THE ALGORITHMIC-AUTOREGULATION (AA) METHODOLOGY AND SOFTWARE: A COLLECTIVE FOCUS ON SELF-TRANSPARENCY

Renato Fabbri - renato.fabbri@gmail.com

University of São Paulo, Institute of Mathematical and Computer Sciences - São Carlos, SP, Brazil

Abstract. There are numerous efforts to achieve a lightweight and systematic account of what is done by a group and its individuals. The Algorithmic-Autoregulation (AA) is a special case, in which a technical community embraced the challenge of registering their own dedication for sharing processes, self-transparency, and documenting the efforts. AA is used since June/2011 by dozens of researchers and software developers, with the support of different software gadgets and for distinct tasks. This article describes these implementations and statistics of their usage including expected natural properties and ontological formalisms which eases comparative analysis and furthers integration.

Keywords: Distributed development, Social participation, Self transparency, Statistics, Anthropological physics

1. INTRODUCTION

The Algorithmic Autoregulation (AA) is a self-transparency mechanism for sharing processes, documenting efforts, and enhancing personal or collective self-transparency. The purposes for using AA are numerous: enabling automated and fair compensation for dedications, facilitating co-working, introducing newcomers, keeping public historical logs of activities, etc. Indeed, other systems have been designed for such a task (see Section 1.1). A brief characterization of AA is:

- The collective origin, purpose and maintenance. This is a free-culture trait, present within many software, and leads to open software and data as described in Section 4.

- Voluntary logging of messages about ongoing work.

- Enables coordinating distributed team work through individual merit.

- More a practice than a software: AA presents variations on the software support and message composition. Often features found are screencasts, peer validation and periodic messaging.

XX ENMC e VIII ECTM
16 a 19 de Outubro de 2017
Instituto Politécnico Universidade do Estado do Rio de Janeiro – Nova Friburgo - RJ

Transparency in this context should be understood as it is in the State and in enterprises: a public account of activities (Wikipedia, 2017); and not directly as transparency in self-knowledge, as is the case in some philosophical and political contexts (McBride, 2003). One should read Fabbri et al. (2013) for a noteworthy overview of AA as a Global Software Development (GSD) undertake.

1.1 Related work

Authors know of no *civil society transparency* platform. There is a number of transparency initiatives for governments (Kim & Lee, 2012), for religious parties (Habermas, 2006) and for private institutions (Jensen & Hayes, 2006). The data analysis performed in this document comprises methods that are derived from Text Mining (TM) and Complex Networks (CN) constituting a hybrid framework of approaches that are somewhat established (Das Sarma et al., 2010) and novel (Perkins, 2014; Costa et al., 2011).

1.2 Historical note

In June the 7th, 2011, Cleodon Silva (Wikipédia, 2017) died by heart failure. In his memory, the Lab Macambira (LM) group was born (Pedro Macambira was one of this pseudonyms). The AA was conceived as the "cardiac pulse" of the group and is in constant usage since July, 2011. It gathers thousands of messages, tenths of users and hundreds of processes. AA messages present contributions, such as commits to the official repositories of Evince, Firefox, Libreoffice, Puredata, and other software (Fabbri et al., 2013). A number of other activities were registered: new software elaboration and coding, writing of articles, Wikis and Etherpads; articulation of civil society, academic and state instances; studies and reviews. Even so, AA is highly biased towards software development, as can be observed in Section 6, Fabbri (2015), and in the GSD article about AA (Fabbri et al., 2013).

1.3 Essay structure

Section 2 describes AA design. Section 3 describes AA uses incident and envisioned. Section 4 describes different software written or used for AA. Section 5 is dedicated to describing the structure of AA data. Section 6 presents statistics about AA in terms of vocabulary and networks. Section 7 concludes with final remarks, further works, and acknowledgements.

2. AA DESIGN

To understand use practices and software support (Sections 3 and 4), one needs to grasp the core design features of AA:

- evenly spaced messages should be sent by the AA user. The time lapse is called a "slot" and the message a "shout". A slot might also mean the time lapse and the message, this is context dependent and will be disambiguated if needed.

- Shouts should report the task being tackled and/or a briefing of what was done in the slot.

- Shouts might be grouped into "sessions". Each session is ideally linked to a short screencast by the user, with a few dozens of seconds for the explanation about the session.

