The "Sketchboard": A Dynamic Interpretative Memory and its Use for Spoken Language Understanding

Gérard SABAH
Language and Cognition Group
LIMSI - CNRS
B.P. 133, 91403 ORSAY Cedex, FRANCE
Tel: 33 1 69 85 80 03, Fax: 33 1 69 85 80 88, E-mail: gs@limsi.fr

ABSTRACT

Blackboards allow various knowledge sources to be triggered in an opportunistic way, but does not allow higher modules to feedback information to lower level modules. The solution presented here remedies this shortcoming, since our *Sketchboard* implements reactive feedback loops.
Within the Sketchboard, modules are considered from two points of view: either they build a result (a *sketch*, possibly rough and vague) or they give back a response to the modules from which they received their input data. This response signals the degree of confidence the module has towards its *own* result. These relations are generalized across all the modules that interact when solving a problem. As higher and higher level modules are triggered, the initial sketch become more and more precise, taking into account the higher modules knowledge. Conceived for natural language processing, the Sketchboard is also useful for spoken language understanding as shown by a detailed example.

1. INTRODUCTION

Blackboards [1] allow various knowledge sources to interact in a very opportunistic way, the solutions being conceptually developed in a parallel way (sophisticated control allows for handling this sequentially) [2, 3]. However, even if this model has allowed for significant progress concerning the understanding of interactions between modules, it does not allow higher modules to feedback information to lower level modules (even with the sophisticated GUARDIAN [4]). We present here the *Sketchboard* that contains a mechanism for reactive feedback loops, generalized across all the modules that interact when solving a problem.
We will not insist here on the neurobiological relevance of this kind of mechanism. This is particularly obvious for the vision process, where the existence of top-down paths (injecting *new information* into the initial message) has been demonstrated at the neurobiological level [5]. Furthermore, [6] has convincingly argued that these auto-referential loops, analoguous to Harth's *creative loops* [7], govern the whole nervous system.
In this paper, we show that our *Sketchboard* leads to a new kind of connection between modules, which allows modules to behave in a similar way as human beings.

The architecture described in [8, 9] is designed to integrate the representations and expertise needed to perform various natural language processing tasks. Expertise is embedded in a multi-layer assemblage of processes with increasing complexity and scope: elementary processes are dedicated to the execution of simple cognitive tasks, compound processes, dealing with a cluster of subprocesses, carry out more elaborate tasks; a hierarchy of supervisors harmonize the whole to fulfill a given global task. This is a multi-agent system with a continuous control and a dynamic management, based on blackboards mixed with message passing (dynamic adaptation of the first plan calculated by the controller). This architecture allows for dynamic management of interactions between the agents involved in solving a given problem. The processes used to solve a problem, and the order in which they are applied, are dynamically computed, depending on the global task (understanding a text, abstracting it or managing a dialogue) as well as on the specific current context. This has proven to be useful, but requires a heavy explicit control. We propose here an implicit control allowing for such a behavior.

2. THE SKETCHBOARD

2.1 The model

Our Sketchboard is an extended blackboard: in addition to being a general input-output zone, responsible for triggering modules and managing their communication, it sets up specific relations between a specialized process and the processes that use its results. These relations allow the latter to feedback information to the former, so that it adapts its behavior automatically. Within the Sketchboard modules are considered from two points of view: (a) either they build a given kind of result (a *sketch*, possibly rough and vague) or (b) they return a response that indicates the degree of confidence concerning their results . These relations are generalized across all the modules that interact when solving a problem, which allows the Sketchboard to set up original relations between the active modules. While higher and higher-level processes are triggered, the first sketch will become more and more precise, since these processes give feedback to the first module, indicating the relevance of what is computed with regard to their own knowledge. At the end of the process, the system's entire knowledge is taken into account. This kind of confidence-driven reactive behavior deviates from classical architectures.

