Seminar: Tutorielle Systeme

Veranstalter: Helmut Horacek, Magdalena Wolska

Ort und Zeit: Mi., 16-18 Hörsaal 3, Geb. E1.3
(Vorbesprechung und Anmeldung 24.10.)

Kursseite: http://www.coli.uni-sb.de/courses/its-ws0708

http://www.ags.uni-sb.de/~horacek/tutor-nl.html

Beschreibung: In dieser Veranstaltung werden intelligente tutorielle Systeme in Bezug auf die Verwendung natürlichsprachlicher Interaktion behandelt (Analyse, Generierung, Dialog, tutorielle Strategien, …)

Vorträge wahlweise in Deutsch oder Englisch (Englisch präferiert)

Gemeinsame Veranstaltung mit Computerlinguistik

Leistungspunkte je nach anwendbarer Studienordnung (siehe Kursseite)

Organization

Those who did not receive an Email by Magdalena Wolska:

- **Please write us an email, to get your address right!**

Those who are not scheduled for talks yet:

- **Please write us an email about your preferences within 1 week!**
- **We will publish the schedule on the website**
- **In case of too high demand, we will arrange extra lectures**
- **We will mail you access information for the preprint of the book**

BSc/MSc Informatik and CuK Students:

- **Please register by 1.12. in http://frweb.cs.uni-sb.de/03.Studium/011.HISPOS!**

What is an intelligent tutoring system (ITS)?

Tutoring task

Categories of human tutoring:

Classroom, one-to-one tutoring

Features of intelligent tutors

Not just a collection of instances of tutor reactions

Abstraction into descriptions of tutorial settings

Mechanisms that mimic aspects of human tutoring

Reasonable handling of unexpected student reactions

Features met by some tutoring systems:

Mixed initiative, interactive learning, instructional modeling, self-improving

Why building and using intelligent tutoring systems?

Benefits of tutoring

Teaching is necessary to acquire knowledge and skills

Constructivist teaching – learning unique for each individual

Students may make mistakes, query processes, monitor their own progress

Inquiry and social discussion

essential for critical thinking, problem solving and reasoning

Motivation for ITS

Overcoming the bottleneck of human tutors

Availability of expertise and teaching material

But: operationalization of tutoring capabilities increasingly difficult

GUIDON – the first intelligent tutoring system

Domain of tutoring

Teaching medical knowledge

Classification and problem solving

Size of the system

400 domain rules

200 tutoring rules

Main achievement as an ITS

Explicit separation of domain knowledge and pedagogical knowledge

A tutoring session with Ms. Lindquist (algebra tutor)

T1: Hello. (After presenting a 'bike-trip' problem).
Please write an expression for the total time the trip took. (symbolize)

S1: Help! (The tutor breaks down the problem into steps)

T2: Let me ask an easier question. Please compute the time actually on the bikes if instead the speed was 20 miles per hour, the time of the break was 2 hours, and the distance ridden was 100 miles?

S2: 7

T3: Correct. What math did you do to get that 7?

S3: 2+5 (Student does not completely articulate)

T4: Okay. But how did you get the 5? (similar to T3)

S4: 100/20

T5: Correct. Now lets try this again, what math did you do to get the 7?
(same question as T3 but inserting the phrase 'lets try this again'

S5: 2+100/20 (correct articulation)

T6: Correct. Now write your answer of '2+100/20' using the variables given in the problem. (generalize like T4 from previous example)

S6: b+m/s

T7: Correct.

Some tutoring skill incorporated in Ms. Lindquist

Teaching technqiues – implementing human strategies

Partitioning/simplifying the problem
Elaborating instantiations
Combining and abstracting partial solutions

Communicative behavior

Encouraging attitude
Accepting partially correct answers
Emphasizing discourse coherence

Innovations

Feedback not based on explicit buggy rules
Chains of hints as progressively more explicit messages

IMPORTANCE OF NATURAL LANGUAGE

Empirical investigations

Natural language communication makes tutoring effective

[Moore 2000]

Presenting reasoning structures in natural language increases subjects' performance

