Towards a General Attention Mechanism for Embedded Intelligent Systems

Helgi Páll Helgason[1], Kristinn R. Thórisson[2], Deon Garrett[3], Eric Nivel[4]

[1]Center for Design and Analysis of Intelligent Agents/School of Computer Science,
Reykjavik University, Menntavegur 1, 101 Reykjavik, Iceland

[2]School of Computer Science, Reykjavik University, Menntavegur 1, 101 Reykjavik, Iceland

[2, 3, 4]Icelandic Institute of Intelligent Machines, Menntavegur 1, 101 Reykjavik, Iceland

[1]helgih09@ru.is; [2]thorisson@iiim.is; [3]deon@iiim.is; [4]eric@iiim.is

Abstract- In the domain of intelligent systems the management of system resources is typically called "attention". Attention mechanisms exist because even environments of moderate complexity are a source of vastly more information than available cognitive resources of any known intelligence can handle. Cognitive resource management has not been of much concern in artificial intelligence (AI) work that builds relatively simple systems for particular targeted problems. For systems capable of a wide range of actions in complex environments, explicit management of time and cognitive resources is not only useful, it is a necessity. We have designed a general attention mechanism for intelligent systems. While a full implementation remains to be realized, the architectural principles on which our work rests have already been implemented. Here we examine some prior work that we find relevant to engineered systems, describe our design, and how it derives from constructivist AI principles.

Keywords- Attention; Resource Management; Architecture; Artificial Intelligence

I. INTRODUCTION

In nature, the phenomenon we call attention stems from the need to manage a set of limited resources; namely, a body and a mind. The human mind is capable of significant amounts of learning, but we cannot learn everything at once. The need to choose what our mind focuses on at any point in time greatly affects many things, including how well we learn, how well we achieve our tasks, and how efficiently our body carries out its tasks. The real world is generally a source of vastly greater amounts of information than any single intelligent agent could ever hope to cognitively ingest and process in any given period of time. So even the smartest being of them all must come equipped with attentional mechanisms of some sort.

Modern attention research can be said to have begun with the work of Colin Cherry on the "cocktail party effect" [2] – referring to the fact that when engaged in a cocktail party conversation you are able to ignore the chatter around you, yet you can hear your name being mentioned by someone standing afar away in the crowd. More generally, it is the process known as selective attention – which includes the ability to remain alert to unexpected but important stimuli while being in a focused state of attention towards other aspects of the environment [14]. These performance characteristics of human attention imply the simultaneous operation of a deliberate, selective steering mechanism and an unconscious reactive mechanism.

We view natural attention as a cognitive function – or a set of them – that allows animals to focus their limited resources on relevant parts of the environment as they perform various tasks, while remaining reactive to unexpected events. Without it we could for example not stay alert to environmental events while finishing an important task, or manage multiple tasks at the same time. Powerful methods for cognitive resource management are critical if we intend to create more capable systems than seen to date – systems of greater autonomy that have the ability to learn to solve novel tasks and adapt to unforeseen changes in environments of real-world complexity, while operating under time constraints – systems we refer to as artificial general intelligence (AGI) systems.

In this paper we present a high-level design of a general attention mechanism that is based on a holistic approach to attention and features data and process prioritization for implementing simultaneous top-down and bottom-up control. First we survey selected work on human attention from cognitive psychology, highlighting ideas we have found useful for implementing attention in an AGI-aspiring system. Then we review implementations of attention within some existing cognitive architectures and discuss their benefits and limitations. Subsequently we look at relevant research in multi-objective optimization and multi-criteria decision making, and propose ways to leverage these for the development of practical and efficient implementations of attention mechanisms. Finally, outline our attention mechanism design, which makes few and fairly high-level requirements for the underlying architecture, but is otherwise architecture-independent. While it would be preferable to achieve a design that is completely independent of system-architecture, and thus applicable to any system, we do not think this to be possible – for the processes of attention are too dependent on core operational features of the system in which it lives. The design proposal presented here is just that – a proposal for a design – but the basic principles on which it rests have already been proven in prior architecture implementations [7, 8, 10, 11]. Our work so far has not only resulted in the new attention mechanism presented here but also greatly affected the kinds of architectures we consider to be relevant to AGI research – architecture and attention are co-dependent. In this respect we discuss how the attention mechanism presented can

be used for managing meta-cognitive operation and architectural self-growth; two fundamental functions of AGI systems (cf. [15]).

