THE IEEE
Intelligent Informatics
BULLETIN

IEEE Computer Society Technical Committee on Intelligent Informatics

December 2013 Vol. 14 No. 1 (ISSN 1727-5997)

Profile

Feature Articles

Announcements

On-line version: http://www.comp.hkbu.edu.hk/~iib (ISSN 1727-6004)

The Technical Committee on Intelligent Informatics (TCII) of the IEEE Computer Society deals with tools and systems using biologically and linguistically motivated computational paradigms such as artificial neural networks, fuzzy logic, evolutionary optimization, rough sets, data mining, Web intelligence, intelligent agent technology, parallel and distributed information processing, and virtual reality. If you are a member of the IEEE Computer Society, you may join the TCII without cost at http://computer.org/tcsignup/.

The IEEE Intelligent Informatics Bulletin

Aims and Scope

The IEEE Intelligent Informatics Bulletin is the official publication of the Technical Committee on Intelligent Informatics (TCII) of the IEEE Computer Society, which is published once a year in both hardcopies and electronic copies. The contents of the Bulletin include (but may not be limited to):

1) Letters and Communications of the TCII Executive Committee

2) Feature Articles

3) R&D Profiles (R&D organizations, interview profile on individuals, and projects etc.)

4) Book Reviews

5) News, Reports, and Announcements (TCII sponsored or important/related activities)

Materials suitable for publication at the IEEE Intelligent Informatics Bulletin should be sent directly to the Associate Editors of respective sections.

Technical or survey articles are subject to peer reviews, and their scope may include the theories, methods, tools, techniques, systems, and experiences for/in developing and applying biologically and linguistically motivated computational paradigms, such as artificial neural networks, fuzzy logic, evolutionary optimization, rough sets, and self-organization in the research and application domains, such as data mining, Web intelligence, intelligent agent technology, parallel and distributed information processing, and virtual reality.

Publisher: *The IEEE Computer Society Technical Committee on Intelligent Informatics*
Address: *Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong (Attention: Dr. William K. Cheung; Email:william@comp.hkbu.edu.hk)*
ISSN Number: *1727-5997(printed)1727-6004(on-line)*
Abstracting and Indexing: *All the published articles will be submitted to the following on-line search engines and bibliographies databases for indexing—Google(www.google.com), The ResearchIndex(citeseer.nj.nec.com), The Collection of Computer Science Bibliographies (liinwww.ira.uka.de/bibliography/index.html), and* ***DBLP*** *Computer Science Bibliography (www.informatik.uni-trier.de/~ley/db/index.html).*

The Technical Committee on Intelligent Informatics (TCII) of the IEEE Computer Society deals with tools and systems using biologically and linguistically motivated computational paradigms such as artificial neural networks, fuzzy logic, evolutionary optimization, rough sets, data mining, Web intelligence, intelligent agent technology, parallel and distributed information processing, and virtual reality. If you are a member of the IEEE Computer Society, you may join the TCII without cost at http://computer.org/tcsignup/.

The IEEE Intelligent Informatics Bulletin

Aims and Scope

The IEEE Intelligent Informatics Bulletin is the official publication of the Technical Committee on Intelligent Informatics (TCII) of the IEEE Computer Society, which is published once a year in both hardcopies and electronic copies. The contents of the Bulletin include (but may not be limited to):

1) Letters and Communications of the TCII Executive Committee

2) Feature Articles

3) R&D Profiles (R&D organizations, interview profile on individuals, and projects etc.)

4) Book Reviews

5) News, Reports, and Announcements (TCII sponsored or important/related activities)

Materials suitable for publication at the IEEE Intelligent Informatics Bulletin should be sent directly to the Associate Editors of respective sections.

Technical or survey articles are subject to peer reviews, and their scope may include the theories, methods, tools, techniques, systems, and experiences for/in developing and applying biologically and linguistically motivated computational paradigms, such as artificial neural networks, fuzzy logic, evolutionary optimization, rough sets, and self-organization in the research and application domains, such as data mining, Web intelligence, intelligent agent technology, parallel and distributed information processing, and virtual reality.

Publisher: *The IEEE Computer Society Technical Committee on Intelligent Informatics*
Address: *Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong (Attention: Dr. William K. Cheung; Email:william@comp.hkbu.edu.hk)*
ISSN Number: *1727-5997(printed)1727-6004(on-line)*
Abstracting and Indexing: *All the published articles will be submitted to the following on-line search engines and bibliographies databases for indexing—Google(www.google.com), The ResearchIndex(citeseer.nj.nec.com), The Collection of Computer Science Bibliographies (liinwww.ira.uka.de/bibliography/index.html), and **DBLP** Computer Science Bibliography (www.informatik.uni-trier.de/~ley/db/index.html).*

The Cognitive Anteater Robotics Laboratory (CARL) at the University of California, Irvine

I. Introduction

Within our grasp is a deep scientific understanding of how the brain's mechanisms give rise to perception, cognition, emotion, action and social engagement with others. Such an understanding will have a revolutionary impact on science, medicine, economic growth, security, and social wellbeing. One way to understand this complicated system is through the construction of working models. Developing neural models that follow the architecture and dynamics of brain networks, combined with building robotic systems that physically ground these models, has great potential to solve one of the Grand Challenges posed by the United States National Academy of Engineering: *Reverse-Engineering the Brain*. Our laboratory attempts to meet this challenge in four complementary ways by: 1) Promoting the field of Cognitive and Brain-Based Robotics. 2) Developing adaptive action selection systems based on principles of vertebrate neuromodulation. 3) Data-mining neuroinformatic and gene expression databases. 4) Constructing large-scale, detailed models of cortical and subcortical processing on parallel computing platforms.

II. Cognitive and Brain-Based Robotics

For over 10 years, we have been promoting the field of Cognitive Robotics, or Neurorobotics. These brain-based robots are physical devices whose control systems have been modeled after aspects of brain processing. The goals of these robots are to better understand cognition through the construction of physical artifacts, and to create practical systems that demonstrate cognitive capabilities. Neurorobotics is built on the notion that the brain is embodied in the body, which is, in turn, embedded in the environment, and that this coupling is necessary for an intelligent system. The field is small, but growing, due to technological advances and increased interdisciplinary research. Our group has developed a series of neurorobotic models that have successfully demonstrated perceptual categorization and conditioning [1], visual binding and scene segmentation [2], texture discrimination with artificial whiskers [3], adaptive motor control [4], spatial memory and navigation [5, 6], and neuromodulation as a general-purpose robot control system [7, 8]. These algorithms have several important features for autonomous robot control in general, such as fluid switching of behavior, gating in important sensory events, and separating signal from noise. Our algorithms and models have been tested on several robotic platforms in our laboratory, and we are currently working with other robotics laboratories around the world to demonstrate their applicability.

As an extension of our previous neurorobotic work in spatial memory and navigation, we are developing cognitive robots capable of contextual learning. A main goal of this research is to create a robot capable of constructing a cognitive map of its environment while foraging for different valued resources under varying environmental conditions. The system should lead to a better understanding of how areas of the medial temporal lobe interact with cortical areas to create flexible episodic memory. Such a system would be a major step forward for autonomous navigation by artificial systems.

Another research direction of our lab, which builds upon our cognitive robotics work, is to deploy teams of cognitive robots. These robot teams, or swarms, can be fairly large in size, and as a result, an inexpensive robot with strong communication capabilities is favourable. To that end, we have developed an open source robotic platform that leverages smartphone technology as a control system [9, 10]. The computing, communication, and sensing capabilities of current smartphones affords an inexpensive yet highly capable robotic platform that can be used for education and research. The platform, called leCarl, consists of an Android phone, R/C car platform, IOIO interface board, and additional sensors (see Figure 1). In the near future, our action selection, learning, and cognitive mapping algorithms will be deployed on a leCarl swarm in a Search and Rescue task.

Figure 1. Android based robotic platform. The Android phone serves as the computing and sensing device. The IOIO provides an interface to add additional sensors, such as IR range finders. The base is installed on the chassis of a R/C truck. The robotic head is composed of a rectangular tube, two servos for the pan and tilt unit, and a phone holder made of foam. Adapted from [9].

III. Adaptive Action Selection Systems Based on Principles of Vertebrate Neuromodulation

The vertebrate neuromodulatory system plays a key role in regulating decision-making and responding to environmental challenges. In particular, the serotonergic system underlies control of stress, social interactions, and risk-taking behavior. The dopaminergic system has been implicated in the prediction of rewards and incentive salience. The cholinergic and noradrenergic systems are thought to play important roles in attention and judging uncertainty. We suggested that the behavior of an autonomous system modeled after the vertebrate neuromodulatory system, might demonstrate the complexity and flexibility associated with higher order animals by monitoring its surroundings, adapting to change, and responding decisively to important environmental events [11]. Since the publication of this paper, our group has demonstrated how these systems can modulate attention in uncertain environments [12], shape decision-making in social situations [13, 14], and be used as an adaptive controller for autonomous robots [8, 15]. Our attentional study showed how the noradrenergic and cholinergic systems interact with each other, and suggested how this could lead to behavioral adaptation in the face of uncertainty [12]. We suggested that basal forebrain activity tracks expected uncertainty and that this shapes attentional search. We also suggested that the locus coeruleus tracks unexpected uncertainty, and this leads rapid responses to changes in the environment.

Game theory can be a powerful tool for testing models and discovering the neural correlates of decision-making in cooperative and competitive situations. In a set of human robot interaction studies using socioeconomic game theory, specifically the Hawk-Dove game, we showed that adaptive agents, whose behavior is guided by simulated dopaminergic and serotonergic systems, could evoke changes in strategy, reward/cost tradeoffs, and reciprocal behavior in subjects [14]. We also showed that division into two groups best described subjects' responses during these games [13]. Lowering subjects' serotonin levels through Acute Tryptophan Depletion caused some subjects to be more aggressive (as expected), but others to be less aggressive (unexpected). We suggest that individual variation, possibly due to genetic differences in serotonin and dopamine action, may be influencing this variability. To further understand this relationship, we turned to another socioeconomic game, called the Stag Hunt, which focuses on cooperation. In the Stag Hunt, subjects can either hunt a low valued hare on their own or form a social contract with another player to hunt a highly valued stag (see Figure 2). We constructed an adaptive agent, based on the interaction between the dopaminergic and serotoninergic systems, which learned to play Stag Hunt and develop strategies based on the human player's tendencies [16]. In this study, we tested the performance of 40 subjects playing against five opponent types (the adaptive agent, and four other set strategies) in a spatiotemporal version of the Stag Hunt game. Subjects put more thought in their movements and in considering the movements of the agent when playing against the adaptive agent. Similar to our Hawk-Dove study, we observed differences between subjects on the individual level, with several responding to the adaptive agent by almost always cooperating, and several others remaining nearly exclusively uncooperative. In future work, we are interested in both the development of the agent strategy and the subjects' reaction to adaptive agents. Moreover, we plan to further investigate the neural correlates of these behaviors through brain imaging, pharmacological manipulations and genetic screening.

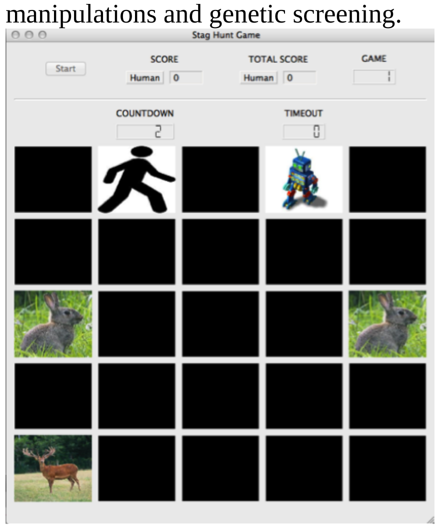


Figure 2. Screenshot of Stag Hunt game board. The game board included a 5x5 grid of spaces upon which the player (stick figure image), agent (robot image), stag (stag image), and hare (hare image) tokens resided. The screen included a button to start the experiment, the subject's score for the round, the subject's overall score for the experiment, the game number within the round, a 3-second countdown to the start of the game, and a 10-second counter monitoring the game's timeout. At the beginning of each game, the locations for the stag, player, and agent tokens were randomly placed along either the top row, bottom row, or middle column at least one square away from each other. The initial positions of the hares were fixed in the locations shown above for all games. The player and agent could move one square at a time towards their goal at the start of the game, while the targets remain fixed. Adapted from [16].

IV. Data-Mining Neuroinformatic and Gene Expression Databases.

In addition to our modeling work, we are taking a neuroinformatic approach to understanding cognitive function. Neuroinformatics is an emerging technique concerned with the management and sharing of neuroscience data. In recent work, we performed an exploratory survey of receptor gene expression associated with classical neuromodulatory systems (i.e., cholinergic, dopaminergic, noradrenergic, and serotonergic) within anatomical origins of these neuromodulatory systems, as well as in the amygdala [17]. Investigation of receptor gene expression in these regions was undertaken using the Allen Mouse Brain Atlas, a growing neuroinformatic resource that contains data sets of extensive mouse gene expression and neuroanatomical data. As a result, this type of exploratory analysis revealed many connectivity relations and receptor localization of these neuromodulatory systems that had not been previously reported (Figure 3). Currently, we are using this approach to understand the structural and functional underpinnings of reward processing by acquiring and analyzing expression data from dopamine and serotonin signaling genes across brain areas associated with the reward circuit.

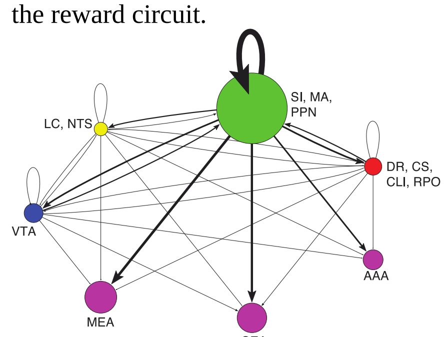


Figure 3. Network model showing overall expression of neuromodulatory receptors and their implied neuromodulatory projections to target areas. Vertices represent brain regions that are either standalone (purple = amygdala regions) or combined regions (yellow = noradrenergic, green = cholinergic, blue = dopaminergic, and red = serotonergic). Directed arcs represent projections going to and coming from a source. The pointed-arrow indicates the target location and the non-arrow end of the arc indicates the origin. The thickness of each arc, as well as the size of vertices,

is proportional to the amount of expression found in the target location. Adapted from [17].

V. Detailed Models of Cortical and Subcortical Processing on Parallel Computing Platforms

Despite recent increases in computer power, constructing a neural model that approaches the size of a human-brain will require several orders of magnitude increases in computation, communication, and memory capacity. Conventional computer hardware may not be the appropriate architecture for modeling a brain. Unlike a conventional computer, the brain is a massively parallel, analog, fault-tolerant, selective system that does not rely on programmed instructions. Alternative computer architectures and programming paradigms, which are neurobiologically inspired, are in need of investigation [18, 19]. Our group has been developing tools to incorporate these brain features into computer models. Specifically, we have constructed large-scale network models that capture the dynamics of neural signaling at the microcircuit (i.e., within brain areas) and macrocircuit (i.e., between brain areas) levels. We have developed a highly efficient implementation of Spiking Neural Networks (SNN) by leveraging the parallel computing power of Graphical Processing Units (GPUs). Our publically available software program, called CARLsim (http://www.socsci.uci.edu/~jkrichma/CARLsim), is a C/C++ based SNN simulator that runs on both generic x86 CPUs and standard off-the-shelf GPUs. With our optimizations, we have demonstrated roughly 25X speedups over cutting edge desktop computers. This simulation environment was released to the modeling community so that researchers would have easy access to large-scale SNN simulations [20]. It has been very popular among computer scientists, neuroscientists, and engineers. Our latest release of simulator software extended this prior model to include more biologically plausible descriptions of synaptic connections and learning rules [21]. In particular, this new simulation environment facilitates the development of very large-scale spiking neural networks that follow the brain's architecture. Using this simulator environment, we developed cortical models of visual form, color, and motion processing in which we replicated color opponency and motion perception results at both the psychophysical and neuronal level (see Figure 4). This simulation environment has also been used to replicate a recent and important finding on how basal forebrain activation can enhance cortical coding of natural scenes [22]. Our spiking neuron model, which included the basal forebrain, thalamus, and visual cortex, suggested that basal forebrain activation switches the firing mode of thalamic neurons, which in turn leads to an increase in within-cell reliability and a decrease in between-cell redundancy in LGN and visual cortex. In near future releases of our spiking simulator, we plan to introduce an automated parameter tuning framework, and a more extensive visual motion perception model. In addition, we are expanding our GPU-accelerated spiking neural network simulator (CARLsim) to run across many GPUs with the use of MPI. We believe this work in the spiking neural network domain will have a broad impact on the neuromorphic engineering community and will one day lead to practical applications deployed on specialized hardware.

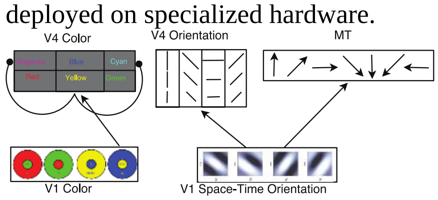


Figure 4. Architecture of the spiking neural network model of visual cortex. In the V1 color layer, there are four color opponent (center+/surround−) responses, which are combined in V4 to respond to six primary colors. The V1 motion energy model projects to edge detecting neurons in V4 and directionally selective neurons in cortical area MT Adapted from [21].

VI. Summary

By combining computational modeling and neuroinformatics with autonomous robots and parallel computing techniques, our group has created a multi-disciplinary approach to understanding the inner workings of the brain and cognition. It is our hope that this approach will continue to benefit both the neuroscience and computer science communities and move us closer to meeting the grand challenge of reverse-engineering the brain.

Contact Information

Derrik E. Asher dasher@uci.edu
Michael C. Avery averym@uci.edu
Michael Beyeler mbeyeler@uci.edu
Liam D. Bucci, lbucci@uci.edu
Kristofor D. Carlson, kdcarlso@uci.edu
Alexis B. Craig, acraig1@uci.edu
Nikil Dutt, dutt@ics.uci.edu
Jeffrey L. Krichmar, jkrichma@uci.edu
Nicolas Oros, noros@uci.edu
Emily L. Rounds, roundse@uci.edu
Andrew Zaldivar, azaldiva@uci.edu

Website:
http://www.socsci.uci.edu/~jkrichma

VII. References

[1] J. L. Krichmar and G. M. Edelman, "Machine Psychology: Autonomous Behavior, Perceptual Categorization, and Conditioning in a Brain-Based Device," *Cerebral Cortex*, vol. 12, pp. 818-830, 2002.
[2] A. K. Seth, J. L. McKinstry, G. M. Edelman, and J. L. Krichmar, "Visual Binding Through Reentrant Connectivity and Dynamic Synchronization in a Brain-based Device," *Cereb Cortex*, vol. 14, pp. 1185-1199, 2004.
[3] A. K. Seth, J. L. McKinstry, G. M. Edelman, and J. L. Krichmar, "Active sensing of visual and tactile stimuli by brain-based devices," *International Journal of Robotics and Automation*, vol. 19, pp. 222-238, 2004.
[4] J. L. McKinstry, G. M. Edelman, and J. L. Krichmar, "A cerebellar model for predictive motor control tested in a brain-based device," *Proc Natl Acad Sci U S A*, vol. 103, pp. 3387-92, Feb 28 2006.
[5] J. L. Krichmar, A. K. Seth, D. A. Nitz, J. G. Fleischer, and G. M. Edelman, "Spatial navigation and causal analysis in a brain-based device modeling cortical-hippocampal interactions," *Neuroinformatics*, vol. 3, pp. 197-222, 2005.
[6] J. L. Krichmar, D. A. Nitz, J. A. Gally, and G. M. Edelman, "Characterizing functional hippocampal pathways in a brain-based device as it solves a spatial memory task," *Proc Natl Acad Sci U S A*, vol. 102, pp. 2111-6, Feb 8 2005.
[7] J. L. Krichmar, "A neurorobotic platform to test the influence of neuromodulatory signaling on anxious and curious behavior," *Front Neurorobot*, vol. 7, pp. 1-17, 2013.
[8] B. R. Cox and J. L. Krichmar, "Neuromodulation as a Robot Controller: A Brain Inspired Design Strategy for Controlling Autonomous Robots," *IEEE Robotics & Automation Magazine*, vol. 16, pp. 72-80, 2009.
[9] N. Oros and J. L. Krichmar, "Smartphone Based Robotics: Powerful, Flexible and

Inexpensive Robots for Hobbyists, Educators, Students and Researchers," *IEEE Robotics & Automation Magazine,* p. Under review, 2013.

[10] N. Oros and J. L. Krichmar, "Neuromodulation, Attention and Localization Using a Novel Android™ Robotic Platform," presented at the ICDL-EpiRob 2012 : IEEE Conference on Development and Learning and Epigenetic Robotics, San Diego, CA, 2012.

