COGENT – An Innovative Architecture for Cognitive Processing

Julius F. Bogdanowicz
Raytheon, jfbogdanowicz@raytheon.com
John Granacki
USC-ISI, Granacki@isi.edu

Overview

The COGENT project is sponsored by the DARPA IPTO office under the Architectures for Cognitive Information processing program. Raytheon is leading a diverse team to define processing requirements for emerging DoD cognitive applications and cognitive system models to develop architecture concepts for a new class of architecture that will provide orders of magnitude improvements in processing capability for this type of processing.

Development Approach

The COGENT team has been using a top-down development approach to conceptualize the COGENT cognitive architecture. This approach is driven by a derivation of computational requirements for emerging DoD cognitive applications, profiling of existing algorithm implementations and the definition of an application independent Cognitive System Model based on the characteristics of a number of research cognitive systems such as SOAR, ACT-R, and Icarus. Figure 1 provides an overview of the development approach.

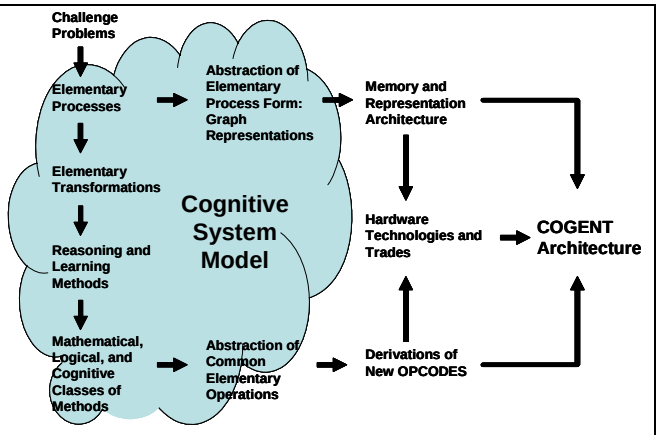


Figure 1: A topdown development approach is being used on the COGENT project.

Application Attributes

We have evaluated a number of applications such as dynamic planning for unattended vehicles, intelligence analysis, and the human computer interface. In general, a cognitive system must be able to deliver real-time response in very dynamic environments, process very large amounts of knowledge and data, and support the modeling of the Warfighter's intent. Applications need a robust cognitive system model. We have adopted an Observe-Orient-Decide-Act + Learn model for COGENT. In general the system must "Sense and Respond" to the environment, "Predict and "Prepare" what actions to take next and exploit both "Reactive" and "Contemplative" processes. Graphs are used to provide a formal, expressive representation for knowledge in the system.

Processing parallelism is available at the coarse (alternative processing contexts), mid level (multiple computational threads) and low level (within a processing thread). Computations are based on inexact information and must be managed to avoid exhaustive search to provide approximate solutions. Latency tolerant processing techniques are needed with sophisticated memory management techniques for access and manipulation of sparse memory representations in short/long term memory. To simplify application development, the computational view of the system should be decoupled from the developer's view. The system will need to support a wide range of computation kernels.

COGENT Processing Architecture

The COGENT evaluated a number of alternative computational architectures for the ACIP program. We have selected a novel re-circulating type architecture which is memory centric. We move the computation to where the data is instead of moving the data. Figure 2 provides a toplevel view of the architecture. The top portion of the figure provides an application independent view of the cognitive architecture supporting the developer and the bottom half of the figure provides a computational view of the hardware. A universal naming approach is used for data, agents and processing contexts. The developer has no knowledge of where data is stored or processed within the computational fabric.

The computational fabric is composed of three major elements: Data distribution Center (DDC), Cognitive Agent & Graph Engine (CAGE), and Prioritize-Filter-Fuse (PFF). The DDC automatically "scatters" the computation to the appropriate CAGE node that contains the data. The CAGE nodes provides support for legacy codes and provides accelerated semantic access to short/long term memory and primitive graph operations. Memory is distributed across the CAGE nodes. CAGE nodes do not communicate directly. The PFF automatically gathers and coalesces

results, prunes stale and unproductive computations and re-circulates unfinished computations to the DDC for re-distribution to the appropriate CAGE node.

A functional simulator has been developed for the COGENT processing architecture. A set of fundamental and elementary operations have been developed for execution on these architectural elements. An analogical reasoning system has been implemented on the simulator. Analogical reasoning is a major component of many cognitive system models and has been used for applications such as dynamic planning. At the beginning of the program, we defined a goal of achieving a minimum speedup of 800x over an Intel 4 baseline. Currently, we have shown that a 64 CAGE node system can achieve a speedup in excess of 1200x.