II. INFORMATION PROCESSING SYSTEMS IN REAL-WORLD ENVIRONMENTS

While traditional software systems operate in the real-world, they certainly do so through greatly simplified representations of the real-world. The operating environment of such systems is typically deterministic and discrete, and represents time in a fairly coarse-grained manner. None of these features describe well the various environments of everyday life – environments of real-world complexity – which are stochastic, continuous, and temporally fine-grained. The real world is a source of vast amounts of constantly changing information. No resource-bounded information processing system currently known can keep up with processing the entirety of this information at depth – including human beings. Biological evolution has provided us with attention mechanisms that allow our limited cognitive resources to be applied to stimuli that are most important to our present situation.

The operation associated with the deliberate, selective mechanism is referred to as top-down attention as it is directed from cognition towards the environment. The unconscious, reactive mechanism that unconsciously reacts to important information across the sensory spectrum is referred to as bottom-up attention as it is directed from the environment towards cognition. Most existing psychological models of attention fall into one of two categories: early selection or late selection. This categorization and the benefits of each class of models have resulted in considerable debate in the psychological literature. In early selection models, the selection of sensory stimuli to receive further processing occurs early in the sensory pipeline on the basis of primitive physical features of the stimuli. This shallow processing permits little or no analysis of meaning.

In late selection models, selection is performed at later stages in the sensory pipeline based on varying degrees of analysis of meaning. One of the best known examples of early selection models is the Broadbent filter model (Broadbent 1958), which selects and eliminates stimuli based on primitive features. The Deutsch-Norman model (Norman 1969) is a representative example of late-selection models. It features gradual processing of stimuli activates representations in memory, followed by a competitive selection process at the level of these representations on the basis of activation.

Early selection models fail to account for some attentional characteristics commonly observed in human behaviour, such as alertness to relevant information in unattended sensory channels. An example of this is when we notice our own name being called from the other side of a crowded, noisy room in the cocktail party scenario. For this to occur, some analysis of meaning in unattended sensory channels is implied.
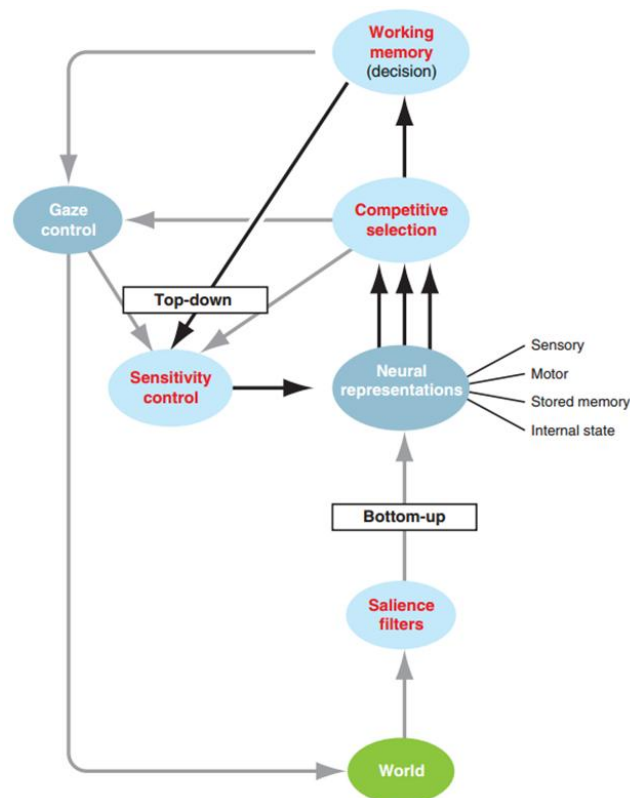

Fig. 1 The Knudsen attention framework (after [6]). Information flows up from the environment and passes through saliency filters that detect important or unusual stimuli. Information that is passed through the filters then activates memory representations that encode knowledge. Memory representations are also activated by top-down sensitivity control, this process is influenced by the contents of working memory and adjusts activation thresholds of representations. Representations compete for access to working memory, with the most active ones being admitted.

