

Role of Non-Axiomatic Logic in a Distributed Reasoning Environment

Mirjana Ivanović¹, Jovana Ivković^{1(✉)}, and Costin Bădică²

¹ Department of Mathematics and Informatics, Faculty of Sciences,
University of Novi Sad, Novi Sad, Serbia
{mira,jovana.ivkovic}@dmi.uns.ac.rs

² University of Craiova, Craiova, Romania
cbadica@software.ucv.ro

Abstract. The aim of this paper is to introduce the design of a novel Distributed Non-Axiomatic Reasoning System. The system is based on Non-Axiomatic Logic, a formalism in the domain of artificial general intelligence designed for realizations of systems with insufficient resources and knowledge. Proposed architecture is based on layered and distributed structure of the backend knowledge base. The design of the knowledge base makes it fault-tolerant and scalable. It promises to allow the system to reason over large knowledge bases with real-time responsiveness.

Keywords: Reasoning architecture · Non-Axiomatic Logic · Artificial General Intelligence · Big Data

1 Introduction

Big Data analytics is a hot topic that could be applied in different real life industries (e.g. [4]). Growing amounts of available information have led to the requirement that every intelligent software system must have: the ability to manipulate huge amounts of data [3]. As a consequence, several practical technologies, including, for example, graph databases and NoSQL (e.g. [11]) and the MapReduce programming model [1], have emerged.

This paper will present the idea of a new system – Distributed Non-Axiomatic Reasoning System (DNARS) [6]. This system is based on a novel distributed reasoning architecture that incorporates Big Data notions, includes fault-tolerance based on data replication, as well as a highly-scalable and distributed backend knowledge base. DNARS uses recent techniques for the purpose of large-scale distributed data processing. Our approach makes it possible for the system to operate on huge knowledge bases, and also service a large number of external clients in real-time.

The DNARS architecture consists of two central components. Its backend knowledge base is used to store the system's knowledge and experience on a large scale. To support this functionality, DNARS includes a set of algorithms packaged in the form of inference engines. In order to realize high-level reasoning

capabilities, DNARS relies on the Non-Axiomatic Logic (NAL). NAL provides a formal way to support reasoning in Artificial General Intelligence (AGI) systems [14, 16].

The expression “non-axiomatic” implies that there is an insufficient amount of resources and knowledge [16] available in the system, and said knowledge can also be inconsistent and uncertain. New evidence of various content can be added to the knowledge base, and thus can change the truth value of the included knowledge statements. Moreover, truth-values do not necessarily have to converge to certain limit, i.e. they can change arbitrarily. Furthermore, only a part of the available statements is usually involved in the problem-solving activities, thus providing locality to the reasoning process.

Accordingly, mechanisms in NAL support efficient handling of inconsistencies and uncertainty in available statements. They also have the ability to reduce the number of statements and thus abridge the available knowledge to enable more efficient reasoning.

The rest of the paper is organized as follows. Section 2 provides a brief introduction into NAL. Section 3 presents an overview of the proposed architecture of the DNARS system. Section 4 briefly discusses some related work, while Sect. 5 concludes the paper.

2 Non-Axiomatic Logic

Non-Axiomatic Logic (NAL) is set apart from other formalisms used in reasoning systems. It contains a set of inference rules, a symbolic grammar, and a semantic theory. NAL sentences consist of subject-copula-predicate form. Both subject and predicate can be represented by atomic (a single word) or compound (which joins atomic and/or compound terms) terms. NAL has an experience-grounded semantics [12] based on the concepts of generalization and specialization. Inheritance, formally captured as $S \rightarrow P$, is the most typical statement in NAL. Terms S and P represent subject and object, while the connector \rightarrow represents the inheritance copula. The meaning of this statement is: *S is a type of P*.

NAL inference rules (syllogistic form) are used for deriving new knowledge. They support question answering or dealing with statement inconsistencies. Inference in NAL system is achieved by means of various available inference rules. Those rules are influenced by the copulas and positions of the common terms in premises.

NAL itself is hierarchically organized into 9 layers. Different inference rules, and/or new features are introduced from level to level. In the first proposal of our DNARS architecture we concentrated on the first four layers.

NAL-1 introduces inference rules on inheritance statements (deduction, induction, and abduction) [16].

NAL-2 is extended by symmetric inheritance represented in the grammar with the similarity copula: $(S \leftrightarrow P) \Leftrightarrow (S \rightarrow P) \wedge (P \rightarrow S)$. As a consequence, three new forward inference rules are introduced in this layer: comparison, analogy, and resemblance [14–16].

NAL-3 brings compound terms in form: $\{T_1 \text{ con } T_2 \text{ con } \dots \text{ con } T_n\}$ where *con* is the connector and $T_1 \dots T_n$ are terms, $n \geq 1$ [14,16]. Connectors are: extensional intersection (\setminus), intensional intersection (\cap), extensional difference ($-$), and intensional difference (\ominus). The rule summarizes the system's experience.

