

Be Brief, And They Shall Learn: Generating Concise Language Feedback for a Computer Tutor

Barbara Di Eugenio, Davide Fossati, *Department of Computer Science, University of Illinois at Chicago, Chicago, IL, 60302, USA*
{*bdieugen,dfossal*}@uic.edu

Susan Haller, *Department of Computer Science, SUNY Potsdam, Potsdam NY 13676, USA*
hallerism@potsdam.edu

Dan Yu, *Albert A. Webb Associates, Riverside, CA 92506, USA*
danyu79@gmail.com

Michael Glass, *Department of Math and Computer Science, Valparaiso University, Valparaiso, IN 46383, USA*
michael.glass@valpo.edu

Abstract. To investigate whether more concise Natural Language feedback improves learning, we developed two Natural Language generators (DIAG-NLP1 and DIAG-NLP2), to provide feedback in an Intelligent Tutoring System that teaches troubleshooting. We systematically evaluated them in a three way comparison that included the original system, which generates overly repetitive feedback. We found that DIAG-NLP2, the generator which intuitively produces the best, corpus-based language, does engender the most learning. Distinguishing features of the more effective feedback are: it obeys Grice's maxim of brevity, it is more directive and uses a specific type of referring expressions. Interestingly, simpler ways of restructuring the original repetitive feedback as done in DIAG-NLP1, such as exploiting the hierarchical structure of the domain, were not effective. Since the design of interfaces to Intelligent Tutoring Systems often includes verbal feedback, we suggest that: if the number of different contexts in which verbal feedback is provided is high, such feedback should be based on corpus studies, and generated by techniques more sophisticated than template filling.

Keywords. Intelligent tutoring systems, natural language interfaces, corpus studies, feedback generation

INTRODUCTION

The next generation of Intelligent Tutoring Systems will be able to engage the student in a fluent Natural Language dialogue. This is the vision of the many researchers who are exploring Natural Language (NL) as the key to bridge the gap between human tutors and current Intelligent Tutoring Systems (ITSs). This area of inquiry started with pioneering work in the '70s and '80s, such as (Carbonell, 1970; Burton & Brown, 1979); was abandoned, perhaps because of the brittleness of the NL components of the time, other than by a handful of researchers such as Martha Evens and colleagues (Evens, Spitzkovsky, Boyle, Michael, & Rovick, 1993; Evens & Michael, 2006); and has been flourishing in the last few years

(Alevén, Koedinger, & Popescu, 2003; Moore, Porayska-Pomsta, Varges, & Zinn, 2004; Graesser, Person, Lu, Jeon, & McDaniel, 2005; Di Eugenio, Fossati, Yu, Haller, & Glass, 2005a, 2005b; Zinn, Moore, & Core, 2005; Litman et al., 2006; Pon-Barry, Schultz, Bratt, & Peters, 2006; Kumar, Rosé, Alevén, Iglesias, & Robinson, 2006; Ohlsson et al., 2007).

Two crucial goals of this collective effort are to ascertain whether an ITS endowed with language capabilities does positively impact learning, and to investigate which specific features of the NL interaction are responsible for the improvement. The reader may think that the former is a necessary condition for the latter: namely, if language does not make a difference to learning, then there is no point in asking which features are effective. However, because the space of possible language features one can focus on and of their combinations is infinite, the relationship between these two questions is unclear and they are difficult to separate.

The pursuit is both theoretical and practical. From a cognitive point of view, research at the intersection of psychology, education and computer science is investigating which features of dialogue are conducive to learning. From a practical point of view, fully-fledged NL interaction with an ITS is still well beyond the state of the art. If only some specific features of human tutoring engender learning, then an ITS that only includes those would be easier to build, and more likely to be effective than an ITS that tries to address the full complexity of human dialogue.

The work presented in this paper is among the first to show that more sophisticated language feedback does make a difference. Specifically, we will show that feedback that focuses on higher levels of abstraction is more effective than more detailed but more repetitive feedback, including feedback that is reorganized along structural rather than functional dimensions. In addition, the more effective feedback includes explicit directives on what to do next, and uses distinct references to a class of domain objects.

Naturally, research on how to provide language feedback in an ITS is subsumed by the general question of how to provide feedback and hints in Interactive Learning Environments. A vast literature exists on this topic, some of which is summarized in (Alevén, Stahl, Schworm, Fischer, & Wallace, 2003). The number of dimensions that can affect learning outcomes is huge, from features of the learner such as gender and aptitude (Arroyo, Woolf, & Beal, 2006), to what type of information the feedback contains (McKendree, 1990; Alevén & Koedinger, 2002), to the timing of feedback (Schooler & Anderson, 1990), to the media and media mixture through which the feedback is provided (Moreno, 2006). In fact, many systems provide verbal feedback of one type or another. Often though, this feedback is canned, i.e. different types of messages are predefined, or minimally changed, in that variables in a few templates get bound differently (e.g. see (Alevén & Koedinger, 2002; Arroyo et al., 2006) just to cite a couple). In addition, the specific texts that are used in feedback messages are often devised by the designers, possibly based on cognitive task analysis, or inspired by tutoring interactions in the domain of interest, but not on the basis of a full corpus study. Current research in NL for ITSs instead is grounded in corpus studies and uses NL technologies that help provide more varied interactions. This is the approach we will adopt in this paper. Not every ITS designer will subscribe to the pre-eminence of language as a means of delivering feedback, and we will not address the issue of how to parcel content into textual and graphics media. However, we hope we will convince the reader of the following. First, if the linguistic feedback needs to vary according to a high enough number of contexts, it should also be based on human linguistic data, not only on informed introspection. Second, if the number of contexts is really high, e.g. in the thousands as

in the work presented here, canned texts or even templates where a few variables get bound to different content are not sufficient. More sophisticated Natural Language Generation (NLG) techniques are called for – NLG is the subfield of Natural Language Processing concerned with producing language according to the goals of the speaker/writer. As we will show, NLG techniques are feasible and effective.

Going back to the general area of analyzing and modeling tutorial dialogue in ITSs, it is by now a pretty robust finding that ITSs with some sort of language interface are better than just a lecture or reading the relevant material in the textbook (Graesser et al., 2004; Evens & Michael, 2006). However, it is not yet clear how sophisticated and interactive the language interface needs to be. Some recent results call into question whether the *interaction* hypothesis, as (VanLehn et al., 2007) calls it, is fully verified or not. This hypothesis, based on work such as (Chi, Bassok, Lewis, Reimann, & Glaser, 1989; Fox, 1993; Chi, Siler, Yamauchi, & Hausmann, 2001), predicts that students will learn more in a one-on-one dialogue tutoring condition, as compared to a less interactive condition that covers the same content. In both (VanLehn et al., 2007) and (Evens & Michael, 2006), the version of the ITS that engages students in dialogue in response to a student's question was not more effective than a version of the same ITS that provided carefully crafted texts to read when faced with the same question. In addition, (Graesser et al., 2004) reports that whereas the increase in effect size in learning gain engendered by the ITS was large compared to simply reading the textbook, it became almost negligible when compared to reading carefully edited scripts.

It would obviously be premature to conclude that full interaction does not help, since there are many robust findings that show that students learn more when engaged in a conversation with human tutors (Graesser, Person, & Magliano, 1995; Lepper, Drake, & O'Donnell-Johnson, 1997; Evens & Michael, 2006). Further, the lack of support for the interaction hypothesis may be due to technical shortcomings of the dialogue engine, including the fact that the dialogue was typed, not spoken (Litman et al., 2006; Moreno, 2006). But even if it turns out that fully-fledged dialogue is not necessary, it is important to note that a language interface, even if not interactive, still appears to be the superior choice. In fact, in the three studies that found that reading explanations engender the same learning as interaction does (Graesser et al., 2004; Evens & Michael, 2006; VanLehn et al., 2007), the bar posed by the control condition was quite high: the texts the students read were carefully crafted, not just lifted out of a textbook. In fact, NLG techniques can generate at least some of those texts on the fly, instead of humans preparing them in advance. For example, the texts that students read in the experiments in (Graesser et al., 2004) were summaries of AUTOTUTOR's curriculum scripts written by humans; an NLG system could have summarized those scripts instead, at least as a first pass.

Our results support the finding that carefully edited texts engender more learning than other competitive conditions. We added more sophistication to the feedback generation component of an existing ITS, which had been originally endowed with the capability of providing very simple, and repetitive, language feedback. We developed two different feedback generation engines, that we systematically evaluated in a three-way comparison that included the original system. Our results show that feedback that focuses on higher levels of abstraction, as human tutors do in our domain, is more effective than more detailed but more repetitive feedback. From a general pragmatics point of view, our results provide experimental evidence for one of Grice's Maxims, specifically, the *Be brief* submaxim of his Maxim of Manner, *Be perspicuous* (Grice, 1975). From the NLG point of view, the *Be brief* submaxim prompts research on

aggregation, i.e. on how lengthy information can be grouped and presented as more manageable chunks (Reiter & Dale, 2000). In addition, the more effective feedback provides more explicit hints on what the student should do next. We included these more explicit directives because of our tutors' behavior, even if they seem to contradict findings that suggest that tutors should limit themselves to prompting and scaffolding (Chi et al., 1989, 2001). Finally, the more effective feedback also changed how certain objects in the domain were referred to. We will discuss the implications of the different features of the more effective feedback later in the paper.

The paper is organized as follows. We will first discuss DIAG, the ITS shell we are using, and the two feedback generators that we developed, *DIAG-NLP1* and *DIAG-NLP2*. Since the latter is based on a corpus study, we will describe that as well. We will then present the formal evaluation we conducted. We will conclude with a discussion of our results.

NATURAL LANGUAGE GENERATION FOR DIAG

DIAG (Towne, 1997) is a shell to build ITSs based on interactive graphical models that teach students to troubleshoot complex systems such as home heating and circuitry. A DIAG application presents a student with a series of troubleshooting problems of increasing difficulty. The student tests *indicators* to try and infer which faulty part (replaceable unit, or RU) may cause the abnormal states detected via the indicator readings. The only course of action for the student to fix the problem is to replace faulty components in the graphical simulation. Figure 1 shows the furnace, one subsystem of the home heating system in our DIAG application. Figure 1 includes indicators such as the gauge labeled *Water Temperature*, RUs such as the *System Control Module*, and complex modules such as the *Oil Burner* that are zoomable and that in turn, contain indicators and RUs.

At any point, the student can consult the tutor via the Consult menu, shown in Figure 2, which is activated via the Consult button (see Figure 1). There are two main types of queries: *Discuss indications* and *Discuss Replaceable Units*, which we will respectively call *ConsultInd(icator)* and *ConsultRU* in the rest of the paper. For *ConsultInd* queries, the students select one or more indicators, and the system provides feedback on those. In principle, a *ConsultInd* should be used when an indicator shows an abnormal reading, to obtain a hint regarding which RUs may cause the problem. The DIAG application discusses the RUs that should be most suspected given the symptoms the student has already observed.

