Features↓ vs. Architectures→ (download this table)	4D/RCS	ACT-R	ART	BECCA	biSoar	CERA-CRANIUM	Chrest	Clarion	CogPrime	CoJACK	Disciple	Epic	FORR	GLAIR	GMU-BICA
Contributors to this table: → *Curator: Alexei Samsonovich* *Last update: 2010-09-20*	*James Albus*	*Andrea Stocco and Christian Lebiere*	*Stephen Grossberg*	*Brandon Rohrer, updated 8/3/2011*	*B. Chandrasekaran and U. Kurup*	*Raul Arrabales*	*Fernand Gobet and Peter Lane*	*Ron Sun*	*Ben Goertzel*	*Frank Ritter & Rick Evertsz*	*George Tecuci*	*Shane Mueller, Andrea Stocco*	*Susan L. Epstein*	*Stuart C. Shapiro*	*Alexei Samsonovich*
Iconic link to diagrams															
Basic Overview															
Knowledge and experiences are represented using	images, maps, objects, events, state, attributes, relationships, situations, episodes, frames	Chunks and productions,	Visual 3D boundary and surface representations; auditory streams; spatial, object, and verbal working memories; list chunks; drive representations for reinforcement learning; orienting system; expectation filter; spectral timing networks	Perceptual experiences are represented features and combinations of features. Knowledge about the world is represented as transitions between experiences.	The representational framework of Soar plus diagrams – the diagrammatic part can also be combined with any symbolic general architecture, such as ACT-R	Single percepts, complex percepts, and mission percepts. Single and complex behaviors.	chunks and productions	chunks, rules, and NNs	CogPrime is a multi-representational system. The core representation consists of a hypergraphs with uncertain logical relationships and associative relations operating together. Procedures are stored as functional programs; episodes are stored in part as "movies" in a simulation engine; and there are other specialized methods too.	Beliefs-Desires-Intentions (BDI) architecture with events, plans/intentions (procedural), beliefsets (declarative) and activation levels	Disciple shell includes general modules for user-agent interaction, ontology representation, problem solving, learning, and tutoring, as well as domain-independent knowledge (e.g., knowledge for evidence-based reasoning). The problem solving engine of a Disciple cognitive assistant (see the top part of Figure 6) employs a general divide-and-conquer strategy, where complex problems are successively reduced to simpler problems, and the solutions of the simpler problems are successively combined into the solutions of the corresponding complex problems. To exhibit this type of behavior, the knowledge base of the agent contains a hierarchy of ontologies, as well as problem reduction rules and solution synthesis rules which are expressed with the concepts from the ontologies.	Production Rules, Working memory entries	*Descriptives* are shared knowledge resources computed on demand and refreshed only when necessary. *Advisors* are domain-dependent decision rationales for actions. Measurements are synopses of problem solving experiences.	SNePS, simultaneously a logic-based, assertional frame-based, and propositional graph based representation	schemas, mental states
Main components	Behavior Generation, World Modeling, Value Judgment, Sensory Processing, Knowledge Database. These are organized in a hierarchical real-time control system (RCS) architecture	Chunk-based declarative memory; buffers; procedural knowledge encoded as productions	Many model brain regions, notably laminar cortical and thalamic circuits	BECCA is a solution to the general reinforcement learning problem. It consists of two parts, an unsupervised feature creator and a model-based incremental learner. Both are incremental and on-line, designed for a physically embodied agent operating in an unstructured environment.	DRS, for diagram representation; perceptual and action routines to get information from and create/modify diagrams.	Physical layer global workspace, Mission-specific layer global workspace, core layer contextualization, attention, sensory prediction, status assessment, goal management.	