Model-based Surprise and Explanation : A Way to Negotiate Concepts

jean louis Dessalles
Télécom Paris - Département Informatique
46 rue Barrault - 75634 PARIS Cedex 13 - France
Tel.: (33 1) 45 81 75 29 - Fax: (33 1) 45 81 31 19
E-✉: dessalles@enst.fr

abstract

We present here an analysis of a specific form of explanation that can be found in naturally occurring conversations, and that may be needed by users of KBS: explanations as answers to surprises that follow a discrepancy between expectations and reality. We describe a tutoring system based on this type of explanation: SAVANT3 systematically looks for reasons to be surprised, so that the student feels compelled to give explanations. We examine the requirements that a system has to meet to be able to produce this kind of explanation based on a preliminary surprise.

keywords: explanation ; conversation; KBS; CAL

Logical Aspects of Surprise

Previous studies on the logical aspects of argumentation in natural conversations (see [dessalles 1985, 1990]) showed that people quite often introduce a new topic by uttering surprise. Consider for instance the following excerpt:

> [ex_canteen]
> context: A, B and C are choosing a table at their staff canteen. A is surprised because there are very few people, when it's normally quite difficult to find a table free. B gives an explanation: most of the students went to the so-called "forum" where companies present job opportunities.
> *A: What happens today? There 's nobody here. It's Wednesday...*
> *B: Today is the forum day.*

This kind of surprise, where one realizes that the situation does not match one's expectations and where *reasons* can be given to justify such expectations, appears as essentially logical. In this excerpt, A had reasons to think that students should have been more numerous: it was wednesday, and they had to attend classes. This expectation is in contradiction with the small number of students. A's surprise thus comes from an expectation.

It is possible to capture such expectations with a simple logical representation (such a logical representation can be made more precise at will, but it is not necessary here):

normal_workday ⟹ **not** students_are_absent

Some expectations are the result of logical reasonings, as was shown for instance by Inhelder & Piaget [1979] when they insisted on the difference between procedures and structures. In one experiment with children, they put pearls one after the other into two containers alternatively. The two containers were shown to contain initially a different number of pearls, and one of them was then hidden. Smaller children considered as possible that they contain the same number of pearls after some time, when older children considered such an event as *impossible*. One of them declared: "Oh yes: as soon as you know, you know for ever!" (*"Ah oui: une fois qu'on sait, on sait pour toujours"*). The authors concluded:

> structures show themselves through inferences made by the child while procedures involve much more empiricism; and the structural nature of these inferences is best revealed by suppressions of contradictions and of incompatibilities.

So people have structures that allow them to draw inferences and thus to have expectations about the world (as was also recently emphasized by Ohlsson [1991]). For our purpose, we can consider these structure as logical models. Of course some expectations are not the result of logical inferences. They may follow for instance statistical measurements. But *model-based surprise* (caused by logical expectations) seems much more likely to be followed by an explanation, and thus we chose to focus on it.

Explanation as a solution to a model-based surprise

Explanations that occur after a model-based surprise are interesting for at least two reasons: first they occur naturally, as we saw (this point is also illustrated in [Heritage 1990]), and thus we may hope that their use could improve the acceptability of human/machine interactions under certain circumstances. But the problem is to reproduce these explanations on artificial systems. This is the second point. As we will see now, this type of explanation is heavily constrained, so that, in certain situations, one can write programs that are able to recognize and to synthesize such explanations.

We expressed the surprise contained in [ex_canteen] with a logical representation that can be rewritten as:

[normal_workday **&** students_are_absent] ⟹ **F**

F stands here for an ever false proposition. Thus [*a* **&** *b*] ⟹ **F** means that *a* and *b* are logically incompatible. The explanation given by B aims at denying *normal_workday*: *if* the forum takes place today, *then* today is not a normal day (classes have been cancelled). Any model-based surprise can be written this way as a logical incompatibility, and thus we are exactly in the situation mentioned by Inhelder and Piaget, where subjects have to "suppress contradictions or incompatibilities". M. Baker [1991] describes also "internal conflicts" as leading to explanatory dialogues, and shows situations in which inconsistencies are related to dialogic cooperation at the sociological level. But our suggestion of using surprise-based explanations in explanatory systems comes more simply from the observation that interlocutors in conversation do utter their internal logical conflicts spontaneously, and that other interlocutors do their utmost to find relevant explanations.

