Distributed knowledge engineering and evidence-based knowledge representation in multi-agent systems

Anton Kolonin

Aigents Group,
Pravdy 6/12, 630090 Novosibirsk, Russian Federation
akolonin@gmail.com
http://aigents.com/

Abstract. Paper describes an approach to multi-agent distributed knowledge management architecture for semantic web implementation and presents a model of structured knowledge representation for such a system, enabling collaborative knowledge engineering in social environments, with cognitive architecture suggested and prototype systems described.

Keywords: cognitive architecture belief knowledge engineering knowledge graph knowledge management multi-agent system ontology peer-to-peer semantic web social collaboration software agents

1 Introduction: From centralized knowledge engineering to distributed model

Nowadays, the most representative implementations of the semantic web [1] are supplied by the top Internet content providers such as Google with its Knowledge Graph (Knowledge Vault) [2]. From the perspective of our earlier work [3], it can be classified as *centralized knowledge globalization*, with all semantic information physically contained within a proprietary semantic database owned by a *knowledge aggregator*. In such a model, an access to it can be granted by means of non-intelligent clients connected to an intelligent server. The vast majority of internal *knowledge* and hence cognitive capabilities of a system reside inside the perimeter of a corporate data center, even if some tiny fraction of it can be offloaded to a public domain (such as Wikidata) or a particular user (in respect to their personal data). It is also implicit that the knowledge aggregator takes responsibility to maintain the truth value of any piece of knowledge in respect to any event or fact in the outer world.

The alternative model is denoted as *decentralized (distributed) knowledge globalization* [3, 4], which assumes the knowledge is semi-evenly distributed across the entire global computational network. This also implies possibility of dynamic redistribution (and possible redundancy) of the knowledge itself, as well as distribution of the points of it processing across a peer-to-peer network, where different nodes may belong to different owners. The *truth value* of a piece of knowledge turns to be *dynamic* and rather *subjective* specific to agents holding a particular segment of the entire knowledge network, as it was originally invented [1] and fits ideally the emerging Internet of Things (IoT) [5]. Motivation and argumentation for developing this model was properly covered earlier, along with description of a distributed semantic version of a social network [4]; herein we discuss a more generic approach to maintenance of distributed semantic knowledge in general and describe our development of a system serving this purpose.

2 Principal goals: Distributed knowledge engineering

For a distributed knowledge engineering environment to emerge in multi-agent software systems, we anticipate it should follow the patterns of social self-organization in human societies. Evolution of distributed computational intelligence is possible as co-evolution with collaborative intelligence of human society. That calls for emergence of a society of computational agents with the following requirements.

There is a need for rich *historical memory* shared by communicating computer agents (e.g. accessible public banks of information available for mutual sharing). It is needed to maintain an open space of semantic graphs which can be formed by means of sharing (donating) the personal semantic graphs by private agents, when each sharing or donation act contains information authored by an agent itself or delegated to an agent for re-distribution and it is considered non-confidential.

Each computer agent should have an *ability to explicitly expose its own knowledge* indicating confidence, proprietary rights and privacy (share-ability in respect to other agents) of it. They also have a right to retain intellectual property on the knowledge they contribute and specify the privacy levels of it so it can be either accessible by peer agent only or forwarded to another agent.

There is a requirement for rich *sensory environment* and accessible *means of gathering novel information*, driving the communication end and enabling development of adaptive intelligent behavior (e.g. search, browsing and messaging against peer computer agents). In order to benefit human users, agents should be capable of adaptive intelligent behaviors finding new patterns and creating new knowledge in multi-factor and dynamically changing environments.

Fertility of diverse behavioral patterns (i.e. computational algorithms) exposed by agents (capable of evolving upon feedback from peer agents) is expected. This is not that much a requirement but more an expected beneficial outcome from the other requirements, assuming agents are equipped with adaptive learning algorithms.