[11] J. L. Krichmar, "The Neuromodulatory System - A Framework for Survival and Adaptive Behavior in a Challenging World.," *Adaptive Behavior,* vol. 16, pp. 385-399, 2008.

[12] M. C. Avery, D. A. Nitz, A. A. Chiba, and J. L. Krichmar, "Simulation of Cholinergic and Noradrenergic Modulation of Behavior in Uncertain Environments," *Frontiers in Computational Neuroscience,* vol. 6, pp. 1-16, 2012-January-31 2012.

[13] D. E. Asher, S. Zhang, A. Zaldivar, M. D. Lee, and J. L. Krichmar, "Modeling individual differences in socioeconomic game playing," in *COGSCI 2012 - The Annual Meeting of the Cognitive Science Society*, Sapporo, Japan, 2012.

[14] D. E. Asher, A. Zaldivar, B. Barton, A. A. Brewer, and J. L. Krichmar, "Reciprocity and Retaliation in Social Games with Adaptive Agents," *IEEE Transactions on Autonomous Mental Development,* In Press.

[15] J. L. Krichmar, "A Biologically Inspired Action Selection Algorithm Based on Principles of Neuromodulation," in *IEEE World Congress on Computational Intelligence*, Brisbane, Australia, 2012, pp. 1916-1923.

[16] A. B. Craig, D. E. Asher, N. Oros, A. A. Brewer, and J. L. Krichmar, "Social contracts and human-computer interaction with simulated adapting agents," *Adaptive Behavior,* 2013.

[17] A. Zaldivar and J. Krichmar, "Interactions between the neuromodulatory systems and the amygdala: exploratory survey using the Allen Mouse Brain Atlas," *Brain Structure and Function,* pp. 1-18, 2012.

[18] J. L. Krichmar, N. Dutt, J. M. Nageswaran, and M. Richert, "Neuromorphic Modeling Abstractions and Simulation of Large-Scale Cortical Networks," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, San Jose, CA, 2011, pp. 334-338.

[19] J. M. Nageswaran, M. Richert, N. Dutt, and J. L. Krichmar, "Towards reverse engineering the brain: Modeling abstractions and simulation frameworks," in *VLSI System on Chip Conference (VLSI-SoC), 2010 18th IEEE/IFIP*, 2010, pp. 1-6.

[20] J. M. Nageswaran, N. Dutt, J. L. Krichmar, A. Nicolau, and A. V. Veidenbaum, "A configurable simulation environment for the efficient simulation of large-scale spiking neural networks on graphics processors," *Neural Netw,* vol. 22, pp. 791-800, Jul-Aug 2009.

[21] M. Richert, J. M. Nageswaran, N. Dutt, and J. L. Krichmar, "An efficient simulation environment for modeling large-scale cortical processing," *Frontiers in Neuroinformatics,* vol. 5, pp. 1-15, 2011-September-14 2011.

[22] M. Avery, J. L. Krichmar, and N. Dutt, "Spiking Neuron Model of Basal Forebrain Enhancement of Visual Attention," in *IEEE World Congress on Computational Intelligence*, Brisbane, Australia, 2012.

Intelligent Web Crawling

(WI-IAT 2013 Tutorial)

Denis Shestakov

Abstract—**Web crawling, a process of collecting web pages in an automated manner, is the primary and ubiquitous operation used by a large number of web systems and agents starting from a simple program for website backup to a major web search engine. Due to an astronomical amount of data already published on the Web and ongoing exponential growth of web content, any party that want to take advantage of massive-scale web data faces a high barrier to entry. We start with background on web crawling and the structure of the Web. We then discuss different crawling strategies and describe adaptive web crawling techniques leading to better overall crawl performance. We finally overview some of the challenges in web crawling by presenting such topics as collaborative web crawling, crawling the deep Web and crawling multimedia content. Our goals are to introduce the intelligent systems community to the challenges in web crawling research, present intelligent web crawling approaches, and engage researchers and practitioners for open issues and research problems. Our presentation could be of interest to web intelligence and intelligent agent technology communities as it particularly focuses on the usage of intelligent/adaptive techniques in the web crawling domain.**

Index Terms—**web crawling, web crawler, intelligent crawling, adaptive crawling, collaborative crawling, Web ecosystem, Web structure, incremental crawling, focused crawling, deep Web**

I. Introduction

WEB crawling [1], [2], a process of collecting web pages in an automated manner, is the primary and ubiquitous operation used by a large number of web systems and agents starting from a simple program for website backup to a major web search engine. For example, search engines such as Google or Microsoft Bing use web crawlers to routinely visit billions of web pages, which are then indexed and made available for answering user search requests. In this way, the characteristics of obtained web crawls such as coverage or freshness directly affect on the quality of web search results served to users. Besides web search, the web crawling technology is central in such applications as web data mining and extraction, web monitoring, social media analysis, digital preservation (i.e., web archiving), detection of web spam and fraudulent web sites, web application testing, finding unauthorized use of copyrighted content (music, videos, texts, etc.), identification of illegal and harmful web activities (e.g., terrorist chat rooms), and virtual tourism.

Due to an astronomical amount of data already published on the Web and ongoing exponential growth of web content, any party (be it an individual, company, government agency, non-profit or educational organization) that want to take advantage of massive-scale web data faces a high barrier to entry. Indeed, only network costs associated with the downloading of web-scale size collection by themselves lead to expenses that are not affordable by the majority of potential players.

For those with flexible budgets, there is a next barrier: operating web-scale crawl, i.e. hundreds of millions of pages, is a challenging task that requires skills and expertise in distributed data retrieval and processing, not to mention large operational costs. Finally, for the parties who nevertheless manage to overcome the above obstacles but interested in specific subsets of web information, the results of crawl are often wasteful, as majority of retrieved pages do not match their criteria of interest.

In this paper, we will overview recent advances made in harvesting the information on the Web, in order to introduce the intelligent systems community to the challenges in this area, with particular stress on intelligent web crawling approaches using adaptive crawling agents as well as the underlying open issues and research problems. We will also address issues in building a spectrum of services and applications collecting and aggregating large amounts of web information, e.g., the role of web crawlers in the Web ecosystem, how intelligent crawling strategies can lead to a better overall quality of crawled data.

II. Web Crawling

This section will introduce the basics of web crawler operations and important web crawling applications, and provide relevant statistics on the Web link structure. Next we will describe the architecture of a web crawler and present a number of crawling strategies including three adaptive crawling approaches.

A. Overview

The underlying mechanism of crawling – namely, given an URL download a corresponding web page, extract all URL links from it and repeat the process for those links that were not visited yet – is naive and simple. However, due to a number of imposing restrictions and resource limitations under which crawlers operate, algorithms and techniques behind a large-scale web crawler are far more complicated than the trivial implementation. For example, in order not to be banned by a web server, a crawler has to avoid sending too many URL requests to a server within a short time period. Since the distribution of pages over web servers is non-uniform, a crawler faces a problem of downloading a large number of pages from only a relatively small number of web servers (comparing to their overall number on the Web).

D. Shestakov is with the Department of Media Technology, Aalto University, Finland e-mail: denis.shestakov@aalto.fi

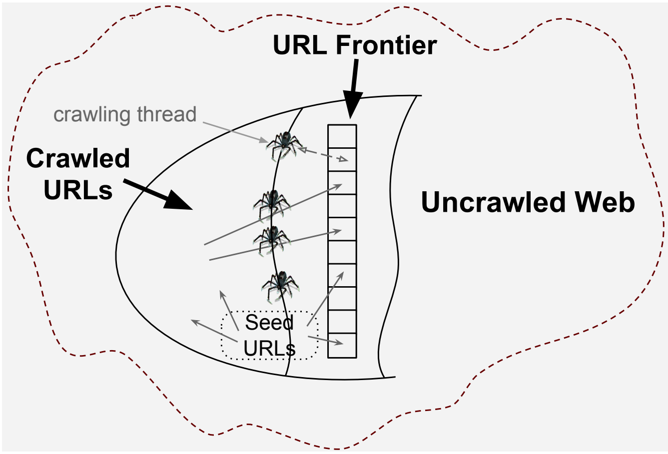


Fig. 1. URL Frontier in crawler's operations.

There are many applications with web crawlers playing a crucial role. The application spectrum ranges from visiting as many web pages as possible by web search engine or web archiver crawlers to the recently appeared trend of using crawlers for web application testing [3]. Needs of commercial web search engines are, however, the most important driving force in design and development of better crawler agents. With a few notable exceptions (e.g., see [4], [5]), academic crawling projects operate on a much smaller scale and apparently employ less sophisticated techniques.

The size and structure of the Web [6] are the most essential aspects that define several key requirements for a web crawler. The exponential growth of the Web suggests that no crawler can cope to cover all the information on the public Web [7], [8]. Moreover, the dynamism of web content guarantees that any collection of crawled documents is stale (not up-to-date) to a certain degree. As normally only limited resources are available, making crawls to be up-to-date involves a trade-off between freshness and coverage of the harvested documents. Similarly, the link structure of the Web [9] is crucial for understanding how crawlers can better prioritize their unseen URL lists.

B. Intelligent Web Crawling

A general-purpose web crawler typically operates in a distributed fashion, with multiple crawl threads that may run under different processes and often at different nodes. The architecture of a crawler [10], [11] includes a number of components, including the URL frontier. It keeps URLs to be visited in some order and returns the one with the highest score to a crawler thread when it seeks for a new URL. The URL frontier is schematically depicted in Figure 1.

There are a number of approaches to prioritize the URLs in the URL frontier. The main goal is to assign a URL some value that corresponds to the "importance" of a web page located at this URL. The URL prioritization strategies clearly depend on the crawling goals. E.g., if a crawler has no domain focus (general) or has to primarily focus on harvesting pages on a certain topic (topical). Another possible concern could be if a crawler should make a snapshot of a certain segment of the Web (batch) or should re-crawl previously visited pages (incremental). In general, one can categorize the existing approaches into six popular strategies used for both general and topical batch crawling: Breadth-First, Depth-First, Backlink count, Best-First, PageRank and Shark-Search [12], [13]. In essence, a crawling strategy defines the assignment of a priority value to a newly extracted URL. Depending on the strategy a number of factors can be taken into account – from a simple time-stamp of adding a link to the frontier to an inherited score value based on relevance scores of several ancestor pages pointing to a page with this link.

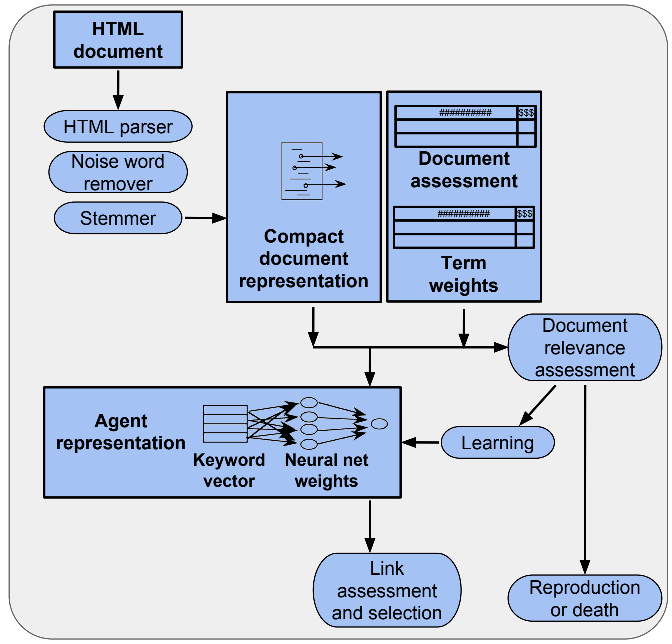


Fig. 2. Architecture of InfoSpider agent.

The abovementioned crawling strategies are static, in the sense that they do not learn from experience or adapt to the context of a topic in the course of crawl. In contrast to them, an intelligent crawler agent uses an adaptive learning model to assign priorities to the URLs in the frontier. In the literature, there exist at least three adaptive crawling approaches: InfoSpiders, ant-based crawling and HMM-supported crawling [14], [15], [16], [17]. While HMM-supported crawling utilizes Hidden Markov Models for learning paths leading to relevant pages, InfoSpiders and ant-based crawling are inspired by evolutionary biology studies and models of social insect collective behaviour correspondingly. Figure 2 shows the architecture of the InfoSpider agent, where agent's representation is supported by neural network.

III. Open Challenges

This section will briefly discuss the role of crawlers in the Web ecosystem and then present some open challenges in web crawling research, such as collaborative web crawling, crawling the deep Web and crawling multimedia content.

Being an important part of the Web ecosystem, crawler agents follow the pull model of resource access, under which a client has to first issue a request for a given resource (compared with the push model where a server can send (push) a content to a client without an earlier request from client-side). While the pull model has several advantages, it also leads to significant inefficiencies in crawlers' performance. The collaborative crawling or "crawling as a common service" approach [18] is the attempt to overcome some of these problems by supplementing a regular general crawler with a scalable filtering layer that allows other parties to crawl by setting conditions for documents of interest and obtaining relevant documents from the prime crawler.

The significant portion of the Web containing publicly-available information from myriads of online web databases (known as the deep Web [19]) is poorly accessible by crawlers. Accessing a deep web resource requires recognizing a search interface (search form) to a database and filling the recognized interface with meaningful values – both tasks are extremely challenging for conventional crawlers. In the literature, there are some relevant techniques for deep web crawling [20], [21], [22].

The Web has evolved from a huge textual repository to a fully-fledged multimedia platform serving web users all media types of content. Images, video, audio are now not just supplementing textual content of web documents but become integral part of many web resources. Most crawlers, however, do not adapt to this change and continue to operate as text harvesting systems. Thus, problems in crawling multimedia content [23], [24] are well-timed and of high importance.

IV. Supporting Materials

The material of this article was presented as a tutorial on the IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT 2013) held in Atlanta, USA in November 2013. The tutorial slides are available at http://goo.gl/woVtQk; note that last part of tutorial provides relevant references to important crawl datasets and self-study materials. The bibliography for web crawling domain can be found in [1], [25].

References

[1] C. Olston and M. Najork, "Web crawling," *Found. Trends Inf. Retr.*, vol. 4, no. 3, pp. 175–246, 2010.

[2] D. Shestakov, "Current challenges in web crawling," in *Proc. ICWE 2013*, 2013, pp. 518–521.

[3] S. Choudhary, M. E. Dincturk, S. M. Mirtaheri, A. Moosavi, G. von Bochmann, G.-V. Jourdan, and I. V. Onut, "Crawling rich internet applications: The state of the art," in *Proc. CASCON 2012*, 2012, pp. 146–160.

[4] H.-T. Lee, D. Leonard, X. Wang, and D. Loguinov, "IRLbot: Scaling to 6 billion pages and beyond," *ACM Trans. Web*, vol. 3, no. 3, pp. 8:1–8:34, 2009.

[5] P. Boldi, A. Marino, M. Santini, and S. Vigna, "Bubing: Massive crawling for the masses," 2013. [Online]. Available: http://vigna.di.unimi.it/papers/P4.8.pdf

[6] J. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins, "The web as a graph: Measurements, models, and methods," in *Proc. COCOON 1999*, 1999, pp. 1–17.

[7] Z. Bar-Yossef and M. Gurevich, "Random sampling from a search engine's index," *J. ACM*, vol. 55, no. 5, pp. 24:1–24:74, 2008.

[8] D. Shestakov, "Sampling the national deep web," in *Proc. DEXA 2011*, 2011, pp. 331–340.

[9] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tompkins, and E. Upfal, "The web as a graph," in *Proc. PODS 2000*, 2000, pp. 1–10.

[10] V. Shkapenyuk and T. Suel, "Design and implementation of a high-performance distributed web crawler," in *Proc. ICDE 2002*, 2002, pp. 357–368.

[11] P. Boldi, B. Codenotti, M. Santini, and S. Vigna, "Ubicrawler: a scalable fully distributed web crawler," *Software: Practice and Experience*, vol. 34, no. 8, pp. 711–726, 2004.

[12] M. Najork and J. L. Wiener, "Breadth-first crawling yields high-quality pages," in *Proc. WWW 2001*, 2001, pp. 114–118.

[13] M. Hersovici, M. Jacovi, Y. S. Maarek, D. Pelleg, M. Shtalhaim, and S. Ur, "The shark-search algorithm. an application: tailored web site mapping," *Computer Networks and {ISDN} Systems*, vol. 30, no. 1-7, pp. 317–326, 1998.

[14] F. Menczer, G. Pant, and P. Srinivasan, "Topical web crawlers: Evaluating adaptive algorithms," *ACM Trans. Internet Technol.*, vol. 4, no. 4, pp. 378–419, 2004.

[15] A. Micarelli and F. Gasparetti, "Adaptive focused crawling," in *The Adaptive Web*, ser. Lecture Notes in Computer Science, 2007, vol. 4321, pp. 231–262.

[16] F. Gasparetti and A. Micarelli, "Swarm intelligence: Agents for adaptive web search," in *Proc. ECAI 2004*, 2004, pp. 1019–1020.

[17] S. Batsakis, E. G. Petrakis, and E. Milios, "Improving the performance of focused web crawlers," *Data & Knowledge Engineering*, vol. 68, no. 10, pp. 1001–1013, 2009.

[18] J. M. Hsieh, S. D. Gribble, and H. M. Levy, "The architecture and implementation of an extensible web crawler." in *Proc. NSDI 2010*, 2010, pp. 329–344.

[19] D. Shestakov, "Deep web: databases on the web," in *Encyclopedia of Database Technologies and Applications, 2nd edition*, 2009, pp. 581–588.

[20] J. Madhavan, D. Ko, L. Kot, V. Ganapathy, A. Rasmussen, and A. Halevy, "Google's deep web crawl," *Proc. VLDB Endow.*, vol. 1, no. 2, pp. 1241–1252, 2008.

[21] D. Shestakov, "On building a search interface discovery system," in *Proc. VLDB Workshops 2009*, 2010, pp. 81–93.

[22] Z. Behfarshad and A. Mesbah, "Hidden-web induced by client-side scripting: an empirical study," in *Proc. ICWE 2013*, 2013, pp. 52–67.

[23] A. Nanopoulos, D. Rafailidis, M. M. Ruxanda, and Y. Manolopoulos, "Music search engines: Specifications and challenges," *Information Processing & Management*, vol. 45, no. 3, pp. 392–396, 2009.

[24] J. Hare, D. Dupplaw, W. Hall, P. H. Lewis, and K. Martinez, "Building a multimedia web observatory platform," 2013. [Online]. Available: http://eprints.soton.ac.uk/352465/

[25] (2013) Web Crawling (Mendeley Group). [Online]. Available: http://www.mendeley.com/groups/531771/web-crawling/

A Survey on Software Agent Architectures

Rosario Girardi and Adriana Leite

***Abstract*—An important decision when developing a software agent is the design of its internal architecture. Several models of deliberative and reactive architectures have already been proposed. However, approaches of hybrid software architectures that combine deliberative and reactive components, with the advantages of both behaviors, are still an open research topic. This paper analyzes the state of the art of software agent architectures from the basic reactive and deliberative models to more advanced ones like hybrid and learning software architectures. A case study on the design of an ontology-driven hybrid and learning software agent architecture is also described.**

***Index Terms*—Software Agents, Software Architectures, Hybrid Agents; Learning Agents; Agent-oriented Development; Software Design**

I. Introduction

DEVELOPING software of high quality is difficult because of the natural complexity of software. Looking for appropriate techniques to confront complexity, software development paradigms have evolved from structured to object-oriented approaches to agent-oriented ones [31].

Having the properties of autonomy, sociability and learning ability, software agents are a very useful software abstraction to the understanding, engineering and use of both complex software problems and solutions like distributed and open systems and to support the decision making process [19][23]. A software agent is an entity that perceives its environment through sensors and acts upon that environment through actuators [38]. Agent attributes allow to approach complexity of software development through appropriate mechanisms for software decomposition, abstraction and flexible interactions between components [31].

During the process of developing a multi-agent system, both the architecture of the agent society and the one of each agent are defined in the global and detailed design phases, respectively, looking for satisfying the functional and non-functional requirements of the system.

A software architecture is a software computational solution to a problem showing how the component parts of a system interact, thus providing an overview of the system structure. It is the product of a software design technique and considered a bridge between requirements engineering and coding where emphasis is on coordination and cooperation over computation. The main elements of a software architecture are components, connectors and cooperation and coordination mechanisms. The components are the computational units, like modules, objects or software agents. The connectors represent the interactions between the architecture components, for instance, the messages exchanged by agents in a multi-agent society. The cooperation and coordination mechanisms define the way in which the elements are arranged, for example, a layered architecture.