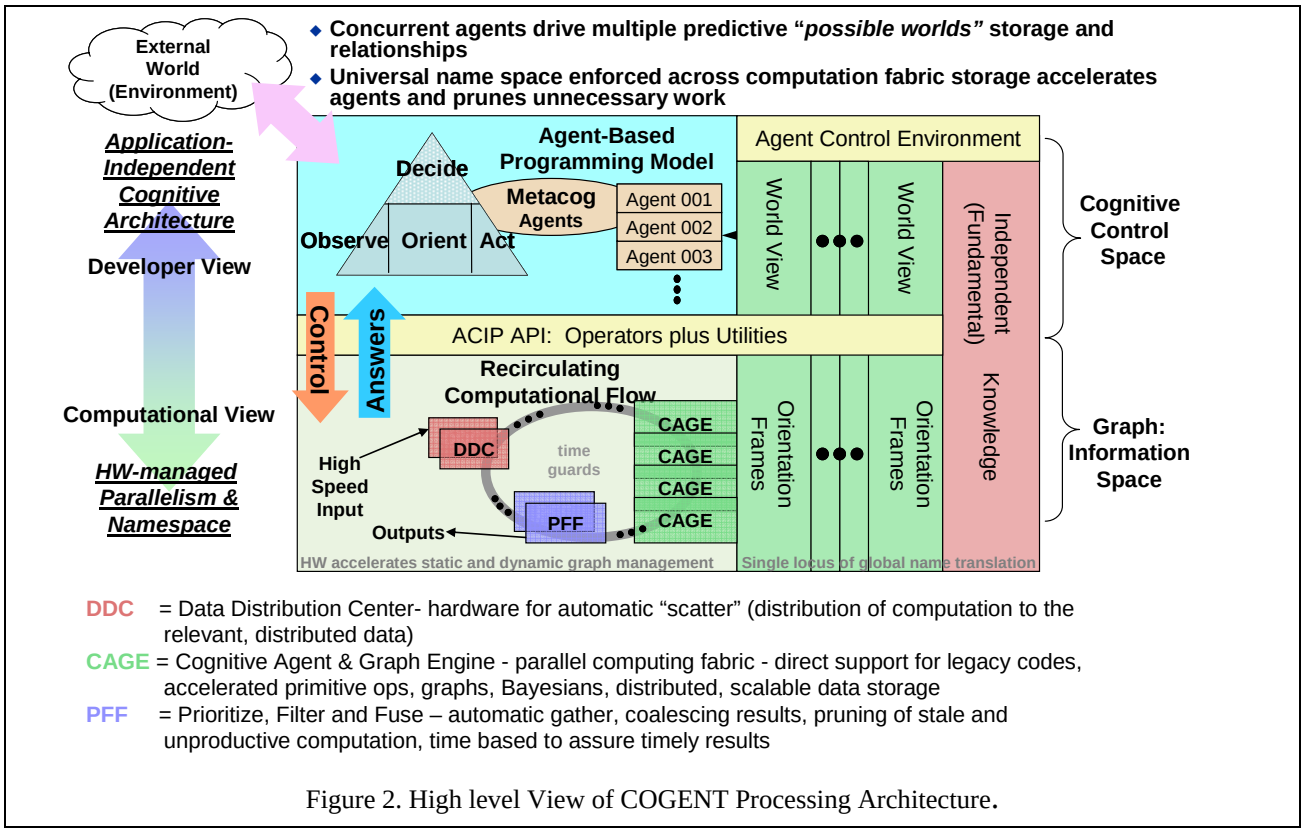
Table 1 provides a comparison of some of the differences between a conventional processor and the COGENT system. Differences in memory organization and access and system control flow are highlighted.

Summary

We have provided a brief high level view of a new processing architecture concept developed for cognitive processing. During Phase 2 of the ACIP program, a detailed design and simulation/emulation of the architecture will be performed.

Table 1. Comparison of COGENT and Conventional Processor Features.

Conventional Processor	COGENT Processor
Relies heavily on cache: dynamic access patterns make cache ineffective	No cache – large knowledge base does not fit; large on chip memories with good bandwidth – processor memory BW > 100 GB/S
Load/store access to memory	HW manipulation of memory access, global ID vs. address, publish and subscribe to information sources
Word focused – no semantics	Semantically accessed memory: Graph based representation of knowledge – HW optimized access and manipulation mechanisms
General computing: RISC instruction set, register file focus, compiler driven program	Optimizations for the cognitive operations – intensive memory focus with staging and location of data HW optimized
Exact, repeatable deterministic functions with low level semaphores	Probabilistic representations, reconstructive memory for runtime synthesized data representations
Ops concept is driven from program counter, interrupts, etc.	Ops concept is driven from data flow and probabilistic data relationships – dynamically adjusted based on experience



DDC = Data Distribution Center- hardware for automatic "scatter" (distribution of computation to the relevant, distributed data)

CAGE = Cognitive Agent & Graph Engine - parallel computing fabric - direct support for legacy codes, accelerated primitive ops, graphs, Bayesians, distributed, scalable data storage

PFF = Prioritize, Filter and Fuse – automatic gather, coalescing results, pruning of stale and unproductive computation, time based to assure timely results

Figure 2. High level View of COGENT Processing Architecture.