

Empfehlungen zur Weiterentwicklung der Mathematik-Software *ISAC*, abgeleitet aus der Beobachtung eines Unterrichtsversuchs

Alan Krempler

13.7.2007

Zusammenfassung

ISAC ist ein laufendes Softwareprojekt zur Entwicklung eines interaktiven und lernfördernden Mathematiksystems. Im Zuge der Praxiserprobung des aktuellen Entwicklungsstandes wurde ein Unterrichtsversuch beobachtet, um daraus Rückschlüsse auf vordringliche Schritte bei der Weiterentwicklung, insbesondere der Dialogkomponente zu ziehen. Zusätzlich zur eigentlichen Fragestellung werden Entwicklungspotentiale im Usability-Bereich identifiziert und benannt.

1 Einleitung

Das Softwareprojekt *ISAC* (<http://www.ist.tugraz.at/projects/isac>) hat eine neuartige Mathematiksoftware zum Ziel, die interaktiven und selbstgesteuerten Zugang zu Mathematikwissen zum Anspruch erhebt. Dabei soll das angewendete Mathematikwissen der Software jederzeit dem Benutzer zugänglich sein und gerade dadurch sowohl professionelle Anwendung als auch exploratives Lernen unterstützen.

Insbesondere für das Anwendungsgebiet des explorativen Lernens ist eine weitgehend selbsterklärende, intuitive Benutzeroberfläche eine wesentliche Voraussetzung. Diese Voraussetzung wurde in einem Unterrichtsversuch an der HTL Ortweinschule in Graz einem Praxistest unterzogen, mit dem Ziel vor allem Hinweise für die weitere Verbesserung des Dialogverhaltens von *ISAC* zu gewinnen.

Die nachfolgenden Empfehlungen resultieren aus der Beobachtung von zwei jeweils dreistündigen Unterrichtssequenzen und den Rückfragen und spontanen Kommentaren der beteiligten Schülerinnen und Schüler.

2 Usability-Voraussetzungen für eine Weiterentwicklung der Dialogkomponente

Im Zuge der Unterrichtsversuche stellte sich heraus, dass, während *ISAC* nach einer kurzen Erklärung und Einarbeitungszeit durchaus flüssig bedient werden konnte, einige Punkte immer wieder zu Rückfragen und damit zur Unterbrechung des Interaktionsflusses führten. Einige dieser wiederholt beobachteten oder nachgefragten Stolpersteine lassen vermuten, dass ein weiterer Ausbau der

Dialogfähigkeit von *ISAC* erst Sinn macht, wenn das System durch Adaptierungen der Benutzeroberfläche tatsächlich ohne fremde Hilfe bedient werden kann:

2.1 Einstieg in eine interaktive Berechnung

Die Auswahl von Rechenaufgaben aus einer Beispielsammlung bereitete keine Schwierigkeiten und in diesem Fall war der Einstieg in eine Berechnung jeweils leicht zu schaffen. Eine weitere Erleichterung könnte eine spezifischere Beschriftung der Bedienungselemente (Buttons) in der jeweiligen Landessprache bringen.

Die Eingabe selbst gewählter Rechenaufgaben jedoch war spontan nicht zu schaffen. Besonders irritierte der Umstand, dass es nicht ohne Weiteres möglich war, ein aus der Sammlung vorgefertigter Beispiele als lösbar bekanntes Beispiel „von Hand“ selbst in eine leere Berechnung einzugeben.

2.1.1 Intuitive Interaktion während der Spezifikationsphase

Bei der Auswahl eines leeren Arbeitsblattes für eine neue Berechnung erscheint ein Fenster mit zwei leeren Eingabezeilen ohne weitere Hinweise. Weder wird klar, wodurch sich die zwei leeren Zeilen in ihrer Bedeutung und Funktion unterscheiden, noch was jeweils einzugeben ist beziehungsweise überhaupt eingegeben werden kann.

Benutzerführung durch erklärende Texte, vorgegebene Eingabeformulare, Auswahlmenüs oder Dialogführung sollte Abhilfe schaffen. Dazu könnte eine bereits von Mario Hochreiter in Ansätzen programmierte Benutzerführung für die Spezifikationsphase aktiviert und ausgebaut werden.

2.1.2 Verzicht auf Schlüsselwörter

Die Anweisung, um welche Art der Problemstellung (Vereinfachen, Gleichung lösen, etc.) es sich handelt, wird derzeit durch Schlüsselwörter als Teil einer Formel eingegeben. Weder die möglichen und vom System erkannten Schlüsselwörter noch die jeweilige Syntax konnten von den Probanden erraten werden und auch die einmal erklärten Schlüsselwörter zu merken lenkte vom explorativen Fluss ab.