More recently, models of attention have been proposed that focus on the interaction of top-down and bottom-up attentional processes. The Knudsen Attention Framework [6] (shown in Figure 1) is a particularly interesting example. It describes the interaction of four interacting processes: top-down sensitivity control, bottom-up filtering, competitive selection and working memory. This framework appears to address all the major necessary parts for attention and represents a promising candidate to inspire efforts aimed at supplying systems with attentional capabilities. The work provides a general solution to the filtering role of attention: When attention is viewed as a single-step process of filtering information, management of resources and their utilization does not easily fit into the picture. If a system is desired to fully leverage available resources, processing decisions must be made in light of current resource availability. Rather than implementing a filtering process based on the content of information, Knudsen proposes a process of prioritization, where information competes for processing resources based on an evaluation of priority collectively implemented by top-down and bottom-up processes. This, and other work prior to ours, highlights functional requirements that we find important. We summarize these as follows:

Information processing systems intended to perform focused tasks in real-world environments, while remaining alert to unexpected events, requiring processes of top-down and bottom-up attention.

Top-down attention processes must target events relevant to current tasks of the system, to guarantee task and goal achievement. However, a system equipped only with top-down attentional processes will utterly fail to target and process important but unexpected events. Such events may be indirectly relevant to current tasks of the system, or may present a threat to goal achievement. Bottom-up attention processes are necessary to target such events. These processes are not (or less) influenced by current tasks of the system, instead they perform evaluation of new information on the basis of novelty to the system. Systems implementing only bottom-up attentional processes face great challenges in accomplishing tasks consisting of multiple steps of action (requiring planning in some form), being largely limited to simple, reactive tasks.

Allocation of resources to attentional processes must be addressed.

The balance in resource allocation between top-down and bottom-up attentional processes must continuously be evaluated and tuned, to avoid excessive bias towards top-down processes, which would introduce operational risk as the capacity of the system to detect unexpected (but possibly important) events would become diminished, and conversely avoiding excessive bias towards bottom-up processes, which would negatively impact task performance as achieving goals could become too time-consuming. A balance of attentional functions cannot be specified in advance – it must take into account the operating context of the system at any given time. Adjustment of this balance must thus be learned through experience.

Late-selection models present a more rational paradigm for attentional processing than early-selection models.

In addition to problems inherent in early selection models, any AI system aspiring towards general intelligence cannot make any assumptions in advance with regards to tasks and environments. Any new information is potentially important for such systems. Primitive, surface-level features can only supply rough clues, at best, with regards to informed resource management. Slow, gradual changes in such shallow features can represent precursors to important events. Ignoring information without some attempts to relate it to prior experience and knowledge of the system introduces operational risk.

III. COGNITIVE ARCHITECTURES AND DECISION THEORY

Few existing cognitive architectures implement attentional functionality explicit in spite of its obvious relevance to practical operation and resource management in real-world environments. Most previous efforts aiming to endow system with attention have focused on particular aspects of the puzzle, such as a particular modality type (cf. Schmidhuber 1991) or its relation to working memory (cf. Skubic 2004), rather than taking a holistic approach. However, some notable exceptions exist that attempt to address the issues and requirements identified in the previous section. For a more general review of the below architectures, see [15].

The Non-Axiomatic Reasoning System (NARS) is a general-purpose reasoning system that addresses real-time processing and assumes insufficient knowledge and resources (Wang, 1995). An implementation of attention is provided based on a computational control strategy called controlled concurrency where special parameters are used to prioritize processing. An urgency parameter represents the priority of a task and decays over time as dictated by a second parameter called durability which is used to represent the temporal scope of the task. Tasks are chosen as recipients of processing resources by a process of competitive selection. The priority of internally-generated tasks is determined and assigned by the system but original goals must be provided and their priorities must be supplied by the developer. This represents potential problems for prioritization – viewed here as an integral role of attention – as part of the process is delegated to an external entity outside the system and may limit the level of autonomy that can be achieved by the system.