For *ConsultRUs*, the student selects one or more RUs that s/he suspects may be faulty and hence the cause of the problem, and obtains feedback on his/her diagnosis. The DIAG application responds with an assessment of that diagnosis and provides evidence for it in terms of the symptoms that have been observed relative to that RU.

The designers of the original DIAG system (*DIAG-orig*) did include basic capabilities to provide language feedback, namely, very simple templates where a few variables get appropriately bound. The top parts of Figures 3 and 4 show the replies provided by *DIAG-orig* to a *ConsultInd* on the *Visual Combustion Check*, and to a *ConsultRu* on the *Water Pump*. In general, *DIAG-orig*'s educational philosophy is to push the student to select the most informative tests, and not to provide too much explicit information when asked for hints. This is reflected in the way *DIAG-orig* answers students' queries. *DIAG-orig*

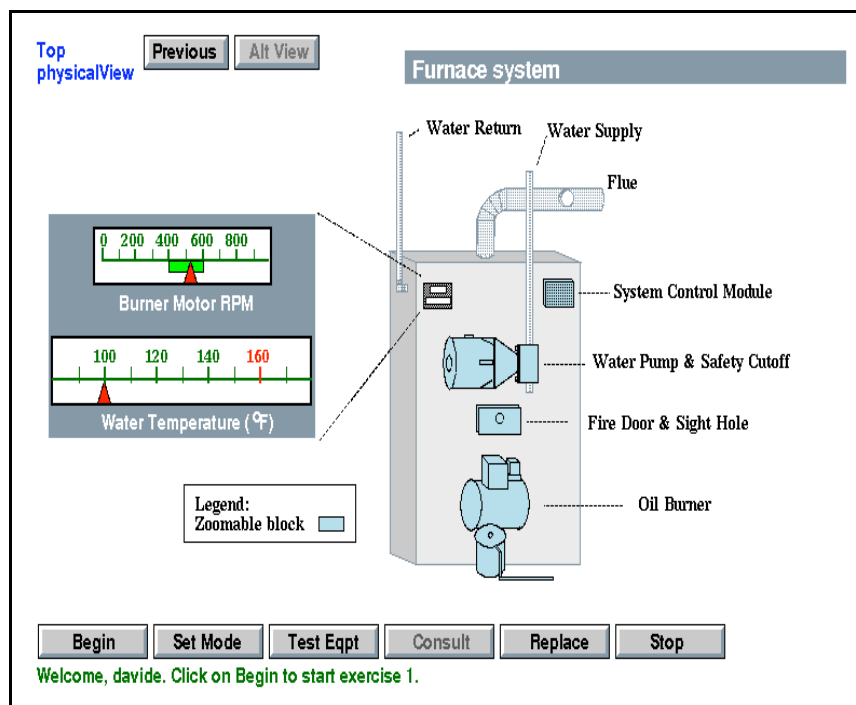


Fig.1. The furnace system.

does not provide direct information on what to do next unless it is asked explicitly via the corresponding button (cf. Figure 2): for this reason, we actually disabled that button in all our experiments.

The highly repetitive feedback by *DIAG-orig* screams for improvement.¹ The first issue we had to face was, which features of the feedback to focus on. We developed *DIAG-NLP1* on the basis of hypotheses, informed by the literature, on what would make for effective language feedback. The failure of those hypotheses points to the value of a corpus study in which to base the language feedback the system provides, which is what *DIAG-NLP2* was based on. The second issue was, how the system should generate that feedback. We had no doubt that we needed to deploy more sophisticated NLG techniques than the templates *DIAG-orig* uses. The alternative of enumerating all the possible contexts in which feedback may be asked for, and of asking experts to produce appropriate feedback messages, was untenable: in fact, the number of different contexts is in the thousands. For example, to answer a *ConsultRU*, *DIAG-orig* revisits the symptoms the student has seen, namely, the indicators the student has examined. While there are only 13 RUs in the system, and there is an assumption, known to the student, that only one of them fails at any time, a student could have inspected any subset of the 17 indicators present in the system. Hence, for every possible fault there are 131,072 potential contexts to be verbalized, corresponding to the cardinality of the powerset of the indicators. While this number could be viewed as an unrealistic worst case scenario, in general our students do indeed inspect a high

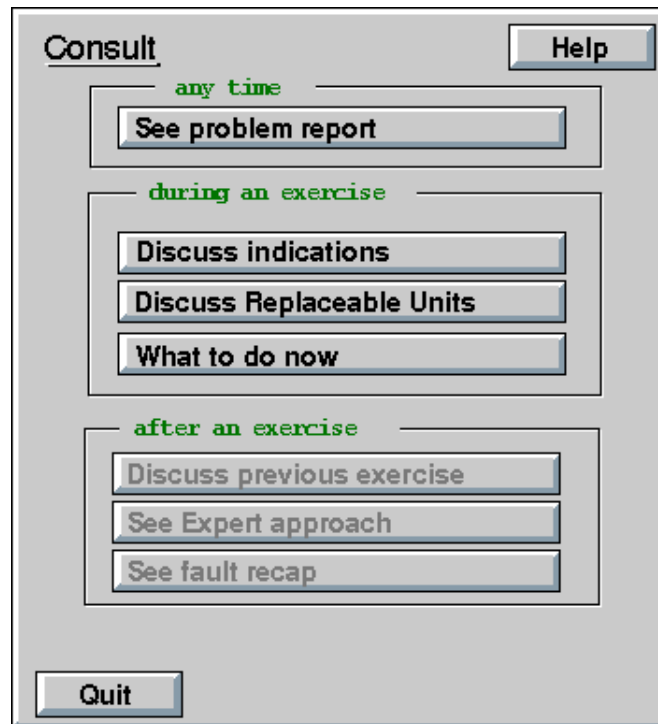


Fig.2. The consult menu.

number of indicators, as indicated by the average number of *ConsultInd*, shown in Table 3. We will come back to this point when we discuss Table 3 in the Evaluation section.

From the NLG point of view, our goal in developing *DIAG-NLP1* and *DIAG-NLP2* was to assess whether relatively simple, rapidly deployable NLG techniques would lead to measurable improvements in learning. Thus, in both cases it is still the DIAG application that performs content determination, and provides *DIAG-NLP1* and *DIAG-NLP2* with a file in which the facts to be communicated are written – a *fact* is the basic unit of information that underlies each of the clauses in a reply by *DIAG-orig*. Figure 5 includes some of the facts that underlie the answers in Figure 3. The attribute *fuf_{er}*² represents the strength of the causal connection between the failure of an RU and an observed symptom. The order of the information in the text file mirrors the order in which the DIAG application assembles the information, which is also directly mirrored in the feedback provided by *DIAG-orig* (see top of Figure 3).

The only way we altered the interaction between student and system was to replace the actual language that is presented in the output window. In *DIAG-NLP1* we mostly explored using syntactic and structural aggregation to improve feedback, whereas *DIAG-NLP2* focuses on functional aggregation, as arose from the corpus study. According to (Reape & Mellish, 1998), syntactic aggregation pertains to grouping syntactic components of two or more clauses, e.g. subjects, predicates, objects, etc: for example, we can syntactically aggregate *Mary is smart* and *Sue is smart* as *Mary and Sue are smart* (hence, the name *aggregation via shared participants*). Functional aggregation pertains to aggregating objects based

The visual combustion check is igniting which is abnormal (normal is combusting).
Oil Nozzle always produces this abnormality when it fails.
Oil Supply Valve always produces this abnormality when it fails.
Oil pump always produces this abnormality when it fails.
Oil Filter always produces this abnormality when it fails.
System Control Module sometimes produces this abnormality when it fails.
Ignitor Assembly never produces this abnormality when it fails.
Burner Motor always produces this abnormality when it fails.

The visual combustion check indicator is igniting.
This is abnormal.
Normal is combusting.

Within the furnace system,
this is sometimes caused if
the System Control Module has failed.

Within the Oil Burner
this is never caused if
the Ignitor Assembly has failed.
In contrast, this is always caused if
the Burner Motor, Oil Filter, Oil Pump,
Oil Supply Valve, or Oil Nozzle has failed.

The combustion is abnormal.
In the oil burner, check the units along the path of the oil
and the burner motor.

Fig.3. Answers to *ConsultInd* by *DIAG-orig*, *DIAG-NLP1* and *DIAG-NLP2*.

<p>Water pump is a very poor suspect. Some symptoms you have seen conflict with that theory. Water pump sound was normal. This normal indication never results when this unit fails. Visual combustion check was igniting. This abnormal indication never results when this unit fails. Burner Motor RMP Gauge was 525. This normal indication never results when this unit fails.</p> <hr/> <p>The Water pump is a very poor suspect. Some symptoms you have seen conflict with that theory.</p> <p>The following indicators never display normally when this unit fails. Within the furnace system, the Burner Motor RMP Gauge is 525. Within the water pump and safety cutoff valve, the water pump sound indicator is normal.</p> <p>The following indicators never display abnormally when this unit fails. Within the fire door sight hole, the visual combustion check indicator is igniting.</p> <hr/> <p>The water pump is a poor suspect since the water pump sound is ok. You have seen that the combustion is abnormal. Check the units along the path of the oil and the electrical devices.</p>

Fig.4. Answers to *ConsultRu* by *DIAG-orig*, *DIAG-NLP1* and *DIAG-NLP2*.

on their functionality (Paris, 1988); it is related to *conceptual aggregation* (Reape & Mellish, 1998), i.e. to reducing the number of different semantic propositions in the message (e.g. aggregating *dove(x)* and *sparrow(x)*, as *bird(x)*; in our domain, e.g. aggregating the *oil nozzle*, *oil pump*, *oil filter* as *the parts on the path of the oil*). In both *DIAG-NLP1* and *DIAG-NLP2*, we use EXEMPLARS (White & Caldwell, 1998), an object-oriented, rule-based generator. The rules (called *exemplars*) are meant to capture an exemplary way of achieving a communicative goal in a given context. EXEMPLARS selects rules by traversing the exemplar specialization hierarchy and evaluating the applicability conditions associated with each exemplar. In both *DIAG-NLP1* and *DIAG-NLP2* EXEMPLARS accesses a Knowledge Base (KB) that encodes static domain information. We had to build this KB by hand, since in *DIAG* applications, domain knowledge is hidden and hardly accessible. In *DIAG-NLP1*, the KB is embedded within the SNePS³ Knowledge Representation and Reasoning System (Shapiro, 2000). In *DIAG-NLP2*, the KB is implemented via a relational database. We will discuss these choices in detail below.

```

ConsultIndicator Indicator
name Visual combustion check
state igniting
modeName startup
normalState combusting
- -
ConsultIndicator ReplUnit
name Oil Nozzle
fufer always
- -
ConsultIndicator ReplUnit
name System Control Module
fufer sometimes
- -
ConsultIndicator ReplUnit
name Ignitor assembly
fufer no effect

```

Fig.5. A portion of a “fact” file to be verbalized.