To enable *peer-to-peer communication* in the environment involving multiple agents as well as people, we need a *unified language* based on the common basic ontology. That means not just syntax of declarative descriptions for data sets or imperative programmatic instructions but a whole range of means to convey the meaning of states, intents and inquiries of communicating agents, based on a common belief system, in syntax, easily parsable by software and comprehensible to humans at the same time. Semantic architecture of a language, regardless of its syntactical representation, be it RDF, Turtle, SPARQL, JSON-LD, Lisp or ORL [6,7,8,9] or a combination of these, should support a wide range of communication paradigms.

The latter language should also provide capabilities such as *fuzzy-ness, subjectivity and partial comprehension. Fuzzy-ness* implies the need to maintain both truth value and confidence level of an assertion, being able to calculate dynamic truth value of an assertion in different inference contexts (with the process of merging congruent assertions supplied with evidence from different communication subjects and amount of confidence specific to the context). *Subjectivity* means that certain assertions can be treated useful only in the context of a particular belief system but not in others (say Google Knowledge Graph's belief may be somewhat different from same of someone else's). It signifies that there is a need to express this belief-owner-specific knowledge in the communication. *Partial comprehension* requirement suggests that any complex message from one agent to another may be only partially comprehended, to the extent the mental models and ontological beliefs of a sender and a receiver overlap, while the remainder of the message can be ignored.

3 Architecture approach: Multiple agent roles and configurations

Overall architecture implementing the above-suggested environment can be drawn with the following scheme, involving various agents playing a typical role or a combination of several such roles.

Within the suggested architecture, *storage agents* provide distributed (and likely redundant) storage of structured information while *collector agents* perform information gathering from unstructured media (such as text files, web pages, raw video, audio, scanned paper hardcopy materials, etc.) as well as getting input signals from the outer world (using input devices such as thermometers, motion sensors, microphones, camcorders, etc.). *User agents* establish forward and backward communications with users and operators while *broker agents* serve routing of the messages between all other agents (e.g. implementing topologies such as *cloud storage* and *federated search*). Finally, actor agents can direct actions towards the surrounding social and physical environments (publishing web pages, sending emails and messages, authoring files or activating devices in the physical world).

Different types of agents (Fig. 1) are typical roles rather than narrow specializations, i.e. the same physical instance of an agent can play different roles simultaneously. At the same time, given specific storage and performance capabilities and connectivity graphs, various topologies can be formed (either by manual configuration or adaptive emergence). For instance, a broker agent plus a set of storage agents implement cloud storage. A broker agent with sets of collector agents and user agents managed by user agents form a *search engine* with *crawler service*. In turn, a set of user agents associated with broker agents form *social network*. Finally, all systems mentioned above can be integrated into a meta-system (such as *federated search* or *distributed crowdsourcing* platform) with help of broker agents with a broad specialization.

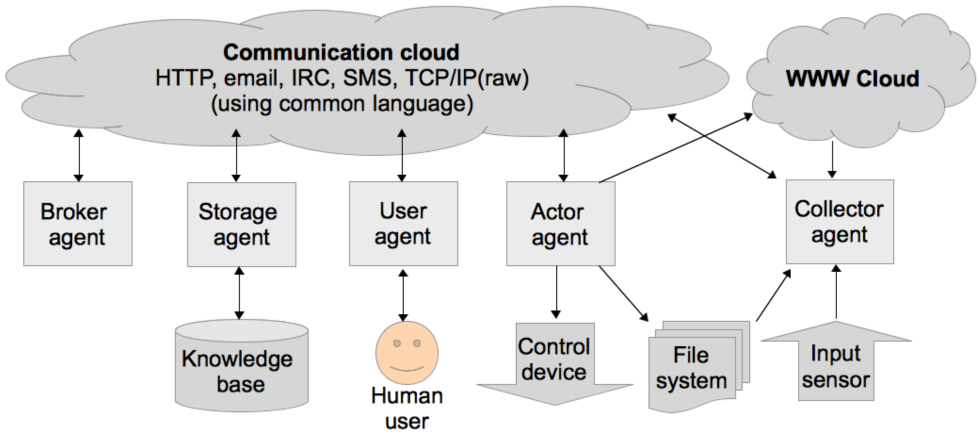

Fig. 1. Roles of distributed intelligence agents.