LIDA (Baars 2009) is a cognitive architecture that targets intelligent, autonomous agents and is based on a theory of human consciousness. Each operating cycle consists of sensing, attending and action selection phases. The attending phase represents dedicated attentional processes, during which the selection of data for further processing is performed by a collection of small programs called attentional codelets. These form coalitions of data that compete for system resources. Filtering and competitive selection for data is achieved in this fashion. The filtering process is not static as the system is capable of learning to improve its data-filtering from experience. However, the availability of resources at each point in time is not addressed

directly by the system, which causes problems in terms of real-time operation and utilization of resources. Furthermore, prioritization is limited to data and control issues such as process prioritization are not addressed.

The CLARION cognitive architecture addresses attention in a meta-cognitive subsystem (MCS) that handles information selection, dynamic learning and adjustments of system control parameters. However, the MCS does not integrate temporal management which causes problems for real-time operation. Furthermore, current resource availability is not directly used in making resource management decisions while attention is involved in process control to some degree as part of the MCS.

The most commonly accepted function of attention is selection of data to be processed. LIDA and CLARION address information selection in dedicated phases of the sensory pipeline while NARS uses competitive selection based on prioritization in which available resources are used to process information in decreasing order of urgency. The above architectures focus almost exclusively on top-down attention and do not address bottom-up attention. Furthermore, issues of process control are implemented separate to attention (except for a weak relation in CLARION). Attention must include the selection of proper processes given selected data at any point in time, based on the operating context of the system; the context includes time-constraints and resource availability. As selected information can potentially be processed in a vast number of different ways, many of which are not useful or rational, we argue that in a complete attention mechanism process control must be addressed as a key function. While there is some overlap in CLARION between attention and process control, none of these architectures follow an integrated approach to data selection and process control. Finally, real-time operation is problematic for LIDA and CLARION, which do not feature integrated temporal reasoning and make processing decisions without consideration of current resource availability. NARS features temporal reasoning that is implemented with relative time values between events; it can represent the order of tasks and events and additional temporal control is available by way of its durability parameter. The relative approach taken to time results from NARS being a reasoning system that does not focus on action or perception. While dealing with time exclusively in a relative way is clearly better than ignoring time altogether, absolute time values allow for more fine-grained and precise control with regards to real-time operation, and can provide a basis for solid temporal grounding of a system.

As we have argued above, attention systems must include both filtering of information and process control strategies. Systems that can deal with many different goals concurrently, some of which may conflict with one another, must consider action selection at a conceptual level, as the problem of determining which high-level goals to pursue at a given time and/or under a given set of environmental circumstances and choosing actions appropriate to furthering those goals. Somewhat more formally, a sound action selection mechanism might be composed of two parts: a module which selects a set of plausible actions and a second module which incorporates outside information (e.g. preferences) to determine which of the plausible actions should be selected for execution.

In recent years, research into multi-criteria decision making and multi-objective optimization has begun looking more closely at these issues. The organizing principle of multi-objective optimization is finding a set of candidate solutions that may be optimal for some sound set of preferences rather than finding a single solution for all circumstances. That is, while we cannot predict exactly what trade-offs an agent may wish to make at any given point, we can establish that actions resulting in poorer performance with respect to every goal the agent might care about can be eliminated from consideration. The aim of multi-objective optimization is then to produce a representative sample of solutions that would be preferred under some potential set of preferences. Bottom-up attentional methods are still required to provide the agent with the ability to respond to unexpected stimuli that may or may not be relevant to the agent's current goals.