The situation of logical conflict is interesting because it is heavily constrained: only few explanations are admissible (even if not necessarily accepted) as solutions to an incompatibility. Let us take first the simple case of an explanation working as a *direct invalidation*. If we express the logical incompatibility this way:

$$[\ p_1\ \mathbf{\&}\ p_2\ \mathbf{\&}\ \ldots\ \mathbf{\&}\ p_n\ ]\ \Longrightarrow\ \mathbf{F}$$

then a direct invalidation consists in denying one of the terms p_i considered as belonging to the contradiction by the person uttering surprise. So any explanation which denies one p_i or which proves that p_i must be false is thus admissible. This was the case in [ex_canteen].

Another possibility for explaining a logically surprising situation is illustrated by the following example:

[ex_toy]
context: E is surprised by the fact that her great child G (two years old) is playing a lot with a broken toy. The mother, F, gives an explanation.
E1- One could think they leave the toys when they are broken. Listen: G played with a car which had no wheels left. I'm not saying he liked it better, but he played with it at least as much as with the others.
F1- In fact it's because he is imagining he is a mechanic, and he is going to repair it.

We can represent E's surprise logically:

$$[\ \text{plays_with}(\ G,\ \mathbf{Toy})\ \mathbf{\&\ not}\ \text{functional}(\mathbf{Toy})\ ]\ \Longrightarrow\ \mathbf{F}$$

where ***Toy*** is instantiated on the car with no wheels left. F's explanation can be understood as an *indirect invalidation*, i.e. an invalidation of another clause including further premises:

$$[\ \text{plays_with}(\ G,\ \mathbf{Toy})\ \mathbf{\&\ not}\ \text{functional}(\mathbf{Toy})\ \mathbf{\&\ not}\ \text{playing_at_repairing}(\ \mathbf{Toy})\ ]\ \Longrightarrow\ \mathbf{F}$$

F's explanation could be paraphrased this way: "if the child did not play at repairing the toy, then it would be indeed surprising that he plays with it. But this is not the case."

This kind of explanation through indirect invalidation is admissible as long as the surprised speaker can accept it as denying a forgotten premise p_{n+1}. In other words, this speaker has to accept that

$$[\ p_1\ \mathbf{\&}\ p_2\ \mathbf{\&}\ \ldots\ \mathbf{\&}\ p_n\ \mathbf{\&}\ p_{n+1}\ ]\ \Longrightarrow\ \mathbf{F}$$

represents the actual incompatibility. The explanation (**not** p_{n+1}) then appears as an invalidation of this augmented incompatibility. We should not wonder that some premises may be "forgotten" by the first speaker. After all, any incompatibility noticed in real life presupposes that the world still exists, that people are at a single location at any time, and so on. But requiring that a given fact p_{n+1} can be recognized as part of the initial incompatibility remains a very strong constraint on what can or cannot be considered as an admissible explanation. In the preceding excerpt, F could have denied hypotheses like:

- not (the child is insane)
- the child can tell the difference between functional and non-functional toys
- functional toys were equally accessible to the child
- not (the child is specifically attracted by this broken toy)

but not:

- the child likes chocolate
- this toy has a name beginning with C
- the climate in Oregon is mild

These constraints which limit the logical form of explanations that may follow a model-based surprise are strict enough to allow artificial systems to utter or to recognize such explanations. We will see now that a system like SAVANT3, which was designed to help students acquire new technical concepts, is able to *recognize* both direct and indirect invalidations. We will then make some suggestions upon the *synthesis* of relevant explanations as reactions to a surprise expressed by the user.

SAVANT3: uttering surprise to make the student explain

Our system SAVANT3 functions according to the surprise-based explanation principle. It tries to make an incompatibility explicit, and thus offers the student the opportunity to explain it, i.e. to invalidate it directly or indirectly.

[interaction with SAVANT3]
context: this dialog is proposed as an introduction to a digital communication course. Some concepts like digital symbols were presented, and then a printed page is displayed. The student may consider the page as a succession of characters (teletext-like coding) or as a succession of pixels (Fax-like coding). In any case, problems may arise.

S1- Can we consider a printed text as a digital message ?
E1- sure
S2- According to what I understood: this typed page can be considered as a digital message. We have to consider symbols allowing a complete description of the page
E2- characters
S3- yes... mmm... According to what I understood: the points (pixels) in the page are not taken as symbols. What else in the page could we code, if we except printed characters?
E3- you mean if there are drawings?
S4- But there may be a problem ... If we consider that one wants to code for figures (schemas, photos, signature)...
E4- but this is a printed page
S5- yes... According to what I understood: one chooses a TELETEXT-like coding. Printed characters are not the only thing to be coded...
E5- no
S6- Could you say more?...
E6- margins, paragraphs
S7- yes... mmm... But wait... According to what you said: one wants to code for positions (paragraphs, etc.). I must say I'm confused !
E7- you use special codes
S8- well, well!... Okay...