Attention, sensory memory, short-term memory, long-term memory	procedural knowledge and declarative knowledge (each in both implicit and explicit forms --- rules and NNs)	The primary knowledge store is the AtomSpace, a neural-symbolic "weighted labeled hypergraph" with multiple cognitive processes acting on it (in a manner carefully designed to manifest cross-process "cognitive synergy"), and other specialized knowledge stores indexed by it. The cognitive processes are numerous but include: an uncertain inference engine (PLN, Probabilistic Logic Networks), a probabilistic evolutionary program learning engine (MOSES, developed initially by Moshe Looks), an attention allocation algorithm (ECAN, Economic Attention Networks, which is somewhat neural net like), concept formation and blending heuristics, etc. Work is under way to incorporate a variant of Itamar Arel's DeSTIN system as a perception and action layer. Motivation and emotion are handled via a variant of Joscha Bach's MicroPsi framework called CogPsi.	Beliefsets for long term memory, plans, intentions, events, goals		Cognitive processor (including production rule interpreter and working memory), long term memory, production memory, detailed perceptual-motor interfaces (auditory processor, visual processor, ocular motor processor, vocal motor processor, manual motor processor, tactile processor)	Advisors are organized into three tiers. Tier-1 Advisors are fast and correct, recommend individual actions, and are consulted in a pre-specified order. Tier-2 Advisors trigger in the presence of a recognized situation, recommend (possible partially ordered) sets of actions, and are consulted in a pre-specified order. Tier-3 Advisors are heuristics, recommend individual actions, and are consulted together. Tier-3 Advisors' opinions express preference strengths that are combines with weights during voting to select an action. A FORR-based system learns those weights from traces of its problem-solving behavior.	1) Knowledge Layer containing Semantic Memory, Episodic Memory, Quantified & conditional beliefs, Plans for non-primitive acts, Plans to achieve goals, Beliefs about preconditions & effects of acts, Policies (Conditions for performing acts), Self-knowledge, Meta-knowledge; 2) Perceptuo-Motor Layer containing implementations of primitive actions, perceptual structures that ground KL symbols, deictic and modality registers; 3) Sensori-Actuator Layer containing sensor and effector controllers.	memory systems: working, semantic, episodic, procedural, iconic (I/O); plus: cognitive map, reward system, the engine
Primary original reference(s)		Anderson, J.R., & Lebiere, C. (1998). The Atomic Components of Thought. Mahwah: Lawrence Erlbaum Associates.	Grossberg, S. (1987). Competitive learning: From interactive activation to adaptive resonance. Cognitive Science 11: 23-63.	B Rohrer, "An implemented architecture for feature creation and general reinforcement learning," Workshop on Self-Programming in AGI Systems, Fourth International Conference on Artificial General Intelligence, Mountain View, CA, Aug 3-6, 2011. (1MB pdf) http://www.sandia.gov/rohrer/doc/Rohrer11ImplementedArchitectureFeature.pdf	Unmesh Kurup & B. Chandrasekaran, "Modeling Memories of Large-scale Space Using a Bimodal Cognitive Architecture." Proceedings of the International Conference on Cognitive Modeling July 27-29, 2007, Ann Arbor, MI. (CD-ROM, 6 pages.)	Arrabales, R. Ledezma, A. and Sanchis, A. "A Cognitive Approach to Multimodal Attention". Journal of Physical Agents. Volume 3. Issue 1. Pages 53-64. January 2009.	Feigenbaum, E. A., & Simon, H. A. (1962). A theory of the serial position effect. British Journal of Psychology, 53, 307-320. Gobet, F. & Lane, P. (2005). The CHREST architecture of cognition: Listening to empirical data. In D. Davis (Ed.), Visions of mind: Architectures for cognition and affect. (pp. 204-224). Hershey, PA: Information Science Publishing.	Sun, R. (2004). The CLARION cognitive architecture: Extending cognitive modeling to social simulation. In: Ron Sun (Ed.), Cognition and Multi-Agent Interaction. Cambridge University Press: New York.			The most representative early paper on Disciple is [74]. Other key references include [75-78]. Most recent representative publications include [79,80]. [74] Tecuci G., Disciple: A Theory, Methodology and System for Learning Expert Knowledge, Thèse de Docteur en Science, University of Paris-South, 1988. [75] Tecuci G., Building Intelligent Agents: An Apprenticeship Multistrategy Learning Theory, Methodology, Tool and Case Studies, San Diego: Academic Press, 1998. [76] Tecuci G., Boicu M., Bowman M., Marcu D., with a commentary by Burke M., An Innovative Application from the DARPA Knowledge Bases Programs: Rapid Development of a Course of Action Critiquer, AI Magazine, 22, 2, pp. 43-61, 2001. [77] Tecuci G., Boicu M., Boicu C., Marcu D., Stanescu B., Barbulescu M., The Disciple-RKF Learning and