World Journal of Advanced Research and Reviews

eISSN: 2581-9615 CODEN (USA): WJARAI

Cross Ref DOI: 10.30574/wjarr

Journal homepage: https://wjarr.com/

(Review Article)

LIDO: A universal language

Taoufik Ben Hassine *

RIADI Lab, National School of Computer Science, Manouba University Campus, Tunisia.

World Journal of Advanced Research and Reviews, 2023, 18(02), 361–364

Publication history: Received on 28 March 2023; revised on 07 May 2023; accepted on 09 May 2023

Article DOI: https://doi.org/10.30574/wjarr.2023.18.2.0826

Abstract

LIDO is a Metamodeling language for developing IoT solutions. It is a set of concepts of high-level abstraction. LIDO makes it possible that people with different technical backgrounds and IoT culture to come together around the development of IoT solutions. LIDO allows people of different disciplines, having different viewpoints, to come together to discuss and reflect on the modeling and implementation of an IoT solution. This article presents the philosophy underlying LIDO that makes it a universal language allowing sustainable development.

Keywords: LIDO; Modelling; Metamodeling language; IoT culture; Universal IoT Language

1. Introduction

Development is only the expression of self and the community of our different capacities to surpass ourselves and our realities. The solutions we advocate for our problems are the expression of our understanding of these problems and our conception of a better reality and a better existing. Solutions are becoming increasingly complex requiring multidisciplinary skills. It is therefore necessary to integrate all vertical viewpoints in a horizontal vision. Development teams are increasingly formed by end customers for whom and therefore by whom solutions are made. These end users are increasingly involved in defining solutions at the goal level and not only at the level of the use of the tool. These users who participate in the development of solutions come from different backgrounds with different concerns and cultures of participation. The right and commitment of all to take part in development reinforces the universality of the developments achieved. To meet, to make a community, to participate, to cooperate, collaborate, we need support and a vector of expression, we need a language. In this article, we resume the discussion around an already defined language called LIDO [1, 2, 3]. We recall in a second section the philosophy underlying this language. The third section describes how to pass from this philosophy to a concrete technologic solution. The conclusion is an intelligent resume of the article.

2. The philosophy of LIDO

LIDO is the result of a long study and multiple reflections around the Internet of Things (IoT) [7] to allow everyone to freely express their thoughts around a problem and its solution. The goal is to allow a multiplicity of solutions and thus increase their wealth. LIDO has been designed to help any developer and end user in their development process regardless of their IoT culture or point of view.

3. Different Cultures

The aim of LIDO is to be a universal language for all and for any development regardless of the cultures of each other. This will strengthen a shared life made up of cooperation agreements and the collaboration of all with all heterogeneous

[*] Corresponding author: Taoufik Ben Hassine

teams. This requires a Meta conception of what we want to be and what we want to achieve together. To this end, LIDO was created from an Internet of Things (IoT) metamodel that leaves no room for details that are difficult to share and is placed at a high level of representation of the realities to be transformed and their transformations.

4. Same Culture, Different Viewpoints

Even in a culturally homogeneous team, the points of view of the participants are different and sometimes contradictory, both in terms of architectural choices and programming techniques. LIDO allows development according to four layers: the Meta Object Facility (MOF) [4], the metamodel of the IoT (MODIDO) [3], the model of the solution derived from the metamodel and the solution implementing its model. We can also have different derivatives of the metamodel, different implementations of the same model hence the great flexibility of LIDO. This allows anyone to share the same concepts and at the same time do his job the way he prefers. This allows everyone to give the best of themselves while sharing the spirit of their team. This allows at the same time individuality and interest for the community. What allows this is a layered architectural design.