XX ENMC e VIII ECTM
16 a 19 de Outubro de 2017
Instituto Politécnico Universidade do Estado do Rio de Janeiro – Nova Friburgo - RJ

- Ideally, each session is sent by email to a random AA user for validation.

- Such systematics should be adapted as needed both to generate relevant documentation and to assist the user to keep track of the outcomes of his own dedications.

Variants of this features were conceived and used. Figure 1 presents a diagram shared and referenced by AA users in the first months of AA practice.
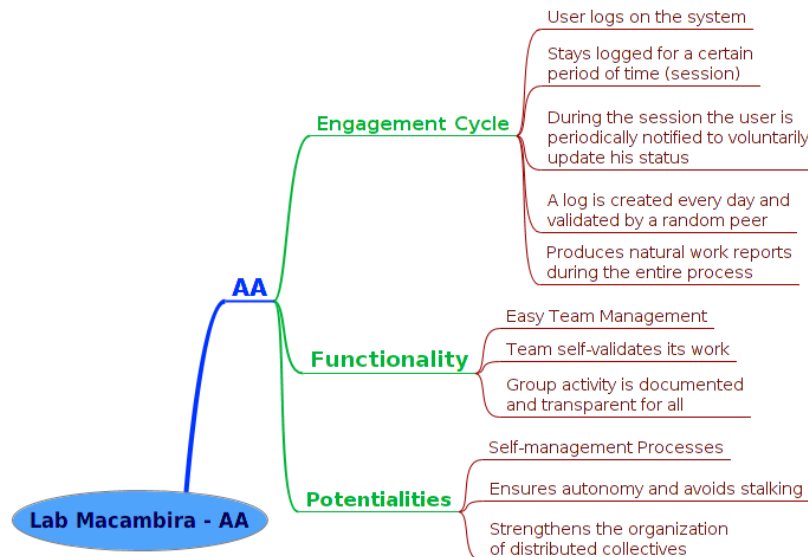

Figure 1: A mind map of the AA methodology shared by users: i) Engagement cycle – the usage of AA; ii) Functionality – the design goals of the system; iii) Potentialities – envisioned benefits of AA by authors of the diagram. As seen in Section 1, core benefits emanate from the self-transparency aspect of AA, with worthy mentions to testifying dedications and sharing processes.

3. USE PRACTICES

Distinct ways to use AA are incident, mostly regarding the design exposed in Section 2. Even those cases which are far from standard can be understood through the angle of AA paradigm. Deviations from the ideal case are always present (see Section 3.5).

3.1 Words and tags

Throughout AA usage, particular words and tags have been used to classify shouts. Of particular interest are:

- Hashtags, such as #aa, #coding and #articulation. These were inherited from Twitter practice.

- Tags starting with "+" sign, such as +django, +sna and +reading. These aimed at a particular usage of tagging within AA, with independence of other systems and easing concurrent use of AA and other social networks.

XX ENMC e VIII ECTM
16 a 19 de Outubro de 2017
Instituto Politécnico Universidade do Estado do Rio de Janeiro – Nova Friburgo - RJ

- Words and abbreviations. Sometimes used in the beginning of shouts, others on the end of them, these also had the purpose of facilitating the categorization of shouts. These cases were sometimes described as tags for entire sessions or for all shouts since tagging, until another tagging was sent by the user (in a shout).

These tagging schemes were also used as a way to enable the "ubiquitous AA", i.e. usage of AA in any social network or communication protocol. The #aao0 tag was used for shouting within Twitter messages and was the most prominent manifestation of the ubiquitous AA. Facebook tagging was also used to indicate posts and comments that were shouts. On some extreme cases, tagging was used in any platform, considering ubiquitous AA implemented, but not yet mined.

3.2 Messages (including shouts)

Messages for AA usage can be of various types. Usually, the type was dictated by the first word of the message. Start messages started a session, while stop messages finished an ongoing session. Push messages sent local sessions (or independent shouts) to a shared database. There was only one automatic message, designated to register a "lost timeslot" of a session (see Section 3.3). Additional messages were used to query for tickets attributed to the user, milestones and other traditional software development managements facilities.

By far the most important AA related message to date is the "shout". Dedicated to expose ongoing tasks, shouts are recurrently envisioned as a structured message, in which the user classifies the shout through special words and tags, and describes ongoing efforts with natural language (usually Portuguese or English). Examples of structured shout proposals are in labMacambira.sf.net (2012) and labMacambira.sf.net (2014). Nevertheless, shouts are used by all AA users, in almost all cases, without such sophisticated structure, but as a plain short natural language description of current efforts, i.e. without classification whatsoever of the message.