Software architectures are represented in graphical diagrams based on architectural styles and design patterns. An architectural style [20] defines a vocabulary of components and connection types and a set of restrictions on how these components may be combined. A design pattern [35] is a reusable solution to a recurring problem. It shows not only the solution but also its restrictions and the context in which to apply this solution. Frequently used architectural styles and design patterns are pipes and filters, object-oriented, layered and blackboard architectures.

Agent architectures emulate human behavior through models of reactive architectures, supporting instinctive or reflexive behavior, and deliberative architectures, supporting different forms of automatic reasoning (deductive, inductive and analogical reasoning, among others).

A reactive architecture is ideal in cases where an immediate action is necessary for a certain perception, and then, its main advantage is the speed of the agent action. Differently, a deliberative architecture is suitable to support more complex decisions where a reasoning process should be executed to find the most appropriate action for a particular perception.

Several models of deliberative and reactive architectures have already been proposed. However, approaches of hybrid software architectures that combine deliberative and reactive components, with the advantages of both behaviors are still an open research topic. These architectures have greater complexity in their definition and use, since they require synchronization between reactive and deliberative components. Another important aspect to be considered on the definition of the internal architecture of an agent is the definition of knowledge bases representing the agent knowledge about itself and the external environment. One of the most effective ways of representing agent knowledge bases are ontologies. An ontology is an explicit specification of a conceptualization [39]. Conceptualization refers to the abstraction of a part of the world (a domain) where are represented the relevant concepts and

Rosario Girardi and Adriana Leite are with the Computer Science Department – DEINF/GESEC，Federal University of Maranhão, Brazil (E-mails: rosariogirardi@gmail.com, adri07lc@gmail.com).

This work is supported by CAPES and FAPEMA, institutions of the Brazilian Government for scientific and technologic development.

their relationships. Ontologies have the advantages of being formal and reusable semantic representations.

Considering that an agent has the granularity of a subsystem, the specification of its architecture has particular importance both for approaching complexity and understanding software solutions.

This article analyzes the state of the art of software agent architectures from the basic reactive and deliberative models to more advanced ones like hybrid and learning software architectures.

The rest of the paper is organized as follows. Section II introduces the basic software agent architectures and section III their main components. Section IV discusses current approaches for advanced architectures such as learning and hybrid ones. Section V concludes the paper.

II. BASIC SOFTWARE AGENT ARCHITECTURES

Basic software agent architectures currently structure just two types of behaviors: reactive or reflexive and deliberative by reasoning. However, there are different proposals for structuring these basic behaviors and its variations [21][38] that are discussed in the following sections.

Fig. 1 illustrates the architecture of a generic agent, which perceives the external environment through sensors and then interprets the perception and transforms it into a sentence. It performs a mapping of the current perception to an action, represented in Fig. 1 by the question mark. This mapping occurs immediately in the case of a reactive behavior or through automatic reasoning otherwise. After encountering the sentence that represents the action, this is interpreted and the action is executed in the environment.

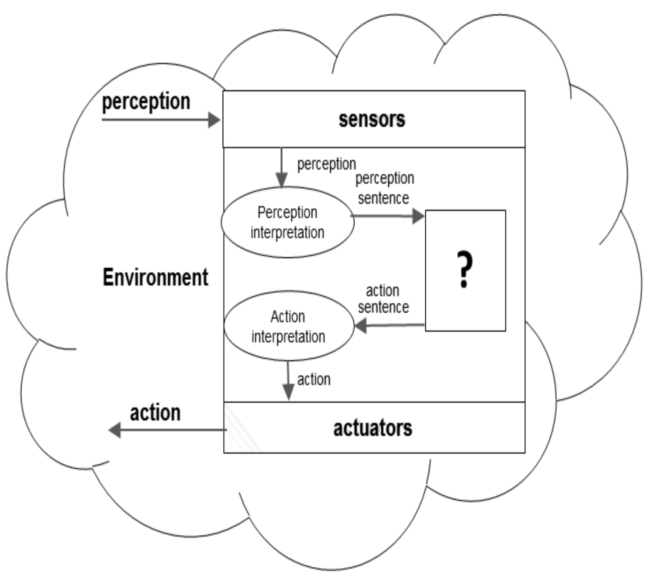


Fig. 1. A generic software agent (adapted from [38])

A. Russel and Norvig Classification

Russel and Norvig [38] define four basic types of architectures for software agents: simple reflex agents, reflex-agents with state, goal-based agents and utility-based agents.

1) *Simple reflex agents*

Simple reflex agents are considered the simplest type of agent where the agent's action is performed as an answer of just the current perception. It has a knowledge base that contains a set of <condition,action> rules. For each perception satisfying the condition, a corresponding action in the knowledge base is selected and performed. For instance, if the knowledge base has the reactive rule “if the car in front is breaking then initiate braking”, when perceived that “The car in front braked” the action “Initiate braking” would be performed. Fig. 2 and Fig. 3 illustrate the structure of a simple reflex agent. First, the perception is interpreted and transformed into a sentence, and then the knowledge base of the agent is updated. If a matching is found between the sentence and the condition of a rule, the corresponding action sentence of the rule is selected. Finally, the sentence that represents the action is interpreted and the agent performs the action in the environment.

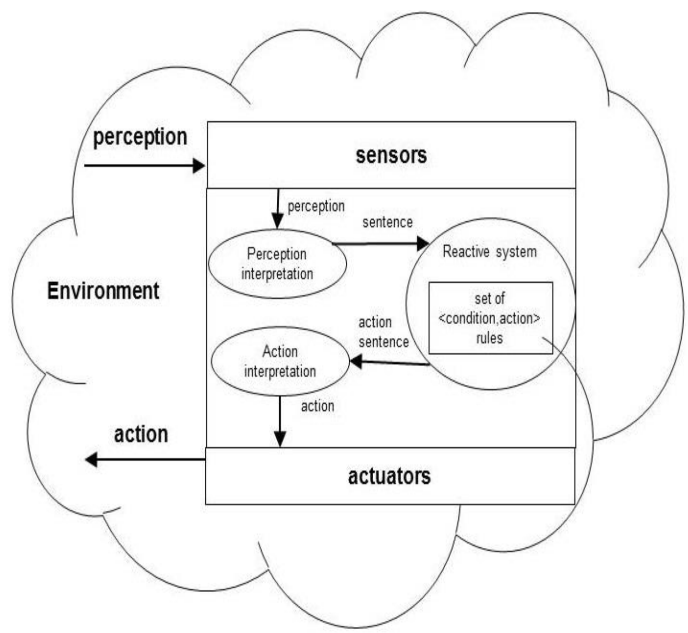


Fig. 2. The structure of a simple reflex agent (adapted from [38])

```
function Skeleton-Agent(percept) returns action
static: memory, the agent’s memory of the world
memory ← Update-memory(memory, percept)
action ← Choose-Best_action(memory)
memory ← Update-memory(memory, action)
return action
```

Fig. 3. Reflex-agent algorithm [38]

Fig. 4 illustrates a vacuum world as an example of a simple reflex agent architecture. This agent environment has two rooms called A and B. Each room can have a dirty or clean state. When the agent perceives that the room A is dirty it should perform the action "clean", then goes to the room B and do the same check. The operation of the vacuum agent is described in the algorithm of Fig. 5.

.

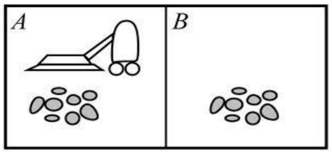


Fig. 4. The vacuum world: an example of a simple reflex agent [38]

function REFLEX-VACUUM-AGENT([location,status])
returns an *action*
if status = Dirt **then return** Suck
else if location = A **then return** Right
else if location = B **then return** Left

Fig. 5. A simple reflex vaccum agent algorithm [38]

2) Reflex agents with state

A reflex agent with state differs from the previous one on the fact that it updates the state of the environment with each new perception. An action in the set of rules is selected according to the perception history and not just from the current perception (Fig. 6 and Fig. 7). Moreover, the agent with state has a world model, that is, contains knowledge about the environment. For example, when an agent crosses a street its reactive behavior will depend on the traffic rules that make part of the agent world model.

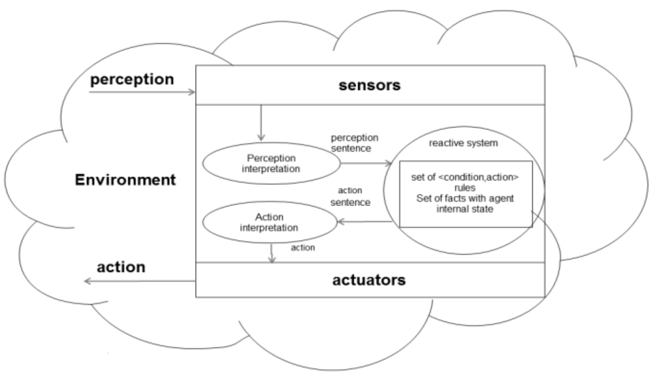


Fig. 6. The structure of a reflex agent with state (adapted from [38])

function Reflex-Agent-With-State(percept) **returns** action
static: state, a description of the current world state
rules, a set of condition-action rules
state ← Update-State(state, percept)
rule ← Rule-Match(state, rules)
action ← Rule-Action (rule)
state ← Update-State(state, action)
return action

Fig. 7. An algorithm with the basic operation of a reflex agent with state [38]

3) Goal-based agents

A goal-based agent maintains the state of the environment and has a goal to be achieved. For that it has to perform an action or a sequence of several actions determined though a reasoning mechanism on the agent knowledge base. This makes them less efficient than reflex agents due to the fact that the processing time required to perform a reasoning process is usually greater than the one required by rule condition-action agents. In Fig. 8 the basic structure of goal-based agent is illustrated and Fig. 9 describes the algorithm with the basic operation of this agent type.

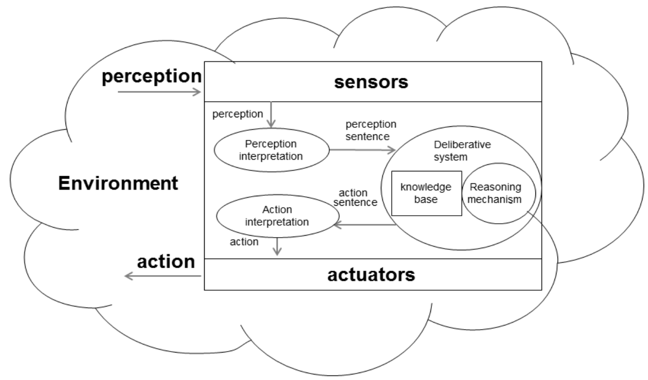


Fig. 8. The structure of a goal-based agent (adapted from [38])

function Goal-Based-Agent (percept) returns action
static: KB, goal, search-space
KB ← UPDATE-KB (KB, percept)
goal ← FORMULATE-GOAL (KB)
search-space ← FORMULATE-PROBLEM (KB, goal)
plan ←SEARCH (search-space, goal)
while (plan not empty) do
action ← RECOMMENDATION (plan, KB)
plan ← remainder(plan, KB)
return action

Fig. 9. Algorithm with the basic operation of goal-based agents [38]

4) Utility-based agents

Utility-based agents are similar to goal-base agents, but they also have an utility measure used to evaluate the level of success when reached the goal. For example, an agent whose goal is to go from "City A" to "City B" can achieve this goal taking two to four hours to complete the route. By defining a measure of time efficiency, the agent will know that completing the route in less time is better. Fig. 10 illustrates the basic structure of such agents and Fig. 11 describes its basic operation.

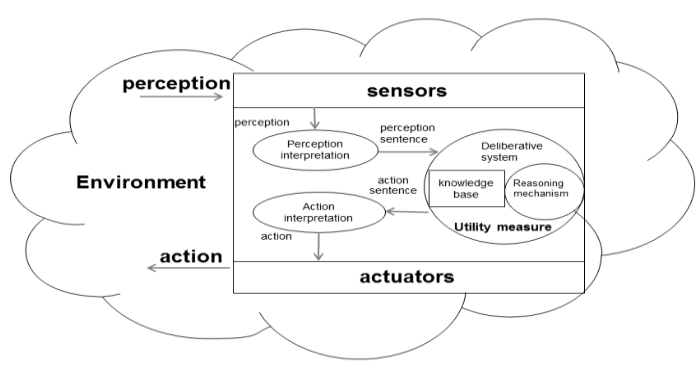


Fig. 10. The structure of an utility-based agent (adapted from [38])

```
function  Goal-Based-Agent (percept) returns action
static: KB, goal, search-space
KB ← UPDATE-KB (KB, percept)
goal ← FORMULATE-GOAL (KB, utility_measure)
search-space ← FORMULATE-PROBLEM (KB, goal)
plan ←SEARCH (search-space , goal)
while (plan not empty) do
action ← RECOMMENDATION(plan, KB)
plan ← REMAINDER (plan, KB)
return action
```

Fig. 11. An algorithm with the basic operation of an utility-based agents [38]

B. Wooldridge Classification

Wooldridge [19][21][22] classifies basic agents into three main categories: deductive reasoning agents, practical reasoning agents and reactive agents.

1) Deductive reasoning agents

Deductive reasoning agents have a symbolic model of their enviroment and their behavior is explicitly represented, typically using logic. The agent handles this representation by deductive reasoning which can require considerable time to be performed. Fig. 12 illustrates the structure of a deductive reasoning agente.

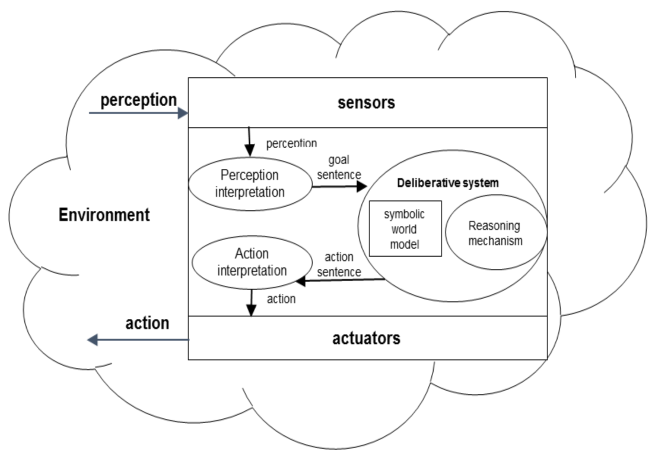


Fig. 12. The structure of a deductive reasoning agent (adapted from [21])

2) Pratical reasoning agents

Practical reasoning agents, also called BDI agents, are their based on the idea that agent acts not only by deductive reasoning. In practical reasoning, the decision process is performed based on beliefs, desires and intentions. Agent beliefs include the agent knowledge about the world, desires are some kind of desirable state to be reached, and intentions are the actions that the agent decides to take to achieve their desires.

To understand BDI agents, in [8] is given the following example: When a person graduates from university with a first degree, he/she is faced with some important choices. Typically, he/she proceeds in these choices by first deciding what sort of career to follow. For example, one might consider a career as an academic, or a career in industry. The process of deciding which career to aim for is deliberation. Once one has fixed upon a career, there are further choices to be made; in particular, how to bring about this career. Suppose that, after deliberation, you choose to pursue a career as an academic. The next step is to decide how to achieve this state of affairs. This process is means–ends reasoning. The end result of means–ends reasoning is a plan or recipe of some kind for achieving the chosen state of affairs. For the career example, a plan might involve first applying to an appropriate university for a PhD place, and so on. After obtaining a plan, an agent will typically then attempt to carry out (or execute) the plan, in order to bring about the chosen state of affairs. If all goes well (the plan is sound, and the agent's environment cooperates sufficiently), then after the plan has been executed, the chosen state of affairs will be achieved.

In Fig. 13 the main elements of a BDI architecture are illustrated. This architecture consists of a set of current beliefs that represent the information the agent has about its environment; by a belief revision function (brf - beliefs review function), that is the entry of a perception and current agent beliefs; a generation options function (generate options), which determines the choices of actions available to the agent (desires), based on their current beliefs about the environment and their current intentions; by a set of current options (desires), representing the actions available to the agent; by a filter function (filter) which represents the agent's deliberation process and determining the intentions of the agent based on their current beliefs, desires and intentions; by a set of current intentions (intentions), representing the current goal of the agent and by a selection function action (action) that determines an action to be performed on the basis of current intentions.

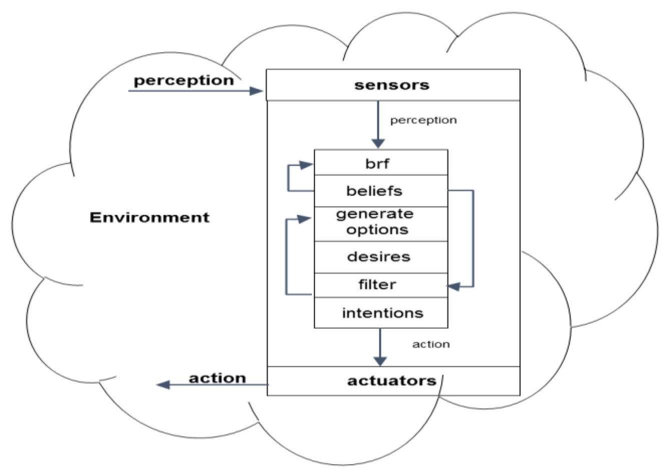


Fig. 13. The structure of a BDI agent (adapted from [21])

The main advantages of BDI architectures, cited in [21], are conceptual proximity to the process of human decision and the informal understanding of the notions of belief, desire and intentions.

3) Reactive agents

Reactive architectures are defined with the aim of filling the main shortcoming of logic-based architectures: processing time. The goal of the architecture is reactive agent that can provide intelligent behavior through a set of quick and simple behaviors. The knowledge of the agent and its behavior is not necessarily represented in logic and the agent does not perform any kind of reasoning.

In Wooldridge reactive architecture (illustrated in Fig. 14) is defined a set of rules for direct mapping of perceptions to actions. Some reactive architectures are organized into layers with different levels of abstraction, where the lower layers have a higher level of priority, i.e., critical actions are performed by these layers. The layers can also be independent, i.e., each layer can process a perception and perform an action. In this case, the actions can be executed in parallel.

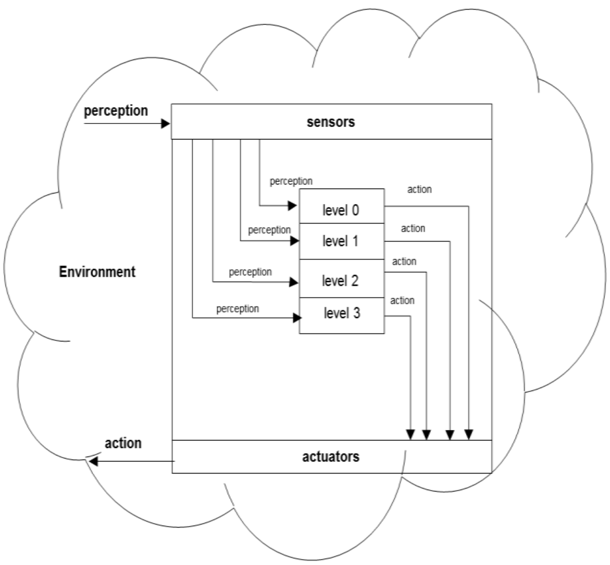


Fig. 14. The structure of a reactive agent (adapted from [21])

C. Kendall Classification

Kendall [3] defined two agent architectures organized into layers: layered agents and reactive agents. A difference this author to others is that it represents the agents through patterns.

1) Reactive agents

Kendall defines the following pattern for structuring a reactive agent (Fig. 15).

Problem: How can an agent react to an environmental stimulus or a request from another agent when there is no symbolic representation and no known solution?

Forces: An agent needs to be able to respond to a stimulus or a request; there may not be a symbolic representation for an application and an application may not have a knowledge based, prescriptive solution.

Solution: A reactive agent does not have any internal symbolic models of its environment; it acts using a stimulus/ response type of behavior. It gathers sensory input, but its belief and reasoning layers are reduced to a set of situated action rules. A single reactive agent is not proactive, but a society of these agents can exhibit such behavior.

Known Uses: Reactive agents have been widely used to simulate the behavior of ant societies and to utilize such societies for search and optimization.

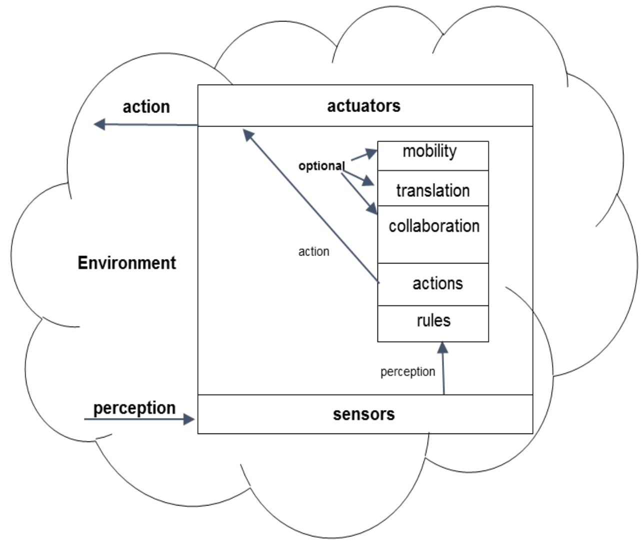


Fig. 15. The reactive agent of the Kendall architecture (adapted from [3])

2) Layered agents

Kendall specified the following pattern for layered agent (illustrated in Fig. 16):

Problem: How can agent behavior be best organized and structured into software? What software architecture best supports the behavior of agents?