The functioning of SAVANT3 is quite simple, and reminds of the entrapment strategy used in WHY ([Collins 1976]). The knowledge given to SAVANT3 on a specific subject consists of a set of incompatibilities:

$$[\, p_{11} \,\&\, p_{12} \,\&\, \ldots \,\&\, p_{1n} \,] \implies \mathbf{F}$$
$$[\, p_{21} \,\&\, p_{22} \,\&\, \ldots \,\&\, p_{2m} \,] \implies \mathbf{F}$$
$$\ldots$$

This knowledge is thus structured as if it contained only consistency production rules (as those of the COVADIS system described in [Ayel 1990]). Some of these clauses may become invalid at a given point of the conversation with the student (when at least one term is false). SAVANT3 looks for a valid clause (e.g. clause *j* if all p_{ji} are known to be true or are still unknown), it utters known terms and pretends to be surprised (as in S7). The student then has the possibility to deny a mentioned term (direct invalidation, as in E4) or an non-mentioned one (indirect invalidation, as in E7).

The functioning of SAVANT3 is illustrated below (cf. [dessalles 1991] for greater detail). At each point during the interaction, the system makes utterances or feigns surprise in order to verify the most promising valid clause (i.e. the clause with highest proportion of known terms). If enough terms in this clause are known, then the program utters surprise. Otherwise it utters a "canned" sentence associated with the next unknown term in the clause (as in S5). This clause may change several times during interaction. Notice that the possibility for SAVANT3 to recognize indirect invalidations comes from the fact that some premises are intentionally "forgotten" when surprised is uttered, as was the case in S7.

S4-	*But there may be a problem ... If we consider that*	*standard text*	
	one wants to code for figures (schemas, photos, signature)...	*wording of a predicate*	contradiction: figures without pixels
E4-	but this is a **printed** page	*keyword detection*	direct
S5-	*yes... According to what I understood:*	*standard text*	invalidation
	one chooses a TELETEXT-like coding.	*wording of a predicate*	
	Printed characters are not the only thing to be coded...	*"canned" sentence associated with a predicate*	contradiction: teletext & positions
E5-	**no**	*keyword detection*	without codes
S6-	*Could you say more?...*	*standard text*	
E6-	**margins**, **paragraphs**	*keyword detection*	

SAVANT3 is able to utter surprise and to recognize relevant explanations given by the student as invalidations, but it is unable to recognize surprise in the student's utterances. Our program PARADISE (see [dessalles 1990]), designed to reconstruct conversations, is able to recognize that some utterances are intended to express a logical incompatibility (as in [ex_canteen]), and is able to suggest explanations as solutions to these incompatibilities. To achieve this, PARADISE considers predicates in the interlocutor's utterance* and looks for a clause in its knowledge that contains most of these predicates. This clause, if found, expresses an incompatibility, and may be considered as the intended meaning of the interlocutor's utterance.

---

* PARADISE has no syntactic capabilities and must be given utterances in a succession of *subject / verb / complement* inputs.

Some extensions and limits of surprise-based explanations

The kind of explanations we are dealing with in this paper may be proposed in any situation in which the functioning of the system does not match the user's expectations. This includes some interactions with knowledge-base systems, for end-users, but also experts during the elicitation and maintenance phases. This concerns also help and advisory systems, as far as the system is able to detect unsatisfied expectations in the user's request.

In any case, implementing surprise-based explanation capabilities requires that the systems has a very good representation of the user's knowledge. For instance, if we want a system to detect surprise, as PARADISE does, in a user's utterance and then to reply by giving an explanation, using a knowledge structured as a set of incompatibilities, then the system has to select a clause which contains terms of the user's request, say r_1 and r_2, and terms that were actualized in the present situation: s_1, s_2.

$$[\, r_1 \,\&\, r_2 \,\&\, s_1 \,\&\, s_2 \,\&\, o_1 \,] \implies \mathbf{F}$$

If such a clause exists, then a good guess would be that

$$[\, r_1 \,\&\, r_2 \,\&\, s_1 \,\&\, s_2 \,] \implies \mathbf{F}$$

is an accurate representation of what the users believes and of his/her surprise: for him/her, r_1 and r_2 cannot be simultaneously true, if we know that s_1 **&** s_2. The system will then try to explain the surprise by invalidating the clause. If one of the terms in the system's clause can be proven false, then the system is able to utter an admissible explanation. For example "*but o_1 is false*" or "*its because not o_1*" would be perceived as *relevant* explanations by the user. The system may also suggest these explanations when a term in the clause is unknown to him or has been learned directly from the user: "*but perhaps not o_1*". When all terms in the clause can be proven true, then the system can find another clause used to prove one of these terms, and recursively try to invalidate this new clause.