NAL-4 introduces arbitrary relations among terms like product (\times). Inheritance between separate components of a compound term is defined in the following way [14,16]: $((S_1 \times \dots \times S_n) \rightarrow (P_1 \times \dots \times P_n)) \Leftrightarrow ((S_1 \rightarrow P_1) \wedge \dots \wedge (S_n \rightarrow P_n))$. R as relational term can be defined as a term connected via inheritance to a product term, i.e. either by $(T_1 \times T_2) \rightarrow R$ or by $R \rightarrow (T_1 \times T_2)$ [14,16].

More information on NAL is available in the references mentioned in this section.

3 DNARS – Distributed Non-Axiomatic Reasoning System

Advantages and capabilities of NAL are planned to be introduced and incorporated in our novel, previously developed multi-agent middleware, named Siebog, which integrates two essential parts:

1. XJAF (Extensible Java EE-based Agent Framework) [8,13], which is a server-side multi-agent architecture that supports clustered environments.
2. A client-side multi-agent system (Radigost) [7,9], mostly based on HTML5 markup language.

Siebog is a multi-agent system that aims to support agents during their lifetime by providing several functionalities, including: maintaining the agent's life cycle, providing infrastructure for message exchange, etc. Siebog operates on top of computer clusters which yields two of its important features: load-balancing and fault-tolerance.

Additionally, it is our plan to enhance Siebog with support for DNARS, so that Siebog agents can have the capability of advanced reasoning.

DNARS (see Fig. 1) consists of the following parts:

- *Resolution engine* that provides clients with answers to their questions.
- *Forward inference engine* that is responsible for producing new knowledge from existing one.
- *Short-term memory* that only includes statements necessary for solving current problems.
- *Knowledge domain* is a sub-set of the knowledge base, consisting of closely related statements.
- *Backend knowledge base* that comprises of the system's entire knowledge base, which is an essential repository of its experience accumulated over time.
- *Event manager* that handles events triggered by changes made in the knowledge base.

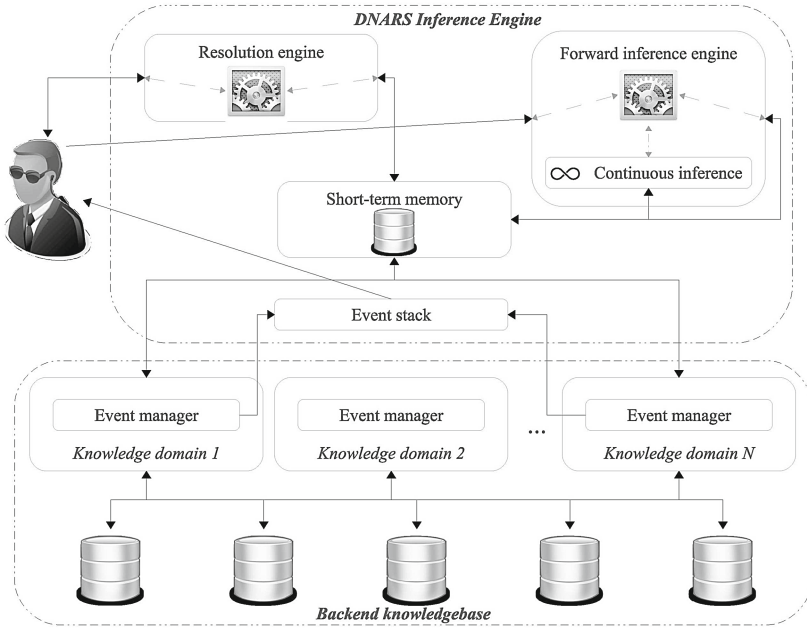


Fig. 1. The architecture of DNARS.

These aforementioned parts could be grouped into two sub-systems: *DNARS Inference engines* and *Backend knowledge base*. The *DNARS Inference engines* sub-system consists of *Resolution* and *Forward inference engines*. The *Backend knowledge base* sub-system includes *Short-term memory*, *Knowledge domain*, *Backend knowledge base* and *Event manager* components. There is a unique *Backend knowledge base* that serves all external clients, while each client is affiliated with its own set of engines. The knowledge base is designed to be scalable, and is organized in a such a way that it can support multiple clients.