In order to actualize the possibility described above, there seems to be a demand to agree on an *open communication standard* for agents of emerging computational intelligence, adopted by the involved players. That standard would include specification of interfaces the intelligent agents would support as well as the language to be used for communication among them with the two basic functions: 1) *Output*: Return requested (by search or browse) knowledge primarily to be implemented by public agents (such as semantic search engines) or adapters to them, but also may be supported by any other large and small, corporate and personal agents which could want to contribute to the semantic search space; 2) *Input*: Accept a piece of knowledge distributed by a peer agent with an option to either reject the input (if does not fit an agent's preferences, i.e. its internal belief system) or incorporate it into the belief system with account to the appropriate copyright and privacy constraints. Both interfaces would have synchronous as well as asynchronous versions so that the *Output* may be either given in respect to a synchronous query, or it may be provided asynchronously upon prior subscription. Respectively, the *Input* can take the form of a channel to accepts the data feed as well as a place to subscribe for content to be delivered to a subscriber.

4 Knowledge representation: Evidence-based social model

Given any agent talk to any other using the same communication language, internal design, the implemented algorithms and programming language of an agent do not matter that much. In order to communicate, however, agents are implied to have some jointly shared system of fundamental knowledge (some belief system) regarding the surrounding world and themselves. They should also have a mechanism for either accepting the knowledge coming to an agent from its outer world (if it is compatible with the agent's belief system), or rejecting it (in the opposite case). Further, for different sorts of accepted knowledge, an agent should be able to make judgments on reliability of different facts, which can be done based on the amount of evidence associated with these facts. Each evidence is considered in terms of trust towards its source. Here we come to the *social evidence-based knowledge representation model*.

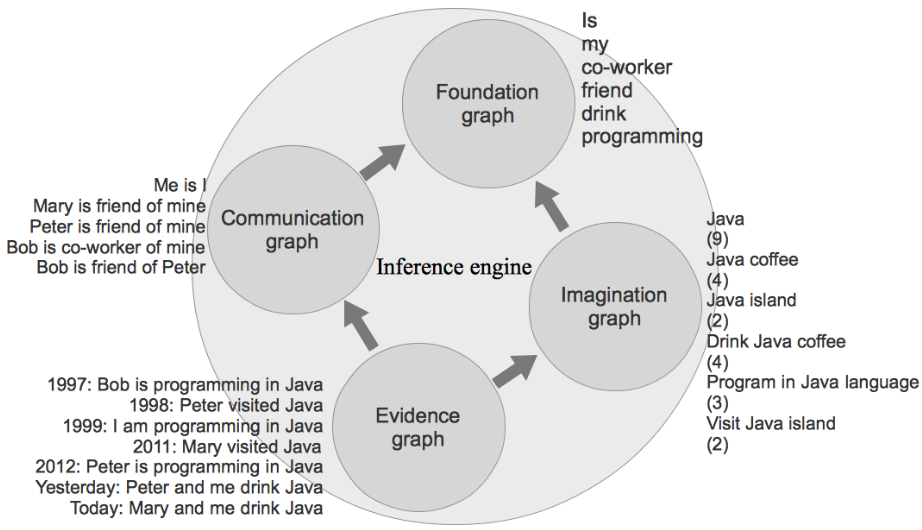

Fig. 2. Specialized subgraphs of the agent knowledge base and the dynamic truth value calculation in the social evidence-based knowledge representation model.

With massively distributed data processing and many-to-many style replication, synchronization of concurrent changes (especially, such as updates and deletes) become a big problem. For instance, if agent A communicates fact P to agent B while B communicates fact Q to A, there is just a counter addition of information to each of the agent's knowledge bases. However, there is a typical scenario where agents argue about something, making conflicting changes to the same data. For instance, agent A tells there are relationships X and Y between P and Q, while agent B argues there is Y and Z but not X who is to be trusted in such case? Obviously, both can agree on presence of Y, while X remains as a personal belief of A and B keeps believing in Z. That is, assuming part of the message can be accepted and the reminder can be declined, it can be possible to make each of the agents more knowledgeable in the course of communication, yet not having to destroy the belief system of each of them.