Intelligent systems must be able to obtain information regarding available courses of action and estimate the outcome of taking each such course of action, but they must also be able to feed the predictions back into the process of deciding on strategies to pursue. There will be too many possible actions to evaluate them all, even in simulation. Therefore, the agent will need to use its current internal state and the state of the environment to decide on an appropriate set of actions to consider. Consideration of these actions (via prediction) may then result in the agent seeing opportunities for fulfilling other goals that were not immediately evident. These new goals need to be available to the agent for the purpose of revising the list of candidate actions to take. We can refer to the set of goals and their relative importance rankings to the agent as its current preferences. Preferences have been incorporated into the action selection strategy in other systems in various ways. If known a priori, preferences may be used to construct a satisfactory weighting for each high-level goal, and an optimal or near-optimal action may be selected accordingly. In this form, preferences impose a total ordering on the available actions, and the agent may simply choose the best one from the candidate set. At the other extreme, approaches such as multi-objective evolutionary algorithms [4] push the incorporation of preferences to the final stage of the process in which a representative sample of the complete set of Pareto-optimal actions is subjected to a preference-based selection. Here, a wide range of alternative sets of preferences are considered in parallel, and the decision maker sees the result of each in parallel. Evolutionary algorithms, because they offer the ability to find the entire trade-off set of optimal actions at once, have proven to be a fertile ground for development of numerous strategies for handling preferences [3]. Perhaps most interesting for intelligent systems are approaches in which the preferences of the agent are dynamically injected during the action selection process, helping guide the search for appropriate actions continually, such as that of Thiele et.al. [9]. One important benefit of this approach is in the detection of "knees" (or "elbows") in the trade-off surface. Knee points are the points of maximal marginal return [1, 5]. Loosely, they are strategies where accepting a slight improvement in performance towards one goal provides an abnormally

large decrease in performance on a conflicting goal. Knee points are almost always preferred solutions where they exist, due to this property (i.e. modifying the knee-point strategy to get slightly better results on your primary goal is accompanied by much worse performance on the secondary goal). As top-down attention mechanisms are essentially a way for the agent to balance its conflicting internal goals, the ability to detect such knee points can be extremely valuable.

Incorporation of explicit preference-handling strategies into AGI attention mechanisms remains an open research area, but we believe that strategies derived from these methods may potentially help solve the problems of implementing AGIs on real hardware with bounded resources while maintaining more of the benefits of late selection.

IV. TOWARDS A GENERAL ATTENTION MECHANISM

Our design of an attention mechanism takes a holistic, inclusive approach to attention, addressing top-down goal-directed control and data selection as well as to bottom-up alertness for handling novel and unexpected events. An overview of the proposed attention mechanism is shown in Figure 2. We have identified a set of architectural constraints that must be satisfied by candidates for implementation of our attention mechanism:
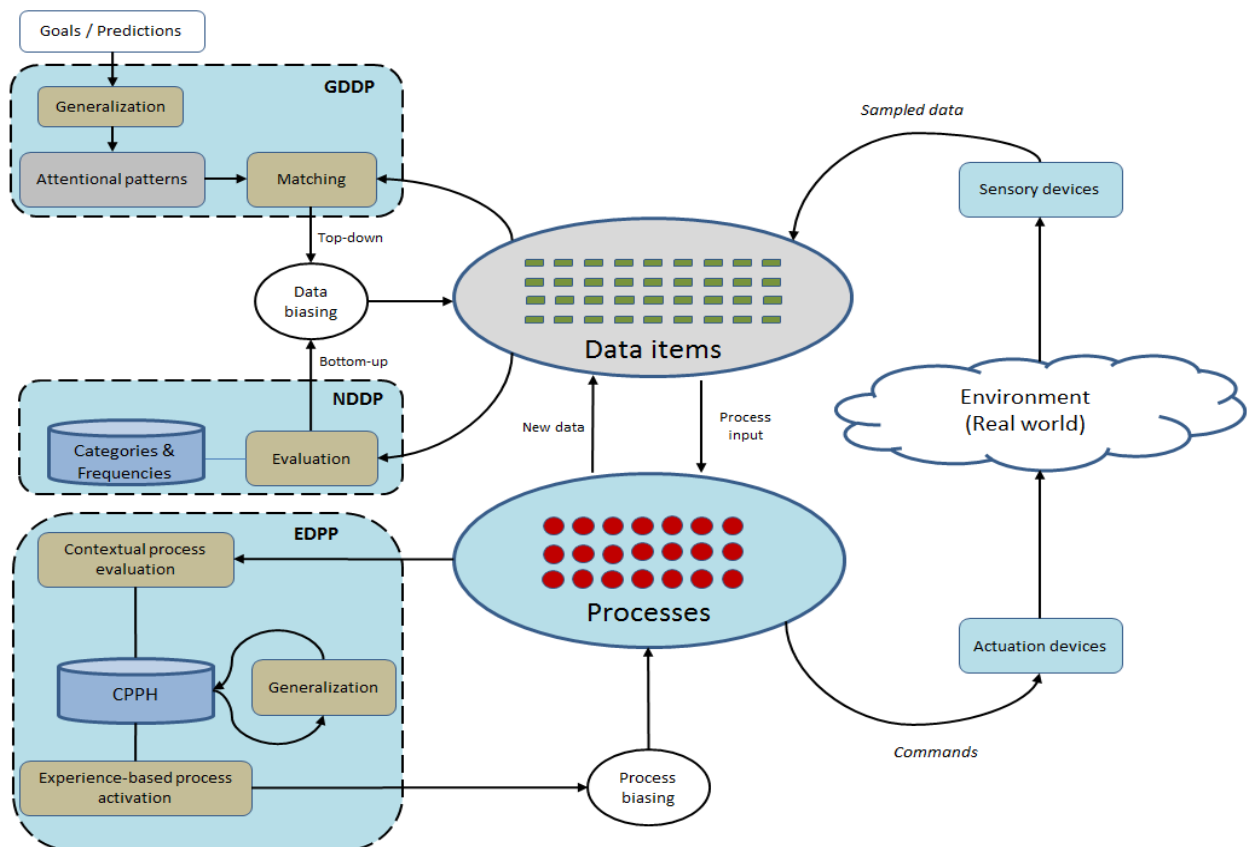