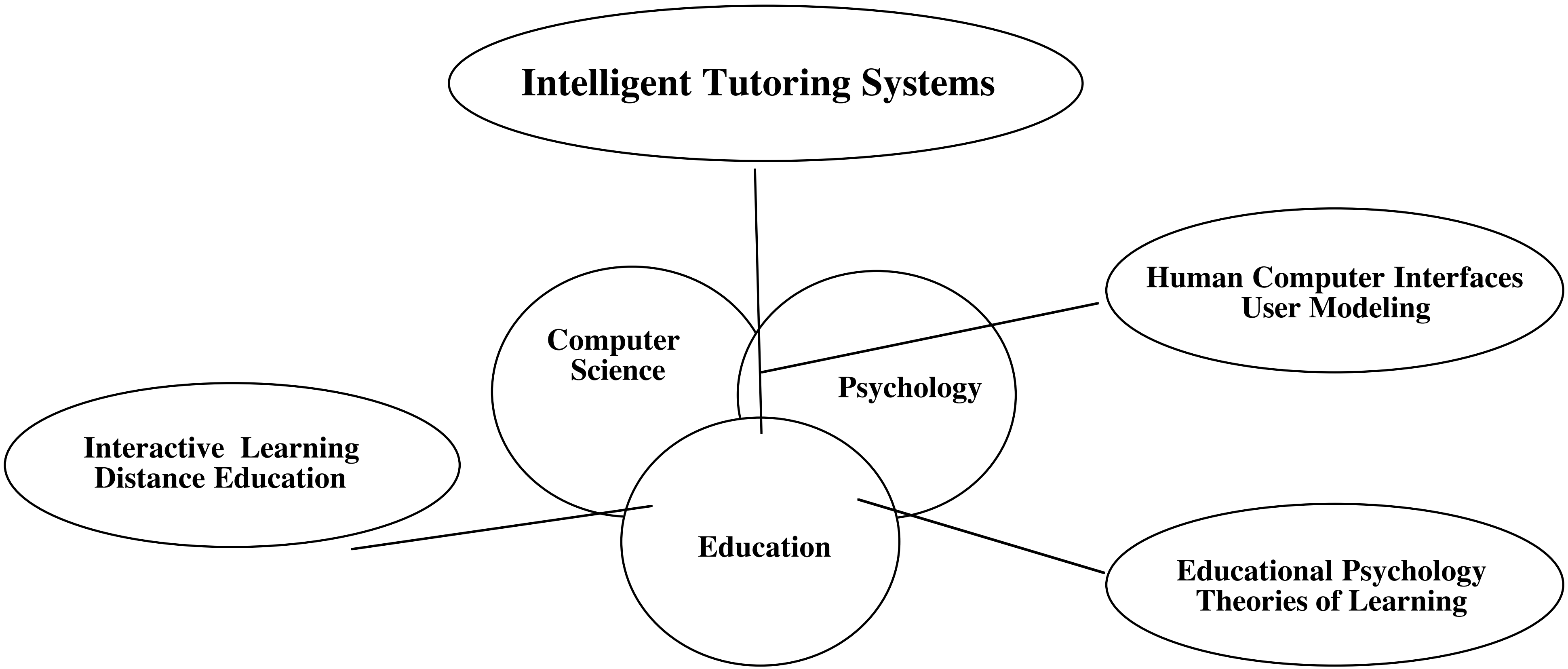
[Di Eugenio et al. 2002]

Natural language evaluative arguments convey object preferences more effectively

[Carenini, Moore 2001]

Preference of natural language variant shown to be *statistically significant*

Building an ITS is an interdisciplinary activity



Components of an ITS

Domain module

Student module

Tutoring module

Communication module

Knowledge relevant for an ITS (1)

Domain knowledge

Model of expert knowledge

Topics, subtopics, definitions or processes

Skills needed to generate algebra equations, administer medications, …

Student knowledge

Describes how tutor reasons about a student's presumed knowledge

Represents each student's mastery of the domain

(acquired skills, time spent on problems, hints requested, possible misconceptions, correct answers, preferred learning style)

Knowledge relevant for an ITS (2)

Tutoring knowledge

Teaching startegies: methods for providing remediation, examples, …

Reasoning about the use of materials, feedback, and testing

(empirical observations, learning theories, technology-enabled)

Communication knowledge

Methods for communication – graphical interface, animated agents, dialog

Communication motivates and supports students

Ensures that a tutor follows a student's reasoning

Cognitive science techniques

Model-tracing tutors

Model of the domain used to interpret student actions and solution paths

Cognitive representation of tasks, mostly a result of careful task analysis

Tutor traces a student's implicit execution of the encoded rules

Assumes that all problem solving steps can be identified and explicitly coded

Assumes that student performs the same reasoning as encoded in the rules

Constraint-based tutors

Pedagogically significant states expressed as constraints

Constraints represent the application of a piece of declarative knowledge

Detect and correct student errors which appear as violated constraints

Constraints represent states the student should satisfy, not the paths involved

Applied in intractable domains, domains that cannot be fully articulated

Teaching strategies

Didactic strategy

Student's problem solving accompanied by piece-wise explanations

Student in some sense navigates in a normative problem-solving space

Socratic strategy

Student's problem solving is guided and supported by hints

Aims at enabling a student's knowledge construction

Socratic teaching generally considered superior:

