A Cognitive-Based Approach to Adaptive Intelligent Multiagent Applications

Charles Hannon

Department of Computer Science
Texas Christian University
Fort Worth, TX
c.hannon@tcu.edu

Abstract. An integrated cognitive-based model (LEAP) and application (SALT) are presented. Building on a new Interlaced Micro-Patterns (IMP) theory and the Alchemy/Goal Mind environment, the LEAP research improves agent-to-human and agent-to-agent communication by incorporating aspects of human language development. The IMP theory further provides a theoretical basis for deep incorporation and sharing of knowledge from different sensor modalities. Research on LEAP points to a better understanding of human language development and the application of this knowledge within intelligent multiagent applications Research with SALT points to how this research supports Smart Home applications and provides feedback to LEAP modeling.

1 Introduction

Many intelligent multiagent applications can be improved by an adaptive agent organization that can not only re-task existing agents, but also add new agent capabilities to deal with changing requirements. While this level of agent adaptability presents a complex problem in design and construction, humans present an archetype for such abilities. In this article we will show how a study of one complex human skill (reading) can be used to drive adaptive multiagent design and how the information used from this study can be used in a Smart Home multiagent application.

To study language use and learning within a reading task, a robust distributed cognitive model called LEAP (Language Extraction from Arbitrary Prose) and a new working theory of cognition called IMP (Interlaced Micro-Patterns) are used. One focus of the LEAP/IMP research is to study how lexical, syntactic, semantic and conceptual information can be learned from a set of English language web-based sources. However, LEAP can also explain how language development occurs within the context of general cognitive development using all sensory modalities. By focusing on both ability and performance within this broader context, LEAP can provide insight into more general use and learning of cognitive skills that can be directly integrated into intelligent multiagent applications that serve to test the current working theories (e.g., IMP) of the models themselves.

© A. Gelbukh, R. Monroy. (Eds.)
Advances in Artificial Intelligence Theory
Research on Computing Science 16, 2005, pp. 239-248

LEAP is developed using the components of the Gold Seekers project depicted in Figure 1. Building on existing non-computational models and other research, the Gold Seekers project attempts to develop working theories (like IMP) that can be used to build modular computational models using Alchemy/Goal Mind [7]. These modules (or Agent Components) can be reused in other agent models to test other aspects of cognition or as the starting point of cognitive-based applications. The Smart-environment Adaptive-agent Language and Tasking (SALT) application builds on the our model research to explore how a dynamically constructed and tasked multiagent model can be used to allow a smart environment to better adapt to its users. SALT forms an integral part of the overall research by providing feedback on how the models handle a 'real world' application.

2 Related Work

Numerous ongoing research projects have applied machine learning techniques directly to the way a smart environment learns user preferences. The SALT application research is focused on adaptation through the way agents communicate and share tasks. For this reason we will focus our related work discussion on the LEAP model.

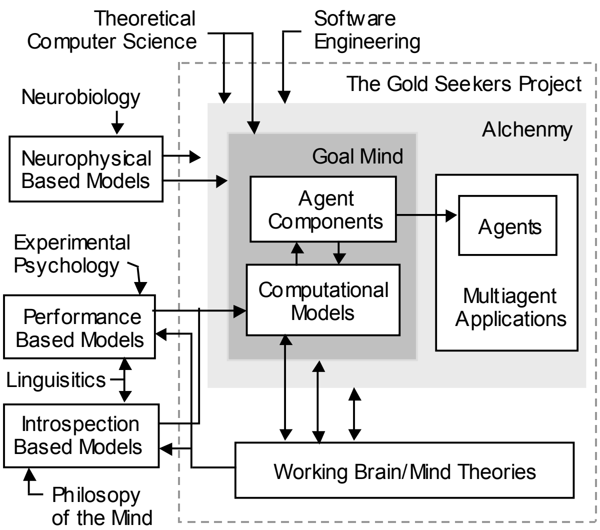

Fig. 1. Computational models are used to produce the agent components making up agents within a multiagent application. Components' design, construction, testing and operation are supported by Goal Mind. The distribution, migration and control of component processes and the resulting agent multi-processes across multiple processors are supported by Alchemy.