Fig. 2 Overview of the proposed attention mechanism

Requirement #1: Data-driven execution model. All processing in the system occurs as a result of the occurrence of data. Each process is executed only when exposed to data that matches its input specification. This absolves the need for fixed control loops or operating cycles and allows for efficient, flexible operation on multiple simultaneous time scales.

Requirement #2: Fine-grained structure. Both atomic processes and data units are assumed to be relatively small and numerous [10]; to perform complex tasks large numbers of these combine together. The system-wide effects of the minute operations of small components – including their effect on resource consumption – are far more tractable to reason about than the effects of large black-box components.

Requirement #3: Predictive capabilities. The system architecture must support the ability to predict, as predictions are control data required for top-down attention (in addition to goals).

Requirement #4: Unified sensory pipeline. Data must be given equal treatment regardless of whether it originated in the operating environment or inside the system. This allows cognitive functions – attention, in particular – to be equally applied to task performance and introspective meta-cognitive processing such as self-reconfiguration.

Requirement #5: Symbolic knowledge representation. The architecture must have a symbolic level of knowledge representation, but no restrictions are imposed on sub-symbolic functionality on which this level may be based.

In systems that satisfy these requirements, small units of data and processing interact where a process executes if it is actively matched with an existing data item on the basis of its input specification. Pattern matching represents a promising method for determining such matches as it allows flexibility in input specifications of processes. The operating environment is continuously sampled by sensors, resulting in generation of new data items, which in turn trigger processes to execute. This results in generation of more data items and commands for the actuators of the system that produce action in the environment on part of the system. Effects of such actions are observed by environmental sampling, closing the perception-action loop.

For resource management, priority parameters are needed for both processes and data items. The central role of attention, regardless of implementation, is to determine appropriate values for these control parameters with regards to the present operating situation. We refer to these priority parameters as saliency for data items and activation for processes. The high-level control policy of the system is to execute processes with highest activation with data items of highest saliency as inputs, however no process will execute without being exposed to compatible data items due to the data-driven nature of the system. All processes can be viewed as data (the contents of which are their code) and data items containing code can be viewed as processes. The main role of attention is to determine and assign priority values to processes and data items; this may be viewed as a biasing task. In our design, three components (each described below) operating in parallel perform these tasks. While the depicted system architecture is likely not the only possible system architecture that is able to satisfy the architectural requirements presented earlier, it serves well to explain the operation of our attention mechanism.