Die vorhandene Funktionalität über ein Auswahlmenü angeboten zu bekommen wird heute von allen Benutzern erwartet und beherrscht. Bis eine der Aufgabe von *ISAC* noch besser angepasste Interaktionsform entwickelt ist, ist jedenfalls Auswahlmenüs der Vorzug vor Schlüsselwörtern zu geben.

2.2 Interaktion während einer Berechnung

Als eines der attraktivsten Features von *ISAC* wurde von allen Probanden die Möglichkeit benannt, Zwischen-Rechenschritte angezeigt zu bekommen, die zum maschinell berechneten Endergebnis führen. Auch die Bedienung des Features stellte sich als ausreichend intuitiv für selbständige Arbeit heraus.

Als Verbesserungspotentiale wurden identifiziert:

2.2.1 Verfügbarkeit von Zwischen-Rechenschritten andeuten

Es ist nicht direkt ersichtlich, an welchen Stellen in der Berechnung weitere Zwischenschritte zur Anzeige zur Verfügung stehen. Es kann also nicht unterschieden werden, aus welchem Grund bei einer versuchten Anforderung keine weiteren Zwischenschritte angezeigt werden:

- Auf Grund von Fehlbedienung
- Auf Grund eines Softwarefehlers
- Weil an der betreffenden Stelle keine weiteren Zwischenschritte zur Verfügung stehen

Daher verschiebt sich die Aufmerksamkeit oft von der der Erforschung der aktuellen Berechnung zur Erforschung der Bedienung von *ISAC*, was den Lernfluss hemmt und Frustrationspotential birgt.

Es könnte programmtechnisch relativ aufwändig sein, das Vorhandensein von zusätzlichen Zwischenschritten an allen in Frage kommenden Stellen in einer Berechnung im Voraus festzustellen. In diesem Fall kann auch eine eindeutige Rückmeldung des Systems, dass die Anfrage zwar verstanden wurde aber nicht erfüllt werden kann, bessere Orientierung für die User bringen.

2.2.2 Begründungen für Rechenschritte anzeigen

Besonders bei komplexeren Formeln ist nicht immer intuitiv ersichtlich, welche Transformation von einer Formel zur nächsten führt. Der mehrfach geäußerte Wunsch, Hinweise auf die angewendete Transformation zu bekommen deckt sich mit einem geplanten Feature von *ISAC*, das also beschleunigt implementiert werden sollte.

2.2.3 Abschluss einer Berechnung transparent machen

Von wenigen Usern wurde angemerkt, dass das Ende einer Berechnung nicht immer ersichtlich ist, das heißt, dass manche Probanden versucht haben, *ISAC* weitere Rechenschritte durchführen zu lassen, obwohl das gewünschte Berechnungsziel bereits erreicht war. Diese Situation ähnelt dem Versuch, nicht vorhandene Zwischenschritte anzeigen zu lassen und könnte analog behandelt werden.

2.3 Darstellung und Eingabe von Formeln

Als eines der größeren Hemmnisse im Umgang mit *ISAC* wurde die ungewohnte Darstellung und Eingabe von Formeln genannt. *ISAC* verwendet derzeit eine textbasierte Eingabe und Darstellung, was bei den Grundrechnungsarten keine besonderen Schwierigkeiten bereitet. Für mathematische Symbole, die auf einer gewöhnlichen Computertastatur nicht vorkommen, werden Schlüsselwörter verwendet, die nicht intuitiv zu erraten sind. Für geübte User stellt das keine Schwierigkeit dar und beschleunigt sogar die Eingabe, für den ersten Kontakt mit *ISAC* stellt das jedoch ein erst zu überwindendes Problem dar.

Unabhängig vom Problem der Schlüsselwörter werden komplexere Formeln schwer les- und handhabbar, weil insbesondere durch das Fehlen von Bruchstrichen die Anzahl der verschachtelten Klammern schnell unpraktische Dimensionen annimmt.

Anzustreben wäre eine grafische Darstellung von Formeln auf dem Arbeitsblatt, um eine eher gewohnte Arbeitsumgebung zu bieten und die Übersichtlichkeit zu verbessern.

Für die Eingabe haben verschiedene Zielgruppen verschiedene Bedürfnisse: geübte User bevorzugen die Geschwindigkeit der Texteingabe, während für ungeübte ein grafischer Formeleditor die Suche nach Symbolen und den Umgang mit verschachtelten Klammern erleichtert. Hier sollte eine individuelle Auswahl möglich sein und zusätzlich zur Texteingabe auch interaktive Eingabe mit Hilfe eines Formeleditors unterstützt werden.