3.1 DNARS Inference Engines

DNARS inference engines sub-system consists of *Resolution* (to answer questions) and *Forward inference* (to derive new knowledge) engines that support:

- Questions with “?”, i.e. “? *copula* P ” or “S *copula* ? ”. Here, the *Resolution engine* searches through the knowledge base in order to find the most appropriate substitute for “?”. The answer should be reached in real time.
- Questions like “S *copula* P”. The engine will first try to find the answer in the knowledge base, but if doesn’t exist, it will use backward inference rules to try and reach the required answer. The answer to this type of questions is also achievable in real time, but only if the answer is already in the knowledge base. If it doesn’t exist as such then the backward inference process will be

performed asynchronously [14,16] and the answer is passed on to the client when it becomes available using the *Event manager*.

Currently, the *Forward inference engine* is meant to enable DNARS with support of a subset of forward inference rules of the first four NAL layers. This approach is a starting point for implementing a first version of a system with practical reasoning abilities.

3.2 Backend Knowledge Base

DNARS is designed to manage big amounts of knowledge efficiently. The *Backend knowledge base* is made up of three layers:

- The first layer consists of the *Knowledge base* which is distributed across a computer cluster using horizontal scaling [5]. Its distributed nature allows the system to store huge amounts of data, and it also provides fault-tolerance due to the state replication of the cluster nodes.
- In the second layer the knowledge base is organized into *Knowledge domains* which allows the system to work with only a part of the knowledge base, depending on the given problem. The domains can be distributed across the cluster nodes (Fig. 1.).
- The top layer consists of the *Short-term memory* and the *Event manager*. *Short-term memory* acts as a fast memory storage. With the completion of the inference cycle, the content of the *Short-term memory* is saved to the appropriate domain. *Event manager* is based on the Observer design pattern, and its purpose is to notify clients of any relevant changes in the knowledge state.

4 Related Work

One of the main intended purposes of DNARS is to serve as an underlying reasoning engine in our multi-agent environment named Siebog [9,13]. This is a departure from the Belief-Desire-Intention (BDI) model which is common in agent technology [10].

Concepts of NAL, when compared to traditional BDI model, offer several advantages. The main characteristic of NAL statements is that they are endowed with truth-values representing confidence of belief, using the true definition of belief. On the other hand, in the BDI model, a developer has to plan for the possibility that a belief might not be true, as agents themselves cannot directly assign confidence to their beliefs.

NAL-based agents provide new features like inconsistency resolution (based on backward inference), learning (based on forward inference), and working with insufficient amount of knowledge and resources (e.g. compound terms [14,16]), while the BDI model does not offer such features.

In contrast to existing BDI systems, the main advantage that DNARS offers is the possibility of applying reasoning over huge knowledge bases.

On the other hand, BDI can be seen as another type of reasoning that can sit on top of NAL. BDI is about “practical reasoning”, i.e. reasoning towards actions. Moreover, in some BDI systems, e.g. Jason, the belief base of the agent architecture is configurable. So, an agent could simply incorporate beliefs from the NAL knowledge base. Systems like that do have problems with speed and scalability, but they can describe “flexible behaviour” by mixing reactivity with proactivity, a feature that is missing from standard backward and forwards chaining reasoning.

OpenNARS [2] is an open-source implementation of NAL [14, 16]. It supports the logic of all existing layers of NAL. Its main parts are the inference engine, the memory module, and lastly the control mechanism which is in charge of the reasoning cycles [14].

Both OpenNARS and DNARS are implementations of non-axiomatic reasoning and, unlike other reasoning and cognitive systems, they use NAL as a foundational formalism. According to available resources, NAL is able to handle insufficient resources and knowledge to a larger extent than other systems.

Recently, an emerging trend in data processing is characterized by different functionalities of the Big Data paradigm, especially in increasing the performance when processing large amounts of complex data. So, by combining Big Data paradigm functionalities with NAL, in DNARS we will achieve better performances when handling big and complex data with limited resources and time. A special problem that we addressed is dealing with knowledge inconsistencies only when they emerge in particular situations (e.g. when multiple answers to a question are available). Another issue that we addressed is the amount of raw information, namely we tried to decrease it by combining separate chunks of information. DNARS is designed with a NoSQL database in mind, so that it could support faster processing of large amounts of data, but at the expense of endangering consistency of the information on a temporary basis. In our proposed architecture we would like to integrate the best benefits of most important techniques and methods for big, complex and inconsistent data processing into a single architecture.

Despite their similarities, there are sharp distinctions between DNARS and OpenNARS. As it has been in development longer, OpenNARS incorporates all nine layers of NAL, and it contains more advanced control mechanisms. The main advantage of DNARS comes with its ability to reason over huge knowledge bases due to its advanced backend knowledge base organization.

5 Conclusions and Future Work

In this paper we considered different emergent technologies and their advanced functionalities, as well as the possibility to include them in our existing multi-agent system Siebog [13]. By using these technologies, we have striven towards making our system more intelligent and able to reason faster over large amounts of knowledge.