3.3 Sessions

AA sessions are collections of shouts. These conventions have had incidence in practice:

- Shouts within a session are input by the user each 15 minutes.

- Considering the 15 minutes grid, the tolerance for shouts in an ideal session is of ± 5 minutes.

- Total duration of 2h per session, in which 8 shouts should outline tasks and technologies.

- A very short screencast is recorded at the end of each session, in which the user described the dedication within dozens of seconds.

- The session is sent to a random user for peer validation in which the session receives a score based on the shouts and the screencast.

Such a session design was very important in the first 6 months of AA, where each of the ± 10 apprentices were performing a session per day. Other users also delivered sessions, but not as regularly. Noteworthy: shouts can be separated by durations different from 15 minutes: example of incident shout separations include 5 minutes, 2 minutes, 30 minutes, 1h. Most shouts are not explicitly related to sessions but still occur in a session-like context. This is

XX ENMC e VIII ECTM
16 a 19 de Outubro de 2017
Instituto Politécnico Universidade do Estado do Rio de Janeiro – Nova Friburgo - RJ

regarded as a consequence of the intuitive usage sessions were aimed for, and as an inheritance of early AA practice. Thus, the term "session" is used as meaning both a session registered as such, and as an arbitrary time-contiguous set of shouts from the same user.

3.4 Accomplishments

Processes registered by AA usage often purposes to accomplish something: write a software or an article; make images, music or research scripts; articulate groups, read technical material, take online classes; etc.

These tasks usually spread through entire sessions. Sometimes, one session embraces multiple tasks. A quite common AA usage is to shout one or just a few messages about current efforts, without much care for regularity or completeness.

3.5 Deviations from the AA paradigm

There are at least three perceived deviations from the idealized AA paradigm, all most often within new users:

- Advertising: shouts containing propaganda about events and groups. This behavior is attributed to both 1) common practice in more commercial platforms, such as Twitter and Facebook; and 2) the outcome of the AA unusual goals and design, which requires acculturation before proper understanding.

- Final product exhibitionism: shouts containing not ongoing processes, but only a media or deed recently completed. Although not considered entirely wrong by users, it does not accomplish the mechanism described in Section 1.

- Introduction to AA, IRC and hacking: handling AA is regarded as empowerment and as an introduction to hacking and open co-working. The first contacts with AA is often marked by playful and test shouts. Although very well esteemed, these messages are also deviations from the AA purpose.

4. Software support

There are mainly three software pieces written to support AA activity. Two of them are a server and client suite each (see Sections 4.1 and 4.2). The third is a fancy dashboard. Among supplementary software support are automated conversational agents (software (ro)bots), used as alternative user interfaces (UIs), with a highlight for the Lalenia bot (see Section 4.4); and an initiative to make AA available in all chat networks (see Section 4.5). All these software items are contextualized in Table 1.

XX ENMC e VIII ECTM
16 a 19 de Outubro de 2017
Instituto Politécnico Universidade do Estado do Rio de Janeiro – Nova Friburgo - RJ

Table 1: All AA versions described and their databases. References marked with (†) are not currently used. Further context is given in Sections 4 and 5.

version name	main languages	user interface	database	code repository	available at
First AA (†)	PHP, Python, Bash	Linux terminal, HTML	MySQL	labMacambira.sf.net (2012, 2011)	-//-
AA 0.1	Python	Linux terminal, HTTP	MongoDB	labMacambira.sf.net (2014)	labMacambira.sf.net (2014,b)
Lalenia bot	Python	IRC	any	labMacambira.sf.net (2011)	labMacambira.sf.net (2011-5)
Ubiquitous AA (†)	Python	IRC, Twitter, Google Chat, Facebook, email, MSN	any	labMacambira.sf.net (2012)	-//-

4.1 First AA: HTTP server, HTML skin and shell client

Although deprecated in favor of AA 01, this first AA software presents the most numerous set of functionalities. The client was completely designed for usage within a GNU/Linux terminal, and the main functionalities are:

- Sending messages to the host.

- Configuration facilities.

- Access to tickets and other software development utilities.

- Timing to ease AA usage, as described in Sessions 2 and 3.3.

Main server functionalities are:

- Receiving shouts and other AA messages through HTTP.