Während heute die meisten HTML-Engines die grafische Darstellung von in MathML codierter Notation direkt unterstützen, ist für die grafische Eingabe größerer Aufwand zu erwarten:

2.3.1 Verfügbarkeit von Formeleditoren

Eine Formeleditor-Komponente zu programmieren übersteigt derzeit die personellen, finanziellen und zeitlichen Möglichkeiten des *ISAC*-Projekts. Daher wird wahrscheinlich die Integration einer bereits vorhandenen Softwarekomponente das Mittel der Wahl sein.

Als freies Softwareprojekt legt *ISAC* besonderen Wert auf Unabhängigkeit von kommerziellen Anbietern und Softwarelizenzen. Während das die Auswahl an verfügbaren Softwarekomponenten beträchtlich einschränkt, sind folgende Formeleditoren als freie Software verfügbar, marktfähig ausgereift und gut dokumentiert:

KFormula ist der Formeleditor von KOffice, des Officepakets der KDE-Desktopumgebung.

Homepage: <http://www.koffice.org/kformula/>

Lizenz: <http://www.koffice.org/faq/#whichlicenseappliesstoKOffice>

OpenOffice.org Math ist der Formeleditor des Officepakets OpenOffice, eines von StarOffice von Sun abgeleiteten freien Officepakets.

Homepage: <http://www.openoffice.org/product/math.html>

Lizenz: <http://www.openoffice.org/license.html>

Beide Editoren unterstützen MathML, das als Datenformat an externen Interfaces von *ISAC* Verwendung finden soll und erleichtern damit die Integration in das System.

3 Software-Voraussetzungen für eine Weiterentwicklung der Dialogkomponente

Im Folgenden sollen einige interne technische Voraussetzungen für den weiteren Ausbau der Dialogkomponente beleuchtet werden:

3.1 Klassifizierung von Dialogsituationen

Die Möglichkeit, die Interaktion von *ISAC* mit den Usern programmgesteuert an die jeweilige Lernsituation anzupassen, ist einer der innovativsten Kernpunkte der *ISAC*-Idee. Eine Voraussetzung dafür ist, dass das System Situationen

identifizieren und klassifizieren kann, um dann sein Interaktionsverhalten daran anzupassen.

Für die Analyse einer didaktischen Situation durch Software wird noch weitere interdisziplinäre Forschung notwendig sein. Die Analyse könnte zum Beispiel den Vergleich des aktuell erforderlichen Wissens mit bereits erlerntem Wissen aber auch mit individuellen Vorlieben und bereits erzielten Erfolgen umfassen.

Die Analyse einer mathematischen Situation, das bedeutet die Klassifizierung einer Situation nach Zusammenhang in der angewandten Lösungsmethode und den aktuell möglichen bzw. sinnvollen Transformationen könnte schon jetzt in Angriff genommen werden.

3.2 Volle Implementation von CalcItem-Objekten

Formeln und verwandte mathematische Objekte werden derzeit intern in einem Textformat als Strings gespeichert. Vorgesehene Interfaces, die Zugriff auf die Struktur zB von Formeln bieten sollen, sind derzeit noch nicht implementiert. Der Zugriff auf die interne Struktur von mathematischen Objekten, beispielsweise im MathML-Format, ist sowohl für die oben genannte grafische Darstellung von Formeln als auch für die Identifikation von Dialogsituationen, wie im vorigen Abschnitt behandelt, eine Voraussetzung und wäre daher beschleunigt zu implementieren.

4 Weiterentwicklung der Dialogkomponente

Die möglichen Entwicklungsrichtungen der Dialogkomponente ergeben sich schrittweise aus dem Fortgang der Arbeit an den im vorigen Abschnitt genannten Softwarevoraussetzungen. Ein erster Schritt ist die Entwicklung der Möglichkeit, Situationen zu klassifizieren, um in einem zweiten Schritt auf dieser Situationsanalyse Entscheidungen zu gründen.

Kriterien für „richtige“ Entscheidungen, um Lernen oder kreativen Arbeitsfluss zu fördern, zu entwickeln, wird Gegenstand weiterer Forschung sein, sobald *TSAC* als System so vollständig, stabil und intuitiv verwendbar ist, dass es als Plattform für weitere umfassende Versuche dienen kann.

5 Zusammenfassung

Die Beobachtungen im Unterrichtsversuch und die Kommentare der Schülerinnen und Schüler haben gezeigt, dass *TSAC* schon als faszinierendes Werkzeug mit weitreichenden Möglichkeiten erkannt wird.

Derzeit sind vordringlich Verbesserungen an der intuitiven Bedienbarkeit des Systems notwendig. Einige dieser Anpassungen haben technische Voraussetzungen, die auch die Weiterentwicklung der interaktiven Dialogkomponente erleichtern werden.

Der nächste Entwicklungsschritt an der Dialogkomponente selbst sollte die Klassifikation von Situationen betreffen, nach der die Dialogkomponente ihr Verhalten ausrichten kann.