5. Metamodeling and Architecture of Models

While a model represents the essence of what is modeled, a metamodel represents the principles that the model follows to conform to that metamodel. In the same way, a meta metamodel defines the contours of a metamodel. Some meta metamodels such as MOF are self-defining, and transcendence is stopped. This defines four (4) layers of modeling. Fig1 illustrates these four layers. The meta-metamodeling layer forms the foundation of the metamodeling hierarchy. The primary responsibility of this layer is to define the language for specifying a meta-model. The layer is often referred to as M3, and MOF is an example of a meta-meta-model. A meta-meta-model is typically more compact than a metamodel that it describes, and often defines several metamodels. It is generally desirable that related meta-models and meta-meta-models share common design philosophies and constructs. However, each layer can be viewed independently of other layers, and needs to maintain its own design integrity. The common basis for all OMG (Object Management Group) [5] meta-models is the OMG Meta-Object Facility [4]. As a framework used for defining specific modeling languages, the MOF specification adopts a four-layer metamodeling architecture, where models specify software, and models are defined as instances of meta-models, which are, in turn, defined as instances of the MOF meta-meta-model. By writing models and meta-models in a common framework, the MOF meta-meta-model, it is easier to perform systematic model/meta-model interchange and integration [34]. MOF provides a meta-meta-model in the upper layer, known as M3 Layer. This M3 model is the language used by MOF to build meta-models, called M2 models. These M2 models describe elements of the layer M1, and therefore, of models M1. The last layer is the M0 layer or the data layer. It is used to describe objects of the real world.

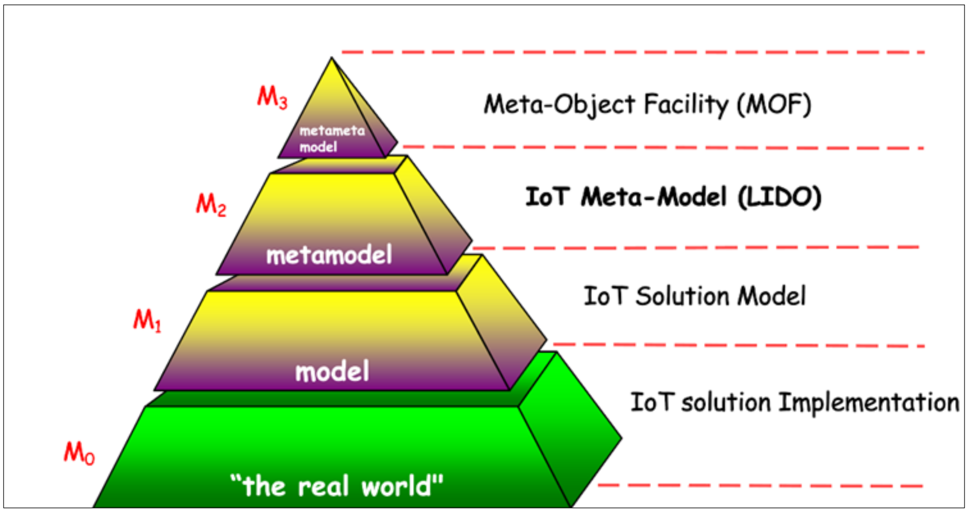

Figure 1 LIDO, an architecture of four (4) Models [1]

In the next section, we teach the reader about the implementation of this philosophy in the development of a company.

6. From philosophy to technology

The question that arises, therefore, is to imagine how we will implement this philosophy of LIDO. We therefore need a method that allows, in the light of this philosophy, to participate in the search for solutions to IoT problems. To this end, an approach has been proposed that offers a development process. In the next paragraph, we present this approach implemented by a development process.

7. The process of Development

The process of development, based on LIDO, is a Model Driven Development (MDD) Process [8]. It is a gradual decrease in the level of abstraction and an increase in the level of concretization. It is a transition from an IoT model to a solution model and then to a solution. The implementation phase comes after the design phase which consists of deriving LIDO to retain the concepts that solve the problem and form the solution. LIDO concepts are expressed in English, but can be expressed in any language, hence the solution is language-free. The vocabulary used by developers in any language they prefer is made up of words composed of syllables each refers to a concept and sometimes to a metaphor both shared, hence the power and universality of LIDO. The conception stage consists of choosing a collection of concepts expressed by LIDO. This collection of concepts must bring the essence or soul of the bricks to be implemented. It is, in a way, a fiction that the director uses to build his film. The actors of this film are none other than the developers. Among them are the end users for whom the dream film is dedicated. The product of the conception stage is a solution model made up of seven (7) views of the IoT solution [1,4]. Indeed, LIDO is a multi-disciplinary language. A solution expressed using LIDO has a user or an interface view, a human-object or a device view, a service view, a communication or a network view, a context view, a localization view and a security view. The implementation stage consists of choosing the technology components that perform the functions required by each view of the solution.