The two language models that LEAP directly build upon are the TALLUS [5] and STRESS [6]. We will briefly contrast the current research with some other cognitive modeling environments and two language-related knowledge bases, before addressing the theory of spreading activation which is a key element in the LEAP model.

2.1 Related Language Modeling and Capture Efforts

A number of on-going research efforts are addressing the cognitive modeling of language at some level. Many of these models address language within the context of other sensor modalities and are aimed at directly supporting an agent-based application. LEAP attempts to; 1) be explanatory, 2) be closely tied to well know cognitive mechanisms such as priming, spreading activation and memory consolidation, and 3) directly support use of its components within multiagent applications. This makes it similar to models built with SOAR [8], ACT-R [1] and ACT-R/PM [3]. The major difference between Alchemy/Goal Mind and these other architectures is that the Alchemy/Goal Mind models are created out of a set of concurrent components which are free to use their own cognitive sub-theories with the main cognitive theory controlling the method in which these components interact. This can be contrasted with the other environments where models are monolithic processes controlled some underlying cognitive mechanism such as ACT-R's symbolic productions and subsymbolic activations.

Some multiagent efforts rely on existing language knowledge bases. Compared to projects like WordNet and Cyc, that attempt to capture language and concept knowledge in large publicly available databases, LEAP currently has an extremely small database of language knowledge. For example, WordNet contains 144,309 unique words organized into synonym sets representing underlying lexical concepts [4]. The Cyc knowledge base contains almost 2 million assertions (rule and fact), 118,000 concepts and a lexicon of 17,000 English root words [11]. Both WordNet and Cyc have been very instrumental in our discovery of the underlying structure of the way language and concept reasoning works, but this does not directly translate to making them useful candidates for knowledge components within an adaptive multiagent application. In contrast to simply using a vast store of language knowledge, LEAP is attempting to capture the way humans learn by the slow consolidation of knowledge into a complex and multifaceted representation of their surrounding world and to use the resulting structure of that representation to discover how we can simulate human development within adaptive agents.

2.2 Spreading Activation

Memory priming via a spreading activation mechanism is a very old concept going all the way back to a direct extension of the Quillian work on Semantic Memory in the 1960's [10]. The original theory, proposed by Collins and Lofus in 1975 proposes that is-a, reverse is-a (what TALLUS and LEAP calls a could-be relation), has-a and part-of semantic relations will be followed to activate parent, children, and other associated nodes within a person's semantic network making it easier to retrieve these concepts immediately after the original concept retrieval [9]. Several psychology experiments in the 1970's and later demonstrated that the priming effects proposed by the spreading activation mechanism were observable [2].

While symbolic AI systems have focused mostly on the priming aspects of spreading activation, a number of connectionist systems have also explored the effect of lateral inhibition where the activation of a concept can cause the retrieval of related concepts to be blocked for a period of time after that activation. Psychology experiments appear to show that lateral inhibition works with spreading activation to allow us to more quickly determine that some statements are counter-to-fact [10].

The most mature cognitive modeling treatment of spreading activation to date is seen in the ACT-R architecture which uses production rules, not a classic semantic network, in its primary knowledge representation. Other semantic network based systems that use it tend to do something similar to ACT-R by calculating the amount of activation for a node based on its distance from the activated node, and then using the resulting number to artificially control the lookup between nodes during inference. As will be shown later, we will take a radically different approach to simulation spreading activation in the LEAP model.

3 Interlaced Micro-Patterns (IMP) Theory

Pattern matching as an important mechanism in the learning, retrieval and recall of simple concepts and procedures have been accepted in both machine learning and cognitive psychology research for some time. The Interlaced Micro-Patterns (IMP) cognitive theory extends the traditional pattern matching mechanism by proposing that if a set of simple patterns are interlaced (i.e., allowed to overlap), the mechanism can be used to learn, retrieve and recall elements of far greater complexity, and thus, could be the driving mechanism of such tasks as language use and learning. The support for IMP as a working theory comes from both a set of thought problems and the results of cognitive modeling work.