In Figure 1, the two modules are in a loop: with R(E), B indicates to A how pleased it is with what it has done with E, the result of A. Then, the first one (A) modifies its result in order to optimize the answer from the second one (B). Put differently, a first sketch E is computed by A, and further refined, such that R(E) — the response from B — be optimal according to a given criterion. Since A does not have the necessary knowledge to interpret a sophisticated message from B (*by definition of modularity*) its behavior is purely reactive: if its previous modification produces a better response from B, it continues modifying its sketch in the same way, otherwise it performs modification the other way round.

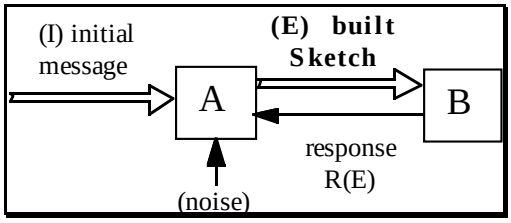

Figure 1. *Elementary connections between two modules*

This mechanism continues until some stability is reached: *this amounts to modify the signal until B's interpretation is as relevant as possible, given the context and the knowledge available.* (Context is represented by active modules and the sketches written onto the Sketchboard. Available knowledge is the knowledge stored in the long term memory and used by these modules. The measure of relevance depends on the module itself).
In its turn (and while the previous loop is still running), B too will receive some feedback from higher level modules. This will influence its own sketch and, as a result, the strength of its answer to A.
Progressively, the various sketches will *simultaneously* become more and more refined. As higher-level modules provide feedback, the relevance of the results produced by a module with regard to higher-level knowledge is established. This results in a non linear process: high level agents act not only as filter for perceptions, but as a device adding features, modifying thus the characteristics of the input (Figure 2).
This being so, the Sketchboard has to fulfill four main tasks: to detect what module is working on what data, to understand its role, to connect the modules that will give rise to elementary connections, and to handle their resulting exchanges. There is no real output from the whole system: the state of the Sketchboard is read by the set of modules that are working on it, and the *impression of understanding* results from some stability of the system.

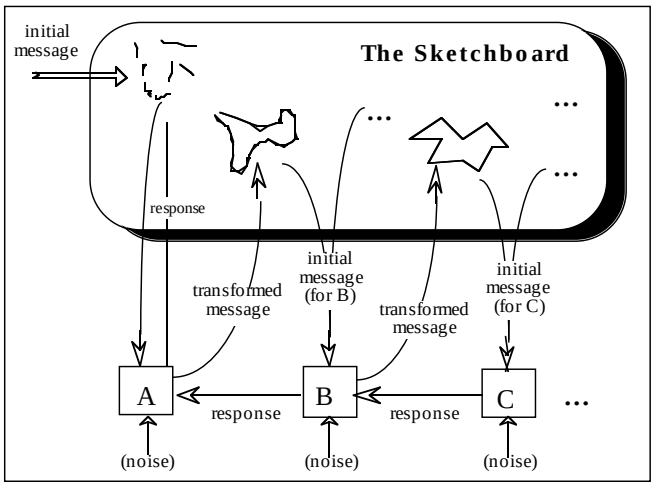

Figure 2. *Interactions between various level processes*

2.2 Multiple interactions and relations to memory

In a similar vein as blackboards, the Sketchboard is divided into several layers in order to simplify the control process: when data appear in a given level, only a limited number of modules may be triggered. Figure 2 shows an example where three modules interact at different levels.
But things may be more complex, for at least two reasons: at a given level, (a) previous data may stay for a variable length of time, and (b) several modules may act in parallel on the same data. To allow for such behavior, a given amount of memory is allocated to each level of the Sketchboard. This allows to maintain active data for a certain amount of time, and to store alternative results from the active modules.
We define thus the *short term memory* as the amount of memory linked to the first processing level: it represents the location where perceptive data are stored and eventually modified, provided that they are still there. The *working memory* is the set of memories linked to the other levels. In the same way, data that are still in such a memory may receive feedback from higher levels and may be compared with data of the same level. Those receiving the highest amount of feedback are most likely to lead to a correct solution.
This allows the Sketchboard to keep track of the various solutions developed in parallel and currently under development. This allows it to give preference to data receiving positive feedback from higher level, and lower the expectations for those receiving negative feedback (or more radically, space becoming short, to forget them).