Forces: An agent system is complex and spans several levels of abstraction; there are dependencies between neighboring levels, with two way information flow; the software architecture must encompass all aspects of agency; the architecture must be able to address simple and sophisticated agent behavior.

Solution: Agents should be decomposed into layers because i) higher level or more sophisticated behavior depends on lower level capabilities, ii) layers only depend on their neighbors, and iii) there is two way information flow between neighboring layers. The architecture structures an agent into seven layers. The exact number of layers may vary. In the sensory layer the agent perceives its environment through sensors. Agents beliefs are based on sensory input, so, in the beliefs layer, perceptions are mapped to logic sentences that are included in the agent knowledge base, which is part of this layer and where the agent maintains models of its environment and itself. In the Reasoning layer, when presented with a problem, the Agent reasons about the symbolic model in the knowledge base to determine what to do by selecting a particular action to perform on the environment. An agent selects a plan to achieve a goal. When the agent decides on an action, it can carry it out directly, but an action that involves other agents requires collaboration. Once the approach to collaboration is determined, the actual message is formulated and eventually translated into other semantic and delivered to distant societies by mobility. Top-down, distant messages arrive by mobility. An incoming message is translated into the agent's semantics, The collaboration layer determines whether or not the agent should process a message. If the message should be processed, it is passed on to actions. When an action is selected for processing, it is passed to the reasoning layer, if necessary. Once a plan placed in the actions layer, it does not require the services of any lower layers, but it utilize higher ones.

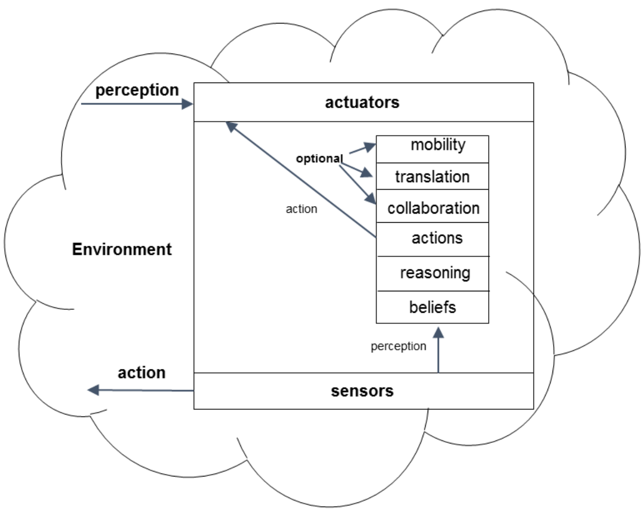


Fig. 16. Layered agent (adapted from [3])

3) *A summary of basic agent architectures*

Table I summarizes different main architecture the architectures of Kendall's discussed in this section.

The simple reflex agent and agent with states defined by Russel and Norvig correspond to Wooldridge reactive agent and the Kendall reactive agent. A main difference is that Wooldridge and Kendall defines that architectures can be arranged in layers with increasing level of abstraction.

The goal-based and utility-based agents defined by Russel and Norvig correspond to the Wooldridge deductive reasoning agent and Kendall layered agent. Russell and Norvig defines a measure of performance and utility. Kendall organizes their deliberative architecture in layers. In general, these architectures are called deliberative architectures and use some kind of reasoning.

Wooldridge defines an architecture called practical reasoning, also neither known as BDI agent, whose main concepts are not approached by Kendall and Russell and Norvig.

TABLE I . CORRESPONDING TERMINOLOGY OF AGENT ARCHITECTURES

Kind of agent	Russel and Norvig	Kendall	Wooldridge
Reactive	Reflex agent	Reactive	Reactive Agent
	Reflex agent with state	Agent	
Deliberative	Goal-based agent	Layered Agent	Deductive Reasoning Agents
	Utility-based agent		Practical Reasoning Agents

III. BASIC AGENT COMPONENTS

An agent has a set of components that are part of its internal architecture, also called agent structure. These components vary according to the agent type.

In this section, basic components of a software agent as reasoning, knowledge base and communication are presented.

A. *The Reasoning Component*

Reasoning is the process of making inferences about a set of assumptions in order to obtain conclusions. There are four main types of reasoning: deduction, induction, abduction and analogy. Deliberative software agents have mechanisms of reasoning and use it to perform more complex actions than reactive ones.

Deduction is the most rigorous kind of reasoning. When deduction premises are true, necessarily also be the conclusion will be also free. For example: "All men are mortal" (premise 1), "Socrates is a man" (premise 2), "Therefore, Aristotle is mortal" (conclusion).

Inductive reasoning part from the observation of objects and of the similarity between its properties. From the observation of the common features of a limited set of objects, an entire category is generalized. For example: "Canaries flies" (Premise 1), "Parrots flies" (Premise 2), "Pigeons flies" (Premise n), "therefore, all birds fly" (Conclusion). Conclusion is not universally true, just likely to be true.

Abductive Reasoning [30] is also called inference by the best explanation. In abduction, as in induction, the conclusion is not an universal truth, but has the probability of being true. Abduction prepares explanatory hypotheses for these observations and hypotheses are evaluated. For example, given the observation that "The street is wet", the following hypothesis can be formulated: "It rained " (hypothesis 1) and "A water truck started pouring water" (hypothesis 2). But when a new observation like "The house roof is wet." Is obtained, the hypothesis 2 should be rejected and hypothesis 1 validated. Abductive reasoning is often used by criminologists, detectives and diagnosis of diseases.

There is another reasoning process called reasoning by analogy, which goes from the particular to the particular. This process occurs with the perception and classification of objects according to similar features and deriving a conclusion from a previous experience in one or more similar situations [18]. For example, consider the set of individuals A, B and C, having the following range of characteristics: white color, thin, blond hair and blue eyes. Given that the individual A has the high stature characteristic, it should be possible to conclude that B and C are likely to have this feature even if not explicit in the definition.

Analogy is a type of reasoning often used in everyday life and has already been successfully applied in various fields of knowledge. For example, the first plans for building vehicles very similar to current helicopters made by Leonardo da Vinci mimicked the mechanisms used by birds to fly [24].

B. The Knowledge Base Component

Another component of a deliberative agent architecture is its knowledge base. The agent knowledge base contains the knowledge of the external environment, the agent perception history and the knowledge of rules for mapping perceptions to actions.

The most common way to represent agent knowledge is symbolic representation, using logic programming languages, like Prolog [17] and Jess [4] and knowledge representation languages supporting semantic networks, frames or ontologies, like RDF [27] and OWL [2]. Ontologies are knowledge representation structures capable of expressing a set of entities in a given domain, their relationships and axioms, being used by modern knowledge-based systems as knowledge bases to represent and share knowledge of a particular application domain. They allow semantic processing of information and a more precise interpretation of data, providing greater effectiveness and usability than traditional information systems [32].

Before being used by an inference engine, perceptions must be transformed into sentences in a knowledge representation language understandable by the agent. After finding, through reasoning, an action representing an appropriate solution for the perception, this action should be interpreted and performed on the environment. For example, in the case of the vacuum cleaner, a sentence that represents the action of moving forward, should rather be interpreted and mapped prior to electronic signals that indicate to the hardware of the vacuum cleaner that he should move. When the agent environment is artificial, like the Internet, the agent perceptions can be just text in natural language or events like a mouse click. Actions can just display text on the screen, send a message or trigger an event like a sound alarm.

It should be distinguished between the agent internal knowledge and the knowledge shared with other agents of the society and / or external entities. The agent internal knowledge is just necessary for performing its own actions. Shared knowledge is required for agents to communicate.

C. The Communication Component

Software agents usually specialize in just certain tasks. Thus, frequently, agents need to communicate to accomplish tasks that are beyond their individual capabilities in order to achieve the overall goal of the multi-agent system. Thus, in these systems, agents need to communicate. Communication among agents is based on the theory of speech acts [37] and for that, Agent Communication Languages (ACLs) have been developed, like KQML (Knowledge Query and Manipulation Language)[7][41] and FIPA-ACL (Foundation for Intelligent Physical Agents - Agent Communication Language) [16] for expressing communication acts and supporting the coordination and cooperation mechanisms of a multi-agent society architecture.

. In the theory of speech acts, a message intent is called performative. For example, the intention of an agent may be a request such as "Can you send me the price of the blue blouse?" or just information like "The summer time in Brazil began on October 20, 2013." According to their main intentions a speech act can inform, question, answer, ask, offer, confirm and share.

IV. ADVANCED AGENT ARCHITECTURES

Basic agent architectures presented in the previous sections already have a high level of maturity, having techniques [11][13][33][34] and frameworks [6][14][15] that support their development. More advanced agent architectures such as learning and hybrid ones are still an open research topic. This section discusses these more advanced software agent architectures.

A. Learning Agent Architectures

The idea behind learning is that perceptions should be used not only to act but also to improve the agent ability to act in the future [38].

Basic software agents have no learning; they act according to the perceptions defined in the agent design. Therefore, for new perceptions the agent must be reprogrammed.

Learning agents can at runtime change their behavior according to changes in the environment. In this type of agents, perceptions should be used not only to act but also to improve the agent ability to act in the future.

According to Russell and Norvig [38], a learning agent has four basic components (Fig. 17): performance, critic, learning and problem generator.

The performance component is what we have previously considered to be a basic agent: perceives and acts on the environment

The learning component is responsible for making the agent behavior improvements. It uses feedback from the critic on how the agent is doing and determines how the performance component should be modified to do better in the future. The critic tells the learning component about the success of the agent according to a fixed performance standard. The critic is necessary because the percepts themselves provide no indication of the agent success.

The last component of the learning agent is the problem generator which is responsib1e for suggesting actions that will lead to new and informative experiences. The point is that if the performance element had its way, it would keep doing the actions that are best, given what it knows. The problem generator main goal is to suggest these exploratory actions. This is what scientists do when they carry out experiments.

An example of the functioning of a learning agent architecture, the automated taxi, is given in [38]. The performance element consists of whatever collection of knowledge and procedures the taxi has for selecting its driving actions. The critic observes the world and passes information along to the learning element. For example, after the taxi makes a quick left turn across three lanes of traffic, the critic observes the shocking language used by others drivers. From experience, the learning element is able to formulate a rule saying this was a bad action, and the performance element is modified by installation of the new rule. The problem generator might identify certain areas of behavior in need of improvement and suggest experiments, such as trying out the brakes on different road surfaces under different conditions.

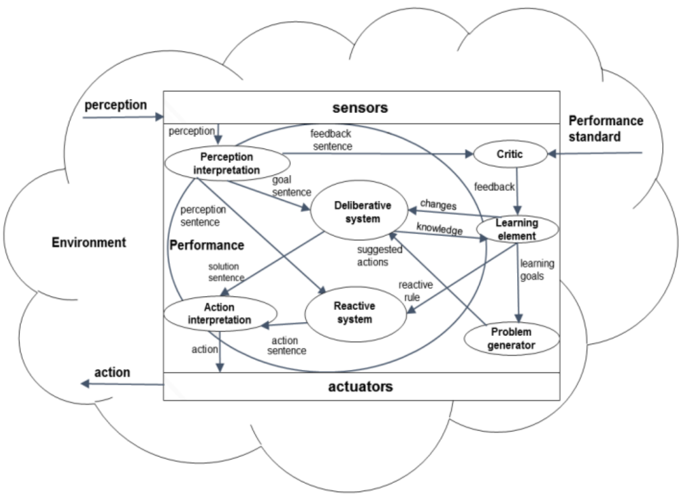


Fig. 17. The structure of a learning agent (adapted from [38])

During the construction of the learning element it is necessary to define the learning technique to be used. Well-known techniques of machine learning are supervised learning, unsupervised learning and reinforcement learning [36].

The problem of supervised learning involves learning a function from a set of inputs and outputs examples which will be later used to produce the correct output given a new input. Examples of these techniques are decision trees, Bayesian networks and neural networks [38]. Supervised learning is associated with two common problems which are classification and regression. Classification is to assign an instance to a class through a classifier previously built. A classifier can be, for example, a decision tree or set of rules. Regression attempts to identify a output and represent it by a numeric value from a set of training data.

A disadvantage of a supervised learning technique is that learning is dependent on the training examples and creating these training examples may require considerable time and effort.

The problem of unsupervised learning involves learning patterns in the input and building a model or useful representations of the data, for example, clusters when no specific output values are supplied. For example, a taxi agent might gradually develop a concept of “good traffic days” and “bad traffic days” without ever being given labelled examples of each [38]. This could be done through the development of clusters.

A cluster is a collection of objects that are similar to each other (according to some pre-defined similarity criterion) and not similar to objects belonging to other clusters. So it could be constructed clusters representing "good traffic days" and "bad traffic days". These clusters could have characteristic values as "time spent route" and "fuel cost". After clusters construction, the agent could learn which days probably has good traffic. Clusters do not always have adequate exits after completion of the training dataset. When this happens, it is necessary to evaluate why the output is not appropriate. Some data output is not correct because there were used insufficient training examples or there were defined non-relevant features for the objects that compose the clusters.

Reinforcement learning is inspired in the behaviorist psychology where an agent learns to act in a way that maximizes rewards in the long term. Reinforcements are obtained through the interaction of the agent with the environment and can be positive (reward) or negative (punishment). In reinforcement learning there is no examples of correct output. The reinforcement obtained through interaction with the environment is used to assess the agent behavior and is associated with a performance standard establishing whether that reinforcement is positive or negative. Better agent performance is obtained through experience.

A reinforcement learning example is of a mouse that moves about a maze trying to get the cheese while avoiding the deadly trap. The mouse does not know beforehand the layout of the maze or the placing of the cheese/trap. At each square, it must choose whether to move up, down, left or right. The mouse will explore the maze to find the cheese. After finding the cheese, it will already know which path took to find the cheese (positive reinforcement), but it still tries to find a shorter way to the cheese (to maximize its performance measure). After various experiences in the maze it will learn the shortest path from the starting point until the cheese.

B. Hybrid Agent Architectures

Hybrid architectures, also known as layered architectures have emerged from the need to gather in a single agent reactive and deliberative behavior. Wooldridge classifies this type of architecture into two groups [21]: hybrid architectures in horizontal layers and hybrid architectures in vertical layers.

In horizontal layers hybrid architectures (Fig. 18), each software layer is connected directly to a sensor and an actuator. In this type of architecture each layer works as an independent agent. One advantage of organizing horizontally layered architectures is the clear separation of the different agent behaviors within the architecture, where each layer can act independently of the other, even in parallel. One of the problems with this architecture type is that the overall behavior of the agent cannot be consistent. To solve this, usually a control layer is also designed to ensure a coherent overall behavior.

In vertical layers hybrid agent architectures, perceptions and actions are handled by more than one layer. In these architectures, layers can be further subdivided into a single-passage (Fig. 19) and two-passages (Fig. 20) architectures. In the architecture of a single-passage, the control flow passes sequentially through each layer until reaching the final layer where the action to be performed in the environment is generated. In the two-passages architecture, information flows to reach the final layer (first pass) and control then flows back down (second pass).

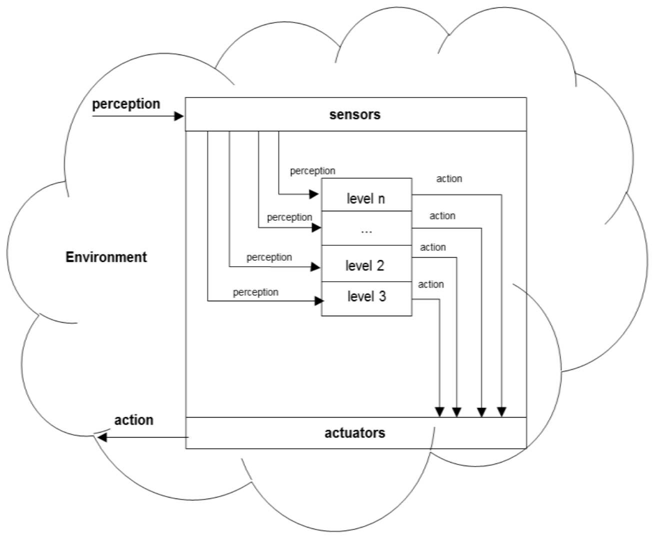


Fig. 18. A horizontal layered agent architecture (adapted from [21])

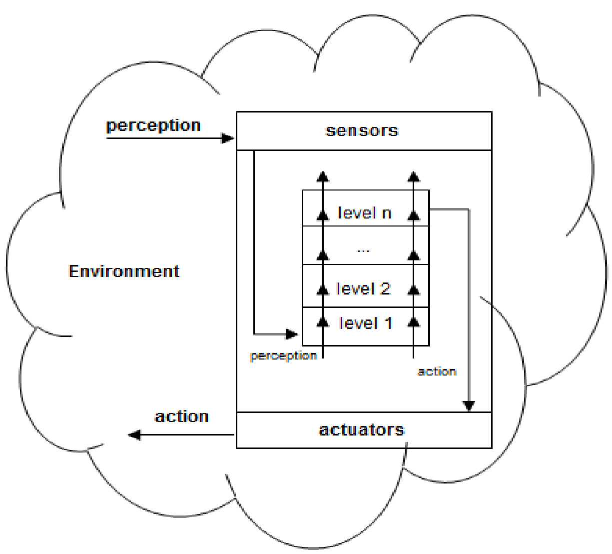


Fig. 19. A vertical layered agent architecture of a single-passage (adapted from [21])

Russell and Norvig [38] also propose a hybrid architecture (Fig. 21). The agent architecture has two main components they called subsystems: a deliberative and a reflex systems. The goal of the architecture is to exhibit more efficient agent behavior by converting deliberative decisions into reflective ones, making the agent actions more fast and efficient.

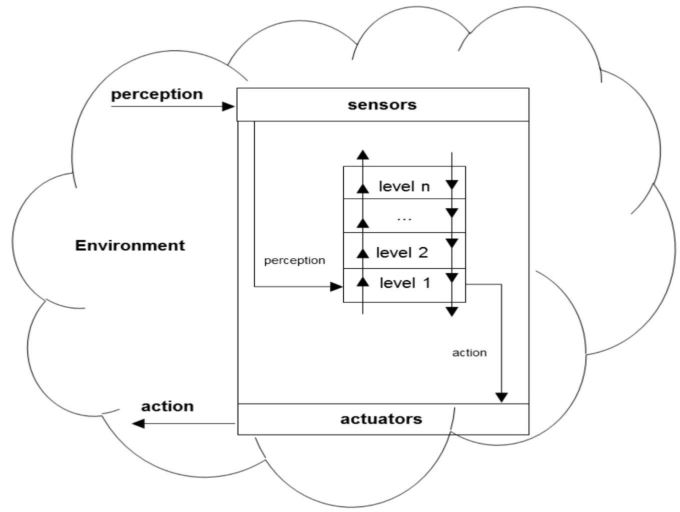


Fig. 20. A vertical layered agent architecture of two-passages (adapted from [21])

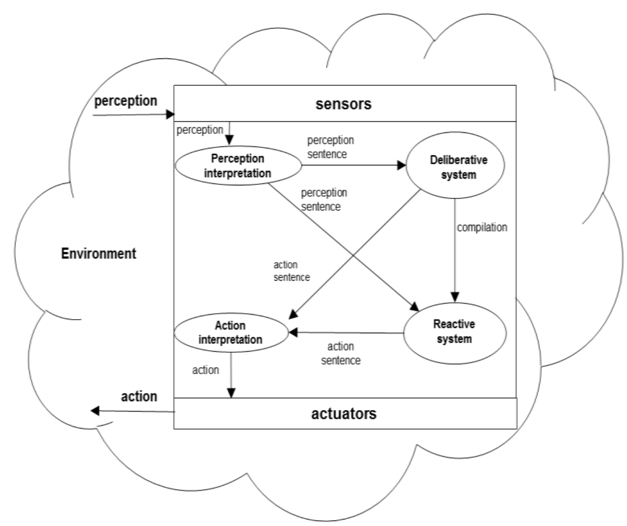


Fig. 21. Another proposal of a hybrid agent architecture (adapted from [38])

Examples of current hybrid architectures are SOAR [5][10], ACT-R (Adaptive Control of Thought–Rational)[9], INTERRAP (Integration of Reactive Behavior and Rational Planning) [29]. A common feature of most proposal of current hybrid architectures is that they are cognitive. A cognitive architecture aims at developing agents capable of acting using human cognitive phenomena such as memory, learning, decision making and natural language processing [28].