The first language model using what would become Alchemy/Gold Mind was TALLUS which was designed to study telegraphic speech (the second true language development phase in humans) within a visual context. Like most language models, TALLUS used a standard generative linguistic theory that proposed that utterances are generated by phrase structure rules that result in the utterance being associated as the leaf nodes of a hierarchical tree structure starting from a root node utterance or sentence. Given a set of generative rules, TALLUS could easily learn new surface forms and their associated concepts, but no believable explanatory mechanism for learning new syntax and their associated conceptual grids could be found.

This model failure resulted in the first thought problem. Why do children find it much easier to learn a natural language than the proposed grammar rules that are suggested to define such a language? Hierarchical syntactic approaches to natural language (NL) align well with the way NL grammars are taught in traditional educational settings, but not with how language development naturally occurs. The teaching of prescriptive grammars may help to stabilize language use across a group of language users, but it seldom controls the complete use of either spoken or written language 'rules' in that group with much of that use being driven by either a conscious or unconscious violation of the prescriptive rules. Many non-generative linguistic theories use this same argument to dismiss generative approaches, but these theories seldom provide a mechanism that could be used in a computational model of language.

So, is there a way to capture the computational strength of generative grammar without it being driven by a hierarchical set of rules? One possible method to do this is to use interlaced micro-patterns. While all possible well-formed utterances conform to some syntactic, semantic and conceptual pattern, the storage of every possible utterance pattern would clearly be too computationally complex to be feasible. However, if all possible sentence patterns were made up of smaller patterns that relied on overlapping elements to ensure correctness, a set of smaller patterns could not only generate correct utterances, but also block the creation of malformed utterances.

To test this approach, the LEAP model was constructed, which has confirmed the viability of the IMP theory for language learning. Further, it has introduced two new questions. Could the IMP theory supply an underlying mechanism for all cognition? And, could differences in the potential size of micro-patterns and their ability to interlace be an underlying control in the level of cognitive abilities exhibited by a biological organism?

3.1 IMPs Relationship to Symbolic AI

It is fairly simple to see how the IMP theory would map to a connectionist approach since the patterns can simply be distributed among the weights of connections; however, Alchemy/Gold Mind is basically a symbolic AI approach so we need to address the symbolic mapping a little further. Due to the large amount of existing research with different Knowledge Representation and Reasoning (KRR) methods, what we do not want is a theory that limits the types of symbolic reasoning possible within an application. Luckily, it can be shown that using the IMP theory as an overall control mechanism does not require such a limitation.

In summary, we can define a system's composite KRR as a set of layered component KRRs with each component's KRR being any desired type. This composite KRR can be stored in Long Term Memory (LTM) and access points within each layer can be activated into Short Term Memory (STM) by a pattern input from an external source (either another layer within the agent or an interface to the external world). In addition to the actual access points activated, other parts of the layer's KR can be activated by a temporal-based spreading activation mechanism when needed and deactivated by removal from the STM when the knowledge is 'timed-out'. Changes to the KRR can occur by updating the KR stored in LTM as a result of changes that occur in STM during activation.

A simple formalization of the effect of using IMP to control a layered KRR can be given if we simplify the KR of an agent to a uniform set of semantic networks. Each of these semantic networks can be viewed as a directed multi-graph,

$$R_n = \text{pair}\,(N_n , A_n),\ A_n = \{(\nu_{ni} , \nu_{nj}) \mid \nu_{ni} , \nu_{nj} \in N_n\} \quad \textbf{(1)}$$

where, n is the level of representation, N_n is a set of nodes, and A_n is a bag of named relationships between these nodes. A sub-representation of this network can be defined as,

$$R'_n = \text{pair } (N'_n, A'_n), N'_n \subseteq N_n, \text{ and} \quad \textbf{(2)}$$
$$A'_n \subseteq A_n \wedge ((\nu_{ni}, \nu_{nj}) \in A'_n \rightarrow \nu_{ni}, \nu_{nj} \in N'_n).$$