8. Solution implementation and Technology

The approach we proposed to support the development team in the construction of their solution is supported by a software tool called "NTIDO" which offers code generation in java. But nothing prevents NTIDO from being thought and reworked for any other programming language such as C++ or Python. Technologies change with context and time (which is a context component) but concepts last much longer and this is the strength of LIDO. LIDO offers a context view that contextualizes the interaction between the components of a solution. The context is described by independent data structures specifying the behavior of the actors (Human and technological components) involved. LIDO is Technologically neutral. In his “Human-Object View (HOV)”, LIDO talks about “Smart Space” as a space for a collection of “Smart Objects (SO)”. But it did never cite technological details. LIDO talks about security aspects in his “Security View”, but it never referred to any technology. However, NTIDO is technologically dependent. It depends on Java technology.

9. Conclusion

In this article, we present the strength keys of LIDO as a universal language. LIDO is a language of concepts. Concepts are more stable and last longer, because paradigms change infrequently. Concepts are better shared because they do not prejudge realities and ways of doing things. IDO is a language close to humans because it allows people, although different, to agree on a common understanding of IoT. This common understanding makes it easier for people to work together, even if they have different visions of IoT and prefer different technologies and ways of doing so. LIDO allows any developer or development team to have at a high level its own vision made of a collection of concepts and at a low level it allows them to opt for specific technology. LIDO is a metamodel and a language for modeling. A part of LIDO is a metamodel that represents a Megavision made up of generic concepts. Another part allows to define a particular model or vision. The NTIDO tool supports an approach based on LIDO that makes it possible to rely on a particular technology.

Compliance with ethical standards

Funding

There is no financial interest to report. This submitted work is original and has not have been published elsewhere in any form or language.

References

[1] Taoufik Ben Hassine, “Approach to building solutions for the Internet of Things”, Our Knowledge Publishing, September 2022, ISBN: 978-620-5-16059-6.

[2] Ben HASSINE Taoufik, KHAYATI Oualid and Ben GHEZALA Henda. “An IoT domain meta-model and an approach to software development of IoT solutions”. In: 2017 International Conference on Internet of Things, Embedded Systems and Communications (IINTEC). IEEE, 2017. p. 32-37.

[3] Taoufik Ben Hassine. “A Language & an Approach for the Development of IoT Solutions”. American Journal of Electrical and Computer Engineering. Vol. 6, No. 1, 2022, pp. 1-14. doi: 10.11648/j.ajece.20220601.11.

[4] Taoufik Ben Hassine, “A security view for LIDO”, World Journal of Advanced Research and Reviews, 2022, 16(02), 660–664.

[5] Frederic P. Miller, Agnes F. Vandome, McBrewster John, “Meta-Object Facility", VDM Publishing, 2010, ISBN: 6131863687, 9786131863684.

[6] Thomas Gillespie, “Object Management Group”, Emereo Pty Limited, 2014, ISBN: 9781488529481, 1488529485.

[7] Samuel Greengard, “The Internet of Things”, MIT Press, May/20/2015, ISBN: 978-0262527736.

[8] Marco Brambilla, Jordi Cabot, Manuel Wimmer, “Model-Driven Software Engineering in Practice”, Springer Cham (2022) ISBN: 978-3-031-02546-4.

[9] Olga Galinina, Sergey Andreev, Sergey Balandin, Yevgeni Koucheryavy, “Internet of Things, Smart Spaces, and Next Generation Networks and Systems”, Springer Cham (Sep/29/2018), ISBN: 978-3-030-01168-0.