Reasoning Agent, Computational Intelligence, 21, 4, pp. 462-479, 2005. [78] Tecuci G., Boicu M., and Comello J., Agent-Assisted Center of Gravity Analysis, CD with Disciple-COG and Lecture Notes used in courses at the US Army War College and Air War College, GMU Press, 2008. [79] Tecuci G., Boicu M., Marcu D., Boicu C., Barbulescu M., Disciple-LTA: Learning, Tutoring and Analytic Assistance, Journal of Intelligence Community Research and Development, 2008. [80] Tecuci G., Schum D.A., Boicu M., Marcu D., Hamilton B., Intelligence Analysis as Agent-Assisted Discovery of Evidence, Hypotheses and Arguments. In: Phillips-Wren, G., Jain, L.C., Nakamatsu, K., Howlett, R.J. (eds.) Advances in Intelligent Decision Technologies, SIST 4, pp. 1-10. Springer-Verlag, Berlin Heidelberg, 2010.	Kieras, D. & Meyer, D.E. (1997). An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. Human-Computer Interaction., 12, 391-438.; Meyer, D.E., & Kieras, D.E. (1997). A computational theory of executive cognitive processes and multiple task performance: Part I. Basic mechanisms. Psychological Review 63: 81-97.	Susan L. Epstein. 1994. For the Right Reasons: The FORR Architecture for Learning in a Skill Domain. Cognitive Science, 18(3): 479-511.		Samsonovich, A. V. and De Jong, K. A. (2005). Designing a self-aware neuromorphic hybrid. In K.R. Thorisson, H. Vilhjalmsson, and S. Marsela (Eds.), AAAI-05 Workshop on Modular Construction of Human-Like Intelligence: AAAI Technical Report, volume WS-05-08
Other key references				www.sandia.gov/rohrer	B. Chandrasekaran. Multimodal Cognitive Architecture: Making Perception More Central to Intelligent Behavior. Proceedings of the AAAI National Conference on Artificial Intelligence, 2006, pp. 1508-1512.		Gobet, F. & Lane, P. (2005). The CHREST architecture of cognition: Listening to empirical data. In D. Davis (Ed.), Visions of mind: Architectures for cognition and affect. (pp. 204-224). Hershey, PA: Information Science Publishing.					Kieras, D. EPIC Architecture Principles of Operation. ftp://www.eecs.umich.edu/people/kieras/EPICtutorial/EPICPrinOp.pdf	Epstein, S. L. and S. Petrovic. 2010 Learning Expertise with Bounded Rationality and Self-awareness. In *Autonomous Search*. Y. Hamadi, E. M., F. Saubion, editors. Springer.		
Most recent representative reference(s)	Albus, J. S., and Barbera, A. J. (2005) "RCS: A cognitive architecture for intelligent multi-agent systems," Annual Reviews in Control, Vol. 29, Issue 1 87-99	Anderson, 2007	http://cns.bu.edu/~steve	www.sandia.gov/rohrer B Rohrer, M Bernard, JD Morrow, F Rothganger, P Xavier, "Model-free learning and control in a mobile robot," 5th Intl Conf on Natural Computation, Tianjin, China, Aug 14-16, 2009. B Rohrer, JD Morrow, F Rothganger, PG Xavier, "Concepts from data," Brain-Inspired Cognitive Architectures Symposium, AAAI Fall Symposium Series, Washington DC, Nov 5-7, 2009.		Arrabales, R. Ledezma, A. and Sanchis, A. "CERA-CRANIUM: A Test Bed for Machine Consciousness Research". International Workshop on Machine Consciousness. Hong Kong. June 2009.	Gobet, F., Lane, P.C.R., Croker, S., Cheng, P. C-H., Jones, G., Oliver, I., and Pine, J. M. Chunking mechanisms in human learning. Trends in Cognitive Science, 5:236-243, 2001. Gobet, F., & Lane, P. C. R. (2010). The CHREST architecture of cognition: The role of perception in general intelligence. The Third Conference on Artificial General Intelligence. Lugano, Switzerland.	http://www.cogsci.rpi.edu/~rsun/clarion.html	Goertzel, Ben (2009). OpenCogPrime: A Cognitive Synergy Based Architecture for Embodied General Intelligence, Proceedings of ICCI-2009. See also http://opencog.org Goertzel, Ben et al. (2010). OpenCogBot: Achieving Generally Intelligent Virtual Agent Control and Humanoid Robotics via Cognitive Synergy, Proceedings of ICAI-10, Beijing	Evertsz, R., Pedrotti, M., Busetta, P., Acar, H., & Ritter, F. E. (2009). Populating VBS2 with realistic virtual actors. In Proceedings of the 18th Conference on Behavior Representation in Modeling and Simulation. 1-8. 09-BRIMS-04.			Epstein, S. L. and S. Petrovic. 2010 Learning a Mixture of Search Heuristics. Metareasoning in *Thinking about thinking*. Cox, M. T. and A. Raja, editors. MIT Press.	Shapiro, S.C., & Bona, J.P. (in press). The GLAIR cognitive architecture. AAAI Technical Report FS-09-01. Menlo Park, CA: AAAI Press	Samsonovich, A. V., Ascoli, G. A., De Jong, K. A., and Coletti, M. A. (2006). AAAI Technical Report WS-06-03, pp. 129-134. Menlo Park, CA: AAAI Press.