SOAR was one of the first proposals of hybrid architectures. It has a development environment and a framework to support the creation of agents according to their definitions. It is also a goal-driven cognitive architecture that integrates reasoning, reactive execution, planning and various learning techniques, aiming at creating a software system having similar cognitive abilities as humans. The SOAR agent architecture is illustrated in Fig. 22. This architecture is composed of a working memory, also known as short term memory, a long-term memory, a reasoning module, modules responsible for managing the agent perceptions and actions and learning modules.

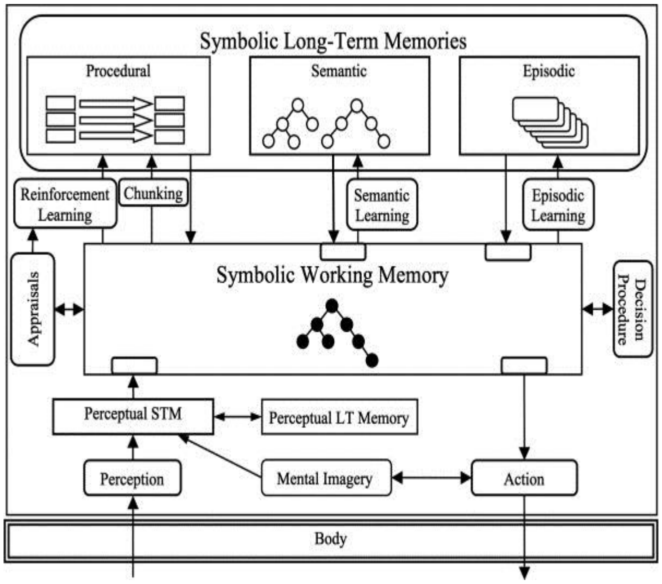


Fig. 22. The SOAR hybrid agent architecture [5]

ACT-R is a programming environment that supports the development of hybrid agents. The ACT-R hybrid architecture (Fig. 23) attempts to emulate cognitive processes of human cognition such as knowledge acquisition and learning through a production system. It consists of a perceptual/motor layer, a cognition layer and a buffer intermediate layer. In the perceptual/motor layer, input and output information processing is emulated through the human visual, motor, speech and hearing modules. In the cognition layer, the memory of the agent is represented by declarative and production modules. The declarative memory corresponds to the reactive part of the system and consists of "chunks". A chunk is an attribute-value structure; with a special type of attribute called "ISA" that determines the type of the chunk. The production memory is formed by condition-action rules and an inference engine to select actions in the knowledge base. The representation of the two types of memories for the agent is what makes ACT-R a hybrid architecture. The intermediate module ACT-R buffers is the working memory of the agent. It corresponds to the knowledge used only when performing a particular task. This knowledge can be retrieved both by the declarative memory and the production memory. Learning in the ACT-R architecture is by "chunking" which basically consists in useful pieces of information (chunks) that are stored in the declarative memory for future use.

The InteRRaP architecture (Fig. 24) is a two-passages hybrid architecture that supports the development of reactive agents and goal-based agents. It is organized into layers together with a control structure and a knowledge base associated with each layer. It consists of five main components: an interface with the world (WIF), a component-based behavior (BBC), a plan-based component (PBC), a cooperation component (CC) and the agent knowledge base. The WIF component enables perception, action and agent communication. The BBC component supports reactive behavior and represents procedural knowledge. The PBC component contains planning mechanisms for constructing agent plans. The CC component contains a mechanism to compose these plans. The knowledge base of InteRRaP consists of three layers. The lowest layer contains facts that represent a model of the agent environment as well as representations of actions and behavior patterns. The second layer contains the agent mental model. The third layer consists of the agent social model, which provides strategies for cooperation with other agents.

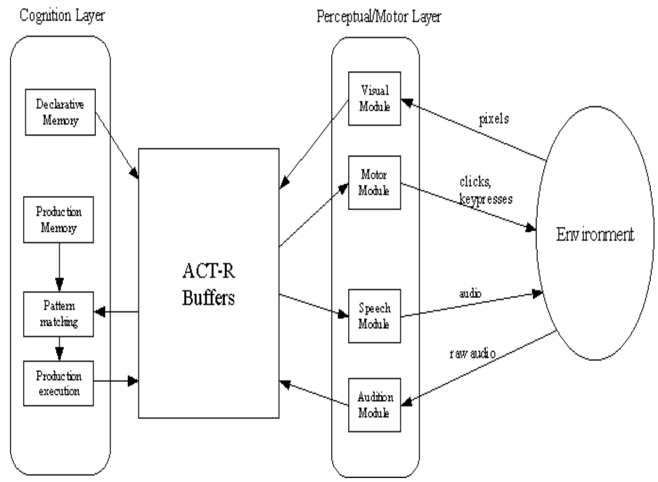


Fig. 23. The ACT-R hybrid agent architecture [26]

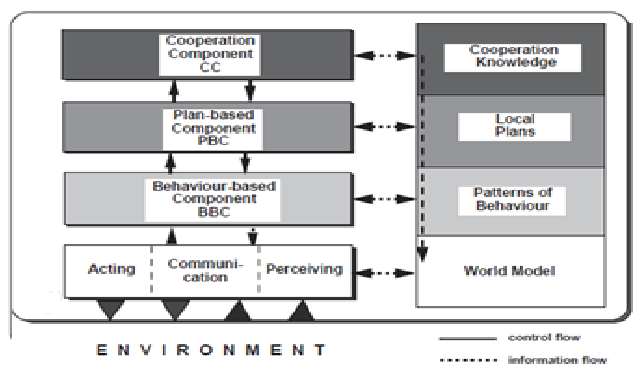


Fig. 24. The InteRRaP hybrid agent architecture [29]

Qinzhou and Lei [12] define a hybrid agent architecture using case-based reasoning and unsupervised learning (Fig. 25) composed of eight modules: a knowledge module in charge of storing a set of cases and a set of rules; a module responsible for the condition-action rules corresponding to the reactive behavior; a perception module responsible for receiving information from the environment, a learning module that uses unsupervised learning algorithms to increase the efficiency of case retrieval; a retrieval module whose main function is to compare and perform the similarity computation between a new case with an old case in the case base; a decision module which corresponds to the deliberative behavior, responsible for performing a process of reasoning on cases using the set of rules from the knowledge module; an execution module responsible for performing actions on the environment; and a communication module responsible for the interactions between agents, which uses the KQML[7].

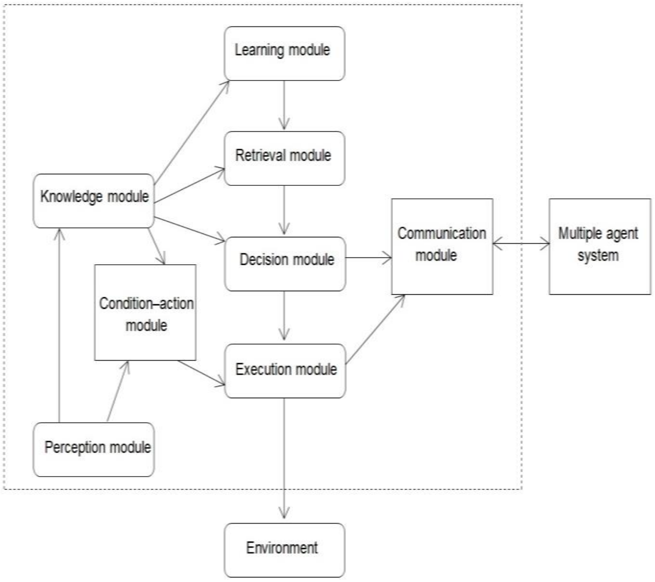


Fig. 25. The Qinzhou and Lei hybrid agent architecture [25]

Table II shows a comparison between the SOAR, ACT-R, INTERRAP and Qinzhou and Lei hybrid architectures by considering the characteristics of their reactive and deliberative components, the applied learning technique and the representation mechanisms of the knowledge base.

Among these architectures, and probably among all hybrid architectures of the state of the art, the broader one is the SOAR architecture which includes various forms of learning and mechanisms for representing the agent knowledge base. SOAR also provides tools for developing agents, updated documentation and examples of implemented agents for various problems.

Table II. A Comparison of Hybrid Agent Architectures

Features / **Architecture**	**Reactive Component**	**Deliberative Component**	**Learning Technique**	**Knowledge base representation**
SOAR	Reactive with state	Deductive	Reinforcement, episodic, chunking and semantic	Condition-action rules, state graphs and semantic memory.
ACT-R	Reactive with state	Deductive	Chunking	Rules, facts and procedural knowledge
INTERRAP	Reactive with state	BDI	No learning	Procedural knowledge
Qinzhou and Lei Architecture	Reactive simple	RBC	Unsupervised learning	Cases

The ACT-R architecture differs from the SOAR architecture because it is more restricted in relation to the alternative representation of the agent knowledge base. However, this architecture has a wider treatment of perceptions, including representing images and voice.

INTERRAP's main advantage over other hybrid architectures presented in this section, is its layered organization, considering that the definition of several independent components inserts complexity in the architecture design, making necessary to manage the interactions between the components.

The architecture of Qinzhou and Lei differs from the others by using case-based reasoning and unsupervised learning to classify similar cases in a class according to the most relevant characteristics of the cases.

V. CASE STUDY

OHAA ("Ontology-driven Hybrid Agent Architectures") is an ontology-driven hybrid and learning agent architecture that combine deliberative and reactive components joining the advantages of both behaviors to improve the decision making process. Thus, the agent may have a reactive behavior or a deliberative one depending on its perception and available behavior.

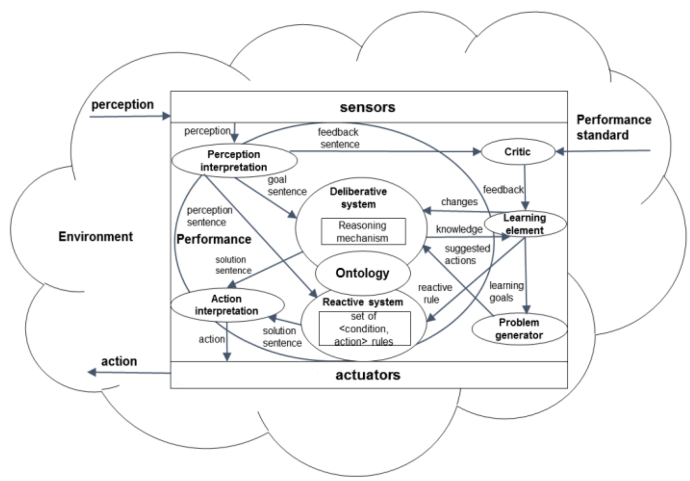


Fig. 26. The structure of OHAA Architecture

Additionally, the architecture allows learning new reactive rules through recurrent solutions to the same perception from the deliberative system, which will be stored in the agent knowledge base. Also, the learning component supports the evolution of deliberative to reactive behavior. Finally, in OHAA, the knowledge base is represented as an ontology thus enabling knowledge improvement through reuse. Fig. 26 illustrates the OHAA basic structure.

A first approach of the OHAA functioning is described as follows.

1. Interpreting the perception;
2. Mapping the perception to a sentence;
3. Asserting the perception sentence in the knowledge base ontology;
4. If there is a rule for the perception, the corresponding

action to perform is selected by the reactive system;
5. If there is no a reflex action corresponding to the perception, this will be treated by the deliberative system that will reason to find the most appropriate action;
6. Upon completion of the action, the agent will perceive a feedback from the environment about the success or failure of the performed action;
7. In the critical component, which is fed by a performance standard of the agent actions from the environment, the feedback perception will be evaluated. If the action was poorly evaluated by the critical, this component informs the learning component;
8. If the action was assessed as good, this behavior will remain in the knowledge base;
9. Following, the learning component makes recommendations for improvements in the actions of the agent;
10. These recommendations are passed to the problem generator component which, in turn can generate a new set of possible actions for the agent;
11. When the agent performs these new actions, suggested by the problem generator component, it will have a feedback perception and the process restarts;
12. Finally, actions repeatedly well evaluated will be transformed into reactive rules;
13. When the agent performs these new actions, suggested by the problem generator component, it will have a feedback perception and the process restarts;
14. Finally, actions repeatedly well evaluated will be transformed by the learning component into reactive rules.

A. A simples example

Consider a genealogy tree as the OHAA environment (Fig. 27). By traversing the tree the agent just perceives who are the parents of a given person.

OHAA also has knowledge about genealogy (Fig. 28) so it can conclude through deductive reasoning who are the kins of a given person. For example, through the knowledge base inference rules of the deliberative system of Fig. 28, the agent can conclude that Bob and Julie are cousins.

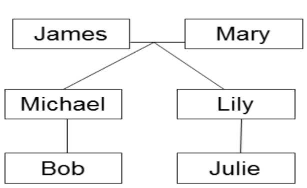


Fig. 27. Genealogy Environment

If the action generated through the inference that conclude that “Bob and Julie are cousins” is repeatedly well evaluated then it could be transformed by the learning component into a reactive rule. When the rule becomes reactive, the agent does not need to reason once the environment is static. Then, the knowledge base of the agent will be updated with the information that “Bob and Julie are cousins”. In the knowledge base (Fig. 29) this reactive rule is represented by the “cousins (bob, julie).” fact.

Deliberative system
parent(X,Y) :- father(X,Y).
parent(X, Y) :- mother(X,Y).
brothers(X,Y):- parent(X,Z),parent(Y,Z).
cousins(X,Y):-brothers(A,B),parent(X,A),parent(Y,B).

Reactive system
father(Michael,James).
father(Lily,James).
father(Bob,Michael).
mother(Michael,Mary).
mother(Lily,Mary).
mother(Julie,Lily).

Fig. 28. Facts and inference rules in the reactive and deliberative systems of the OHAA knowledge base

Deliberative system
parent(X,Y) :- father(X,Y).
parent(X,Y) :- mother(X,Y).
brothers(X,Y) :- parent(X,Z),parent(Y,Z).
cousins(X,Y) :- brothers(A,B),parent(X,A),parent(Y,B).

Reactive system
father(michael,james).
father(lily,james).
father(bob,michael).
mother(michael,mary).
mother(lily,mary).
mother(julie,lily).
cousins(bob,julie).

Fig. 29. The representation of a learned reactive rule in the OHAA knowledge base

VI. CONCLUDING REMARKS

This paper presented a study about basic and advanced software agent architectures. The main features of reactive and deliberative agent architectures, their internal components and how they relate to each other were described. Hybrid architectures which combine both reactive and deliberative agent behavior and learning architectures allowing the improvement of the agent behavior were also analyzed.

Considering that they simulate better intelligent human behavior, current work focuses on the design of hybrid and learning architectures, providing frameworks and design tools for agent construction.

Additionally, the OHAA hybrid architecture has also been introduced. The OHAA Architecture combines deliberative and reactive components joining the advantages of both behaviors to improve the decision making process. A learning component also was defined in OHAA, responsible for learning new agent behaviors and for transforming deliberative behaviors into reactive ones.

An example of OHAA utilization and a comparative study of main approaches of hybrid agent architectures have been also discussed.

The hybrid architecture OHAA is still in an early stage. Current work looks for detailing the architecture components and evaluating its effectiveness through the design and implementation of an initial prototype and the development of a case study in the family law legal field using case-based reasoning [1] and instance-based learning [24].

Further work will specify a technique and implementing a tool for constructing agents using the OHAA architecture. More evaluation experiments will be conducted using deductive reasoning and reinforcement learning [40].

REFERENCES

[1] A. Aamodt, E. Plaza, "Case-Based Reasoning: Foundation Issues, Methodological Variations, and System Approaches", *AICOM*, Vol. 7: 1,1994, pp. 39-59.

[2] B Motik, "OWL 2 web ontology language: Structural specification and functional-style syntax", *W3C recommendation*, v. 27, 2009

[3] E. A. Kendall, P. V. Krishna, C.V. Pathak, C. B. Suresh, "Patterns of intelligent and mobile agents". In *Proceedings of the second international conference on Autonomous agents*, 1998, pp. 92-99.

[4] E. Friedman-Hill, "*JESS in Action. Greenwich*", CT: Manning, 2003.

[5] E. Laird, "*The Soar cognitive architecture*", MIT Press, 2012.

[6] F. Bellifemine, G. Caire, D. Greenwood, "*Developing Multi-Agent Systems with JADE*", Wiley, 2007, ISBN: 978-0-470-05747-6.

[7] F. Tim, Y. Labrou, J. Mayfield, "*KQML as an agent communication language*". Computer Science Department. University of Maryland Baltimore County. Baltimore, USA, 1993.

[8] G. Weiss, "Multiagent Systems Intelligent" In: *Robotics and Autonomous Agents series*, The MIT Press, 2013.

[9] J. Ball, "Explorations in ACT-R Based Cognitive Modeling-Chunks, Inheritance, Production Matching and Memory in Language Analysis," In: *Proceedings of the AAAI Fall Symposium: Advances in Cognitive Systems*, 2011, pp. 10-17.

[10] J. Laird, "*The Soar 9 Tutorial*", University of Michigan, 2012.

[11] J. Mylopoulos, J. Castro, M. Kolp, "The Evolution of Tropos". In: *Seminal Contributions to Information Systems Engineering*. Springer Berlin Heidelberg, 2013, p. 281-287.

[12] K. Riesbeck, C. Schank, "*Inside Case-based Reasoning*", Psychology Press, 2013.

[13] K. Vivekanandan, D. RAMA, "Analysing the Scope for Testing in PASSI Methodology". *International Journal*, v. 3, n. 1, 2013.

[14] L. Braubach, A. Pokahr, "Jadex Active Components Framework-BDI Agents for Disaster Rescue Coordination", In: *Software Agents, Agent Systems and Their Applications,volume 32 of NATO Science for Peace and Security Series*, IOS Press, 2012.

[15] L. Braubach, A., Pokahr, "Developing Distributed Systems with Active Components and Jadex", *Scalable Computing: Practice and Experience*, v. 13, n. 2, 2012.

[16] L. Shujun, M. Kokar, "*Agent Communication Language. Flexible Adaptation in Cognitive Radios*", Springer New York, 2013, pp. 37-44.

[17] L. Sterling, E. Shapiro, "*The Art of Prolog: Advanced Programming Techniques*", 2nd Edition, MIT Press, 1994, 688 pages, ISBN 0-262-19338-8.

[18] L.W. Lloyd. "*Legal Reason: The Use of Analogy in Legal Argument*", New York: Cambridge University Press, 2005.

[19] M. Wooldridge, P. Ciancarini. "Agent- oriented Software Engineering: The State of the Art", In: *"Agent-Oriented Software Engineering"*, Springer-Verlag, Lecture Notes in AI Volume 1957, 2000.

[20] M. Shaw, Garlan, D. "*Software architecture: perspectives on an emerging discipline*", Prentice Hall, 1996.

[21] M. Wooldridge, "*An Introduction to Multiagent Systems*", 2nd ed.,Wiley Publishing, 2009.

[22] M. Wooldridge, "Intelligent Agents", In: *"Multi-agent Systems – A Modern Approach to Distributed Artificial Intelligence"*, G. Weiss (ed.), The MIT Press, 1999.

[23] N. R. Jennings, "On Agent-based Software Engineering", *Artificial Intelligence Journal*, 2000, pp. 277-296.

[24] N. Jiang et al., "Application of bionic design in product form design", In: *Computer-Aided Industrial Design & Conceptual Design (CAIDCD)*, 2010, pp. 431-434.

[25] N. Qinzhou, H. Lei, "Design of case-based hybrid agent structure for machine tools of intelligent design system", In: *2012 Software Engineering and Service Science (ICSESS 2012)*, pp.59-62, 2012.

[26] N. Taatgen, A. Niels, C. Lebiere, J. R. Anderson, "Modeling paradigms in ACT-R", *Cognition and multi-agent interaction: From cognitive modeling to social simulation*, 2006, pp. 29-52.

[27] O. Lassila, R. Swick, "RDF/XML syntax specification". *W3C Recommendation*, v. 22, 2010.

[28] P. Langley, J. Laird, J. Rogers, "Cognitive architectures: Research issues and challenges", In: *Cognitive Systems Research*, v. 10, n. 2, p. 141-160, 2009.

[29] P. Müller, M. Pischel, "*The agent architecture InteRRaP: Concept and application*" 2011.

[30] P. Thagard, E.C. SHELLEY, "Abductive Reasoning: Logic, Visual Thinking, and Coherence", In: *M.L*, 1997.

[31] R. Girardi, "An Analysis of the Contributions of the Agent Paradigm for the Development of Complex Systems", In: *Joint meeting of the 5th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2001) and the 7th International Conference on Information Systems Analysis and Synthesis (ISAS 2001)*, 2001, pp. 388-393.