However such results can only be obtained under very specific conditions:

- user's requests are provoked by deceived model-based expectations
- user's requests contain such expected elements
- user and system manipulate the same concepts
- the system contains a "complete" set of clauses linking these concepts: consistency rules and strategic rules (as was shown by Clancey [1987]) have to be included
- the system constains a knowledge that is not necessarily used during inferences, but that is necessary to justify inference rules (as was shown in XPLAIN by Swartout [1983])
- the system is able to isolate a subset of clauses which accurately represents the user's expertise

A possible consequence of this is that an explanatory module that would include surprise-based explanation capabilities should be autonomous, as emphasized by B. Safar [1992]. But a transposition of the mechanisms outlined above onto KBS explanatory modules raises many problems. One of them is that the backward chaining underlying this mechanism will not necessarily match the trace of the KBS inferences. A possible solution would be that

the explanation module avoids using inference rules that were not actually present in the trace. But many aspects of these transposition problems are still to be investigated.

Conclusion: a way to negotiate conceptual knowledge

Logical relevance, as it can be described in spontaneous explanations produced during natural conversations, seems to be a desirable characteristic of explanations that may be given by artificial systems. Model-based surprise, when it can be recognized with a good probability, either by the user (as is needed in SAVANT3) or by the system (e.g. by a help system), can lead to logically relevant explanations. Alternating surprises and explanations should be an interesting way through which KBS could negotiate conceptual knowledge (which corresponds to the structures mentioned above, as opposed to procedural knowledge), even during task-oriented interactions. SAVANT3 relies on such a negotiation.

Every KBS user has expectations, and (s)he needs a conceptual explanation when the situation does not match them. We tried here to indicate a possible way to give logically relevant explanations by recognizing and invalidating user's expectations.

References

Ayel Marc, Rousset M-Christine (1990): *La cohérence dans les bases de connaissances*, Editions CEPADUES, Toulouse 1990

Baker Michael (1991b): *An Analysis of Cooperation and Conflict in students' clooaborative explanations for Phenomena in Mechanics*, in Tiberghien A., Mandl H.: *Knowledge Acquisition in Physics & Learning Environments*, Springer Verlag, 1991

Clancey William J. (1987): *Methodology for Building an Intelligent Tutoring System*, dans Kearsley Greg P.: Artificial Intelligence & Instruction - Applications and Methods, Addison-Wesley Publishing Company, Menlo Park, USA 1987, 193-228

Collins Allan (1976): *Processes in Acquiring Knowledge*, in Schooling and Acquisition of Knowledge, Anderson, Spiro a, Hillsdale NJ 1976

Dessalles Jean-Louis (1985): *Natural strategies of concept acquisition*, proceedings of COGNITIVA 85, CESTA, Paris 1985, 713-719

Dessalles Jean-Louis (1990): *The simulation of conversations*, dans Kohonen Teuvo, Fogelman-Soulié Françoise: COGNITIVA 90 - Proceedings of the Third Cognitiva Syposium (Madrid), North Holland, Amsterdam 1991, 483-492

Dessalles Jean-Louis (1991): Conversation Assisted Learning: The SAVANT3 Dialog Module, dans Forte Eddy N.: Proceedings of Calisce'91, Presses Polytechniques et Universitaires Romandes, Lausanne 1991, 159-165

Heritage John (1987): *Interactional Accountability: a Conversation Analytic Perspective*, dans Conein Bernard, De Fornel Michel, Quéré L. : Les formes de la conversation, CNET, Paris 1990, 23-49 (T1)

Ohlsson Stellan (1991): *Interview, by J.Sandberg and Y.Barbard*, AICOM vol 4, n° 4, 1991, 137-144

Inhelder Bärbel, Piaget Jean (1979): *Procédures et Structures*, Archives de psychologie, XLVII, 181, 165-176

Safar Brigitte, Berthault Pascale, Sylvestre J. (1992): *Place des explications dans la conception d'une interface intelligente entre une base de données et un usager*, 12èmes Jounées Internationales Avignon'92, Avignon 1992

Swartout William R. (1983): *XPLAIN: a System for Creating and Explaining Expert Consulting Programs*, Artificial Intelligence 21, 1983, 285-325