DIAG-NLP1: Syntactic and structural aggregation

*DIAG-NLP1*⁴ (i) introduces syntactic aggregation and what we call *structural* aggregation, namely, grouping parts according to the structure of the system; (ii) generates some referring expressions; (iii) models a few rhetorical relations; and (iv) improves the format of the output.

In *DIAG-NLP1*, we focused on these specific features on the basis of our informed introspection on what would improve the feedback. We call it *informed* introspection because our intuitions about the usefulness of aggregating *DIAG-orig*'s feedback according to syntactic features, to system structure, and by means of a better layout were supported by different bodies of research, including work on syntactic aggregation in NLG (Dalianis, 1996; Huang & Fiedler, 1996; Reape & Mellish, 1998; Shaw, 2002), and work on how layout affects comprehension, e.g. (Wright, 1997; Shriver, 1997; Power, Scott, & Bouayad-Agha, 2003).

The middle parts of Figures 3 and 4 show the revised output produced by *DIAG-NLP1*, e.g. in Figure 3 the RUs of interest are grouped by the system modules that contain them (Oil Burner and Furnace System), and by the likelihood that a certain RU causes the observed symptoms. In contrast to the original answer, the revised answer highlights that the *Ignitor Assembly* cannot cause the symptom.

In *DIAG-NLP1*, EXEMPLARS accesses the SNePS Knowledge Representation and Reasoning System (Shapiro, 2000), a system with a logic that targets natural language understanding and common-sense reasoning. A SNePS network is said to be *propositional*, because all propositions in the network are represented by nodes. In SNePS, there are *nodes* and labeled directed edges called *arcs*. All nodes represent concepts, either objects or propositional concepts. When information is added to the network,

it is added as a node with arcs emanating from it to other nodes. We load a SNePS network a priori with static information about the home heating system, for example, that the oil nozzle is a component of the oil burner. Then transitory information is added to the network to make a response to each of the student's consultation requests.

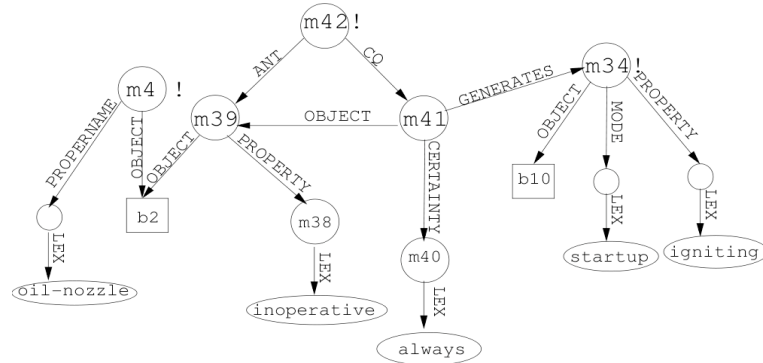


Fig.6. SNePS Network for: If m39 [b2 is inoperative], then m41 [m39 causes m34! (b10 igniting in startup mode) with certainty always].

Figure 6 shows an example of a SNePS network that is constructed to generate part of the middle response in Figure 3. SNePS uses an assertion flag (represented with a !) to distinguish what is asserted as true from other propositions. **m42!** is a rule that asserts that if **m39** (**ANT** = antecedent) then **m41** (**CO** = consequent). **m39** is the proposition that object **b2** is inoperative, and **m41** is the proposition that the situation represented by **m39** always causes **m34!**. **m34!** asserts that object **b10** is igniting in startup mode. In summary, **m42!** asserts that if **b2** is inoperative, then its failure always causes **b10** to not flow in startup mode. Another network segment **m4!** asserts that **b2** is called an *oil-nozzle*. While **m4!**, **m81!** and **m34!** are asserted as true, propositions **m39** and **41** are not asserted, indicating that they are not known to be true.

SNePS makes it easy to recognize structural similarities and use shared structures. The subnetworks we build for each possible aggregation of the specific feedback to be communicated make it easy to recognize the presence of different scalar values, like “never” and “sometimes”, or “never” and “always”. As a consequence, rhetorical relations like *contrast* and *concession* can be inserted to highlight distinctions between dimensional values (see Figure 3, middle). Furthermore, SNePS propositions can be treated as discourse entities, added to the discourse model and referred to (see *This is ... caused if ...* in Figure 3, middle). To generate referential expressions, *DIAG-NLP1* uses the GNOME algorithm (Kibble & Power, 2000). Further details on *DIAG-NLP1* and its usage of SNePS can be found in (Haller, Di Eugenio, & Troilo, 2002; Haller & Di Eugenio, 2003).

DIAG-NLP2: abstraction and functional aggregation

Since *DIAG-NLP1* was developed on the basis of the authors' informed introspection, we felt the need to verify that aggregation rules in *DIAG-NLP1* were empirically grounded. As it turns out, we found

that in the human data syntactic and structural aggregation is almost absent, but that a different kind of aggregation, based more on abstraction and function than on structure, takes place. This shows the value of a corpus study to ground the verbal feedback provided by an ITS. It is in fact *DIAG-NLP2*, the corpus based version of the feedback generator, that is more effective.

Corpus study

We collected 20 tutoring interactions between a student using the DIAG application on home heating and two human tutors, for a total of 253 tutor turns, of which 216 were in reply to *ConsultRU* and 37 in reply to *ConsultInd* (the type of student query is automatically logged). The tutor and the student are in different rooms, sharing images of the same DIAG tutoring screen. When the student consults the system via the Consult menu (see Figure 2), the tutor sees, in tabular form, the information that *DIAG-orig* would use in generating its advice (the same “fact” file that the application gives to *DIAG-NLP1* and *DIAG-NLP2*, see Figure 5) and types a response that substitutes for *DIAG-orig*'s. The tutor is presented with this information because of our goal of uncovering empirical evidence for aggregation rules in our domain. Although we could not constrain the tutor to mention only the facts that *DIAG-orig* would have communicated, in this case we can analyze whether and how the tutor uses the information provided by *DIAG-orig*.

We developed a coding scheme (Glass, Raval, Di Eugenio, & Traat, 2002) and annotated the data. As the annotation was performed by a single coder,⁵ we lack measures of intercoder reliability. This is a shortcoming of our corpus study, and hence what follows should be taken as observations rather than as rigorous findings. However, a different source of validation for our corpus study comes from our evaluation of *DIAG-NLP2*. Since *DIAG-NLP2* is based on these observations and its language fosters the most learning, it indirectly confirms that the phenomena uncovered in the human data are generalizable, similar to what high enough intercoder reliability would suggest.

Similar to other coding schemes for tutorial interaction (Pilkington, 1997; Kim, 1999; Pilkington, 2001; Core, Moore, & Zinn, 2002; Person, 2006), two important components of our coding scheme are: *tutoring move*, what sort of tutoring actions tutors perform; and *topic*, what the tutors talk about, that we limit to the objects and relations in the domain. In addition, we code for *referring modality/aggregation* and *relation to DIAG output*. The latter two groups of tags/attribute values (each tag has from one to five additional attributes that need to be annotated too) were developed with an eye to the specific phenomenon we were attacking in this project, aggregation. We were interested in how tutors refer to objects in the domain, because this has bearing on whether/how they aggregate over those objects. The *relation to DIAG output* was used to mark whether tutors would convey all the information that *DIAG-orig* would: our hypothesis, later confirmed, was that they would not.

Figure 7 shows examples of some of the tags – the SCM is the System Control Module (in the interest of clarity we omit some tags and all attribute/value pairs).

Tutoring move. Several efforts have been devoted to systematically classifying teaching and tutoring discourse, starting from (Sinclair & Coulthard, 1975) and continuing with many others (Chi et al., 2001; Evens & Michael, 2006; Person, 2006), including our own classification in a second project in which we explored the differences between expert and non expert tutors (Lu, Di Eugenio,

[*judgment* [RU the ignitor] is a poor suspect] since [*indication* combustion is working] during startup. The problem is that [RU the SCM] is shutting [RU the system] off during heating.

[*explanation* The [RU SCM] reads [*indicator* input signals] from [RU sensors] and uses [*indicator* the signals] to determine how to control [RU the system].]

[*problem-solving* Check [RU the sensors].]

Fig.7. Examples of a coded tutor reply.

Kershaw, Ohlsson, & Corrigan-Halpern, 2007). In the DIAG-NLP project, we were only interested in a rough characterization of tutoring moves, since we had no plans to set up a full dialogue interface, for which we would have needed a more refined analysis. Thus, we settled on three different tutoring moves:

- *Judgment*. The tutor evaluates what the student did, as in *The ignitor is a poor suspect* in Figure 7.
- *Explanation*. The tutor imparts domain knowledge with one or more statements, as in the middle sentence in Figure 7.
- *Problem solving*. The tutor suggests the next course of action, as the last sentence in Figure 7.

Topic. We tag objects in the domain with their class *indicator* or *RU*. In Figure 7, *input signals* is an indicator,⁶ while *the ignitor*, *the SCM*, *the system* and *the sensors* are replaceable units. Very often the states of indicators and RUs are mentioned as well, denoted by *indication* and *operationality*, respectively. For example, in Figure 7 *combustion is working* is an *indication* that indirectly refers to the indicator *visual combustion check*. Here is an example of the *operationality* tag:
[operationality [RU the nozzle] is not spraying].

Referring/aggregation. As discussed earlier, one of our major interests in this project was to see whether a more concise presentation of information would positively affect learning. Aggregation is performed on propositions, however the outcome often affects the objects those propositions predicate over. For example, one possible aggregation for *the boy in the red coat is running* and *the girl in the red skirt is running*, is *the children in red are running*, where aggregation in fact deeply affects the two original Noun Phrases rather than the Verb Phrases, which are aggregated by simply changing the number of the auxiliary. Hence, we were interested in how tutors would refer to the objects in the domain, including of course whether they would use aggregation on them.

These phenomena are captured via attributes associated to the *indicator* and *RU* tags, rather than via independent tags. The first is *directness*, with values *explicit* (the name is given explicitly), *implicit* (indicators or RUs are referred to implicitly) or *summary* over several unspecified indicators or RUs (*input signals* and *sensors* are labelled as *directness=summary*). The second is *aggregation*, with values *aggregate-object*, i.e. objects that physically include other components, such as *the system* or *the oil burner*; and *linguistic-aggregate*, i.e. plurals and conjunctions, such as *input signals* and *sensors* in Figure 7.

Relation to DIAG output. Contrary to all other tags, in this case we annotate the input that the DIAG application gave the tutor, i.e. lists of facts such as the one in Figure 5. We tag each fact as *included/excluded/contradicted*, according to how it has been dealt with by the human tutor.