[32] R. Girardi, "Guiding Ontology Learning and Population by Knowledge System Goals", In: *Proceedings of the International Conference on Knowledge Engineering and Ontology Development*, Ed. INSTIIC, Valence, 2010, p. 480 – 484.

[33] R. Girardi, A. Leite, "Knowledge Engineering Support for Agent-Oriented Software Reuse", In: *M. Ramachandran. (Org.). Knowledge Engineering for Software Development Life Cycles: Support Technologies and Applications*. Hershey: IGI Global, v. I, p. 177-195, 2011.

[34] R. Girardi, A. Leite, "The Specification of Requirements in the MADAE-Pro Software Process", *iSys: Revista Brasileira de Sistemas de Informação*, v. 3, p. 3, 2010.

[35] R. Johnson, R. Helm, J. Vlissides, "*Design Patterns: Elements of Reusable Object-Oriented Software*". Addison-Wesley Professional, 1994.

[36] R. Sutton, R. Richard, A. Barto, G. Andrew, *"Reinforcement learning: An introduction"*, Cambridge: MIT press, 1998.

[37] S. Kuroda, "*A formal theory of speech acts. Linguistics and philosophy*", v. 9, n. 4, p. 495-524, 1986.

[38] S. Russel, P. Norvig, "*Artificial Intelligence: A Modern Approach* ", 3nd ed., Prentice-Hall, 2009.

[39] T. Gruber, "Toward principles for the design of ontologies used for knowledge sharing", *International Journal of Human-Computer Studies*, v. 43, n. 5, p. 907-928, 1995.

[40] T. Mitchell, "*Machine Learning*", McGraw-Hill, Book, USA, 1997.

[41] X. Wu, J. Sun, "Study on a KQML-based intelligent multi-agent system" In: *Intelligent Computation Technology and Automation (ICICTA)*, 2010 International Conference on. IEEE, 2010. p. 466-469.

Ontology for Multimedia Applications

Hiranmay Ghosh, *Senior Member, IEEE,* Santanu Chaudhury and Anupama Mallik

***Abstract*—This paper provides an overview of the contents of a tutorial on the subject by one of the authors at WI-2013 Conference. The domination of multimedia contents on the web in recent times has motivated research in their semantic analysis. This tutorial aims to provide a critical overview of the technology, and focuses on application of ontologies for multimedia applications. It establishes the need for a fundamentally different approach for a representation and reasoning scheme with ontologies for semantic interpretation of multimedia contents. It introduces a new ontology representation scheme that enables reasoning with uncertain media properties of concepts in a domain context and a language "Multimedia Web Ontology Language" (MOWL) to support the representation scheme. We discuss the approaches to semantic modeling and ontology learning with specific reference to the probabilistic framework of MOWL. We present a couple of illustrative application examples. Further, we discuss the issues of distributed multimedia information systems and how the new ontology representation scheme can create semantic interoperability across heterogeneous multimedia data sources.**

***Index Terms*—Multimedia, Ontology, Learning, Semantic Modeling, MOWL, Abductive Reasoning, Distributed Systems**

I. Introduction

Use of multimedia data on the web has surpassed that of textual data in the recent times. According to a recent survey [1], 300 million photos are uploaded on the *Facebook* every day and 4 billion hours of video have been watched on *Youtube* per month during the year 2012. These numbers do not include the growing volume of media data generated by surveillance cameras, TV broadcasting stations round the world, satellites, medical imaging devices, document scanners and other digitization initiatives, such as cultural heritage preservation.

The phenomenal rise in consumption of audio-visual data has led to research interest in their semantic processing. Some application examples include creation of personal photobooks [2], [3], news aggregation from multiple sources [4], [5] and digital preservation of cultural heritage [6], [7]. This paper intends to present an insight into the challenges in large-scale semantic processing of multimedia data and the approaches to resolve them. As the media content processing technology advances through content-based, concept-based and ontology-based solutions, the specific requirements for knowledge representation scheme for multimedia applications have been discovered. We present a new multimedia ontology representation scheme [8] that addresses these needs. We show that this new scheme can cope up with the challenges of semantic modeling of multimedia data in different contexts. Learning ontology from real-life data is yet another challenge that is dealt with in this paper with a Bayesian learning framework. Further, we illustrate the effectiveness of the new ontology representation scheme with a couple of illustrative application examples. A major motivation for explicit knowledge representation is integration of information from multiple information sources. We discuss how the new ontology representation scheme is more effective in achieving semantic interoperability across heterogeneous multimedia data sources than the existing approaches.

Hiranmay Ghosh is a Principal Scientist with Innovation Labs, TATA Consultancy Services Limited and is an Adjunct Faculty with Electrical Engineering Department, Indian Institute of Technology, Delhi.
E-mail: hiranmay@ieee.org
Santanu Chaudhury is a Professor with Electrical Engineering Department, Indian Institute of Technology, Delhi
E-mail: santanuc@ee.iitd.ac.in
Anupama Mallik is a Project Scientist with Electrical Engineering Department, Indian Institute of Technology, Delhi
E-mail: ansimal@gmail.com

II. Semantic Web and Ontology

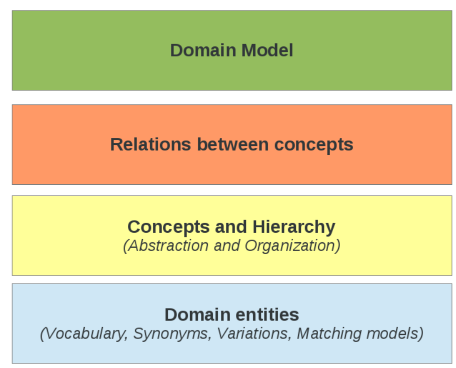


Fig. 1. Layers of abstraction in ontology

The architecture of Semantic Web [9] envisions a world where machines can semantically analyze the data on the web, enhancing the scope of human machine collaboration in specific application contexts. The architecture is based on a *syntactic* layer, where XML is used for describing the data in a uniform way, and a *semantic* layer which relates data items from multiple sources to establish their meanings. An *ontology* that represents an abstract model of a domain, is an essential ingredient of the semantic layer. In context of Information Science, the term "ontology" connotes *formal representation* of knowledge of an *abstraction* of a domain [10]. An ontology defines the "concepts" dealt with in a domain, and establishes their "properties" in context of that domain. Figure 1 depicts the layers of abstraction represented by an ontology. The lowest layer defines the domain entities, i.e. the vocabulary with synonyms, language variations (e.g. "car" or "voiture"), and the matching rules (e.g. use of word-root). The next higher layer brings in abstraction, where concepts are defined and organized into hierarchies. Further up, the properties of the concepts and their mutual relations are defined and the domain model evolves.

Explicit representation of domain knowledge results in its separation from the program logic. The advantages include generalization and reuse of software agents in multiple domain contexts, convenient knowledge engineering and easier maintenance of knowledge-based applications. The formal specification of domain knowledge enables reasoning with them and discovery of new facts. The relations in an ontology represent rules that can be expressed as First-Order-Logic (FOL). Description Logics (DL) proves to be a convenient tool for logical deductions with such rules. Several techniques for formal knowledge representation had been proposed during the previous decades [11]. W3C has standardized the *Web Ontology Language (OWL)* [12] as the language for ontology representation for semantic interoperability of data on the web in 2004.

In principle, the *concepts* in a domain represent abstract entities and transcended any form of their expressions. But, for any practical use, they need to be represented with some means of communication. Since text is the symbolic representation of human experience and is closest to the abstract model of the world, linguistic constructs (mostly, nouns, verbs and phrases) are used to express the domain model in an ontology. The use of linguistic constructs in representing ontology makes them readily suitable for interpreting text documents in a domain context. Typical uses of ontology in text retrieval and information extraction include query expansion using synonyms, hyponyms (sub-concepts) and hypernyms (super-concepts), creating templates for information extraction, identification of associated concept instances in text documents and resaoning with the discovered facts to find new facts, not explicitly available in the documents.

III. Evolution of multimedia content processing

Multimedia content processing started with content based retrieval systems [13], [14] in early 1990's. These systems provided a *query-by-example* interface and used low level image features, e.g. color and shape, to establish similarity between the query and the database images. It was soon understood that the media features do not represent the semantic contents of the images. The phenomenon is referred to as the *"semantic gap"* in the literature. Several knowledge-based methods have evolved [15] to address this issue. The methods generally involve supervised and unsupervised learning techniques with global or local features. A *bag-of-words* approach [16] creates a "visual vocabulary", when classical information retrieval algorithms can be applied with the "visual words" discovered in a media artifact. Higher level image semantics have been discovered with structural models, e.g. a "beach scene" comprises "sky" at the top and "water" and "sand" below, each of which is characterized by some media features [17]. On the other hand, establishment of the context, e.g. a beach scene, enhances the recognition of constituent objects with similar media features, such as the water and the sky. A part-based human action recognition scheme that exploits context information has been proposed in [18]. Most of the proposed systems attempt to solve domain-specific media interpretation problems with implicit domain knowledge. "Open systems" generally rely on relevance feedback and user profiling data to personalize and to improve on the results.

IV. Ontology for Multimedia Data Interpretation

Incorporation of implicit domain knowledge in multimedia systems and resulting diversity in interpretations hinder semantic integration of information from multiple repositories. With the developments in semantic web technologies, ontology was used to interpret metadata, either manually created or machine generated, in an attempt to achieve semantic interoperability of multimedia artifacts from multiple collections [19]. A logical next step was to extend the ontology with symbolic media properties of the concepts, e.g. a set of color values like "red", "blue", etc. Qualitative relations were established between these media properties, e.g. red is *opposite to* green, but is *close to* brown [20]. Such symbolic property attributions provided limited capability to reason with media properties with concepts. These systems relied on commonality of media annotations, which were available in well-curated media collections in specific domains, e.g. a federation of collaborating museums. Uncontrolled media collections, e.g. those on social networks, do not comply with such requirements. The widespread use of social networks for information sharing has triggered interest in deriving semantics out of crowd-sourced annotations and knowledge organizations [21].

While initial work in creating such ontologies used ad-hoc description schemes, development of MPEG-7 standard [22] provided a mechanism for syntactic compatibility in multimedia content descriptions and motivated creation of ontologies linked to MPEG-7. Since MPEG-7 allows for arbitrary semantic descriptors, a comprehensive visual concept ontology has been proposed in [23] to standardize the vocabulary. To overcome the lack of semantics of XML based MPEG-7 MDS, several research groups created ontologies to formalize the meaning of the multimedia content descriptors. While the different ontologies differed in their coverage and their mode of creation, they can be broadly classified into two classes [24]. Some of the ontologies, e.g. [25], extend themselves to the semantic descriptors of MPEG-7, thereby creating a complete semantic and media based description of multimedia artifacts and collections. This approach poses a challenge for aligning the ontological descriptions for diverse and independently developed repositories. The other MPEG-7 ontologies, e.g. [26], [27], [28], do not include semantic descriptors but focus on media based structural descriptions of the contents. They interoperate with external domain ontologies. This approach has the benefit of using a common domain ontology to interpret media based descriptions of the contents from diverse independent repositories. An architecture for ontology based multimedia data fusion is shown in figure 2.

The approaches for multimedia ontologies described so far create semantic models of repository contents using their MPEG-7 descriptions, but do not attempt to produce a collection independent domain model incorporating multimedia attributes. Another problem with these approaches is the use

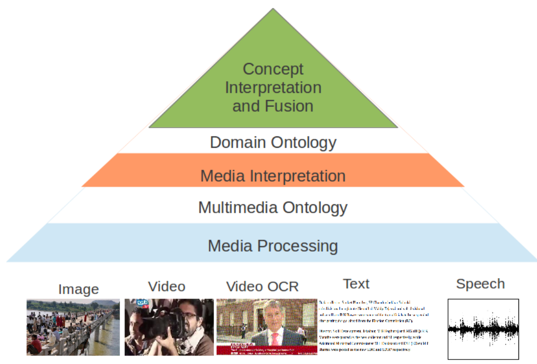


Fig. 2. Ontology based fusion of multimodal data

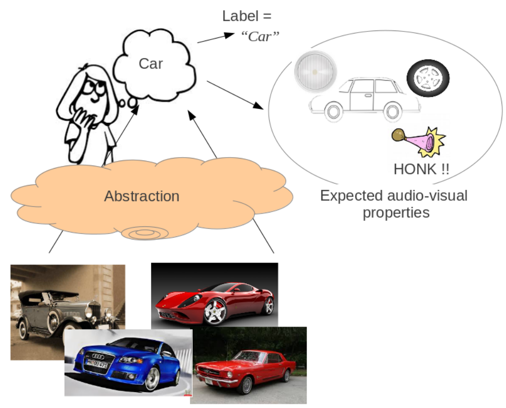


Fig. 3. Perceptual modeling of concepts

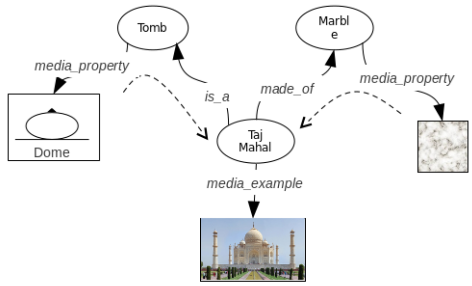
of conventional ontologies that comes with crisp DL based reasoning, which cannot handle the uncertainties associated with the media manifestations of concepts. Following a different approach, domain ontology is extended by [29] with "visual prototypes" or image examples, each of which represent a unique manifestations of a concept. A *query-by-example* search paradigm is used to identify the concepts from the visual contents in a repository. While it is a first step to extending domain ontology to the realm of multimedia, it is quite restrictive in media property specifications. Use of crisp logic for reasoning with interpretation of media contents is another limitation of this approach. Further, an ontology should support reasoning with media properties of concepts, like with the other properties, in a domain context. Media properties of concepts have some special semantics, which are not recognized. We shall shortly discuss the special semantics of media properties.

V. Concept of a "Concept": Perceptual Modeling of Domain

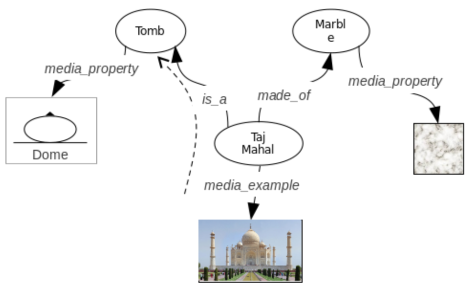
The shortcomings of existing approaches to multimedia ontologies primarily arise from the use of domain description and reasoning techniques that have been developed with text processing applications in view. None of these approaches look into the fundamental needs for knowledge representation in the realm of multimedia data collections. In this context, we note that while text documents are *conceptual* descriptions of human experience, media documents are *perceptual* records of the world, and both are quite dissimilar in nature. The textual descriptions convey the information more crisply than the media instances though they are susceptible to variations in human interpretation and filtering. On the other hand, the media instances are factual records of the world and generally contain a lot more information than text, but they are also likely to contain a lot more noise due to environmental factors. Thus, a conceptual domain model alone cannot cope up with the task of media data interpretation. It needs to be extended to include a *perceptual model*, which may need some different reasoning techniques. The perceptual model of a domain can be the key to bridge the semantic gap between the concepts and their manifestations as media features in multimedia documents.

Though seemingly different, the conceptual model of a domain is not disconnected from the perceptual model, but is derived from the latter [30]. Concepts and concept taxonomies are generated from many observations of the world, mental analysis of their similarities and dissimilarities and the resulting abstractions. An abstract concept is labelled with a natural language construct for the purpose of expression and communication. For example, observation of many cars leads to discovery of some of their common audio-visual properties, which is an abstraction of the concept and which is labeled with a construct, e.g. "car" in a natural language (see figure 3). Further, observation of subtle differences in such audio-visual properties leads to refinement of the concept and formation of concept taxonomy, e.g. "racing car", "vintage car", etc. As a consequence, possibility of manifestation of a concept in a media instance leads to expectation of some common perceptible audio-visual properties. These properties, when observed, leads to a belief in the existence of the concept. For example, a car may be recognized by perceiving one or more of its characteristic audio-visual patterns, e.g. a typical body shape, round wheels and head-lamps, its honk, and so on.

The above observations suggest that the conceptual world is bound to the perceptual world with causal relations. An abstract concept *causes* some perceptible media patterns to appear in multimedia documents. The observation of the media patterns provides evidence towards the concepts in a domain-context. An ontology for multimedia applications needs to encode such causal relations and enable reasoning with them. Further, the media manifestations of concepts are often uncertain and contextual in nature. Thus, it is necessary to incorporate a probabilistic reasoning paradigm with such ontologies. It should also be possible to reason with the media properties in the context of the domain. For example, a monument made of a certain kind of stone is likely to manifest the color and texture properties of the latter. Similarly, the example image of a specific monument is also an example for the generic class to which the monument belongs to (see figure 4). This form of media property inheritance rules are quite distinct from the general property inheritance rules in a concept taxonomy. Moreover, the elementary media properties



(a) Media property propagation



(b) Media example propagation

Fig. 4. Media property and example propagation rules

of a concept often exhibit spatial and temporal relations with each other with some variations in context of the domain. It should be possible to define such spatio-temporal properties in formal yet flexible way in the ontology.

VI. Multimedia Web Ontology Language (MOWL)

A. A conceptual introduction

A new paradigm for perceptual domain modeling with media properties of concepts and for reasoning with the domain model has been proposed in [8] to address the specific needs of knowledge representation for multimedia applications. Since the current ontology languages, e.g. OWL, do not support such model, a new language *Multimedia Web Ontology Language (MOWL)* has been proposed by the authors. The domain model is based on causal relation between the concepts and their media manifestations. Abductive reasoning model with Bayesian network has been proposed for concept recognition to cope up with the uncertainties associated with the causal model.

MOWL supports two types of entities, namely the *concepts* that represent the abstract real world entities and the *media objects* that represent the manifestation of concepts in the media world. For example, while a car can be a concept, its body shape can be a media object. As a special case, visual prototypes as in [29] or example media instances of concepts can also be considered as media objects. Like in other ontology languages, the concepts and the media objects may be organized in a taxonomical hierarchy. The concepts and media objects can have properties. A special class of properties that associates media objects with concepts represent the causal relations in the domain. The uncertainties in such causal relations are captured through a set of conditional probability tables. Another class of properties that relate the concepts signify media property propagation. Such properties can be defined in a domain context. These relations are also probabilistic in nature.

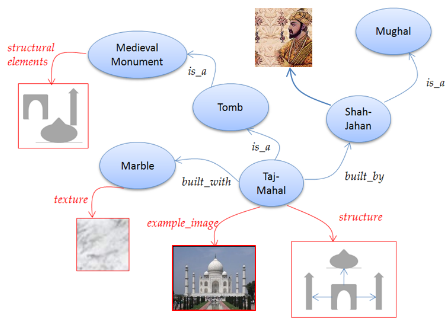
The properties of media objects that represent media manifestation of concepts, can be specified at various levels of complexity. In its simplest form, it can be specified with one of the MPEG-7 elementary audio-visual tools [22]. At the other end of the spectrum, complex media features, e.g. that characterize a dance posture, may need a specially trained classifier. In such cases, a procedural specification or a pointer to an intelligent agent implementing such function may be specified. Another type of complex media property specifications is characterized by spatio-temporal arrangement of simpler media objects. The relative positions of the constituent media objects can have natural variations in different media instances. For example, the relative positions of the dome and the minarets of a monument can be quite different when seen from different perspectives as illustrated in figure 5. MOWL offers constructs to create formal definition of such arrangement with flexibility. The definitions are based on a fuzzy variant of interval algebra, which is consistent with and can be executed with an extended MPEG-7 Query Engine proposed in [31]. Media examples that represent different manifestations of a concept as in [29] can also be associated with media object instances, when an example-based search is used for their detection.

Fig. 5. The Tajmahal seen from two perspectives*(source: http://sv.wikipedia.org/wiki/Taj_Mahal)*

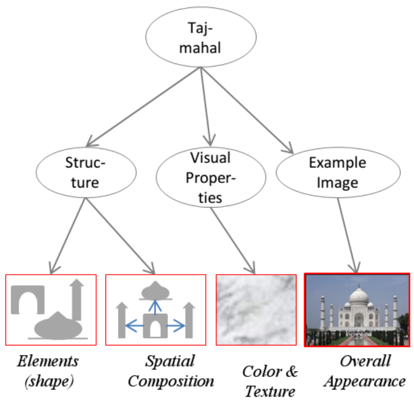
B. Reasoning with MOWL

The causal world model of MOWL prompts an abductive model of reasoning for concept recognition. It is carried out in two steps. In the first step, an *Observation Model (OM)* for a concept is created from the ontology. The OM constitutes the media properties of the concept as well as those of some other related concepts in the domain as determined by the media property propagation rules. The OM is organized as a Bayesian network with the concept at the root node and the expected media properties for that concept at the leaf nodes. Figure 6 shows a possible OM created from a multimedia ontology for the monument “Tajmahal”.