Within the social evidence-based knowledge representation model, *truth value* of any piece of information can be calculated as a sum truth value of its evidence records communicated by peer agents multiplied by the trust levels for each of these peer agents. To achieve this, the entire semantic hyper-graph representing knowledge of an agent can be split in four major sub-graphs (Fig. 2).

The *foundation graph* layer is a cornerstone cognitive base of each of the agents. Without that, the two agents speaking the same language syntactically, would not understand each other if their foundation graphs differ significantly. It is assumed that a foundation graph does not need any fuzzy inference applied to it and there may be some special rules (specific to each agent design) as to how that part of the knowledge is formed. The most favorable approach is to have portions of the imagination graph (discussed further) exceeding the given thresholds of evidence to be hardwired to the foundation graph. Reasoning on this part of knowledge might be called *orthodox*, *stereotypic* or *closed-minded* thinking.

The *imagination graph* is a pool of novel evidence-based knowledge coming to an agent via communication channels. Given the trust levels specific to particular communication peers providing the inputs, as well as amounts of positive and negative evidence supplied for assertions in this graph, the agent is capable to draw its own assertions and either communicate them back to the outer world or upload to the foundation graph eventually. This part of an agent's brain can be considered as *dynamic*, *non-stereotypic* or *open-minded* core.

The *communication graph* layer describes social interaction channels of an agent and also provides the basis for account of subjectivity, so that each fact in the imagination graph is supplied by trust given to a particular communication agent at a time. This is effectively the *social core*, or *personal social network* of an agent, maintaining trust levels for each of peer agents in two dimensions. First how much confidence can be given to incoming information communicated by the peer, in general. Second if there are any confidential restrictions implied for information communicated to the peer for instance, for private knowledge only, or for public share, etc.

The *evidence graph* effectively records temporal facts of evidence exposed by peer agents from communication graph to draw cumulative assertions in imagination graph on that basis. This pool of facts serves as an evidence base for the inference engine calculating the *truth values* with account to subjective grounds as well as with temporal analysis capabilities. Each piece of information here is timestamped and labeled by a peer communicating it. Data stored here can be subject of *evidence compression* with either clustering of fractional time slices into larger time intervals or aggregating evidences from individual peers into larger groups of peers. Further, *evidence can be forgotten*, with either transition of knowledge (derived from the evidence) from the imagination graph to the foundation graph or its complete removal if no supporting evidence was found for a long time (*evidential garbage collection*). The major drives for the forgetting process are physical resources constrains (so the system assures the amounts of all data fit the existing memory) and the basic goal to maintain the most reliable knowledge fitting the system's internal belief to a greater extent.

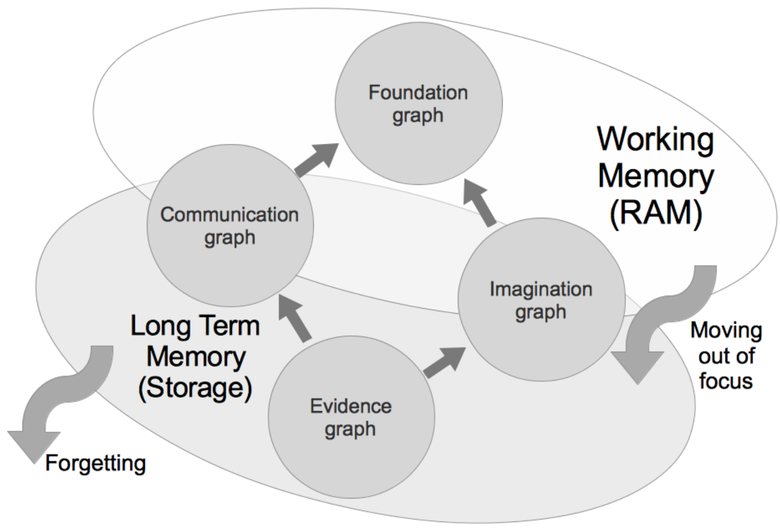

Fig. 3. Specifics of implementing the agent's subgraphs residing in different kinds of memory within the cognitive architecture.