All possible sub-representations at a level n is, of course, the power set of R_n; however, this set has little meaning in the IMP theory since only the activated subrepresentations are of interest. Given all possible activated sub-representations at a level n, defined as,

$$\Phi_n = \{R'_n \mid R'_n \subset R_n \wedge \text{active}(R'_n) \rightarrow \text{True}\}, \quad \textbf{(3)}$$

connections between representation levels can also be viewed as a directed multi-graph,

$$K_{i,j} = \text{pair } (\Phi_{i,j}, \Gamma_{i,j}), \Phi_{i,j} = R'_i \cup R'_j, \text{ and} \quad \textbf{(4)}$$
$$\Gamma_{i,j} = \{(R'_i, R'_j) \mid R'_i, R'_j \in \Phi_{i,j}\},$$

where, i and j are levels of representation being connected and $\Gamma_{i,j}$ is a set of named relationships between these levels.

The number of representation levels (R_n) and number of level connections ($K_{i,j}$) can vary based on application. A traditional agent-based method for using the overall representation structure would be a set of m stacks of representation levels 1 to k with the top-level (level 1) of each stack being an interface representation and the kth level of each stack being either a common conceptual structure or a set of connected conceptual structures.

Given a set of available general inference rules at each level (ρ_n) and between two levels ($\rho_{i,j}$), the extent of general inference at each level (ι_n) and across levels ($\iota_{i,j}$) can be naively described as,

$$\iota_n \approx |\rho_n| \text{ and } \iota_{i,j} \approx |\rho_{i,j}|, \quad \textbf{(5)}$$

assuming no serious difference exist in the number of pre and post conditions of each rule. The total extent of representation at each level also can be naively described as,

$$\gamma_n \approx |N_n| \bullet \max \{\nu_{ni}, \nu_{nj}\} d(\nu_{ni}, \nu_{nj}), \quad \textbf{(6)}$$

which given the amount of accessible (or activated) knowledge at each level being $\beta_n = \cup\Phi_n$, leads to an activated representation extent of,

$$\alpha_n \approx |\beta_n| \bullet \max \{\nu_{ni}, \nu_{nj}\} d(\nu_{ni}, \nu_{nj}) \mid \nu_{ni}, \nu_{nj} \in \beta_n \text{ and} \quad \textbf{(7)}$$
$$\alpha_{i,j} \approx |\Phi_{i,j}| \bullet \max \{R'_i, R'_j\} d(R'_i, R'_j).$$

The activation potential at any level can be described as,

$$\eta_n \approx \Sigma_{\{i = 1 \text{ to } k\}}\ \alpha_{i,j} \bullet \iota_{i,j}, \quad \textbf{(8)}$$

and its inference potential as,

$$\kappa_n \approx \alpha_n \bullet \iota_n . \quad (9)$$

Assuming that we only allow a pattern matching activation mechanism to work between levels, the extent of cross-layer general inference ($\iota_{i,j}$) can be viewed as approaching the value one for all levels. Thus, the activation potential of all levels becomes approximately equal to their part of the cross-layer activated representation extent, $\alpha_{i,j}$, which is simply their own activated representation extent α_n. Thus, a pattern matching interface between layers reduces the overall inference potential in each layer to a function of the number of activated access points and its own inference extent. To the outside world, any results of a layer's inference engine look like an Artificial Neural Network's (ANN) forward or backward activation potentials.

4 The LEAP Model

The LEAP model is a distributed model for learning lexical, syntactic, semantic and conceptual information about English from web-based sources. It is currently made up of twenty Goal Mind components (each a multithreaded LINUX process) built on the environment's production system and semantic network libraries and its standard PostgreSQL 'C' language interface.

At the surface language layer, LEAP uses a set of seven lexical analyzers and a special purpose Stimuli Routing Network (SRN) used to filter some closed categories. It can discover new instances of open part-of-speech (PoS) categories and new patterns of word use within the input utterances. The concept reasoner's Situational Dependences Semantic Network (SDSM) [5] uses spreading activation to allow a very large network to exist in compressed form in the PostgreSQL database (simulating Long Term Memory or LTM) while small pieces of the network can be uncompressed into a dynamic memory structure within each of the concept reasoners (simulating Short Term Memory or STM).