2.3 Implementation

A prototype of the Sketchboard has been implemented using Smalltalk. When a module A writes data into the Sketchboard, this data is stored through a ValueHolder (thus, allowing that any modification of the value be passed on to all its users). A list of semaphores is attached to this ValueHolder. A Sketchboard manager is in charge of building the set of such units.
As soon as another module B is interested in such a value, the Sketchboard sends a message to the corresponding ValueHolder, which consequently adds a semaphore to its list of semaphores. Simultaneously, a communication channel between A and B is set up or updated (a dictionary that associates to A the strength of B's answer). Thus, as soon as A receives data concerning the strength of the answer of a module that uses its result it will recompute a new value of this result, in a way that aims to optimize the answer of the higher module.
When this step is performed, the semaphores are signalled, which triggers again the higher level processes which will transmit again their updated answer to A... When a kind of stability is detected, the loop is stopped.

3. EXAMPLE OF THE USE OF THE MODEL

We show here how the Sketchboard explains the fact that perceptions are context dependent and capable to supplement truncated messages. As an example, we will start from a psycholinguistic experiment that shows that what we really perceive are not phonemes but a set of features on the basis of which we build an interpretation. [10] describes an experiment where a phoneme is replaced by a noise (showed here as *). While hearing *it was found that the *eel was on the axle (*or *orange* or *table…)*, people are convinced to hear *wheel / peel / meal…* (respectively). The Sketchboard can explain this phenomenon and behave the same way.

In the context of speech understanding, let us consider four basic collaborating processes involved in the understanding of sentences: a phoneme recognizer, a word builder, a sentence parser and a semantic interpreter [11]. Let us show how they interact through the Sketchboard structure when the sentence to be processed is *THE *EEL WAS ON THE AXLE* (the * of WHEEL being the non-recognized phoneme).

The first module builds a hypothesis about the non-recognized phoneme. Without any other information, the recognition process may converge towards any solution, depending on the initial input and random variations.

But, this hypothesis is used by, and receives feedback from, the subsequent modules. All of them are related and looping while the relevant hypotheses are processed by higher levels and while further phonemes are taken into account. General feedback coming from the processing of *axle (orange / table…)* may arrive before the end of the processing of **eel*. Therefore it may influence perceptive processes and produce interpretations of the absent phoneme itself.

More precisely, let us consider that, in the figure 1 above, (a) the result of the module A being the amplitude (I_k) at the k_{th} frequency, (b) that B's input is a set M of such elements ($\{I_k / k = 1,n\}$) and (c) B's output is a probability vector concerning each phoneme: $\{x_i / i = 1,32\}$ (the sketch constructed by B).

An absolute measure for B's satisfaction concerning its result E is given by the entropy of this probability set [1]. This is the answer B returns to the lower-level processes (this answer originally depends upon M):

$$R(M) = \sum_{i=1}^{26} x_i * \log(x_i)$$

To take this feedback into account, every process will modify its result according to the following principle: if the last change in I_k has produced a good change in R(M), it will continue along this line and proportionally to its goodness. If the last change in R(M) was wrong it will go the other way round.

1. In fact, the entropy is $-\sum_i x_i * \log(x_i)$. It measures the dispersion of n elements associated with probabilities: it is minimum and equal to 0 when only one element is certain, and maximum and equal to log(n) when the n elements have the same probability. As we want a quantity to be maximum, we will take here the opposite of the entropy.

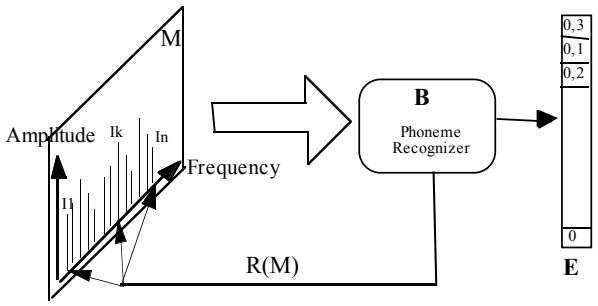

Figure 3. *A simple example of feedback*

The quantity [*lastChangeIn* I_k * *lastChangeIn R(M)*] has exactly this property.Therefore, the $p+1^{th}$ value of the k^{th} intensity depends on its values at the two previous iterations: the difference between the two previous values of B's answer multiplied by the difference between the two previous values of the intensity are added to the p^{th} value. The whole set is optimized by applying a formula inspired from the *Alopex* algorithm [7]:

$$\mathbf{I}_k^{p+1} = \mathbf{I}_k^p + [R(M^p) - R(M^{p-1})] * [\mathbf{I}_k^p - \mathbf{I}_k^{p-1}] + \text{RandomNb}$$

This amounts (abusively) to consider each amplitude independently from the others: at each step the variation may or may not be in the right direction (as the variables are, in fact, bound), but it is somehow more likely to be right than wrong. The random number stands for noise. It does not guarantee success, but it tends to disturb from ridges and minor peaks and increases the probability of reaching the optimal value for R(M).

As soon as the first phoneme of the word is available, the word builder begins to work and sends feedback to the still active letter recognizers. At the beginning of the process, the answer of the word builder will roughly be proportional to the probability to find a word beginning with this phoneme. These probabilities vary with the arrival of the interpretations of further phonemes (…i: l), hence the responses of the various active word recognizers will convey this information. Finally, the somehow vague recognition of the first phoneme produces a fuzzy recognition of the second word of the sentence (*wheel, peel* and *meal* being among the best candidates).

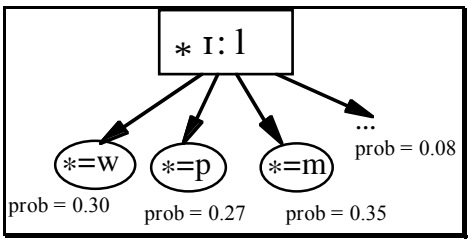

Figure 4. *Three contextual interpretations of the ambiguous phoneme at the word level*

In other words, the interaction of these first two levels through the Sketchboard will modify the behavior of the phoneme recognizer: it increases the response concerning the three phonemes w, p and m and decreases the other responses (this means, the fact that recognizing a w, a p or a m allows to find existing words confirms their adequacy as candidates).

While these processes are running, the syntactic parser has also begun to work. The feedback it gives to the word level increases the likelihood of the next word being a noun (according to the syntactic structure : NP (*determiner noun) VP (copule PP (prep NP)*). Since the three interpretations corresponds to nouns, this have no significant influence on the phoneme recognition.
These three results are written onto the Sketchboard, which then builds new interactions between syntactic and semantic processes. The sizes of the various working memories are supposed to be such that the interactions between the recognition of the phoneme ∗ and the other levels are still active. When the word *axle* is taken into account, three parallel ways of interpretation are entering a loop, corresponding respectively to *THE WHEEL / THE PEEL / THE MEAL WAS ON THE AXLE*. Feedback from the semantic level is significantly stronger for the first one than for the last two, and this will have cascaded backward influences on the previous level and increase the likelihood of the recognition of the phoneme ∗ as w. The evaluation of the result of the semantic interpreter is based on the satisfaction of the selectional restrictions (coming from the verb) and an evaluation of the distances between the concepts appearing in the sentence. The complexities of the founded paths in the semantic net are also taken into account.
This is shown by the different behaviours of the system when confronted to different sentences (*the *eel (peel) was on the orange* or *the *eal (meal) was on the table*).