This paper has presented a new general-purpose reasoning architecture – Distributed Non-Axiomatic Reasoning System (DNARS). DNARS uses NAL as

the basis for its formal reasoning. NAL is a specialized formalism that offers a well-defined syntax, experience-grounded semantics, as well as a set of inference rules. Overall, it takes into consideration the possibility of reasoning over insufficient resources and knowledge [16]. We have faced numerous challenges while designing DNARS, but we have managed to achieve an advantage when compared to other similar systems. An innovative organization of the backend knowledge base, as well as an original implementation of NAL inference rules in a highly-scalable, fault-tolerant and distributed environment makes it possible for DNARS to efficiently process large amounts of data, while being able to service a large number of external clients.

During our research, numerous challenging and open scientific questions were recognized, thus offering us several different directions of realization of high-quality reasoning services in real world environments. In the future, we plan to improve our proposed architecture by adding the remaining layers of NAL to DNARS. Although the first four layers currently considered in DNARS are sufficient for simple reasoning activity, the rest of the layers would provide DNARS with new reasoning capabilities of a higher-level.

Acknowledgments. This work was partially supported by Ministry of Education, Science and Technological Development of the Republic of Serbia, through project number OI174023: “Intelligent techniques and their integration into wide-spectrum decision support”, as well as a collaboration agreement between University of Novi Sad (Serbia), University of Craiova (Romania), SRIPAS and Warsaw University of Technology (Poland).

References

1. Dean, J., Ghemawat, S.: Mapreduce: simplified data processing on large clusters. *Commun. ACM* **51**(1), 107–113 (2008)
2. Hammer, P., Lofthouse, T., Wang, P.: The OpenNARS implementation of the non-axiomatic reasoning system. In: Steunebrink, B., Wang, P., Goertzel, B. (eds.) *AGI 2016*. LNCS, vol. 9782, pp. 160–170. Springer, Cham (2016). doi:[10.1007/978-3-319-41649-6_16](https://doi.org/10.1007/978-3-319-41649-6_16)
3. Hovy, E., Navigli, R., Ponzetto, S.P.: Collaboratively built semi-structured content and artificial intelligence: the story so far. *Artif. Intell.* **194**, 2–27 (2013)
4. Lee, T., Lee, H., Rhee, K.H., Shin, U.S.: The efficient implementation of distributed indexing with hadoop for digital investigations on big data. *Comput. Sci. Inf. Syst.* **11**(3), 1037–1054 (2014)
5. Michael, M., Moreira, J.E., Shiloach, D., Wisniewski, R.W.: Scale-up x scale-out: a case study using nutch/lucene. In: *IEEE International Parallel and Distributed Processing Symposium, IPDPS 2007*, pp. 1–8. IEEE (2007)
6. Mitrović, D.: *Intelligent Multiagent Systems based on Distributed Non-Axiomatic Reasoning*. Ph.D. thesis, University of Novi Sad, Faculty of Science (2015)
7. Mitrović, D., Ivanović, M., Bădică, C.: Delivering the multiagent technology to end-users through the web. In: *Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS 2014)*, p. 54. ACM (2014)
8. Mitrović, D., Ivanović, M., Budimac, Z., Vidaković, M.: Supporting heterogeneous agent mobility with alas. *Comput. Sci. Inf. Syst.* **9**(3), 1203–1229 (2012)

9. Mitrović, D., Ivanović, M., Budimac, Z., Vidaković, M.: Radigost: Interoperable web-based multi-agent platform. *J. Syst. Softw.* **90**, 167–178 (2014)
10. Rao, A.S., Georgeff, M.P., et al.: BDI agents: from theory to practice. In: *ICMAS 1995*, pp. 312–319 (1995)
11. Robinson, I., Webber, J., Eifrem, E.: *Graph Databases: New Opportunities for Connected Data*. O'Reilly Media Inc., Sebastopol (2015)
12. Rodriguez, M.A., Geldart, J.: An evidential path logic for multi-relational networks. In: *AAAI Spring Symposium: Technosocial Predictive Analytics*, pp. 114–119 (2009)
13. Vidaković, M., Ivanović, M., Mitrović, D., Budimac, Z.: Extensible Java EE-based agent framework – past, present, future. In: Ganzha, M., Jain, L. (eds.) *Multiagent Systems and Applications*. ISRL, vol. 45, pp. 55–88. Springer, Heidelberg (2013). doi:[10.1007/978-3-642-33323-1_3](https://doi.org/10.1007/978-3-642-33323-1_3)
14. Wang, P.: *Rigid Flexibility*, vol. 55. Springer, Dordrecht (2006)
15. Wang, P.: Analogy in a general-purpose reasoning system. *Cogn. Syst. Res.* **10**(3), 286–296 (2009)
16. Wang, P.: *Non-Axiomatic Logic: A Model of Intelligent Reasoning*. World Scientific, Singapore (2013)