When a word comes in from the models HMI or HTML reader, all lexical analyzers look up the word in their PoS form table and send either an active or inhibit PoS stimulus message based on this lookup. If the word is not found (i.e., either it is not in the PoS form table or has too low a belief to be used), an analyzer uses reports from other analyzers to try to find a PoS pattern in its PoS pattern table that would indicate that the word may be of its PoS type. If a pattern is found, the word is either added to the PoS form table with a very low belief or the belief of the existing form is incremented based on this example that the word matches the expected pattern. If the word is found but the surrounding words' PoS do not match an existing pattern, a pattern is either added to PoS pattern table with a very low belief or the belief of the existing pattern is incremented based on this example of a valid pattern.

When a concept is looked up, it is copied from the database (LTM) to the dynamic memory (STM) of a concept reasoner and given the maximum time to live by setting its countdown timer to the maximum allowed value. In addition, all other nodes connected via a set number of outbound relations are also activated (moved to memory) and given a time to live based on their distance from the concept that was directly accessed. The concept reasoner is only allowed to inference across active nodes, but when it makes a valid connection between two concepts, it both reports the finding and resets the time to live values for all nodes in the inference path. Running in the background of each concept reasoner is a temporal collector that decrements the countdown timer of each node during each time-slice and removes any node whose counter alarms (hits zero) from STM.

5 The SALT Application

Building on the LEAP and earlier models, the SALT application explores how to dynamically construct and task the control system for a smart environment. It has long been recognized that smart environments need to learn the preferences of their users, but to move them from the lab to mainstream use they will also need to adapt to different and changing hardware environments. Each instance of a smart environment will need to fit into an unique location where size, cost and other factors will determine the hardware being used. These systems will need to be able to accept new smart components as they become available. Further, the control system must be able to 'work around' failed hardware components to ensure both user comfort and safety.

In the current SALT application, five agents are used to test how these agents can learn to communicate and distribute system tasking using a simplified language based on human language lexical, syntactic and semantic constructs. Using seventy-two Goal Mind components, the model current focuses more on language use than the interface to system hardware or the smart environment control, but past Goal Mind research indicates that more robust hardware interfaces and control structures can be added by less than doubling the number of processes in these agents. Running an application with about hundred and fifty processes is well within the capacity of Alchemy to handle on a relatively small Beowulf cluster.

In the current SALT model, a Control and Human Interface agent provides both the Human Machine Interface (HMI) and overall smart environment task distribution. A Kitchen agent controls kitchen appliances and provides meal planning and food inventory control. An Entertainment agent controls entertainment equipment and provides setup based on user preference. An Environmental Control agent monitors A/C and safety components and attempts to match user preferences to safety and efficiency constraints. An Inhabitant and Robot Tracking agent provides the system with situation awareness about mobile elements of the environment using a multiple-camera-based vision system.

Many aspects of the SALT application can exploit the power of the IMP theory. A direct application of the LEAP research in SALT is in its HMI. By integrating the HMI directly to the agents' other sensor modalities, the resulting language interface can be very adaptive. In the future we will build on this to allow the dynamic allocations of agents within the system to support different environments and hardware conditions.

6 Initial Results and Future Work

Both the LEAP and SALT research are presented here to provide an overview of how the Gold Seekers environment allows the meaningful integration of both the theory and application of cognitive approaches. To date, the LEAP results are better understood and will be the focus of this section.

Reading tests with the LEAP model have been conducted on a number of children's stories and web-based news articles. Our current focus is on the size of micro-patterns needed for LEAP to support surface and deep structure language learning and use. To support testing, the learning results, which are stored in a PostgreSQL database are compared against a version of WordNet also store in a PostgreSQL database.

Current LEAP results fall into two basic categories, detailed analysis of individual readings and general observations about language development and the reading task. As expected, the current data from news articles shows that a great deal more language priming is needed to learn at the same rate as with children's stories. These have led to several general observations. First, that using IMP, there is no good way to short-circuit the normal development process starting with simple stories and work up to more and more complex articles. Second, that reading development needs input from other sensor modalities.