The knowledge representation outlined above leads to possible technical implementation architecture, to a certain extent inspired by OpenCog [10]. The major specific feature of the architecture is support for the social evidence-based knowledge representation model, taking on board practical considerations of physical memory capacity in modern computing devices and the requirements for reasonable response times and energy consumption for end-user devices such as personal computers, tablets and smartphones.

The architecture needs to address the following problem. On the one hand, the graph-based operations are very sensitive to input/output performance if executed with traditional relational or object-oriented databases especially with highly-connected graphs. So the optimal implementation would rely on in-memory operations on graphs, as proved earlier by experiments and implementation of a trainable text classification and attribution system [11]. However, the call for the evidence-based knowledge representation model implies the need for a tremendous amount of linked data to get involved which makes the problem harder.

The trade-off described above would get solved with mechanisms of moving knowledge items in-focus and out-of-focus and forgetting the irrelevant and outdated knowledge aligned with the existing hardware constraints (Fig. 3). That is, we assume that the foundation graph as well as most of (if not all) communication graphs and imagination graphs reside in RAM corresponding to *working memory* of human brain at least, as long as parts of the entire graph are connected to any items that need current attention. Respectively, moving knowledge items out of the attention focus would correspond to moving them out of an in-memory cache (while they still reside in *long term memory* corresponding to slow persistent storage), or such process can be enforced by restrictions on consumable memory with new data requiring attention pushed into working memory. Further, the items not recalled for a long time (not linked at all, or having an insufficient number of links, or not being moved to the attention focus for a substantial amount of time), can be garbage collected and so removed even from the long term memory saving the storage space.

There is enhancement to the design described above, adding another level of complexity to it *resource-constrained truth value*. A variation of working memory size for the same amount of knowledge may lead to a variation of truth values calculation confidence. That is, given the number of evidence data exceeding the size of working memory, only the most confident or recent data may be kept for processing, having the reminder left outside of the inference scope, restricted by physical constraints. Hence, the precision or confidence of truth value calculation for a fact may depend on the amount of memory in agent's possession or the amount of memory allocated to agent inference functions at a given moment of time.

5 Practical implementation: Webstructor Project

Based on the requirements, Webstructor project (`http://www.webstructor.net/`) has been under ongoing development since 1995.

In 1995-1996, *semantic graph* was employed to describe domain ontologies and operational spaces of systems carrying out data management, inter-personal interactions, interactive form processing, report generation and action script development. On this basis, a respective software platform was created and used to draw wide range of applications including personal diary, time management, business accounting, inventory/sales automation, customer relationship management and others. A system drawback was poor run-time performance, given full normalization of any data and executable code down to nodes and links of a semantic graph stored in a relational database.

In 1997-1999, based on a similar semantic graph model, *object-relational language* (ORL) [9] for inter-agent communication was developed to enable creating a corporate business automation system for the stock exchange domain. It was used to describe the whole application domain including data model, entry forms, reports and all business rules and functions, and create an operable business application.

In 2001, the *agent software for peer-to-peer* knowledge creation and interchange was created as part of the Webstructor project. The computational agents were developed to operate as web server-side Servlets, browser-side Applets or standalone Applications, exchanging the knowledge in many-to-many fashion encoded in ORL statements, with user interfaces capable to browse, search and maintain the knowledge visually in forms of graphs or an interactive ORL console (so the same language was made usable by humans).The gateway between ORL and Lisp was developed and the entire Open Cyc ontology was uploaded to the Webstructor agent system.

In 2006, the Webstructor semantic engine was employed to build a system for 3D visualization, navigation and sharing of complex scientific data. Within the distributed agent system, it enabled visualization, navigation and amendment of virtual object properties in a hyperspace in a collaborative peer-to-peer network.

The existing Webstructor implementation model is simplified, so that only the fundamental graph and the communication graph are present which implies a *full trust* for agent's interactions, assuming any data involved in exchange are an *absolute truth.* There are three different types of agents present in Webstructor now. Servlet agent runs on the web server and performs the broker and storage roles being able to serve multiple Applets and Servers over HTTP protocol, passing information through between agents and providing intermediate storage at the same time. Applet agent runs in the web browser and provides user access to the whole system. Server agent simultaneously plays the roles of storage, broker and user, so it can be employed to create full-blown distributed peer-to-peer networks.