In the second step, each media instance is processed with appropriate feature extraction routines to detect the media properties specified at the leaf nodes of the OM. A leaf node is instantiated when the corresponding media pattern is detected, resulting in a belief revision in the Bayesian network. The posterior probability of the root node as a result of such media property detections signifies the belief in the concept in a multimedia document instance.



(a) Sample domain ontology



(b) A possible Observation Model

Fig. 6. Ontology and Observation Model

C. Discussions

The main difference of MOWL with MPEG-7 based multimedia ontology representation schemes is that the former can be used to model a domain with media properties of concepts, independent of any collection. In this sense, it is similar to the approach presented in [29]. While the latter allows visual prototypes as the only mechanism of media property specification of concepts, MOWL generalizes it to different types of property specifications, including audio-visual examples and MPEG-7 descriptors. Thus, MOWL can be used to interpret MPEG-7 based content descriptions, wherever available. Unlike normative definition of spatio-temporal relations that are used to express the structural composition of events in MPEG-7 informally, MOWL provides for formal yet flexible definition of such relations. Further, the method for media property specification in MOWL can virtually be extended to any type of media properties by using procedural specification. Note that association of different types of media properties with a concept in MOWL provides a natural solution to multi-modal concept recognition and cross-media associations.

Another important attribute of MOWL is reasoning with media properties in a domain context. Media property propagation rules help in creation of Observation Models for concepts incorporating context information. We shall discuss its importance in more details in the next section. While use of Bayesian reasoning is not uncommon for concept recognition in multimedia instances, dynamic creation of the Bayesian network in a domain context is a novelty in MOWL.

VII. Modeling multimedia semantics

Ontological reasoning in the multimedia domain addresses the problem of exploiting information embedded in multimedia assets and making the underlying meaning of the multimedia content explicit. However, the process of attaching meaning to multimedia content is not simple, not even well determined. For example, meaning of an image is not just determined on the basis of image data but also on the situation or context under consideration. Multimedia web ontology language provides a mechanism to attach semantics to the content by specifying possible content-dependent observables of concrete or abstract concepts. For example, we can associate several observable multimodal features, e.g. visual body-shape and typical *huff and puff* audio track with the categorical concept of steam engine. Ontological reasoning scheme of MOWL also facilitate specification of possible contexts for the steam engine, e.g. feature specifications for a pair of railway tracks or human activities in a railway station. Using these specifications, we can search for possible occurrence of steam engine in multimedia assets such as videos, provided that we have appropriate signal analysis algorithms for detection of huff and puff sound and other specified features. Feature detectors essentially embody techniques for distinguishing specific type of signal instances. Machine learning techniques can be used for building such classifiers and detectors. These classifiers and feature detectors provide the initiation point for semantic modeling of multimedia content in the context of ontological reasoning. MPEG-7 standard provides a scheme for specifying such descriptors but does not address the problem of generation of descriptors. These descriptors can encode semantic models at different levels of abstraction. For example, waterfront, as in LSCOM vocabulary [23], can be specified as the corresponding image classifier in the MOWL ontology at the lowest level. This is the key distinguishing feature of MOWL which enables semantic model construction in a hierarchical fashion linking higher level concepts with low level multimedia data.

As an example, we examine the way using which we can represent the concept of human action using the framework described above. We shall use the scheme proposed in [18] for detecting human action in images. Usually verbs indicate human actions; action part is associated with objects related to the action. For example, verb "riding" associated with "bike" indicates human action of riding bike; replacing bike by horse indicates riding horse. In MOWL, the node "riding" can have two specialization nodes bike-riding and horse-riding indicating two different actions. We can associate image based observables to these nodes using the scheme proposed in [18]. Given an image of a human action, many attributes and parts contribute to the recognition of the corresponding action. Actions are characterized by co-occurrence statistics of objects. For example, the "riding attribute is likely to occur together with objects such as "horse and "bike, but not with, say "laptop. Similarly, the "right arm extended upward" is more likely to co-occur with objects such as "volleyball. These interactions of action attributes and parts have been modeled as action bases for expressing human actions in [18]. A particular action in an image can therefore be represented as a weighted summation of a subset of these bases. The parent node can be represented as weighted summation of union of the subsets of children. In fact, error between reconstruction and test image can be normalized to contribute evidential support. MOWL also provides for specifying observable features for human action in other modalities like text with the same nodes. These features can be used for establishing context with reference to the text associated with an image for a multi-modal multimedia document.

VIII. Learning Multimedia Ontology

An ontology representing concepts and relations of the domain can be hand-coded with inputs from a team of domain experts. Such an ontology may be biased by the opinions of the experts and may not reflect the domain model accurately. This motivates learning of ontology from real-world examples. At another extreme, an ontology learnt from the sample data may not reflect the *human knowledge* of the domain and may be unwieldy. Thus, refinement of a hand-coded ontology with real-world data as an iterative process is considered to be a pragmatic solution to the problem [32].

Machine learning of ontology is essentially a statistical learning process. Probabilistic framework of MOWL is well amenable to it. An Observation Model created from a MOWL ontology models the causal relation between the concept and its possible media manifestations in the form of a Bayesian network. There has been several approaches to ontology learning using Bayesian network. These methods can be used to redefine an Observation Model and in turn, to refine the ontology.

A class of work on Bayesian network learning concentrate on redifining the CPT's in the Bayesian network without changing the network topology. Another class of work, generally referred to as full Bayesian network learning, attempts to discover new relations between concepts (and might drop some existing ones). This approach impacts the network topology. Refining a MOWL ontology can take either of the two forms. A method to update the CPT's in MOWL ontology from implicit user feedback in an retrieval application has been proposed in [33]. In this example, user click-through data has been used to collect implicit user feedback and the ontology is tuned to reflect a specific user's information preferences. A method for full Bayesian network learning in context of a cultural heritage archive has been proposed in [34]. In this example the relations and CPT's of a hand-crafted ontology have been updated using a labelled set of videos depicting classical music and dance.

IX. Application Examples

A. Digital Heritage Preservation

Ontologies have been used in digital museum projects [35], [19] to reason with the domain entities for effective utilization of the digital assets. A shortcoming in these systems is that they cannot reason with the multimedia representations of the artifacts and depend completely on the annotations. In order to deal with this problem, MOWL has been used to model the domain ontology for annotation and semantic navigation in an audio-visual archive *Nrityakosha* of Indian classical dance [7]. Figure 7 depicts an architecture of the system.

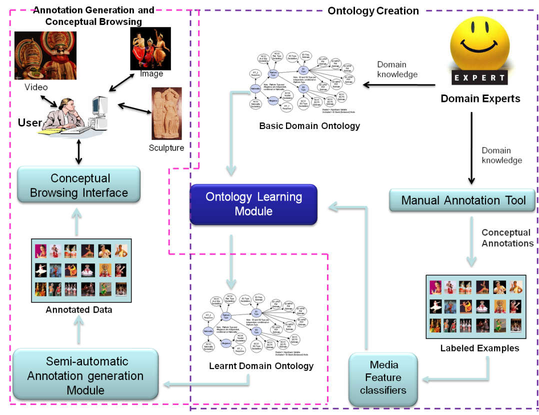


Fig. 7. Architecture of *Nrityakosha*

The domain model of *Nrityakosha* relates various entities, such as dance forms and the accompanying music as well as the myths and the roles that are depicted in those dances. The various concepts *manifest* in some *portrayals*, such as attire of the artistes, dance steps, body postures and musical themes, which are characterized by some audio-visual patterns in the media artifacts. While there is a well-defined grammar for Indian classical dance, individual artistes make their experiments and exercise some freedom resulting in variations to the dance steps. The perceptual and causal model of MOWL has definite advantage over existing ontology languages for such concept recognition tasks. The dance steps are often characterized by a temporal sequence of dance postures with some uncertainties, which can be formally and flexibly expressed with MOWL. Media property propagation rules allow property attributes to "flow" from concepts in mythical stories and roles to the dance steps and postures. While the ontology is initially handcrafted, it has been refined using the ontology learning method described in the previous section with a corpus of labelled data.

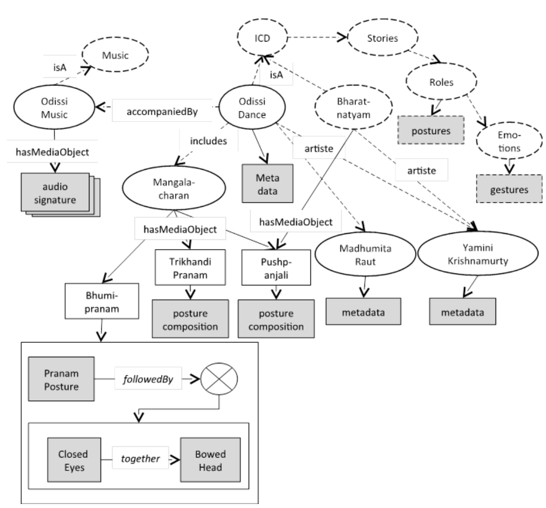


Fig. 8. An ICD Ontology Snippet

To illustrate the use of MOWL in modeling the domain, let us consider a classical dance form *Odissi* that is typically characterized by an opening act of *Mangalacharan* (invoking the gods). *Mangalacharan* is performed as a combination of three dance steps, each of which manifests in a series of postures. Each of these elementary postures can be detected using a trained set of classifiers. Thus, the dance form *Odissi*, the act *Mangalacharan* and its constituent steps can be modeled as concepts. They are evidenced by the *observable* postures and their sequences, which can be modeled as media objects. The classifiers used to recognize the postures can be expressed as "procedural specification" in MOWL. A few concepts, media objects and their relationships are depicted in figure 8. The edges connecting the concepts with their expected media manifestations are causal and are marked with uncertainties.

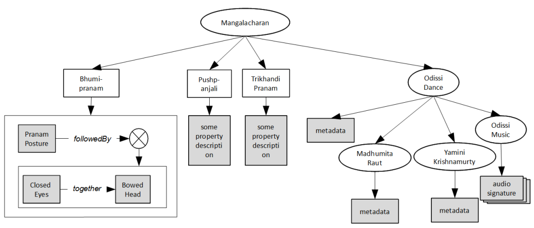


Fig. 9. Observation Model for *Mangalacharan*

Observation Models for concepts like *Mangalacharan* can be constructed from this ontology (see figure 9) and be used for concept recognition. Note that the OM comprises renderings of constituent dance steps (which are temporal sequence of postures) as well as contextual evidences of *Odissi* dance form. A major advantage of this approach is that a concept is recognized with a multitude of evidences, including contextual ones. As a result, failure of a feature detector because of environmental noise has little impact on the overall recognition performance. Further, the elementary postures constituting a higher level concept, e.g. a dance step, have more definitive features than the latter, and it is possible to build more accurate classifiers for them. Deployment of such classifiers and reasoning with their spatio-temporal composition improves the performance of detection of the higher level concepts. Robust concept recognition for audio-visual assets has been used in *Nrityakosha* for their semantic annotation and for establishing their semantic linkages.

B. Product Recommendation for Feature-rich Commodities

Content based filtering technique for product recommendation involves semantic matching of user profile and product features. The semantic associations of features with product categories are quite complex in many domains, such as fashion. Ontology based approaches for apparel recommendation have been presented in [36], [37]. The crisp ontological classification and the first-order reasoning rules deployed in these systems are inflexible to capture the subjectivity and uncertainty associated with choice of apparels. Moreover, they fail to deal with the "look and the feel" (visual and tactile properties), which are important selection parameters for the garments An apparel recommendation system based on perceptual modeling scheme of MOWL is presented in [38].

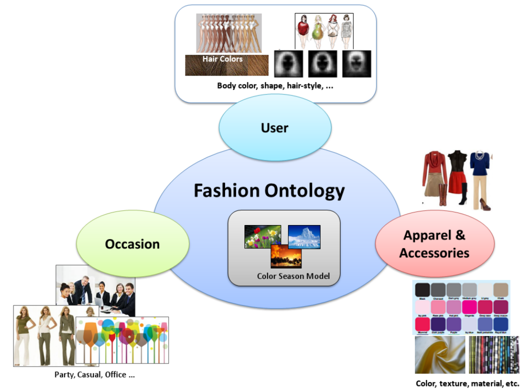


Fig. 10. Ontology for garment recommendation

Figure 10 shows a high-level view of the fashion ontology that incorporates knowledge about human users, occasion to wear and the garments. Visual attributes have been associated with humans and garments. Garments have been organized in several categories and several visual attributes have been assigned to them. The recommendation rules are based on *Color Season Model* [39] and other information sources.

The recommendation problem is handled in two steps in the system. First, an OM for user visual profile is created and the latter is determined based on observations on user body parameters such as skin color and body shape. Then an OM for the garment (to be recommended) is created by incorporating the discovered user profile. This OM has garment properties, e.g. color, texture, material, etc. as its observable property nodes. The garment catalog is consulted and the garment attributes (both visual and semantic) are analyzed to instantiate the property nodes in the OM. The garments that have highest posterior probability based on analysis of the garment properties qualify for recommendation. Figure 11 show the recommendation results for *Sarees*[1] for an Indian celebrity for different occasions.

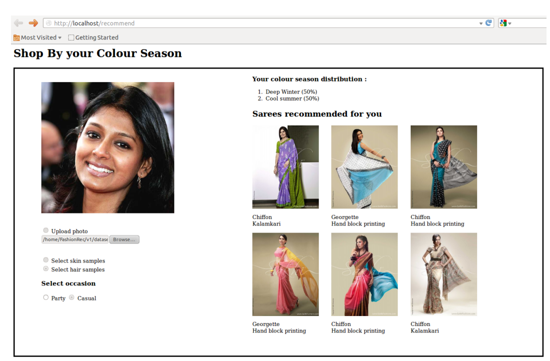


Fig. 11. Results for Apparel Recommendation

This approach provides quite a few benefits as compared to SVM based [36] or SWRL rule-based [37] recommendation. In the first place, the domain rules need not be exhaustively enumerated and it is sufficient to encode the rules connecting the broad classes. MOWL helps in reasoning with the media

[1] An ethnic wear for women popular in South Asia

properties of the concepts like garments, humans and occasions. Further, the abductive reasoning used in MOWL is robust to make recommendations even if all garment properties are not listed in the catalog. Most importantly, the causal probabilistic reasoning enables ranking of the recommendations allowing for user preferences.

X. Distributed Multimedia Applications

Many applications need to integrate information from multiple independent sources, including social media, to meet the user needs. Examples include travel services [40], news aggregation [5], medicine [41] and cultural heritage applications [42]. A multi-agent system [43] is a convenient tool to model such systems. The architecture of typical agent based system used for information gathering from multiple sources is shown in figure 12. The User agent does a pre-processing of the user request before forwarding it to the Broker agent. The Broker agent interprets the request with a background domain knowledge encoded in the form of an ontology and interacts with the Resource agents to retrieve the necessary information, often iteratively. There is usually a good deal of redundancy in Resource agents on the web. Some criteria may be applied to select a limited number of Resource agents to participate in the information gathering process. Finally, the Broker agent semantically integrates the information from multiple sources before reverting to the user.

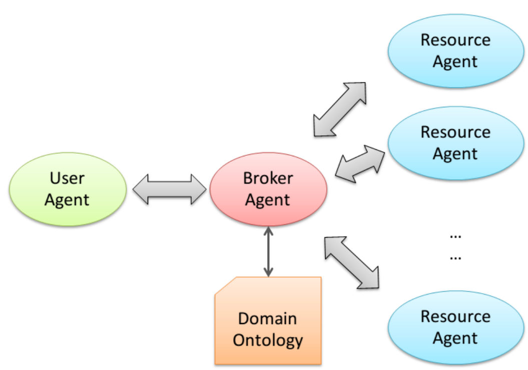


Fig. 12. Architecture on a multi-agent distributed information system

The data to be dealt with in many of the cited domains are often in multimedia format motivating their semantic integration in specific application contexts. MPEG-7 linked ontologies discussed in section IV attempt such integration. As shown in figure 2, semantic integration is effected in these systems at the conceptual level, based on the semantic descriptors for the contents (man-made or machine produced) or other forms of metadata [40] and not based on the information content in the media forms. MOWL provides an opportunity for integration based on analysis of media contents. The domain knowledge available in the Broker agent, when encoded in MOWL, can incorporate a perceptual model of the domain. A user request, when interpreted with such domain knowledge, produces an Observation Model that can be used to interpret media contents by the different Resource agents.

An Observation Model created from a non-trivial domain knowledge generally includes many leaf nodes (observable media properties), signifying many different manifestations of the concept. Generally, it is not necessary to observe all such media properties to have a sufficient belief in the concept. The posterior belief in a concept tends to saturate after a few observations and observation of further media properties does not add significantly to the belief value. Thus, it is desirable to create an *observation plan* by choosing an appropriate set of media properties that can result in sufficient posterior belief in the concept at a minimal computational cost. While the effectiveness of an evidence (media pattern) in identifying a concept depends on the domain knowledge, the computational cost and feasibility for its detection depends on the contents and the data organization in the Resource agents. A method to create resource-specific observation plans, considering both the aspects, using a distributed planning algorithm has been proposed in [44]. The Resource agent that has the potential to produce reliable results within some constraints of computational cost bids for participation. An interesting consequence of such planning is that, while an Observation Model for a concept, say steam locomotive, will contain both audio and visual patterns, the observation plan for an image repository will use some of the visual patterns only, but that for a video repository can use both. While different observation plans are executed by different Resource agents, all of them are derived from the same domain ontology. This facilitates information integration of multi-modal data from multiple sources.

The knowledge about the context is often distributed across multiple agents. For example, while the Broker agent encapsulates the domain ontology, the User agent might model a user profile that incorporates the knowledge about the user's implicit preferences [33]. The Resource agents may include a semantic data model for the contents in their repository [25]. In general, these independently created ontologies employ disparate data models. They need to be aligned to ensure their interoperability. The ontology alignment problem can be stated as discovery of equivalence and subsumption relationship between pairs of entities from two independent ontologies and application of the discovered mapping rules [45]. The equivalence of concepts are generally discovered by establishing context similarity (structure of the ontology graph around the concept), the equivalence of individuals are generally based on commonalty of properties.

An interesting approach to establish relation between entities in different ontologies that has not yet been explored well is by comparing their perceptual properties. Perceptual modeling of domain using MOWL presents such opportunity. While the terminology used to describe a concept can be different in different ontologies and the ontological relations for the concept can be domain dependent, the perceptual properties of a concept are expected to be invariant. Thus, two concepts can be said to be equivalent if the Observation Models for two concepts are similar [46]. Note that the Observation Model of a concept incorporates media properties of related concepts and can thus be used to compare the structural context of the concepts.

XI. Conclusions

Despite significant advances in media content analysis over the last couple of decades, a solution to the problem of "semantic gap" still eludes the researchers. Semantic analysis of media forms is still a subject of vigorous research. Fusion of multi-modal data from heterogeneous and distributed resources poses a much bigger challenge. It appears that an ontology put on top of media analysis services is not a suitable solution. Multimedia Web Ontology Language is a first step towards semantic analysis and integration of multimedia data from information sources in an open Internet environment. While MOWL presently currently deals with audio and visual data, the theoretical framework has the generality to deal with any form of sensor data, thus paving the way for semantic fusion of multi-modal multi-sensor data. The framework further needs to be extended to incorporate other facets of multimedia event models as proposed in contemporary literature [47], [48].

References

[1] Pingdom, "Internet 2012 in numbers," http://royal.pingdom.com/2013/01/16/internet-2012-in-numbers/, 2013.

[2] H. Wang, L.-T. Chia, and S. Liu, "Image retrieval ++–web image retrieval with an enhanced multi-modality ontology," *Multimedia Tools Appl.*, vol. 39, no. 2, pp. 189–215, Sep. 2008. [Online]. Available: http://dx.doi.org/10.1007/s11042-008-0202-7

[3] M. Rabbath, P. Sandhaus, and S. Boll, "Automatic creation of photo books from stories in social media," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 7S, no. 1, pp. 27:1–27:18, Nov. 2011. [Online]. Available: http://doi.acm.org/10.1145/2037676.2037684

[4] O. Conlan, I. O'Keeffe, and S. Tallon, "Combining adaptive hypermedia techniques and ontology reasoning to produce dynamic personalized news services," in *Proceedings of the 4th international conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, ser. AH'06. Berlin, Heidelberg: Springer-Verlag, 2006, pp. 81–90.