A note on segmentation and on the granularity of the annotation. Tutors' turns were not segmented in advance of the annotation, and the mark-up tool (Glass & Di Eugenio, 2002) allowed for word-by-word annotation. However, the coding manual contained guidelines on the spans of text to select according to the different tag types. For tutoring move, the minimal unit of annotation was a main clause; some further guidelines concerned whether to include other main clauses and/or subordinate clauses as part of the same move. For topic, the basic tags (*indicator* and *RU*), when explicit, are assigned to NPs including appositions and restrictive relative clauses, while *indication* and *operationality* can span any type of constituent. Since *aggregation* is marked not via independent tags but via attributes on *indicator* or *RU* tags, issues of segmentation do not arise. Finally, since DIAG's output is preformatted (see Figure 5), annotators are instructed to tag it line by line.

Turning now to some quantitative analysis of the data, tutors provide explicit problem solving directions in 73% of the replies, evaluate the student's action in 45% of the replies, and provide background knowledge in 37% of the replies (clearly, many of the replies include multiple moves, as in Figure 7 which includes all three). In contrast, and by design (Towne, 1997), a DIAG application never provides explicit problem solving directions or explicit domain knowledge in its feedback when answering *ConsultInd/ConsultRU* queries. Students would have to explicitly ask for problem solving directions via the *what to do next* button in the input menu, cf. Figure 2 (recall that this button was disabled in all our experiments). Instead, our tutors provide explicit problem solving directions in more than two thirds of their answers to *ConsultInd/ConsultRU* queries. Even if this behavior of our tutors seems to contradict research that shows tutors are most effective when they let students construct knowledge by themselves (Chi et al., 2001; Evens & Michael, 2006), it is not as uncommon as those previous results would lead one to believe – we will provide more evidence in this regard in the Discussion section. Therefore, problem solving suggestions are included in the output *DIAG-NLP2* generates.

As expected, tutors are much more concise than *DIAG-orig*, e.g. they *exclude* much of the information that *DIAG-orig* would provide. To reply to the 253 queries in our corpus, *DIAG-orig* would have communicated 749 facts – as mentioned earlier, a *fact* is the basic unit of information that underlies each of the clauses in a reply by *DIAG-orig*, see Figure 5. Of these, 63% are excluded. Interestingly, tutors never mention RUs that cannot or are not as likely to cause a certain problem, such as, respectively, the *ignitor assembly* and *the SCM* in Figure 5.

Another important finding from the data concerns how tutors prefer to refer to objects. Two values of *directness* encode whether the tutor explicitly talks about the indicator (e.g. *The water temperature gauge reading is low*), or implicitly via the object to which the indicator refers (e.g. *the water is too cold*). Out of 193 indicators, 45, i.e. 23%, are marked as *explicit*, 110 out of 193, i.e. 57%, are marked as *implicit*, and 38, i.e. 20%, are marked as *summary*. This, and the 137 occurrences of *indication*, prompted us to refer to objects and their states, rather than the indicators whose readings represent the states of those objects. The situation is reversed for replaceable units: out of 551 replaceable units, 410

are marked as explicit (74.4%), 40 as implicit (7.3%), and 101 (18.3%) as summary. Hence, we will refer to RUs directly, via their names.

As mentioned above, aggregation is encoded indirectly via the *summary* value of the attribute *directness*, and via the *aggregate-object/linguistic-aggregate* values of the attribute *aggregation* proper. 101 out of 551 RUs, i.e. 18%, are labelled as summary; 38 out of 193 indicators, i.e. 20%, are labelled as summary. In addition, there are 46 occurrences of *aggregate object*. The value *linguistic-aggregate* for *aggregation* on indicators and RUs and the value *summary* for *directness* always co-occur on plurals: *input signals* and *sensors* are also labelled as *linguistic-aggregate* (not shown in Figure 7).⁷ These percentages, though seemingly low, represent a considerable amount of aggregation, since in our domain some items have very little in common with others, and hence cannot be aggregated, e.g. the only way to aggregate *the oil burner* and *the water pump* would be to refer to them as replaceable units. Further, tutors eschew long lists of parts and aggregate them functionally rather than syntactically – i.e. they prefer to abstract away from the individual parts and give a functional description of them. For example, the same assemblage of parts, i.e. oil nozzle, oil supply valve, oil pump, oil filter, etc., is described functionally by the tutors as *the other items on the fuel line* or as *the path of the oil flow*, rather than syntactically as e.g. *DIAG-NLP1* does (see middle of Figure 3).

Feedback Generation in *DIAG-NLP2*

1. IND ← queried indicator
 2. Mention the referent of IND and its state
 3. IF IND reads abnormal,
 - (a) REL-RUs ← choose relevant RUs
 - (b) AGGR-RUs ← AGGREGATE(REL-RUs)
 - (c) Suggest to check AGGR-RUs
- AGGREGATE(RUs)
1. Partition REL-RUs into subsets by system structure
 2. Apply functional aggregation to subsets

Fig.8. *DIAG-NLP2*: Feedback generation for *ConsultInd*.

The algorithms in Figures 8 and 9 show the control flow in *DIAG-NLP2* for feedback generation for *ConsultInd* and *ConsultRU*. With respect to *DIAG-NLP1*, they constitute a more substantive planning module which manipulates the information before passing it to EXEMPLARS. In contrast, in *DIAG-NLP1*, the fact file provided by *DIAG-orig* gets converted into a SNePS representation, which is then minimally manipulated before being processed by EXEMPLARS. Recall that *DIAG-NLP2*, like *DIAG-NLP1*, is given the facts *DIAG-orig* would communicate by the backend system. Thus when we refer to the student's knowledge, we have access to it because the ITS engine keeps track of what tests the student has performed, and uses this knowledge to decide which facts to communicate.

1. $RU \leftarrow$ queried RU
REL-IND \leftarrow indicator associated to RU
2. IF RU warrants suspicion
 - (a) state RU is a suspect
 - (b) IF student knows that REL-IND is abnormal
 - i. remind him of referent of REL-IND and its abnormal state
 - ii. suggest to replace RU
 - (c) ELSE suggest to check REL-IND
3. ELSE
 - (a) state RU is not a suspect
 - (b) IF student knows that REL-IND is normal
 - i. use referent of REL-IND and its normal state to justify judgment
 - (c) IF student knows of abnormal indicators OTHER-INDs
 - i. remind him of referents of OTHER-INDs and their abnormal states
 - ii. $AGGR-RUs \leftarrow \emptyset$
 - iii. FOR each OTHER-IND
 - A. $REL-RUs \leftarrow$ RUs associated with OTHER-IND
 - B. $AGGR-RUs \leftarrow AGGREGATE(REL-RUs) \cup AGGR-RUs$
 - iv. Suggest to check AGGR-RUs

Fig.9. *DIAG-NLP2*: Feedback generation for *ConsultRU*.

The algorithms in Figures 8 and 9 embody observations and findings discussed earlier in the paper. First, we noted that human tutors do not discuss RUs that either cannot cause the problem, or only rarely cause it. Step 3a in Figure 8 chooses, among all the RUs that *DIAG-orig* would talk about, only those that would definitely result in the observed symptom. Second, our corpus study shows that human tutors talk about the state of objects, rather than about the indicators that represent the state of those objects. This is implemented via Step 2 in Figure 8, and Steps 2(b)i, 3(b)i, 3(c)i in Figure 9. Specifically, Step 2 in Figure 8 generates *The combustion is abnormal* in Figure 3. Steps 3(b)i and 3(c)i in Figure 9 generate *The water pump sound is OK* and *You have seen that the combustion is abnormal* in Figure 4. Third, earlier in the paper, we justified the need for NLG techniques on the basis of a combinatorial argument: e.g. for each RU any possible subset of indicators may have been seen, and hence, a corresponding number of different messages needs to be generated. This potential combinatorial explosion is addressed by Step 3c in Figure 9 and by the AGGREGATE procedure.

As concerns aggregation per se, Step 2 in the AGGREGATE procedure in Figure 8 uses a simple heuristic to decide whether and how to use functional aggregation. For each RU, its possible aggregators

and the number n of units it covers are listed in a table (e.g. *electrical devices* covers 4 RUs, *ignitor*, *photoelectric cell*, *transformer* and *burner motor*). If a group of REL-RUs contains k units that a certain aggregator *Agg* covers, if $k < \frac{n}{2}$, *Agg* will not be used; if $\frac{n}{2} \leq k < n$, *Agg* preceded by *some of* will be used; if $k = n$, *Agg* will be used. This is how the last sentence in *DIAG-NLP2*'s reply in Figure 4 is generated.

Unlike *DIAG-NLP1*, *DIAG-NLP2* does not use SNePS, but a relational database storing relations, such as the ISA hierarchy (e.g. *burner motor* IS-A RU), information about referents of indicators (e.g. *room temperature gauge* REFERS-TO *room*), and correlations between RUs and the indicators they affect. In *DIAG-NLP2* we choose to move away from SNePS because of maintainability and transparency considerations. SNePS has a steep learning curve, which makes it difficult to maintain it if there is any turnover in personnel (which is the norm among graduate students). Thus, given our needs for KR were not too complex, we turned to a more standard approach, i.e. relational databases: they provide a representation that is simple, easy to read, easy to maintain, easy to access from Java via JDBC (a standard interface), yet powerful enough to perform some automated reasoning. The only feature we lost by not using SNePS was the ability to generate references to propositions. Given our results, it appears this feature did not affect learning.

A second difference from *DIAG-NLP1* is that in *DIAG-NLP1* EXEMPLARS directly produces strings, via simple templates. In contrast, the *DIAG-NLP2* algorithms produce one or more *Sentence Structure* objects. EXEMPLARS transforms those into Deep Syntactic Structures, which in turn are translated into English sentences by the sentence realizer RealPro (Lavoie & Rambow, 1997).

EVALUATION

Our empirical evaluation is a three group, between-subject study: one group interacts with *DIAG-orig*, one with *DIAG-NLP1*, one with *DIAG-NLP2*. The 75 subjects (25 per group) were all science or engineering majors affiliated with our university. Each subject read some short material about home heating, went through one trial problem, then continued through the curriculum on his/her own. The curriculum consisted of three problems of increasing difficulty. As there was no time limit, every student solved every problem. Reading materials and curriculum were identical in the three conditions.

While a subject was interacting with the system, a log was collected including, for each problem: whether the problem was solved; total time, and time spent reading feedback; how many and which indicators and RUs the subject consults *DIAG-orig* about; how many, and which RUs the subject replaces. We will refer to all the measures that were automatically collected as *performance measures*.

At the end of the experiment, each subject was administered a questionnaire divided into three parts, the first two on assessment and the third on usability. Specifically, the first part consists of three questions and tests what the student learned about the domain and the diagnosing task. The second part concerns whether subjects remember their actions, specifically, the RUs they replaced.

To summarize, we found that subjects who used *DIAG-NLP2* scored significantly higher on the questionnaires, cumulatively and on the two harder questions, and were significantly more correct (higher precision) in remembering what they did, than subjects that interacted with either of the other two systems, *DIAG-orig* and *DIAG-NLP1*.⁸ As regards performance measures, there are not such clear cut

Table 1

Assessment: Answers to Questions

	Question 1	Question 2	Question 3	Average
<i>DIAG-orig</i>	0.728	0.740	0.641	0.704
<i>DIAG-NLP1</i>	0.800	0.726	0.507	0.678
<i>DIAG-NLP2</i>	0.856	0.934	0.854	0.882

results. As regards usability, subjects prefer *DIAG-NLP1/DIAG-NLP2* to *DIAG-orig*, however results are mixed as regards which of the two they actually prefer.

