzeug integriert werden, ohne dessen Benutzbarkeit zu verschlechtern. Einen Schwerpunkt bildet hierbei die Integration von externen Analyse- und Feedbackclients.

Über die in diesem Artikel konkret nachgewiesene Möglichkeit hinaus, hochflexible eLearning-Argumentationsframeworks über Autorensysteme auch durch Nicht-Informatiker konfigurierbar zu gestalten, liefert dieser Artikel auch zwei weitere Erkenntnisse. Ein interessantes Ergebnis,

das auch für zukünftige andere Entwicklungen relevant sein kann, besteht in der offenbar generellen Ablehnung gegenüber einer Einarbeitung in Form von Handbüchern in ein spezielles Werkzeug seitens der von uns involvierten Experten, obwohl diese durchaus eine Motivation zur Systemnutzung haben sollen – dies unterstreicht die hohe Relevanz von einfachen, selbst-erklärenden Benutzerschnittstellen auch für Expertensysteme. Ein zweites Resultat ist die verwendete Methodik – der dreistufige Ansatz bestehend aus Labor- und Feldstudien mit unterschiedlichen Teilnehmern hat sich als tragfähig erwiesen, wissenschaftliche Erkenntnisse zum getesteten Prototyp zu gewinnen wie auch gleichzeitig zu dessen Weiterentwicklung beizutragen.

7 Literatur

- [1] Fischer, F; Kollar, I; Mandl, H; Haake, JM (2006). Scripting Computer-Supported Collaborative Learning. Springer.
- [2] Hake, RR (1998): Interactive-engagement versus traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses. *American Journal of Physics*, 66(1).
- [3] Hevner, AR; March, ST; Park, J; Ram, S (2004): Design Science in Information Systems Research. *MIS Quarterly* 28(1):75-105.
- [4] Kirschner, PA; Buckingham Shum, SJ; Carr, CS (Eds.) (2003): Visualizing Argumentation: Software Tools for Collaborative and Educational Sense-Making.
- [5] Krug, S (2010): *Web Usability: Rocket Surgery Made Easy*. Addison-Wesley.
- [6] Kuhn, D (1991): *The skills of argument*. Cambridge University Press.
- [7] Loll, F; Pinkwart, N (2011): Guiding the Process of Argumentation: The Effects of Ontology and Collaboration. In: Spada, M; Stahl, G; Miyake, N; Law, N (Eds.), *Connecting Computer-Supported Collaborative Learning to Policy and Practice: CSCL2011 Conference Proceedings. Volume I – Long Papers*: 1437-1448. Hong Kong.
- [8] Loll, F; Pinkwart, N (2010): Ein generisches Framework zur Erstellung von argumentationsunterstützenden Systemen. In: Schumann, M; Kolbe, LM; Breitner, MH (Hrsg.), *Tagungsband der Multikonferenz Wirtschaftsinformatik 2010*:1437-1448. Göttingen.
- [9] Loll, F; Scheuer, O; McLaren, BM, Pinkwart, N (2010): Learning to Argue Using Computers - A View from Teachers, Researchers, and System Developers. In: V. Aleven, J. Kay, J. Mostow, (Eds.), *Lecture Notes in Computer Science (6095) - Proceedings of the 10th International Conference on Intelligent Tutoring Systems (ITS)* (pp. 377 - 379). Berlin, Germany, Springer Verlag.
- [10] Loll, F; Pinkwart, N; Scheuer, S; McLaren, BM (2011): How tough should it be? Simplifying the development of argumentation systems using a configurable platform. In: Pinkwart, N; McLaren, BM (Eds.), *Educational Technologies for Teaching Argumentation Skills*. Bentham Science Publishers Ltd.
- [11] Scheuer, O; Loll, F; Pinkwart, N; McLaren, BM (2010): Computer-supported argumentation: A review of the state of the art. *International Journal of Computer-Supported Collaborative Learning* 5(1):43-102.
- [12] Toulmin, SE (1958): *The uses of argument*. Cambridge University Press.

8 Anhang

8.1 Expertenumfrage – Fragebogen

1. I have been able to configure the LASAD framework by means of the authoring tool.
2. I have been able to configure the LASAD framework by means of the authoring tool 'in an adequate time.
3. I think the authoring tool of the LASAD framework is easy to use.
4. Did you experience any problems? Can you describe the problems?
5. How do you think can the authoring tool be improved?
6. Do you have any additional comments?

Die Fragen 1-3 mussten auf folgender Skala beantwortet werden: 1 = *Strongly agree*, 2 = *Disagree*, 3 = *Neither agree nor disagree*, 4 = *Agree*, 5 = *Strongly agree*. Die Fragen 4-6 waren offen gestellt und eine Freitext-Antwort konnte eingegeben werden.

8.2 Laborstudie – Aufgabenstellung T₁

User definition:

Please create the following three users:

Username	Password	Role
Bob	Bob123	Guest
Pia	456Pia	Guest
Tom	Tom654	Developer

Ontology definition:

Please create a new ontology named "Hypothetical Reasoning Ontology". The ontology should consist of three nodes and one relation. Each node should have a different border style.

The nodes should be titled as following:

1. Hypothetical
(in red color, including a text field as well as an internal link to a given text)
2. Fact (in orange color, including a text field)
3. Test
(in blue color, including two labeled text fields (labels: "if" and "then") as well as a drop down menu labeled "Confirmed?" with two options: "Yes" and "No")

The following relation should be available:

1. is connected to (in green color, not directed and without any explanations, that is, only the line without any additional panel that show the title)

Template definition:

Please define a template named "Hypothetical Reasoning Template". The template should be based on the newly created ontology "Hypothetical Reasoning Ontology" and allow only one user at a time. Furthermore, the following given text should be available:

Line No.	Content
1	Argumentation is essential in many aspects of life – in private as well as in business
2	However, most people struggle to engage in reasonable argumentation. Thus it is important to support argumentation learning from early childhood on.
3	One way to teach argumentation is computer-based argumentation tools.

Session definition:

Finally, the following pre-defined sessions based on the template “Hypothetical Reasoning Template” should be created:

1. Hypothetical Argumentation – Session 1 (Restricted to user “Bob”)
2. Hypothetical Argumentation – Session 2 (Restricted to user “Pia”)

8.3 Laborstudie – Aufgabenstellung T₂

You plan to conduct a small lab study with three participants. In this lab study, the participants should argue about global warming. Therefore, the system should provide multiple workspace elements: First of all, they need to state hypotheses. These hypotheses must be supported by means of data. The data will be entered by the users as text and supported via web references to external material. In addition, each data should get a believability score ranging from 0 to 5 assigned. To connect the data to the hypotheses, there must be two kinds of relations possible: pro and con. In the con case, there should be a text field available that allows the user to clarify why it is con. All relation should have a direction. Further, nodes and relations should have a color on their own. Each node should have a different border to make them easier distinguishable.

The participants should use anonymous standard user accounts. These accounts should be named lab1 to lab3. In addition, there should be another user account named mod1 with additional rights for teachers. For all accounts, the passwords should be equal to the username.

All participants and the teacher should be able to work in the same argument session at the same time and there will be two runs. To allow for an easy coordination of activities they should be able to see who is active in the session. Additionally, the participants should be able to chat with each other in the system. Further, it should be clear who created which part of the argument graph und who is currently working on which part of the argument graph. The latter can be done via a tracking of the cursor movements.