Goal-Driven Data Prioritizer (GDDP): Goals and predictions of the system are specified in operational terms, identifying states (inside the system or in the operating environment) that are desired (goals) or expected (predictions). The specification of such states undergoes generalization, a process in which concrete values and highly specific information are replaced by variables with the result being a special type of pattern we refer to as an attentional pattern. Attentional patterns target data items to varying levels of specification, ranging from all information relating to a particular entity to all information from a specified modality (e.g. auditory). For example, if the system has a goal of having object O1 in position P1, an attentional pattern is created that matches all information related to O1 (e.g. all data units referring to O1). This works in an identical fashion for predictions. The unified nature of the sensory pipeline allows identical methods for targeting external and internal data. When a data item matches an attentional pattern, it receives positive bias (increase in saliency) relative to priority of the goal that spawned the matching attentional pattern. Data units not matching any active attentional template will not receive bias from this process.

Novelty-Driven Data Prioritizer (NDDP): Novel and unexpected events, in terms of prior system experience, are expected to occur during operation. The nature of top-down attention, guided by active goals and predictions, makes such events unlikely to be caught by the GDDP – this is why the NDDP is necessary. The NDDP is responsible for determining a quantitative measure of novelty and unexpectedness for incoming data items and modifying their saliency values accordingly. The underlying idea is that novel data are interesting to the system and likely to be useful in some way; for example learning to detect events that pose a threat to the achievement of active goals. Unlike the GDDP, the operation of the NDDP does not attempt to determine actual relevance of data, but rather gives novel information a greater chance of being processed. Evaluation of novelty and unexpectedness is based on the operational experience of the system. Data or patterns of data that have been observed by the system on prior occasions receive lower bias (resulting in decreased saliency) than data that have never been observed. To accomplish the evaluation under tight temporal (and memory) constraints, it is necessary to compress prior experience of the system in some way, preferably in data structures that allow for efficient look-up and comparison. Dynamic categorization of data coupled with frequency information represents one possible solution. The NDDP must constantly update its control data based on incoming information in order to evaluate novelty and unexpectedness in a satisfactory manner. Habituation is an emergent operational property of this process; novel or unexpected information will automatically cease to be so after having been observed on an increasing number of occasions.

Experience-Driven Process Prioritizer (EDPP): While relationships between goals and processes are difficult to determine directly, the operational experience of the system may be used to extract such information by maintaining a history of the contribution of each individual process to the achievement of individual goals. This is a non-trivial task; many goals will be achieved by interaction and collaboration of a number of processes. However, the problem is tractable using a form of back-propagation from goal achievement backwards through the operational chain in which it resulted (using some form of ampliative reasoning - c.f. Wang 1995). The EDPP must be capable of determining similarity between goals, as strictly identical goals are unlikely to be repeated on multiple occasions. When a new goal is generated within the system, this process must search some compressed forms of the operational history in order to find a sufficiently similar goal that was previously achieved. The best such match (if one is found) results in positive biasing (increased activation) of processes that contributed to goal achievement on previous occasions. This can be achieved by recording each contribution of a process towards the achievement of a goal in a dedicated data structure called the Contextualized Process Performance History (CPPH). Generalization processes then generate abstract goals from this information, making existing knowledge more flexible and applicable to a much wider range of goals.

While the design itself does not feature dedicated processes for real-time operation, it facilitates real-time operation based on small processing units and having the ability to generate predictions (including temporal ones) with regards to its own operation. The significance of small processing units with homogenous computational complexity is that most processes take

roughly the same amount of time to execute, making temporal aspects of performance more predictable, and that the system is highly interruptible and pre-emptive, as it never has to wait for time-consuming processes to complete before reacting to new data. An important feature of the design approach presented here is that it can be applied directly to systems that manage their own growth and expansion – constructivist architectures [12, 13]. As the sum of internal system activity constitutes a large amount of information, an attention mechanism is required to manage resources for self-reconfiguration – in much the same way as for other task performance. The mechanism presented here already assumes a unified sensory pipeline; attention operates identically on externally and internally generated data. By generating goals targeting internal states that advance directed self-reconfiguration of the system, AGI systems can be envisioned that use this attention mechanism to simultaneously perform tasks in complex environments and manage their own growth, while operating under real-time constraints with limited resources.