Implementation	http://www.isd.mel.nist.gov/projects/rcslib/ C++, Windows Real-time, VXworks, Neutral Messaging Language (NML), Mobility Open Architecture Simulation and Tools (MOAST) http://sourceforge.net/projects/moast/ Urban Search and Rescue Simulation (USARSim) http://sourceforge.net/projects/usarsim/	Lisp, TCL/Tk	Nonlinear neural networks (feedback, multiple spatial and temporal scales)	MATLAB, Python	Implemented in the Soar framework	.Net Framework, code written in C#. Runtime based on CCR (Concurrency and Coordination Runtime) and DSS (Decentralized Software Services), part of Robotics Developer Studio 2008 R2. http://www.conscious-robots.com/en/robotics-studio/2.html	Lisp, Java	Java	The OpenCogPrime system implements CogPrime within the open-source OpenCog AI framework, see http://opencog.org. The implementation is mostly C++ for Linux, some components in Java; also a Scheme shell is used for interacting with the system.	overlay to JACK®	Disciple was initially implemented in Lisp and is currently implemented in Java	Original version: Common Lisp; EPIC-X: C++	Common Lisp, Java	Common Lisp	Matlab, Python
Funding program, project and environment in which the architecture was applied (added by Jim Albus)	Please see the long list: http://members.cox.net/bica2009/cogarch/4DRCS.pdf			Sandia National Labs, internal R&D funding								Office of Naval Research	National Science Foundation		DARPA IPTO BICA, virtual indoor/outdoor environments
Support for Common Components															
Working memory?	yes	Not explicitly defined	Yes. Recurrent shunting on-center off-surround network that obeys LTM Invariance Principle and Inhibition of Return rehearsal law	Yes. Current percepts are folded together with a decayed version of recent percepts. This creates a short history of salient percepts that functions as working memory.	Diagrammatic working memory added	yes	Yes, although we call it short-term memory. Auditory short-term memory and visuo-spatial short-term memory are implemented.	separate structure	yes	The active beliefs and intentions	The framework of Disciple supports features and components that are common for many cognitive architectures, including working memory (reasoning trees), semantic memory (ontologies), episodic memory (reasoning examples), and procedural memory (rules). Communication is based on natural language patterns learned from the user.	Yes	Yes		yes: includes mental states of the Self
Semantic memory?	yes	Encoded as chunks	Yes, limited: Associations between chunks	Yes. Semantic information for a feature is obtained from the history of the agent's experience.		no (to be implemented)	Yes. Implemented by the network of chunks in long-term memory.	yes. in both implicit and explicit forms (chunks/rules and NNs)	yes	encocoded as (weighted) beliefs in beliefsets, uses ACT-R's declarative memory equations		Not explicitly	In many descriptives	Yes, in SNePS	yes: includes schemas
Episodic memory?	yes	Not explicitly defined	Yes, limited: Builds on hippocampal spatial and temporal representations	Yes. Individual transitions from experience to experience are an episode.		no (to be implemented)	No, although "episodic" links are used in some simulations.	yes. in both implicit and explicit forms (chunks	yes	Not explicitly defined, but would be encoded as beliefs in		No	Yes, in task history and summary measurements	Yes, temporally-related beliefs in SNePS	yes: includes frozen mental state assemblies