Long term benefits achieved, harder for students (takes more time)

Support knowledge transfer (applying skills to related tasks and domains)

Communication of tutors – a classification

Mixed-initiative dialog

Only few NL tutors support full mixed initiative dialog

Single-initiative dialog

Limited discourse context considered

Directed dialog

Tutor controls the dialog, prompts students for explicit information, …

Finessed dialog

Dialog simulated through menu-based input, logical forms, …

Two excerpts of sessions with the geometry tutor

S1: Angles are equal.
T1: Are any two angles congruent? What tells you that there are two congruent angles?
S2: Angles in a triangle are equal.
T2: That is a bit imprecise. Are ALL angles in ALL triangles congruent?
S3: Angles in an isosceles triangle are equal.
T3: Correct, except to: are all three angles in an isosceles triangle congruent?
S4: Base angles in an isosceles triangle are equal. (correct)

…

S5: They are 180
T5: Could ou be more specific? Who are *they*? And what tells you they are 180?
S6: The angles are 180
T6: You are heading in the right direction, but you need to state this rule more precisely. First, where does the 180 come from? Are all angles 180?
S7: The angles in a triangle are 180.
T7: You are awfully close. But is each angle measure equal to 180? Precisely what is 180?
S8: The sum of the angles in a triangle are 180. (correct)

Issues in the communication of tutors

Dialog strategies

Only few NL tutors support full mixed initiative dialog

NL analysis

Limited discourse context considered

NL generation

Tutor controls the dialog, prompts students for explicit information, …

The role of emotion in tutoring

Recognising the mood of the student

Interpreting sequences of student actions

(some even monitor facial expresssions)

Selecting/adapting system actions to boredom, frustration, enthusiams, …

Teaching environments

Use of animated agents

Examining the effects of varions forms of agents

Preventing misuse

Gaming – just clicking to get maximum feedback in minimal time

Cheating – producing absurd or off-the-topic contributions

Evaluating tutoring systems

Techniques

Specifying conditions to be tested

Experiments with a featured group and a control group

Computing the effect of the difference – statistically significant?

Problems

Experiments are expensive – picking most important properties

Capturing system/teaching properties to be tested

Examples

Geometry curriculum with ITS significatly better than traditional form

Linguistically adequate presentations improve performance signifcantly

ITS as a scientific field

Research community

Annual conferences since 20 years, approx. 200 participants

A dedicated journal, presence in related conferences and journals

Domains of application

Facilitating learning in groups (classroom)

Ono-to-one tutoring in physics, mathematics, programming, formal design …

Success

Significantly improved learning with ITS

Cognitive tutors for algebra and geometry in use in more than 1300 US schools

The DIALOG project (SFB 378)

Goal

Participating in a flexible natural language tutorial dialog

Empirically investigating dialogs in teaching mathematical proofs

Architecture – modular design

***Learning environment* – getting acquainted with some lesson material**

***Mathematical proof assistant* – checks appropriateness of student's utterances**

***Proof manager* – maintains representation of constructed proof object**

***Natural language processing* – NL expressions interleaved with formulas**

attempts the interpretation of imprecise, ambiguous and faulty utterances

***Dialog manager*– maintains state of dialog and determines system reaction**

including an embedded hinting algorithm

***Knowledge resources* – domain and pedagogical knowledge (hint taxomony)**

WIZARD-OF-OZ EXPERIMENT 1

Goal

Collecting a corpus on tutorial dialogs in the naive set domain

Testing tutorial strategies developed

Experiment phases

Preparation and pre-test on paper

Tutoring session mediated by *Wizard-of-Oz* tool

Post-test on paper and evaluation questionnaire

Tasks to prove

(1) $K((A \cup B) \cap (C \cup D)) = (K(A) \cap K(B)) \cup (K(C) \cap K(D))$

(2) $A \cap B \in P((A \cup C) \cap (B \cup C))$

(3) If $K(B) \supseteq A$**, then** $K(A) \supseteq B$

Experience gained

Socratic strategy not as effective as hoped (long-term effects unexplored)

Distracted by lengthy clarification subdialogs resolving low-level issues

WIZARD-OF-OZ EXPERIMENT 2

Goal

Collecting a corpus on tutorial dialogs about relations (a more advanced topic)

Exploring human hinting strategies in a socratic style

Experiment phases

Getting acquainted with the domain and environment on the computer

Tutoring session mediated by *Wizard-of-Oz* and editing tools

Evaluation questionnaire

Tasks to prove

(1) $(R \circ S)^{-1} = R^{-1} \circ S^{-1}$

(2) $(R \cup S) \circ T = (R \circ T) \cup (S \circ T)$ **(for relations R, S and T over a set M)**