Two practical applications are present a visual ontology editor and a spatial data visualization system, both enabling peer-to-peer knowledge sharing. The visual ontology editor provides capabilities to edit various graphs with options to associate vertices with web resources, colors, shapes and image information. This can be used to edit hierarchical graphs as well as recurrent networks. There is also a possibility to create higher-order networks suitable to express logical formulae, for instance. Besides handling input and output data in ORL format, the same content can be imported from CycL language. In addition to graphical editing capabilities, the application provides an interactive console which can be used to manipulate knowledge by means of ORL language.

6 Conclusion: Opportunities and challenges

On the practical side, assuming industry agreement on an *open cross-platform multi-agent communication protocol (language)*, there is a possibility for a *distributed computational intelligence agent software* to run on every smartphone and personal computer. The software would look like a personal knowledge management assistant, capable to create knowledge content (i.e. authoring things and their properties), establish communications with other agents (as a knowledge consumer or as a knowledge provider or in both roles) and implement a distributed knowledge storage cell role for the entire agent system.

Within the same inter-agent communication infrastructure, application patterns such as a *distributed storage*, *social network*, *federated search* and others can be constructed by users upon the need dynamically or emerged on run-time in the course of operations. The topology of the communication graph can be an emergent structure and a part of the entire distributed system knowledge rather than a rigid pre-defined schema.

There are two major problems to be addressed. Primarily, it is essential to develop efficient *technology for dynamic truth value determination* based on context-specific knowledge sets (contextual subgraphs), incorporating multiple contextual restrictions such as participants of the conversation or temporal interval of the problem being explored. Secondarily, it is necessary to come up with a well-understood and accepted procedure (bound to an open protocol employed by community) which would enable *merging knowledge sets* of one agent conveyed to another, accounting for fussiness, preserving subjectivity, with possibility of partial comprehension.

References

1. Berners-Lee, T., Fischetti, M.: Weaving the Web : The Original Design and Ultimate Destiny of the World Wide Web by its Inventor, Harper San Francisco (1999)
2. Dong, X., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S., Zhang, W.: Knowledge Vault: A Web-Scale Approach to Probabilistic Knowledge Fusion, SIGKDD (2014)
3. Kolonin, A.: The Emerging World Wide Mind. Artificial General Intelligence conference. Beijing (2013), `http://www.webstructor.net/news/20130801/`
4. Tramp, S., Frischmuth, P., Ermilov, T., Shekarpour, S., Auer, S.: An architecture of a distributed semantic social network. Semantic Web, pp 77-95 (2014)
5. Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M.: Internet of Things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems 29, Elsevier (2013)
6. Decker, S., Melnik, S., Van Harmelen, F., Fensel, D., Klein, M., Broekstra, J., Erdmann, M., Horrocks, I.: The Semantic Web: The roles of XML and RDF. IEEE Internet Computing. 15 (3), pp. 63-74 (2000)
7. World Wide Web Consortium: Turtle Terse RDF Triple Language" (2012)
8. Lanthaler, M., Gtl, C.: On Using JSON-LD to Create Evolvable RESTful Services. Proceedings of the 3rd International Workshop on RESTful Design (WS-REST 2012) at WWW2012 (2012)
9. Kolonin, A.: Object-Relational Language and modern tradeoffs in software technology. International Andrei Ershov Memorial Conference, PSI 2006, Program understanding workshop, Russia, Novosibirsk (2006), `http://webstructor.net/docs/orl/`
10. Goertzel, B., Duong, D.: OpenCog NS: A Deeply-Interactive Hybrid Neural-Symbolic Cognitive Architecture Designed for Global/Local Memory Synergy, Biologically Inspired Cognitive Architectures II: Papers from the AAAI Fall Symposium (FS-09-01) (2009)
11. Kolonin, A.: High-performance automatic categorization and attribution of inventory catalogs. Proceedings of All-Russia conference Knowledge Ontology Theories (KONT-2013), Novosibirsk, Russia (2013), `http://www.webstructor.net/papers/Kolonin-HP-ACA-IC-text.pdf`