[5] R. Bansal, R. Kumaran, D. Mahajan, A. Khurdiya, L. Dey, and H. Ghosh, "Twipix: a web magazine curated from social media," in *Proceedings of the 20th ACM international conference on Multimedia*, ser. MM '12. New York, NY, USA: ACM, 2012, pp. 1355–1356. [Online]. Available: http://doi.acm.org/10.1145/2393347.2396482

[6] M. Hatala, L. Kalantari, R. Wakkary, and K. Newby, "Ontology and rule based retrieval of sound objects in augmented audio reality system for museum visitors," in *Proceedings of the 2004 ACM symposium on Applied computing*, ser. SAC '04. New York, NY, USA: ACM, 2004, pp. 1045–1050.

[7] A. Mallik, S. Chaudhury, and H. Ghosh, "Nrityakosha: Preserving the intangible heritage of indian classical dance," *J. Comput. Cult. Herit.*, vol. 4, no. 3, pp. 11:1–11:25, Dec. 2011. [Online]. Available: http://doi.acm.org/10.1145/2069276.2069280

[8] A. Mallik, H. Ghosh, S. Chaudhury, and G. Harit, "Mowl: An ontology representation language for web based multimedia applications," *ACM Transactions on Multimedia Computing, Communications and Applications*, (In press).

[9] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific American*, May 2001.

[10] T. R. Gruber, "A translation approach to portable ontologies," *Knowledge Acquisition*, vol. 5, no. 2, pp. 199–220, 1993.

[11] F. van Harmelen, V. Lifschitz, and B. Porter, Eds., *Handbook of Knowledge Representation*. Elsevier, 2008.

[12] "Owl web ontology language guide," http://www.w3.org/TR/owl-guide/, 2004.

[13] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by image and video content: The qbic system," *Computer*, vol. 28, no. 9, pp. 23–32, Sep. 1995. [Online]. Available: http://dx.doi.org/10.1109/2.410146

[14] J. R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. Jain, and C. fe Shu, "The virage image search engine: An open framework for image management," in *Proceedings of the SPIE Conference on Visual communication and Image Processing*, 1996, pp. 76–87.

[15] Y. Liu, D. Zhang, G. Lu, and W.-Y. Ma, "A survey of content-based image retrieval with high-level semantics," *Pattern Recogn.*, vol. 40, no. 1, pp. 262–282, Jan. 2007. [Online]. Available: http://dx.doi.org/10.1016/j.patcog.2006.04.045

[16] J. Sivic, "Efcient visual search of videos cast as text retrieval," *IEEE Trans. on PAMI*, vol. 31, no. 4, pp. 591–605, April 2009.

[17] J. Fan, Y. Gao, H. Luo, and G. Xu, "Statistical modeling and conceptualization of natural images," *Pattern Recogn.*, vol. 38, no. 6, pp. 865–885, Jun. 2005. [Online]. Available: http://dx.doi.org/10.1016/j.patcog.2004.07.011

[18] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei, "Human action recognition by learning bases of action attributes and parts," in *Proceedings of the 2011 International Conference on Computer Vision*, ser. ICCV '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 1331–1338. [Online]. Available: http://dx.doi.org/10.1109/ICCV.2011.6126386

[19] M. Doerr, "The cidoc conceptual reference module: an ontological approach to semantic interoperability of metadata," *AI Mag.*, vol. 24, no. 3, pp. 75–92, Sep. 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=958671.958678

[20] A. Gangemi, N. Guarino, C. Masolo, A. Oltramari, and L. Schneider, "Sweetening ontologies with dolce," in *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*, ser. EKAW '02. London, UK, UK: Springer-Verlag, 2002, pp. 166–181. [Online]. Available: http://dl.acm.org/citation.cfm?id=645362.650863

[21] H. Xu, X. Zhou, M. Wang, Y. Xiang, and B. Shi, "Exploring flickr's related tags for semantic annotation of web images," in *Proceedings of the ACM International Conference on Image and Video Retrieval*, ser. CIVR '09. New York, NY, USA: ACM, 2009, pp. 46:1–46:8. [Online]. Available: http://doi.acm.org/10.1145/1646396.1646450

[22] F. Nack, J. v. Ossenbruggen, and L. Hardman, "That obscure object of desire: Multimedia metadata on the web, part 2," *IEEE MultiMedia*, vol. 12, no. 1, pp. 54–63, Jan. 2005. [Online]. Available: http://dx.doi.org/10.1109/MMUL.2005.12

[23] J. R. Smith, "Largescale concept ontology for multimedia," *IEEE Multimedia Magazine*, vol. 13, no. 3, pp. 86–91, July–September 2006.

[24] S. Dasiopoulou, V. Tzouvaras, I. Kompatsiaris, and M. G. Strintzis, "Enquiring mpeg-7 based multimedia ontologies," *Multimedia Tools Appl.*, vol. 46, no. 2-3, pp. 331–370, Jan. 2010. [Online]. Available: http://dx.doi.org/10.1007/s11042-009-0387-4

[25] C. Tsinaraki, P. Polydoros, and S. Christodoulakis, "Interoperability support between mpeg-7/21 and owl in ds-mirf," *IEEE Trans. on Knowl. and Data Eng.*, vol. 19, no. 2, pp. 219–232, Feb. 2007. [Online]. Available: http://dx.doi.org/10.1109/TKDE.2007.33

[26] J. Hunter, "Enhancing the semantic interoperability of multimedia through a core ontology," *IEEE Trans. Cir. and Sys. for Video Technol.*, vol. 13, no. 1, pp. 49–58, Jan. 2003. [Online]. Available: http://dx.doi.org/10.1109/TCSVT.2002.808088

[27] R. Arndt, R. Troncy, S. Staab, L. Hardman, and M. Vacura, "Comm: designing a well-founded multimedia ontology for the web," in *Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference*, ser. ISWC'07/ASWC'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 30–43. [Online]. Available: http://dl.acm.org/citation.cfm?id=1785162.1785166

[28] K. Dalakleidi, S. Dasiopoulou, G. Stoilos, V. Tzouvaras, G. Stamou, and Y. Kompatsiaris, "Knowledge-driven multimedia information extraction and ontology evolution," in *Knowledge-Driven Multimedia Information Extraction and Ontology Evolution*, G. Paliouras, C. D. Spyropoulos, and G. Tsatsaronis, Eds. Berlin, Heidelberg: Springer-Verlag, 2011, ch. Semantic representation of multimedia content, pp. 18–49. [Online]. Available: http://dl.acm.org/citation.cfm?id=2001069.2001071

[29] M. Bertini, A. D. Bimbo, G. Serra, C. Torniai, R. Cucchiara, C. Grana, and R. Vezzani, "Dynamic pictorially enriched ontologies for digital video libraries," *IEEE MultiMedia*, vol. 16, no. 2, pp. 42–51, Apr. 2009. [Online]. Available: http://dx.doi.org/10.1109/MMUL.2009.25

[30] H. Kangassalo, "Conceptual level user interfaces to data bases and information systems," in *Advances in information modelling and knowledge bases*, H. Jaakkola, H. Kangassalo, and S. Ohsuga, Eds. IOS Press, 1991, pp. 66–90.

[31] S. S. Wattamwar and H. Ghosh, "Spatio-temporal query for multimedia databases," in *Proceedings of the 2nd ACM workshop on Multimedia semantics*, ser. MS '08. New York, NY, USA: ACM, 2008, pp. 48–55. [Online]. Available: http://doi.acm.org/10.1145/1460676.1460686

[32] F. Wu and D. S. Weld, "Automatically refining the wikipedia infobox ontology," in *Proceedings of the 17th international conference on World*

Wide Web, ser. WWW '08. New York, NY, USA: ACM, 2008, pp. 635–644. [Online]. Available: http://doi.acm.org/10.1145/1367497.1367583

[33] H. Ghosh, P. Poornachander, A. Mallik, and S. Chaudhury, "Learning ontology for personalized video retrieval," in *Workshop on multimedia information retrieval on The many faces of multimedia semantics*, ser. MS '07. New York, NY, USA: ACM, 2007, pp. 39–46. [Online]. Available: http://doi.acm.org/10.1145/1290067.1290075

[34] A. Mallik and S. Chaudhury, "Acquisition of multimedia ontology: an application in preservation of cultural heritage," *IJMIR*, vol. 1, no. 4, pp. 249–262, 2012.

[35] L. Schneider, "Designing foundational ontologies - the object-centered high-level reference ontology ochre as a case study," in *in Conceptual Modeling 2003, 22nd International Conference on Conceptual Modeling, I.-Y.* Springer, 2003, pp. 91–104.

[36] S. Liu, J. Feng, Z. Song, T. Zhang, H. Lu, C. Xu, and S. Yan, "Hi, magic closet, tell me what to wear!" in *Proceedings of the 20th ACM international conference on Multimedia*, ser. MM '12. New York, NY, USA: ACM, 2012, pp. 619–628.

[37] D. Vogiatzis, D. Pierrakos, G. Paliouras, S. Jenkyn-Jones, and B. J. H. H. A. Possen, "Expert and community based style advice," *Expert Systems with Applications: An International Journal*, vol. 39, no. 12, pp. 10 647–10 655, September 2012.

[38] S. Ajmani, H. Ghosh, A. Mallik, and S. Chaudhury, "An ontology based personalized garment recommendation system," in *Workshop on Personalization, Recommender Systems and Social Media*, November 2013.

[39] B. Kentner, *Color me a season: A complete guide to finding your best colors and how to use them*. Ken Kra Publishers, 1979.

[40] Y. T. F. Yueh, D. K. W. Chiu, H.-f. Leung, and P. C. K. Hung, "A virtual travel agent system for m-tourism with semantic web service based design and implementation," in *Proceedings of the 21st International Conference on Advanced Networking and Applications*, ser. AINA '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 142–149. [Online]. Available: http://dx.doi.org/10.1109/AINA.2007.25

[41] H. Kosch, R. Slota, L. Böszörményi, J. Kitowski, J. Otfinowski, and P. Wójcik, "A distributed medical information system for multimedia data - the first year's experience of the parmed project," in *Proceedings of the 8th International Conference on High-Performance Computing and Networking*, ser. HPCN Europe 2000. London, UK, UK: Springer-Verlag, 2000, pp. 543–546. [Online]. Available: http://dl.acm.org/citation.cfm?id=645564.658465

[42] X. Pan, T. Schiffer, M. Schröttner, R. Berndt, M. Hecher, S. Havemann, and D. W. Fellner, "An enhanced distributed repository for working with 3d assets in cultural heritage," in *Proceedings of the 4th international conference on Progress in Cultural Heritage Preservation*, ser. EuroMed'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 349–358.

[43] E. Shakshuki, M. Kamel, and H. Ghenniwa, "A multi-agent system architecture for information gathering," in *Proceedings of the 11th International Workshop on Database and Expert Systems Applications*, ser. DEXA '00. Washington, DC, USA: IEEE Computer Society, 2000, pp. 732–. [Online]. Available: http://dl.acm.org/citation.cfm?id=789091.790428

[44] H. Ghosh and S. Chaudhury, "Distributed and reactive query planning in r-magic: An agent-based multimedia retrieval system," *IEEE Trans. on Knowl. and Data Eng.*, vol. 16, no. 9, pp. 1082–1095, Sep. 2004. [Online]. Available: http://dx.doi.org/10.1109/TKDE.2004.40

[45] V. Ermolayev and M. Davidovsky, "Agent-based ontology alignment: basics, applications, theoretical foundations, and demonstration," in *Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics*, ser. WIMS '12. New York, NY, USA: ACM, 2012, pp. 3:1–3:12. [Online]. Available: http://doi.acm.org/10.1145/2254129.2254136

[46] B. Maiti, H. Ghosh, and S. Chaudhury, "A framework for ontology specification and integration for multimedia applications," in *Proceedings of Knowledge Based Computer System (KBCS 2004)*, December 2004.

[47] V. Mezaris, A. Scherp, R. Jain, M. Kankanhalli, H. Zhou, J. Zhang, L. Wang, and Z. Zhang, "Modeling and representing events in multimedia," in *Proceedings of the 19th ACM international conference on Multimedia*, ser. MM '11. New York, NY, USA: ACM, 2011, pp. 613–614. [Online]. Available: http://doi.acm.org/10.1145/2072298.2072391

[48] P. Appan and H. Sundaram, "Networked multimedia event exploration," in *Proceedings of the 12th annual ACM international conference on Multimedia*, ser. MULTIMEDIA '04. New York, NY, USA: ACM, 2004, pp. 40–47. [Online]. Available: http://doi.acm.org/10.1145/1027527.1027536

RELATED CONFERENCES, CALL FOR PAPERS/PARTICIPANTS

TCII Sponsored Conferences

WI 2014
The 2013 IEEE/WIC/ACM International Conference on Web Intelligence
Warsaw，Poland
August 11-14, 2014
http://wic2014.mimuw.edu.pl/wi/homepage

The 2014 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2014) will be a part of the 2014 Web Intelligence Congress (WIC 2014). WIC 2014 will be held at the historical Central Campus of the University of Warsaw, Warsaw, Poland, between 11th and 14th of August 2014.

Web Intelligence focuses on scientific research and applications by jointly using Artificial Intelligence (AI) (e.g., knowledge representation, planning, knowledge discovery and data mining, intelligent agents, and social network intelligence) and advanced Information Technology (IT) (e.g., Semantic Web, Wisdom Web, Web search, Web Mining, recommender systems) for the next generation of Web-empowered products, systems, services, and activities.

The series of Web Intelligence (WI) conferences was started in Japan in 2001. Since then, WI has been held yearly in several countries, including: Canada, China, France, USA, Australia and Italy. WI conference is recognized as the World's leading forum related to Web Intelligence. In 2014, it will be organized in Warsaw as a Special Event commemorating the 25th anniversary of the Web.

IAT 2014
The 2013 IEEE/WIC/ACM International Conference on Intelligent Agent Technology
Warsaw，Poland
August 11-14, 2014
http://wic2014.mimuw.edu.pl/iat/homepage

The 2014 IEEE/WIC/ACM International Conference onIntelligent Agent Technology (IAT 2014) will be a part of the 2014 Web Intelligence Congress (WIC 2014). WIC 2014 will be held at the historical Central Campus of the University of Warsaw, Warsaw, Poland, between 11th and 14th of August 2014.

Intelligent Agent Technology explores advanced intelligent systems and their broad applications in computer science and engineering, big data mining, biomedical informatics, health informatics, social networks, education, robotics, security, etc.

The series of Intelligent Agent Technology (IAT) conferences was started in Japan in 2001. Since then, IAT has been held yearly in several countries, including: Canada, China, France, USA, Australia and Italy. IAT series of conferences is recognized as one of the most important events related to Multi-Agent Systems with an emphasis on their connections to Information Technology.

ICDM 2014
The Twenty-First IEEE International Conference on Data Mining
Shenzhen, China
December 14-17, 2014
http://icdm2014.rutgers.edu/

The IEEE International Conference on Data Mining series (ICDM) has established itself as the world's premier research conference in data mining. It provides an international forum for presentation of original research results, as well as exchange and dissemination of innovative, practical development experiences. The conference covers all aspects of data mining, including algorithms, software and systems, and applications. In addition, ICDM draws researchers and application developers from a wide range of data mining related areas such as statistics, machine learning, pattern recognition, databases and data warehousing, data visualization, knowledge-based systems, and high performance computing. By promoting novel, high quality research findings, and innovative solutions to challenging data mining problems, the conference seeks to continuously advance the state-of-the-art in data mining. Besides the technical program, the conference features workshops, tutorials, panels and, since 2007, the ICDM data mining contest.

ISMIS 2014
The 20th International Symposium on Methodologies for Intelligent Systems
Roskilde, Denmark
June 25-27, 2014
http://www.isl.ruc.dk\ismis2014

International Symposium on Methodologies for Intelligent Systems is an established and prestigious conference for exchanging the latest research results in building intelligent systems. Held twice every three years, the conference provides a medium for exchanging scientific research and technological achievements accomplished by the international community. The scope of ISMIS is intended to represent a wide range of topics on applying Artificial Intelligence techniques to areas as diverse as decision support, automated deduction, reasoning, knowledge based systems, machine learning, computer vision, robotics, planning, databases, information retrieval, etc. The focus is on research in intelligent systems. The conference addresses issues involving solutions to problems that are complex to be solved through conventional approaches and that require the simulation of intelligent thought processes, heuristics and applications of knowledge. The integration of these multiple approaches in solving complex problems is of particular importance. ISMIS provides a forum and a means for exchanging information for those interested purely in theory, those interested primarily in implementation, and those interested in specific research and industrial applications.

ICTAI 2014
The Twenty-fourth IEEE International Conference on Tools with Artificial Intelligence
Limassol，Cyprus
November 10-12, 2014
http://cecs.wright.edu/atrc/ictai13

The annual IEEE International Conference on Tools with Artificial Intelligence (ICTAI) provides a major international forum where the creation and exchange of ideas related to artificial intelligence are fostered among academia, industry, and government agencies. The conference facilitates the cross-fertilization of these ideas and promotes their transfer into practical tools, for developing intelligent systems and pursuing artificial intelligence applications. The ICTAI encompasses all technical aspects of specifying, developing and evaluating the theoretical underpinnings and applied mechanisms of the AI-based components of computer tools such as algorithms, architectures and languages.

Related Conferences

AAMAS 2014
The Twelfth International Conference on Autonomous Agents and Multi-Agent Systems
Paris, France
May 5-9, 2014
http://aamas2014.lip6.fr

The AAMAS conference series was initiated in 2002 in Bologna, Italy as a joint event comprising the 6th International Conference on Autonomous Agents (AA), the 5th International Conference on Multiagent Systems (ICMAS), and the 9th International Workshop on Agent Theories, Architectures, and Languages (ATAL).

Subsequent AAMAS conferences have been held in Melbourne, Australia (July 2003), New York City, NY, USA (July 2004), Utrecht, The Netherlands (July 2005), Hakodate, Japan (May 2006), Honolulu, Hawaii, USA (May 2007), Estoril, Portugal (May 2008), Budapest, Hungary (May 2009), Toronto, Canada (May 2010), Taipei, Taiwan (May 2011) and Valencia, Spain (June 2012). Saint Paul, Minnesota, USA(2013). AAMAS 2014 will be held in May in Paris, France

AAMAS is the largest and most influential conference in the area of agents and multiagent systems, the aim of the conference is to bring together researchers and practitioners in all areas of agent technology and to provide a single, high-profile, internationally renowned forum for research in the theory and practice of autonomous agents and multiagent systems.

AAAI 2014
The Twenty-Seventh AAAI Conference on Artificial Intelligence
Quebec, Canada
July 27-31, 2014
http://www.aaai.org/Conferences/AAAI/aaai14

The Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI-13) will be held July 27–31, 2014 in Québec City, Québec, Canada. The purpose of this conference is to promote research in artificial intelligence (AI) and scientific exchange among AI researchers, practitioners, scientists, and engineers in affiliated disciplines. AAAI-14 will have a diverse technical track, student abstracts, poster sessions, invited speakers, tutorials, workshops, and exhibit/competition programs, all selected according to the highest reviewing standards. AAAI-14 welcomes submissions on mainstream AI topics as well as novel crosscutting work in related areas.

SDM 2014
The Thirteenth SIAM International Conference on Data Mining
Philadelphia, Pennsylvania, USA
April 24- 26, 2014
http://www.siam.org/meetings/sdm14/

Data mining is an important tool in science, engineering, industrial processes, healthcare, business, and medicine. The datasets in these fields are large, complex, and often noisy. Extracting knowledge requires the use of sophisticated, high-performance and principled analysis techniques and algorithms, based on sound theoretical and statistical foundations. These techniques in turn require powerful visualization technologies; implementations that must be carefully tuned for performance; software systems that are usable by scientists, engineers, and physicians as well as researchers; and infrastructures that support them.

This conference provides a venue for researchers who are addressing these problems to present their work in a peer-reviewed forum. It also provides an ideal setting for graduate students and others new to the field to learn about cutting-edge research by hearing outstanding invited speakers and attending tutorials (included with conference registration). A set of focused workshops are also held on the last day of the conference. The proceedings of the conference are published in archival form, and are also made available on the SIAM web site.

IJCAI 2015
The Twenty-Third International Joint Conference on Artificial Intelligence
Buenos Aires, Argentina
August 3-9, 2015
http://lanyrd.com/2015/ijcai

IJCAI is the International Joint Conference on Artificial Intelligence, the main international gathering of researchers in AI. Held biennially in odd-numbered years since 1969, IJCAI is sponsored jointly by IJCAI and the national AI societie(s) of the host nation(s).. AI can play a key role in addressing environmental, economic, and societal challenges concerning sustainable development and a sustainable future. AI techniques and methodologies can be exploited to help address sustainability problems and questions, for example to increase the efficiency and effectiveness of the way we manage and allocate our natural and societal resources. The study of sustainability questions will also enrich and transform AI, by providing new challenges. The conference will include a special track dedicated to papers concerned with all these aspects.

IEEE Computer Society
1730 Massachusetts Ave, NW
Washington, D.C. 20036-1903

Non-profit Org.
U.S. Postage
PAID
Silver Spring, MD
Permit 1398