(3) $(R \cup S) \circ T = (T^{-1} \circ S^{-1})^{-1} \cup (T^{-1} \circ R^{-1})^{-1}$

Experience gained

Mistakes of various kinds (see the categories on the next slides)

Tutor reactions addressing errors opportunistically

INTERPRETATION OF SLOPPY EXPRESSIONS

(1) $A \cup B$ must <u>be in</u> $P((A \cup C) \cap (B \cup C))$, since $(A \cap B) \cup C \supseteq$ of $A \cap B$

(2) If A is a subset of C and B a subset of C, then both sets <u>together</u> must also be a subset of C

Relations ambiguous between *element* and *subset*, resp. *union* and *intersection*

(3) $K((A \cup B) \cap (C \cup D)) = (K(A \cup B) \cup K(C \cup D))$, De Morgan Rule 2 applied to <u>both complements</u>

(4) $A \cap B$ on the <u>left side</u> is $\in$ of $C \cup (A \cap B)$, which is extended only by C

Intended referents not mentioned explicitly, scopus preferences apply

Metonymic interpretations required

DOMAIN ONTOLOGY

Domain knowledge representation

- **Complete logical definitions represented in λ-*calculus***
- **Inheritance used to percolate shared information, *no* hierarchical organization**
- **Only proof-relevant knowledge expressed**
- *Discrepancy to linguistic requirements*

Discrepancy bridged through intermediate representation

- **Imposing hierarchical organization**
- **Linking *vague* and *general* terms to domain terms**
- **Additionally modeling *typographic* features (markers, orderings)**

ANALYSIS PHASES

Preprocessing and parsing

- Mathematical expressions sustituted with default lexical entries
- Syntactic parsing and building linguistic meaning representation

Domain and discourse interpretation (using the semantic lexicon)

Symbolic representation built and passed to the proof manager

Consulting the tutoring manager with results obtained

EXAMPLES REVISITED

(1) $A \cup B$ must <u>be in</u> $\mathcal{P}((A \cup C) \cap (B \cup C))$, since $A \cap B \in$ of $(A \cap B) \cup C$

only ELEMENT interpretation is *relevant,* SUBSET is *incorrect*

(2) If A is a subset of C and B a subset of C, then both sets <u>together</u> must also be a subset of C

only UNION interpretation is *relevant,* INTERSECTION merely *correct*

(3) $K((A \cup B) \cap (C \cup D)) = (K(A \cup B) \cup K(C \cup D))$, De Morgan Rule 2 applied to <u>both complements</u>

only separate rule application possible, not their composition, thus disambiguated

(4) $A \cap B$ on the <u>left side</u> is $\in$ of $C \cup (A \cap B)$, which is extended only by C

judged as *incorrect*, since argument types clash with ELEMENT relation

HANDLING ERRORS – CORPUS EXAMPLES

	Example formula	*Error category*
(1)	$P((A \cup C) \cap (B \cup C)) = PC \cup (A \cap B)$	**3**
(2)	$(p \cap a) \in P(a \cap b)$	**2**
(3)	$(x \in b) \notin A \quad K(A) \supseteq x$	**2**
(4)	$P((A \cap B) \cup C) = P(A \cup B) \cup P(C)$	**1**
(5)	$P(A \cap B) \supseteq (A \cap B)$	**1**
(6)	**if** $K(B) \supseteq A$ **then** $A \notin B$	**2**

3: *Structural* errors (1): Missing space between *P* and *C*, and enclosing parentheses

2: *Type* errors (2,3,6): Typographical (2), argument type (3), operator type (6)

1: *Logical* errors (4,5): Set inclusion for equality (4), membership for subset (5)

ERROR RECOGNITION

Components contributing

Formula analyzer

Defined repertoire of operators and variables, with arity and type restrictions

Proof manager

Tries to find a proof for the assertion, within the defined context

Error categories

***Structural (syntactic) errors* – Formula analyzer cannot built an analysis tree**

***Type (semantic) errors* – Formula analyzer reports a type mismatch**

***Logical (truth-value) errors* – Proof manager disproves the assertion**

ERROR CORRECTION

Components contributing

- **Formula analyzer**
  - **Performs structural modifications to enable building an analysis tree**
- **Formula modifier**
  - **Tries to apply cognitively plausible changes to the flawed formula**

Formula modifications

- **Local and cognitively justified changes**
- **Guided by error category and flawed portion of the formula**
- **Searching for modifications that improve the correctness state of the formula**