We will now discuss all the different measures in detail. In the tables that follow, boldface indicates significant differences, as determined by an analysis of variance performed first via ANOVA, followed by Tukey post-hoc tests if necessary (significance is measured as $p \leq 0.05$, as customary).

Table 1 reports mean scores on the answers to the three questions, individually and average across the three questions.⁹ Additionally, performance on individual questions is presented visually in Figure 10 (note that the origin is at 0.50, not at 0). Specifically, the three questions are:

1. Describe the main subsystems of the furnace.
2. What is the purpose of (a) the oil pump (b) the system control module?
3. Assume the photoelectric cell is covered with enough soot that it could not detect combustion. What impact would this have on the system?

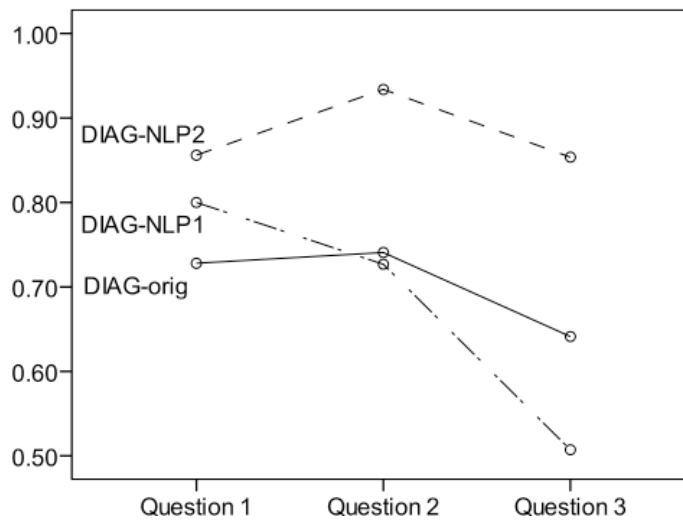


Fig.10. Scores on the three questions, by system.

As concerns cumulative scores, differences were found between the three systems ($F = 10.411$, $p = 0.000$). Tukey post-hoc tests revealed that subjects who interacted with *DIAG-NLP2* had a higher

Table 2
Assessment: RU Precision and Recall

	RU Precision	RU Recall
<i>DIAG-orig</i>	0.778	0.53
<i>DIAG-NLP1</i>	0.698	0.47
<i>DIAG-NLP2</i>	0.909	0.40

cumulative test score ($M = 0.882$, $SD = 0.11$) than subjects who interacted with each of the other two systems ($p \leq 0.001$ in both cases). We also found an effect of problem ($F = 7.364$, $p = 0.001$). Tukey post-hoc tests revealed that the performance on Question 3 is significantly lower ($M = 0.667$, $SD = 0.339$) than the performance on the other two questions. Finally, we found a significant interaction between system and problem ($F = 2.439$, $p = 0.048$).

Turning now to scores on each individual questions, significant differences were found on questions 2 and 3 ($F = 8.481$, $p = 0.000$, and $F = 7.909$, $p = 0.001$, respectively) and marginally significant differences on Question 1 ($F = 2.774$, $p = 0.069$). Tukey post-hoc tests confirm that it is subjects who interact with *DIAG-NLP2* who have higher scores on questions 2 and 3 ($M = .934$, $SD = 0.095$ and $M = 0.854$, $SD = 0.27$ respectively) than subjects who interacted with either *DIAG-orig* ($p = 0.003$ for Question 2, and $p = 0.043$ for Question 3), or *DIAG-NLP1* ($p = 0.001$ for both questions); for Question 1, Tukey post-hoc tests are non significant. The questions are increasingly difficult. The fact that the difference in scores is marginally significant for Question 1 but highly significant for questions 2 and 3 confirms that *DIAG-NLP2* is helping students acquire “deep” knowledge more effectively than *DIAG-orig* and *DIAG-NLP1*. The interaction we found between system and problem, as mentioned above, also appears to show that the better language *DIAG-NLP2* provides helps the most on the ‘deeper’ questions.¹⁰ We are using *deep knowledge* in an intuitive sense here, but at least Question 3 relates to transfer. To show this, we analyze our questions in terms of the taxonomy of knowledge and of cognitive processes proposed in (Anderson et al., 2001). They categorize knowledge into four categories: a. factual, b. conceptual, c. procedural, d. metacognitive. Further, they discuss six different kinds of cognitive processes: from simplest to most difficult, *remember*, *understand*, *apply*, *analyze*, *evaluate* and *create*. Other than *remember*, all other cognitive processes indicate *transfer*. All types of knowledge other than *factual* involve cognitive processes other than *remember*. Going back to our questions now, the first one tests factual knowledge and could be answered by simply *remembering* the appropriate diagrams. The second question relates to conceptual knowledge and as a minimum requires *understanding*, in particular its subcomponent *interpreting* (“being able to express presented information in another form”). Finally, the third question is at the border of conceptual and procedural knowledge, and precisely focuses on the causal relationships between malfunctions of RUs and malfunctions of the entire system – to answer it, subjects need to use several of the more complex subprocesses within *understanding*, such as *inferring* and *explaining*.

We now turn to a measure of *task recall* for our subjects, as illustrated in Table 2. The link we postulate here between the specific type of recall we measured and learning is admittedly speculative and would require further investigation. However, it is informed by research we will briefly describe. *Recall* is used in many disciplines as a correlate of the variable of interest, e.g. various types of verbal

Table 3
Performance Measures

	Time	RU Replaced	ConsultInd	Avg. Reading Time	ConsultRU	Avg. Reading Time
<i>DIAG-Orig</i>	30'17"	8.88	22.16	8"	63.52	5"
<i>DIAG-NLP1</i>	28'34"	11.12	6.92	14"	45.68	4"
<i>DIAG-NLP2</i>	34'53"	11.36	28.16	2"	52.12	5"
<i>Human Tutors</i>	38'54"	8.1	1.85	16.0"	10.8	12.0"

recall have been found to correlate with dementia (Kaltreider, Cullum, Lacritz, Brewer, & Filley, 1999), or with learning disabilities and language skills (Alloway & Gathercole, 2005). Closer to our concerns, sentence recall, or, more in general, text recall has been shown to correlate with learning in reading comprehension and in second language acquisition (Riley & Lee, 1996; Williams, 1999; Chang, 2006). In addition, (Anderson et al., 2001) defines *recall* as a subcomponent of *remembering*, namely, *finding relevant knowledge in long-term memory*. Importantly, *remembering* does not just apply to factual knowledge, but also to e.g. procedural knowledge, since before applying a procedure one needs to *recall* those procedures.

The term *recall* in the literature is used to refer to different measures. Here, we turn to information retrieval, where two measures, *precision* and *recall*, are precisely defined (from now on we will use *recall* in the information retrieval sense). Given a target set of documents, *precision* measures the percentage of target documents present among the retrieved documents, while *recall* measures the percentage of target documents that have been retrieved. In our case, we ask subjects to write down which RUs they replaced, and we compute these two measures with respect to the log the system has of the RUs they actually replaced. Precision is the ratio of the number of correct RUs they mention to all the RUs they say they replaced; whereas recall is the ratio of the number of correct RUs to the number of RUs they have actually replaced, e.g. suppose that a subject has replaced the oil filter, the oil nozzle and the oil pump, these three RUs will be recorded in the log. The subject however writes down that he replaced the oil filter and the water pump. In this case, precision is $\frac{1}{2}$ and recall is $\frac{1}{3}$.

Differences among the systems were found as concerns RU Precision ($F = 4.719, p = 0.012$). Tukey post-hoc tests revealed that subjects who interacted with *DIAG-NLP2* were more correct in remembering the RUs they had replaced ($M = .909, SD = 0.143$) with respect to subjects who interacted with *DIAG-NLP1*, but not with respect to subjects who interacted with *DIAG-orig*. There are no significant differences as far as RU recall is concerned.

We now turn to performance measures. They are reported in Table 3, cumulatively across the three problems, other than average reading times. The last line reports performance measures logged by the system during the data collection with human tutors.¹¹ We will come back to the difference between the human and the software conditions after we discuss the comparison between the three systems.

Subjects who used one of the three systems, *DIAG-orig*, *DIAG-NLP1* or *DIAG-NLP2*, don't differ significantly in the time they spend solving the problems, or in the number of *RU replacements* they perform. The system's assumption (known to the subjects) is that there is only one broken RU per problem, but the simulation allows subjects to replace as many as they want before they come to the correct solution. It would be desirable if subjects learned to replace as few RUs as possible; there is no

penalty in the system for replacing a working RU, but in real life this corresponds to wasting time and money.

The next four entries in Table 3 report the number of queries that subjects ask, and the average time it takes subjects to read the feedback. Note that reading times are approximate. Since we don't have eye-tracking equipment, we compute reading time as the time elapsed between when the feedback window pops up and when the subject clicks the OK button to dismiss the window – the subject has to dismiss the window otherwise no other buttons are clickable. Differences among the systems were found as concerns the number of *ConsultInd* queries ($F = 8.905, p = 0.000$), and average reading time for *ConsultInd* queries ($F = 15.266, p = 0.000$). Tukey post-hoc tests revealed that subjects who interacted with *DIAG-NLP1* asked fewer *ConsultInd* queries ($M = 6.92, SD = 8.261$), and subjects who interacted with *DIAG-NLP2* read the feedback for *ConsultInd* queries significantly faster ($M = 2'', SD = 1''$). The latter result is not surprising, since the feedback in *DIAG-NLP2* is much shorter than in *DIAG-orig* and *DIAG-NLP1*. Neither the reason nor the significance of subjects asking fewer *ConsultInd* questions of *DIAG-NLP1* are apparent to us – it happens for *ConsultRU* as well, to a lesser, not significant degree. Feedback in *DIAG-NLP1* is much closer to that of *DIAG-orig* than to that of *DIAG-NLP2*. Hence, we would expect *DIAG-orig* and *DIAG-NLP1* to cluster, instead subjects ask a comparable number of queries in *DIAG-orig* and *DIAG-NLP2*.