Most of the work with children's stories demonstrate similar results so as an example of a detailed analysis we will focus on a single story, *Robert the Rose Horse*. The story contains approximately 1100 words and 200 utterances. This gives a mean-length-of-utterance (MLU) of about 5.5. Ignoring closed categories, the vocabulary is about 90 words. From this and other children's stories studied, it is clear that authors focus on the reduction of word length, lexical complexity and the MLU, but do not necessarily attempt to reduce the syntactic complexity of the resulting utterances.

Using a database primed with 15 words of the core vocabulary and no patterns, LEAP was able to detect 26 patterns. Increasing the core vocabulary to 20 words by adding 5 nouns produced one additional noun pattern, while increasing the core vocabulary to 22 words by adding 2 verbs produced 10 additional patterns. In all cases the patterns were non-conflicting between PoSs. Most patterns show a minimal amount of repeatability within a story, but a few are highly repeatable due to the prose structure of children's stories. These differences in repeatability is not common in news articles. Continuing to add nouns and verbs to the core vocabulary continues to show the same data trend where verbs influence the number of patterns found more than other PoSs. As a result of the spreading activation mechanism used in the concept reasoner, associations between new and exiting concepts can be more easily identified. In *Robert the Rose Horse* this method was used to learn that 'rose' is a type of 'flower' and 'bank' is a thing that can be 'robbed'.

Applying LEAP results to SALT is driven by earlier work with the TALLUS model. In TALLUS, we were able to greatly simplify the language generation task by focusing on telegraphic speech patterns. The agent communication in SALT currently relies on the same reduction in language complexity. There is clearly a point where using a non-formal adaptive language adds too much overhead to an overall agent, but a SALT-like environment is targeted for highly intelligent agents where this is not a major factor. Initial work with SALT has shown that given a small shared vocabulary and a set of patterns that represent a simple syntax, agents can learn a shared language. We are still working on each agent's semantic ties to this language.

Both the Gold Seekers' toolset and the current models support the ability to dynamically create new agents which would allow a SALT-like application to add new agents as new hardware is added and support other changes to the overall environment. While this is clearly an interesting line of research, the short-term focus of both LEAP and SALT is in improving the integration of their language use.

7 Conclusion

Current work with IMP, LEAP and SALT demonstrate that they are providing valuable information about human language development and adaptive agent communication. As other uses of the IMP theory are explored, it is hoped that it will provide a general mechanism for adaptive intelligence within a multiagent environment. The SALT-based research should continue to provide an even better platform for testing the concepts proposed by the LEAP research. While the integration of the LEAP and SALT research paths provide complex challenges, the result of such integration appears to be worth such complexity.

References

1. Anderson, J. R. and Lebiere, C. *Atomic Components of Thought*. Hillsdale, NJ: Lawrence Erlbaum Associates, Pub., 1998.
2. Anderson, J. R. *Cognitive Psychology and its Implications*. New York: W. H. Freeman and Company, 1995.
3. Byrne, M., "ACT-R/PM and Menu Selection: Applying a Cognitive Architecture to HCI", *International Journal of Human Computer Studies*, 1999.
4. Fellbaum, C (Editor), *WordNet: An Electronic Lexical Database*, Cambridge, MA, MIT Press, 1998.
5. Hannon, C. and D. J. Cook. "Developing a Tool for Unified Cognitive Modeling using a Model of Learning and Understanding in Young Children." *The International Journal of Artificial Intelligence Tools*, 10 (2001): 39-63.
6. Hannon C. and D. J. Cook. "Exploring the use of Cognitive Models in AI Applications using the Stroop Effect." Proceedings of the Fourteenth International Florida AI Research Society Conference, May 2001.
7. Hannon, C., A Geographically Distributed Processing Environment for Intelligent Systems. *In Proceedings of PDPS-2002*. 355-360, 2002.
8. Newell, A. *Unified Theories of Cognition*. London: Harvard University Press, 1990.
9. Martindale, C., Cognitive Psychology: A neural-Network Approach, Belmont, CA, Brooks/Cole, 1991.
10. Smith, G. W., *Computers and Human Language*, New York: Oxford Press, 1991.
11. Witbrock, Michael, D. Baxter, J. Curtis, et al. An Interactive Dialogue System for Knowledge Acquisition in Cyc. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, Acapulco, Mexico, 2003.