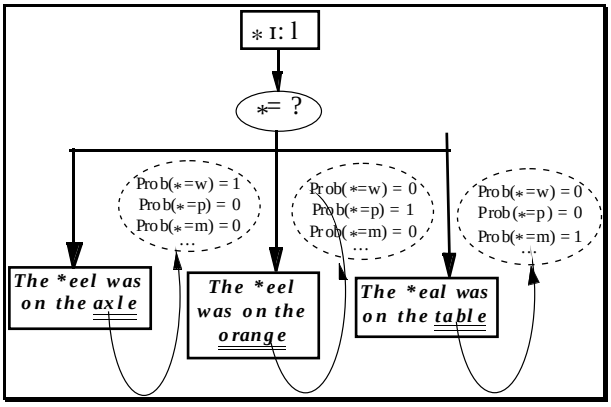

Figure 5. *Three different contextual interpretations*

Since *wheel* and *axle* are more closely related than *wheel* and *orange* or *table*, in the first sentence, the semantic interpreter will primarily have a preference for*wheel* rather than for *orange* or *table*. It will thus imply that the word recognizer prefers this word too. This new result will in its turn feedback to the loop taking place between the phoneme- and the word recognition (supposedly still active). In consequence, the recognition of the phoneme w will be increased. For the second sentence, the same process will imply a preference for a p, since peel and orange are more closely related than the other couples of words. Again the same process will make the system prefer a m in the third sentence.
These simple examples illustrate how the Sketchboard allows very naturally higher levels to influence lower levels (e.g., the processes of perception and pattern recognition).

4. CONCLUSION

We have proposed a new data structure, the *Sketchboard*, an extension of Blackboards, allowing different modules to collaborate while solving a problem. This model allows feedback from higher levels to lower levels of processing. The Sketchboard also explains the possibility of "inverting" a sensory process (e.g., how the thought of an object can produce a perception of that object). Indeed, the proposed algorithm provides a top-down control by which this process can take place: conceptual levels provide feedback to lower levels and the Sketchboard looping mechanism acts on these lower levels. The whole system will converge towards a state as if the object in the mind has really been perceived.
Our proposal, which we believe to have true psychological relevance, has been implemented in Smalltalk. The program runs on simple examples; its integration within the CARAMEL model for Natural Language understanding is currently under study. If the results turn out to be conclusive, the Sketchboard will replace the classical sequence of parsing and generating levels by research strategies and contextual use of information, depending on the situation and the current goal.
Though we have not been concerned here with robustness, it should be pointed out that the kind of processing resulting from the use of the Sketchboard is very robust and flexible. In that respect, it is similar to processing done in the connectionist framework but remaining in the symbolic area. Moreover, while neural nets explain well certain kinds of feedback (e.g., between phonetic and lexical levels), they have difficulties in explaining others (from syntax or semantics towards the lexicon for example), which is not the case for our model.

5. REFERENCES

[1] L.D. Erman, *et al.*, *The HERSAY-II speech understanding system : integrating knowledge to resolve uncertainty*. Computing surveys, 1980. 12(2): p. 213-253.
[2] B. Hayes-Roth, *A blackboard architecture for control*. Artificial intellligence, 1985. 26: p. 252-321.
[3] P. Nii, *Blackboard Systems: the Blackboard Model of Problem Solving and the evolution of Blackboard Architectures.* AI magazine, 1986. August: p. 82-106.
[4] B. Hayes-Roth, *et al.*, *Guardian: A prototype intelligent agent for intensive-care monitoring*. Artificial intellligence in Medicine, 1992. 4: p. 165-185.
[5] S. Ramón y Cajal, *Neuronismo o reticularismo? La prueba objectivas de la unitad anatomica de la celulad nervioses.* Arch. neurobiologicas, 1933. 13: p. 217-291.
[6] R. Restak, *The Brain: the last frontier*. 1979, New York: Doubleday, Garden City.
[7] E. Harth, *The creative loop; how the brain makes a mind*. 1993, New York: Addison-Wesley.
[8] G. Sabah. CARAMEL : A computational model of natural language understanding using a parallel implementation. *ECAI*. 1990. Stockholm.
[9] G. Sabah and X. Briffault. Caramel : a Step towards Reflexion in Natural Language Understanding systems. *IEEE International Conference on Tools with Artificial Intelligence*. 1993. Boston.
[10] R. A. Warren and C. Obusek, *Speech perception and phonemic restoration.* Perception and psychophysics, 1971. 9: p. 358-362
[11] J. Vapillon, X. Briffault, K. Chibout and G. Sabah An Object-Oriented Linguistic Engineering Environment using LFG and Conceptual Graphs. "ENVGRAM" ACL'97 Workshop. 1997. Madrid.