With the numbers from Table 3 in hand, we can now better assess the combinatorial argument we gave earlier in the paper. We argued that, since a student could inspect any subset of the 17 indicators present in the system, for every possible fault there would be 131,072 potential contexts to be verbalized, corresponding to the cardinality of the powerset of the indicators. The sheer number of indicators students inspect, as given by the column *ConsultInd* in Table 3, is quite high, at least in *DIAG-orig* and *DIAG-NLP2*. Since that number is cumulative through the three problems the students solve, we can conclude that students inspect between 7 and 9 indicators per problem, other than in *DIAG-NLP1*. Further, we checked whether this subset of indicators tends to always contain the same 7 (or 9) indicators. We found that, out of the original 17 indicators, only two are indeed never or almost never consulted (e.g. in *DIAG-orig*, they are visited respectively zero and 3 times *in total* across the 25 subjects and 3 problems); however, each of the other 15 indicators is visited on average between once and twice per subject, across the three problems – hence, the amount of variability across which specific subset of indicators each subject consults is quite high. It is therefore legitimate to equate the number of possible contexts per fault, i.e. per problem, to the number of all the possible ways we can choose a set of 7 (or 9) indicators out of the 15 indicators that are indeed consulted by the students. This corresponds to the binomial coefficient, which still results in thousands of contexts, even if we take the lower number 7: $\binom{15}{7} = 6435$. Even if considerably lower than the original number of contexts we estimated, this number is still too high to envision authoring messages by hand. Additionally, our subjects do go back to revisit the same indicator in the same problem: if we allow for repetitions within the set of 7 indicators, the total number of possible contexts climbs back up to 116,280.

We will now compare performance measures across the systems and the human tutors. The interactions with human tutors do not properly constitute a 'human condition', since they were collected previous to those with the *DIAG* systems, and in fact, they are the foundations for the design of *DIAG-*

Table 4
User preferences among the three systems

	prefer	neutral	disprefer
<i>DIAG-NLP1</i> to <i>DIAG-orig</i>	28	5	17
<i>DIAG-NLP2</i> to <i>DIAG-orig</i>	34	1	15
<i>DIAG-NLP2</i> to <i>DIAG-NLP1</i>	24	1	25

Table 5
DIAG-NLP2 versus *DIAG-NLP1*

	prefer	neutral	disprefer
Consult Ind.	8	1	16
Consult RU	16	0	9

NLP2. However, in our opinion the comparison between features of the two types of interactions is meaningful for several reasons. First, the 20 subjects who interacted with the 2 human tutors were taken from precisely the same science and engineering major population at UIC as the students who later interacted with the three *DIAG* systems. Second, they used the same interface and went through the same curriculum of 4 problems, the first used as a trial problem as discussed earlier. In fact, since our human tutors were shown what *DIAG-orig* would have said, in a sense they were more constrained than human tutors employed in other comparisons between human tutors and ITSs, e.g. those in (VanLehn et al., 2007). This said, some of the measures in Table 3 give rise to highly significant differences ($p \leq 0.0001$) as concerns number of *ConsultInd* ($F = 13.877$), number of *ConsultRU* ($F = 13.451$) and reading time for *ConsultRUs* ($F = 11.817$). Additionally, there is a marginally significant difference as concerns reading time for *ConsultInd* ($F = 2.197, p = 0.094$). Post-hoc Tukey tests reveal that subjects in the human conditions asked fewer *ConsultRUs* ($M = 10.8, SD = 5.1$) and read them more carefully ($M = 12'', SD = 9''$) than in any of the the software conditions. As concerns the number of *ConsultInd* ($M = 1.85, SD = 2.32$), there is a significant difference between the human condition and each of *DIAG-orig* and *DIAG-NLP2*, but not with *DIAG-NLP1*. It is then apparent that when interacting with our human tutors, students ask far fewer questions, and they read them much more carefully (even if they still replace more RUs than one would wish for). It has been observed elsewhere that students don't read the output of instructional systems (Heift, 2001). Alternatively, or in addition, they may be gaming the system (Baker et al., 2006). We will come back to these issues in the Discussion section.

We also collected usability measures - this has become increasingly frequent in ITS evaluations, e.g. see (Evens & Michael, 2006). These measures are important as well, since in a real setting, students should be more willing to sit down with a system that they perceive as more friendly and usable. Subjects rate the system along four dimensions on a five point scale: clarity, usefulness, repetitiveness, and whether it ever misled them (the scale is appropriately arranged: the highest clarity but the lowest repetitiveness receive 5 points). There are no significant differences on individual dimensions. Cumulatively, *DIAG-NLP2* (at 15.08, out of a highest possible rating of 20 points) slightly outperforms the other two (*DIAG-orig* at 14.68 and *DIAG-NLP1* at 14.32), however, the difference is not significant.

In the last part of the questionnaire, subjects compare two pairs of versions of feedback: in each pair, the first feedback is generated by the system they just worked with, the second is generated by one of the

Table 6
Reasons for system preference

	natural	concise	clear	contentful
<i>DIAG-NLP1</i>	4	8	10	23
<i>DIAG-NLP2</i>	16	8	11	12

other two systems. Subjects say which version they prefer, and why – they can judge the system along one or more of four dimensions: natural, concise, clear, contentful. The first two lines in Table 4 show that subjects prefer the NLP systems to *DIAG-orig* (marginally significant, $\chi^2 = 9.49, p < 0.1$), and the last line, that an equivalent number of subjects prefer *DIAG-NLP1* to *DIAG-NLP2*, or vice versa. However, a more detailed analysis (Table 5) shows that subjects prefer *DIAG-NLP1* for feedback to *ConsultInds*, but *DIAG-NLP2* for feedback to *ConsultRus* (marginally significant, $\chi^2 = 5.6, p < 0.1$). Since students ask many more *ConsultRus* than *ConsultInds*, this preference for *DIAG-NLP2* regarding *ConsultRus* may contribute to *DIAG-NLP2*'s effectiveness. Finally, Table 6 shows that subjects find *DIAG-NLP2* more natural, but *DIAG-NLP1* more contentful ($\chi^2 = 10.66, p < 0.025$). This comparison is of course only suggestive, since a more telling usability evaluation would require subjects to use two systems for a substantive amount of time.

Finally, subjects can add free-form comments. Only about half did so, and the distribution of topics and of evaluations is too broad to be telling – for example, some comment on the graphical interface, some on the home heating domain, and just very few on the language feedback per se.

DISCUSSION AND CONCLUSIONS

In the introduction, we posed the two main questions that underlie all research on the automated generation of effective feedback messages in ITSs, specifically: 1. whether the interaction positively affects learning; and, 2. which specific features of the interaction are responsible for learning.

Our answer to Question number 1 is yes, at least as far as regards our task, simulation-based diagnostic training. We showed that *DIAG-NLP2*, the ITS endowed with more sophisticated language feedback, as based on human tutoring data in this domain, engenders significantly more learning than the other two systems: the original system *DIAG-orig*, which provides extremely repetitive feedback, and the first NL version we built, *DIAG-NLP1*, which aggregates the feedback in terms of syntax and structure of the simulated system. While *DIAG-orig* could be considered as a non competitive baseline against which to compare *DIAG-NLP2*, *DIAG-NLP1* instead constitutes a real competitor, since its feedback was conceived on the basis of much previous research. From a practical point of view, we also showed that relatively simple, rapidly deployable NLG techniques did lead to measurable improvements in students' learning.

Question number 2 is the most interesting from a cognitive/theory of learning point of view. The feedback in *DIAG-NLP2* differs from the feedback in *DIAG-orig* and *DIAG-NLP1* along three dimensions: functional aggregation, that stresses an abstract and more conceptual view of the relation between symptoms and faulty parts; being more strongly directive in suggesting what to do next; and, using referents of indicators instead of indicators (i.e. saying *the water is cold* rather than *the water temperature*

gauge is low). It is hard to pinpoint which of these three features is responsible for better learning; possibly, it is the fact they occur together that is effective.¹² Specifically, functional aggregation is justified from a general pragmatics point of view, namely, as an incarnation of one of Grice's Maxims, specifically, the *Be brief* submaxim of his Maxim of Manner, *Be perspicuous* (Grice, 1975). In fact, functional aggregation is a more efficient incarnation of that submaxim than the other kinds of aggregation we experimented with, syntactic and structural: it shortens the text more, and it raises the description at a higher level of abstraction. As concerns being more directive, the reader may disagree, or, at a minimum, be surprised that such feedback can positively affect learning. Findings such as (Chi et al., 1989; Fox, 1993; Hume, Michael, Rovick, & Evens, 1996; Chi et al., 2001; Aleven & Koedinger, 2002) show that students learn best when they construct knowledge by themselves, and therefore, that tutors should limit themselves to prompting and scaffolding. However, the analysis of tutoring dialogues, both ours and others, shows that in fact many expert tutors do use direct instruction to a much larger extent than one would expect (not all expert tutors do, though (Evens & Michael, 2006)). For example, we studied the difference between expert and non expert tutors in a sequence learning domain (Kotovsky & Simon, 1973). The expert tutor, who is more effective than the other tutors, provides explicit problem solving directions 17% of the time, and much more than the other two tutors do (Lu et al., 2007) (other frequent moves of this tutor, out of 13 we coded for, are prompting 18% of the time, and summarizing 17% of the time). Person's 14 expert tutors provide direct instruction 38% of the time, with *prompt*, the next most frequent move out of 13 cognitive scaffolding moves, occurring 16% of the time (Person, 2006). Thus, we cannot rule out that direct instruction, at appropriate rates and in appropriate contexts, is actually effective.¹³ We are in fact investigating the role of direct instruction in our current projects, discussed below. Finally, the referring expressions we use – that refer directly to an invisible object, rather than to the visible indicator that encodes that object's state – can be seen as *immediate situation use* (Hawkins, 1978).¹⁴ Descriptions of this kind, such as *the dog* in the prototypical *Beware of the dog* notice, refer to an object in the situation in which the definite description is uttered. For this usage to be felicitous, it is not necessary that the referent of the description be visible. In our case, the immediate situation consists of the graphical simulation, in fact, our descriptions are uttered in a multimodal context. Much is known about the effect of referring expressions on comprehension, and the literature on this topic is huge, including much work in psycholinguistics (among many others, (Clark, 1992, 1996)), and work on referring expressions and multimodality (Oviatt, DeAngeli, & Kuhn, 1997; Kehler, 2000; Chai, Prasov, Blaim, & Jin, 2005). However, we are not aware of any studies on the specific type of referring expressions we employ, where there is a preference for a direct reference to an invisible object rather than to the visible embodiment of that object, or of studies that link referential expressions and learning in a problem solving setting. Intriguingly, in our DIAG application the referring expressions in question are shorter than or as long as the ones referring to the indicator that represents the object's state. We can then hypothesize that Grice's submaxim of brevity is at work for referring expressions as well, although we know that this maxim does not necessarily translate in the shortest possible referring expression (Dale & Reiter, 1995; Jordan, 2000).