V. CONCLUSIONS

Surprisingly limited amount of work focusing on attention exists in the field of AI, although we have demonstrated that existing attention models from cognitive psychology can be mapped to AGI architectures in a useful way. We have identified a large overlap between the processes of attention and control mechanisms of the underlying architecture. This strongly indicates that retrofitting an existing architecture with resource management capabilities such as the ones described here will be highly problematic. Attention reveals itself on closer examination as a ubiquitous function of a cognitive architecture, influencing operation and structure across all levels. Finally, we proposed connections to existing work in incorporating preferences and multi-criteria decision making as a way of extending attention mechanisms in AGI with additional beneficial properties. The result of this work, whose original goal was the design of an attention mechanism, also resulted in a near-complete control mechanism for cognitive architectures.

REFERENCES

[1] Bechikh, S., Lamjed, B.S., Ghédira, K. (2010). Searching for knee regions in multi-objective optimization using mobile reference points. In Proceedings of the 2010 ACM Symposium on Applied Computing. pp. 1118-1125.

[2] Cherry, E. C. 1953. Some experiments on the recognition of speech, with one and two ears. Journal of the Acoustical Society of America, pp. 975-979.

[3] Coello-Coello, C. 2000. Handling preferences in evolutionary multiobjective optimization: a survey. In Proceedings of the 2000 IEEE Congress on Evolutionary Computation.

[4] Deb, K. Multi-Objective Optimization Using Evolutionary Algorithms. Wiley Academic Press. 2001.

[5] Deb. K, and Gupta, S. Understanding knee points in bicriteria problems and their implications as preferred solution principles. Engineering Optimization. 43(11): pp. 1175-1204, 2011.

[6] Knudsen, E. I. 2007. Fundamental components of attention. Annu Rev Neurosci, volume 30. Pages 57-78.Knudsen, E. I. 2007. Fundamental components of attention. Annu Rev Neurosci, vol. 30, pp. 57-78.

[7] Nivel, E. 2007. Ikon Flux 2.0. Reykjavik University technical report, available online at http://www.ru.is/media/skjol-td/RUTR-CS07006.pdf.

[8] Nivel, E., Thórisson, K. R. 2008. Self-Programming: Operationalizing Autonomy. In Proceedings of the 2nd Conf. on Artificial General Intelligence.

[9] Thiele, L., Miettinen, K., Korhonen, P., and Molina, J. A Preference-Based Evolutionary Algorithm for Multi-Objective Optimization. Evolutionary Computation. 17(3): 411-436, 2009.

[10] Thórisson, K. R., and Nivel, E. 2009. Achieving artificial general intelligence through peewee granularity. In Proc. of the 2nd Conf. on Artificial General Intelligence, pp. 220-221.

[11] Thórisson, K. R., and Nivel, E. 2009. Holistic intelligence: Transversal skills and current methodologies. Proc. of the 2nd Conf. on Artificial General Intelligence, pp. 222-223.

[12] Thórisson, K. R. 2009. From Constructionist to Constructivist A.I. Keynote, Technical Report, FS-90-01, AAAI press, Menlo Park, California.

[13] Thórisson, K. R. (2012). A New Constructivist AI: From Manual Construction to Self-Constructing Systems. In Wang & Goertzel (eds.), Theoretical Foundations of Artificial General Intelligence, 2012.

[14] Wood, N., Cowan, N. (1995). The cocktail party phenomenon revisited: How frequent are attention shifts to one own name in an irrelevant auditory channel? Journal of Experimental Psychology: Learning, Memory, and Cognition, vol. 21.

[15] Thórisson, K. R., Helgason, H. P. 2012. Cognitive Architectures and Autonomy: A Comparative review. Journal of Artificial General Intelligence, vol. 3, pages 1-30.