Naturally, DIAG-NLP2 is still not equivalent to a human tutor. When we discussed performance measures (see Table 3), we observed that students appear not to read the output of any of our DIAG systems, as already observed for other Interactive Learning Environments (Heift, 2001). There may be

various reasons for this behavior. First, the replies from the human tutors must certainly be better than those by *DIAG-NLP2*, also because they freely refer to previous replies; instead, the dialogue context is just barely taken into account in *DIAG-NLP2* (by steps 2(b)i, 3(b)i and 3(c)i in Figure 9) and not taken into account at all in *DIAG-orig* and *DIAG-NLP1*. Alternatively, or in addition, this may be due to the *face* factor (Brown & Levinson, 1987; Moore et al., 2004), i.e. one's public self-image. We observed that, when interacting with any of the systems, some subjects simply ask for hints on every RU, thus displaying that they are not making any effort to solve the problem on their own - this partly explains the high number of *ConsultRUs* and *ConsultInd* in Table 3. This behavior could also be an instance of what (Baker et al., 2006) calls *gaming the system*, namely,

attempting to succeed in an educational task by systematically taking advantage of properties and regularities in the system used to complete that task, rather than by thinking through the material. In Cognitive Tutor software, gaming the system includes systematic guessing, and repeatedly asking for an additional hint until the software gives the student the answer (<http://joazeirodebaker.net/ryan/gaming.html>)

Turning to the design of interfaces for ITSs and more in general, for ILEs, some readers may be skeptical of the value of Natural Language per se. Still, most environments do include some language feedback couched in one way or the other, e.g. as simple as captions to graphics, or as complex as hints of different sorts. We hope we have convinced even the more skeptical readers that: 1. language feedback can be effective, even when a full interaction is not necessary or feasible; 2. corpus studies are necessary to understand the content and form of that feedback, since even the most informed analysis of what those messages should contain cannot fully reflect what human tutors would do in those circumstances; and 3. if the number of possible contexts in which different feedback needs to be provided is high, NLG techniques that go beyond template filling should be used.

The *DIAG-NLP* project has come to a close. We are satisfied that we demonstrated that even not overly sophisticated NL feedback can make a difference; however, the fact that *DIAG-NLP2* has the best language and engenders the most learning prompted us to explore more complex language interactions. We have been pursuing new exciting directions in two new domains. In the first domain, abstract sequence learning, we have been exploring the difference between expert and non expert tutors (Lu et al., 2007; Lu, 2007). In the second, learning of basic data structures and algorithms in Computer Science (Fossati, Di Eugenio, Brown, & Ohlsson, 2008), we are moving towards a different paradigm of data analysis that focuses on effective tutoring *sessions*, rather than on effective *tutors*. This shift in perspective is motivated by the fact that it is impossible to define tutoring expertise a priori, and that not every tutor succeeds with every student (Ohlsson et al., 2007).

ACKNOWLEDGEMENTS

This work was supported by the Office of Naval Research (awards N00014-99-1-0930, N00014-00-1-0640, and N00014-07-1-0040), and in part by the National Science Foundation (award IIS 0133123). We are grateful to CoGenTex Inc. for making EXEMPLARS and RealPro available to us. Thanks to the other students who helped us with the data collection, the implementation and/or evaluation of the

DIAG-NLP systems over the years, M. Trolio, M. Traat, R. Serafin and Y. Hou. Finally we thank three anonymous reviewers whose comments greatly contributed to improving the paper.

REFERENCES

- Aleven, V., & Koedinger, K. R. (2002). An effective metacognitive strategy: learning by doing and explaining with a computer-based cognitive tutor. *Cognitive Science*, 26, 147-179.
- Aleven, V., Koedinger, K. R., & Popescu, O. (2003). A tutorial dialog system to support self-explanation: Evaluation and open questions. In H. U. Hoppe, F. Verdejo & J. Kay (Eds.) *Artificial Intelligence in Education Shaping the Future of Learning through Intelligent Technologies* (p. 39-46). Amsterdam: IOS Press.
- Aleven, V., Stahl, E., Schworm, S., Fischer, F., & Wallace, R. (2003). Help seeking and help design in interactive learning environments. *Review of Educational Research*, 73(3), 277-320.
- Alloway, T. P., & Gathercole, S. E. (2005). The role of sentence recall in reading and language skills of children with learning difficulties. *Learning and Individual Differences*, 15(4), 271-282.
- Anderson, L., Krathwohl, D., Airasian, P., Cruikshank, K., Mayer, R., Pintrich, P., et al. (2001). *A taxonomy for learning, teaching, and assessing: a revision of Bloom's taxonomy of educational objectives*. New York: Longman.
- Arroyo, I., Woolf, B. P., & Beal, C. R. (2006). Addressing cognitive differences and gender during problem solving. *International Journal of Technology, Instruction, Cognition and Learning*, 4, 31-63.
- Baker, R., Corbett, A., Koedinger, K. R., Evenson, E., Roll, I., Wagner, A., et al. (2006). Adapting to When Students Game an Intelligent Tutoring System. In M. Ikeda, K. D. Ashley & T.-W. Chan (Eds.) *Proceedings of the Eighth International Conference on Intelligent Tutoring Systems* (pp. 392-401). Berlin: Springer.
- Brown, P., & Levinson, S. (1987). *Politeness: Some universals in language usage*. Cambridge University Press.
- Burton, R. R., & Brown, J. S. (1979). Toward a natural language capability for computer-assisted instruction. In H. O'Neill (Ed.) *Procedures for instructional systems development* (p. 272-313). New York: Academic.
- Carbonell, J. (1970). AI in CAI: an artificial intelligence approach to computer-aided instruction. *IEEE Transactions on Man-Machine Systems*, 11, 190-202.
- Chai, J., Prasov, Z., Blaim, J., & Jin, R. (2005). Linguistic Theories in Efficient Multimodal Reference Resolution: an Empirical Investigation. In *IUI-05, The 10th International Conference on Intelligent User Interfaces* (pp. 43-50). San Diego, CA.
- Chang, Y.-F. (2006). On the use of the immediate recall task as a measure of second language reading comprehension. *Language Testing*, 23(4), 520-543.
- Chi, M. T. H., Siler, S. A., Yamauchi, T., & Hausmann, R. G. (2001). Learning from human tutoring. *Cognitive Science*, 25, 471-533.
- Chi, M. T. H., Bassok, M., Lewis, M. W., Reimann, P., & Glaser, R. (1989). Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science*, 13(2), 145-182.
- Clark, H. H. (1992). *Arenas of language use*. Chicago, IL: The University of Chicago Press.
- Clark, H. H. (1996). *Using language*. Cambridge University Press.
- Core, M. G., Moore, J. D., & Zinn, C. (2002). *Draft: Tutorial annotation scheme*. (University of Edinburgh. Manuscript)
- Dale, R., & Reiter, E. (1995). Computational Interpretations of the Gricean Maxims in the Generation of Referring Expressions. *Cognitive Science*, 18, 233-263.
- Dalianis, H. (1996). *Concise natural language generation from formal specifications*. Unpublished doctoral dissertation, Department of Computer and Systems Science, Stockholm University. (Technical Report 96-008)

- Di Eugenio, B., Fossati, D., Yu, D., Haller, S., & Glass, M. (2005a). Aggregation improves learning: experiments in Natural Language Generation for Intelligent Tutoring Systems. In *ACL05, Proceedings of the 42nd Meeting of the Association for Computational Linguistics* (pp. 50-57).
- Di Eugenio, B., Fossati, D., Yu, D., Haller, S., & Glass, M. (2005b). Natural Language Generation for Intelligent Tutoring Systems: a case study. In C.-K. Looi, G. McCalla, B. Bredeweg & J. Breuker (Eds.) *Artificial Intelligence in Education Supporting Learning through Intelligent and Socially Informed Technology*. Amsterdam, The Netherlands.
- Di Eugenio, B., Glass, M., & Trolio, M. J. (2002, July). The DIAG experiments: Natural Language Generation for Intelligent Tutoring Systems. In *INLG02, The Third International Natural Language Generation Conference* (pp. 120-127). Harriman, NY.
- Evens, M. W., & Michael, J. A. (2006). *One-on-one tutoring by humans and machines*. Mahwah, NJ: Lawrence Erlbaum Associates.
- Evens, M. W., Spitzkovsky, J., Boyle, P., Michael, J. A., & Rovick, A. A. (1993). Synthesizing Tutorial Dialogues. In *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society* (pp. 137-140). Hillsdale, New Jersey: Lawrence Erlbaum Associates.
- Fossati, D., Di Eugenio, B., Brown, C., & Ohlsson, S. (2008). Learning Linked Lists: Experiments with the iList System. In B. P. Woolf, E. Aïmeur, R. Nkambou & S. P. Lajoie (Eds.) *ITS 2008, the 9th International Conference on Intelligent Tutoring Systems*. Berlin: Springer.
- Fox, B. A. (1993). *The human tutorial dialogue project: Issues in the design of instructional systems*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Glass, M., & Di Eugenio, B. (2002, July). MUP: The UIC standoff markup tool. In *The Third SigDIAL Workshop on Discourse and Dialogue*. Philadelphia, PA.
- Glass, M., Raval, H., Di Eugenio, B., & Traat, M. (2002). *The DIAG-NLP dialogues: coding manual* (Tech. Rep. No. UIC-CS 02-03). University of Illinois - Chicago.
- Graesser, A. C., Person, N. K., & Magliano, J. P. (1995). Collaborative dialogue patterns in naturalistic one-to-one tutoring. *Applied Cognitive Psychology*, 9, 495-522.
- Graesser, A. C., Person, N., Lu, Z., Jeon, M., & McDaniel, B. (2005). Learning while holding a conversation with a computer. In L. PytlíkZillig, M. Bodvarsson, & R. Brunin (Eds.) *Technology-based education: Bringing researchers and practitioners together*. Information Age Publishing.
- Graesser, A. C., Lu, S., Jackson, G., Mitchell, H., Ventura, M., Olney, A., et al. (2004). AutoTutor: A tutor with dialogue in natural language. *Behavioral Research Methods, Instruments, and Computers*, 36, 180-193.
- Grice, H. (1975). Logic and Conversation. In P. Cole & J. Morgan (Eds.) *Syntax and semantics 3. speech acts*. Academic Press.
- Haller, S., & Di Eugenio, B. (2003). Minimal Text Structuring to Improve the Generation of Feedback in Intelligent Tutoring Systems. In I. Russell & S. Haller (Eds.) *FLAIRS 2003, the 16th International Florida AI Research Symposium* (pp. 382-386). Menlo Park, CA: AAAI Press.
- Haller, S., Di Eugenio, B., & Trolio, M. J. (2002). Generating Natural Language Aggregations Using a Propositional Representation of Sets. In S. Haller & G. Simmons (Eds.) *FLAIRS 2002, the 15th International Florida AI Research Symposium* (pp. 365-369). Menlo Park: AAAI Press.
- Hawkins, J. A. (1978). *Definiteness and indefiniteness: a study in reference and grammaticality prediction*. London: Croom Helm.
- Heift, T. (2001). Error-specific and individualized feedback in a web-based language tutoring system: Do they read it? *ReCALL Journal*, 13(2), 129-142.
- Huang, X., & Fiedler, A. (1996). Paraphrasing and Aggregating argumentative text using text structure. In *Proceedings of the 8th International Workshop on Natural Language Generation* (pp. 21-30). Sussex, UK.

- Hume, G. D., Michael, J. A., Rovick, A. A., & Evens, M. W. (1996). Hinting as a tactic in one-on-one tutoring. *Journal of the Learning Sciences*, 5(1), 23-47.
- Jordan, P. W. (2000). *Intentional influences on object redescriptions in dialogue: Evidence from an empirical study*. Unpublished doctoral dissertation, Intelligent Systems Program, University of Pittsburgh.
- Kaltreider, L. B., Cullum, C. M., Lacritz, L. H., Brewer, K., & Filley, C. M. (1999). Brief Recall Tasks and Memory Assessment in Alzheimer's Disease. *Applied Neuropsychology*, 6(3), 165-169.
- Kehler, A. (2000). Cognitive Status and Form of Reference in Multimodal Human-Computer Interaction. In *AAAI 00, The 15th Annual Conference of the American Association for Artificial Intelligence* (pp. 685-689). AAAI Press/MIT Press.
- Kibble, R., & Power, R. (2000). *Nominal generation in GNOME and ICONOCLAST* (Tech. Rep. No. ITRI-00-02). Brighton, UK: Information Technology Research Institute, University of Brighton.
- Kim, J. H. (1999). *A Manual for SGML Mark Up of Tutoring Transcripts (CIRCSIM transcripts annotation manual)*. (Illinois Institute of Technology. Manuscript)
- Kotovsky, K., & Simon, H. (1973). Empirical tests of a theory of human acquisition of information-processing analysis. *British Journal of Psychology*, 61, 243-257.
- Kumar, R., Rosé, C. P., Aleven, V., Iglesias, A., & Robinson, A. (2006). Evaluating the Effectiveness of Tutorial Dialogue Instruction in an Exploratory Learning Context. In M. Ikeda, K. D. Ashley & T.-W. Chan (Eds.) *Proceedings of the Seventh International Conference on Intelligent Tutoring Systems* (pp. 666-674). Berlin: Springer.
- Lavoie, B., & Rambow, O. (1997). A Fast and Portable Realizer for Text Generation Systems. In *Proceedings of the Fifth Conference on Applied Natural Language Processing* (pp. 265-268). ACL Press.
- Lepper, M. R., Drake, M. F., & O'Donnell-Johnson, T. (1997). Scaffolding techniques of expert human tutors. In K. Hogan & M. Pressley (Eds.) *Scaffolding student learning: Instructional approaches and issues*. Cambridge, MA: Brookline.
- Litman, D. J., Rosé, C. P., Forbes-Riley, K., VanLehn, K., Bhembe, D., & Silliman, S. (2006). Spoken versus typed human and computer dialogue tutoring. *International Journal of Artificial Intelligence in Education*, 16, 145-170.
- Lu, X. (2007). *Expert tutoring and natural language feedback in intelligent tutoring systems*. Unpublished doctoral dissertation, University of Illinois - Chicago.
- Lu, X., Di Eugenio, B., Kershaw, T., Ohlsson, S., & Corrigan-Halpern, A. (2007). Expert vs. non-expert tutoring: Dialogue moves, interaction patterns and multi-utterance turns. In *CICLING07, Proceedings of the 8th International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 456-467). Mexico City, Mexico.
- McKendree, J. (1990). Effective feedback content for tutoring complex skills. *Human-Computer Interaction*, 5, 381-413.
- McLaren, B. M., Lim, S., Yaron, D., & Koedinger, K. R. (2007). Can a Polite Intelligent Tutoring System Lead to Improved Learning Outside of the Lab? In R. Luckin, K. R. Koedinger & J. Greer (Eds.) *Artificial Intelligence in Education Building Technology Rich Learning Contexts that Work* (pp. 443-450). Amsterdam: IOS Press.
- Moore, J. D., Porayska-Pomsta, K., Varges, S., & Zinn, C. (2004). Generating Tutorial Feedback with Affect. In V. Barr & Z. Markov (Eds.) *FLAIRS04, Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference* (pp. 917-922). Menlo Park: AAAI Press.
- Moreno, R. (2006). Does the modality principle hold for different media? A test of the *method-affects-learning* hypothesis. *Journal of Computer Assisted Learning*, 22(3), 149-158.
- Ohlsson, S., Di Eugenio, B., Chow, B., Fossati, D., Lu, X., & Kershaw, T. (2007). Beyond the code-and-count analysis of tutoring dialogues. In R. Luckin, K. R. Koedinger & J. Greer (Eds.) *Artificial Intelligence in*

- Education Building Technology Rich Learning Contexts that Work* (pp. 349-356). Amsterdam: IOS Press.
- Oviatt, S., DeAngeli, A., & Kuhn, K. (1997). Integration and Synchronization of Input Modes during Multimodal Human-Computer Interaction. In *CHI 97, Proceedings of Conference on Human Factors in Computing Systems* (pp. 415-422).
- Paris, C. L. (1988). Tailoring object descriptions to the user's level of expertise. *Computational Linguistics*, 14(3), 64-78.
- Person, N. (2006). *Why study expert tutors?* Presentation at ONR Contractors' Conference on Instructional Strategies. University of California at Santa Barbara, Santa Barbara, Ca.
- Pilkington, R. M. (1997). *Analyzing educational dialogue: the DISCOUNT scheme* (Tech. Rep. No. 019703). Computer Based Learning Unit, The University of Leeds.
- Pilkington, R. M. (Ed.). (2001). *Special Issue on Analysing Educational Dialogue Interaction (Part II)* (Vol. 12). International Journal of Artificial Intelligence in Education.
- Pon-Barry, H., Schultz, K., Bratt, E. O., & Peters, S. (2006). Responding to student uncertainty in spoken tutorial dialogue systems. *International Journal of Artificial Intelligence in Education*, 16, 171-194.
- Power, R., Scott, D., & Bouayad-Agha, N. (2003). Document structure. *Computational Linguistics*, 29(4), 211-260.
- Reape, M., & Mellish, C. (1998). Just what is aggregation anyway? In *Proceedings of the European Workshop on Natural Language Generation* (pp. 20-29). Toulouse, France.
- Reiter, E., & Dale, R. (2000). *Building Natural Language Generation Systems*. Cambridge University Press.
- Riley, G., & Lee, J. (1996). A comparison of recall and summary protocols as measures of second language reading comprehension. *Language Testing*, 13, 173-189.
- Schooler, L. J., & Anderson, J. R. (1990). The Disruptive Potential of Immediate Feedback. In *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society* (pp. 702-708). Lawrence Erlbaum Associates.
- Shapiro, S. C. (2000). SNePS: A Logic for Natural Language Understanding and Commonsense Reasoning. In L. M. Iwanska & S. C. Shapiro (Eds.) *Natural Language Processing and Knowledge Representation*. AAAI Press/MIT Press.
- Shaw, J. (2002). A Corpus-based Analysis for the Ordering of Clause Aggregation Operators. In *COLING02, Proceedings of the 19th International Conference on Computational Linguistics* (pp. 1-7). Morristown, NJ: ACL Press.
- Shriver, K. (1997). *Dynamics in document design*. John Wiley & Sons, Inc.
- Sinclair, J., & Coulthard, M. (1975). *An Analysis of Discourse: The English Used by Teachers and Pupils*. Oxford University Press.
- Towne, D. M. (1997). Approximate reasoning techniques for intelligent diagnostic instruction. *International Journal of Artificial Intelligence in Education*, 8, 262-283.
- VanLehn, K., Graesser, A. C., Jackson, G. T., Jordan, P. W., Olney, A., & Rosé, C. P. (2007). When are tutorial dialogues more effective than reading? *Cognitive Science*, 31(1), 3-62.
- Wang, N., Johnson, W. L., Mayer, R. E., Rizzo, P., Shaw, E., & Collins, H. (2008). The politeness effect: Pedagogical agents and learning outcomes. *International Journal of Human-Computer Studies*, 66(2), 98-112.
- White, M., & Caldwell, T. (1998). EXEMPLARS: A Practical, Extensible Framework for Dynamic Text Generation. In *Proceedings of the Ninth International Workshop on Natural Language Generation* (pp. 266-275). Morristown, NJ: ACL Press.
- Williams, J. N. (1999). Memory, attention and inductive learning. *Studies in Second language Acquisition*, 21, 1-48.
- Wright, P. (1997). The way readers are influenced by layout. In *LAMPE'97: Lausanne Atelier sur Modeles de Page Electronique (Workshop on Electronic Page Models)*. Lausanne, Switzerland.

Zinn, C., Moore, J. D., & Core, M. G. (2005). Intelligent information presentation for tutoring systems. In M. Zancanaro & O. Stock (Eds.) *Multimodal intelligent information presentation* (pp. 227-254). Dordrecht: Kluwer Academic Publishers.

Notes

¹This should not be taken as a criticism of the original designers of the DIAG authoring tool, since their focus was not on providing language feedback.

²The name comes from DIAG.

³SNePS stands for *Semantic Network Processing System*.

⁴*DIAG-NLP1* actually augments and refines the first feedback generator we created for DIAG, *DIAG-NLP0* (Di Eugenio, Glass, & Trolio, 2002). *DIAG-NLP0* only covered (i) and (iv).

⁵There was an attempt at double coding but it yielded too little data to compute intercoder reliability.

⁶In the furnace system representation, these signals are physically represented as indicators to be read and tested.

⁷We disregard cases in which *linguistic-aggregate* is applied to conjunctions or disjunctions, since it is not clear what to make of cases such as *the water, oil and ignitor* or *the cutoff valve or the module*.

⁸We do not have a measure of learning in terms of learning gain, i.e. as a difference between a pre- and a post-test. In our domain, a true pre-/post-test pair should consist of a diagnostic problem. However, the curriculum we had been given consisted of only 4 problems, and it was not possible to develop more problems because of logistic/personnel issues. Thus, we used the first problem for the subject to get acquainted with the system, and the remaining three for the curriculum. The lack of usable problems is also the reason why we don't assess troubleshooting skills directly. Once we decided we would test learning via essay-like questions, it did not seem appropriate to pose them in a pre-/post-test fashion, since we did not want those three questions to affect the way subjects interacted with the system.

⁹The answers were scored by one of the authors, following written guidelines.

¹⁰We should also mention though that such interaction may be a statistical artifact, since there is only one true independent variable in our evaluation, the system.

¹¹When we collected our naturalistic data, students were not asked to answer the questionnaire.

¹²The feedback the various systems provide is not part of the information they log. This has the unfortunate consequence that we cannot run correlations between subject scores and these three features, since without the log of the actual feedback we cannot reconstruct them.

¹³Some researchers have a different take on directness, and are concerned with politeness effects. (Wang et al., 2008) finds that students learn more when feedback is less direct and more polite, but (McLaren, Lim, Yaron, & Koedinger, 2007) finds only a non-significant trend in the same direction. However, we do not believe that directness should necessarily be equated with curtness. Additionally, it is very hard to compare these three studies – (Wang et al., 2008; McLaren et al., 2007) and ours – since there are many other differences among their settings, tasks, and approaches.

¹⁴Thanks to Massimo Poesio, p.c., for this characterization.