- Registering shouts and other AA messages, received through HTTP, in a MySQL database.

Core HTML interface functionalities are:

- Exhibiting shouts and sessions to other users by common HTML pages.

- Enabling interaction of AA users for reviews and screencast attachments to sessions.

A full description of the features extrapolates the information above and the scope of this article, as do implementation details. Further information of this and other versions of AA are contextualized in Table 1 and Fabbri et al. (2013).

Throughout Jul/2011-Mar/2014, this first version of AA was used directly or routed from bots (see Section 4.4) and other gadgets (Section 4.5).

4.2 AA 0.1

Although there was no online AA software support in the months of April and May, 2014, there was AA activity, as seen in Fabbri (2015). This motivated the creation of a minimum version of AA to support this visceral usage. This minimum version is AA 0.1 (labMacambira.sf.net, 2014). This implementation targeted registering the shout messages independently. All other characteristics were left to data mining and tagging performed by the users. The minimum client has only one feature: a simple HTML call. Integrated trivially as bash commands by scripts and bots. The minimum server has more features:

XX ENMC e VIII ECTM
16 a 19 de Outubro de 2017
Instituto Politécnico Universidade do Estado do Rio de Janeiro – Nova Friburgo - RJ

- it receives the shout with an associated nick and registers it to a MongoDB instance with the time stamp in which the message arrived.

- Returns all shouts as a string or as JSON.

- It has a Heroku Flask app, integrated to an online MongoDB. All these are free online services.

The minimum interface/skin features are part of the server, but listed here for organization:

- Lightest HTML.

- Export as JSON.

4.3 pAAinel

A fancy skin for visualizing AA activity is pAAinel. Core features are visualization of:

- Latest AA shouts.

- Latest IRC messages.

- Embedded Black Duck Open Hub (former Ohloh) analytics.

- Latest commits to LM main repositories.

This Django (Python) software was written for first AA and has not been adapted to AA 0.1.

4.4 Lalenia interface

To ease and enhance the usage and social aspects of AA, the lalenia IRC bot was used as an AA client. This enabled shouts to be logged by IRC users while on the same channel as lalenia. Core features are:

- Users on the same channel as lalenia can log shouts by using the prefix ";aa " to a regular message.

- Returns confirmation that the shout was logged. Returns information about AA and software and concepts if successfully logged. Returns an error message if message was not logged.

- Both first version and 0.1 have supybot plug-ins (lalenia is a supybot (Nanotube, 2015)).

Within information in lalenia logs[1], there were found 1,654 AA shouts that were not in either MySQL or MongoDB databases. Actually, to ease mining, any shout in channel log whose message is identical to any message in all 114,040 messages from MongoDB or MySQL was discarded. Therefore, there was probably a few more shouts in #labmarambira IRC channel log then what is reported here.

---

[1]About the #labmacambira@Freenode IRC channel log, as stated by lalenia, on #labmacambira there have been 172554 messages, containing 6964778 characters, 1066138 words, 4202 smileys, and 10181 frowns; 178 of those messages were ACTIONs.There have been 31906 joins, 680 parts, 31109 quits, 0 kicks, 4 mode changes, and 133 topic changes.

XX ENMC e VIII ECTM
16 a 19 de Outubro de 2017
Instituto Politécnico Universidade do Estado do Rio de Janeiro – Nova Friburgo - RJ

4.5 Ubiquitous AA

Following the route enabled by AA social network interfaces (Section 4.4 reports an example for IRC), the Ubiquitous AA is the expansion of AA to be usable in all social networks and, indeed, to any media in which activity can be registered. There are two approaches to ubiquitous AA:

1. a software that connects to many messaging services as a bot. One implementation connected to all IRC, Twitter, Google Chat, Facebook, email and MSN. This bot usually receives messages and register them as AA shouts, but can encompass more elaborate communication procedures (labMacambira.sf.net, 2012).

2. Tags with which messages are bind to AA activity. This automatically enables AA usage in all social platforms. Messages can be mined for reports and other community usage.

Ubiquitous AA has mythological aspects: it is understood as the receiver part of the Yupana Kernel, a mythological entity that receives and spreads all things (Begalli et al., 2011); and is also understood as a necessary step to human unification (labMacambira.sf.net, 2012; Fabbri, 2013).

5. Data

AA data is scattered among different databases and logs (see Section 4). A coherent integration of these resources is done by means of a dedicated OWL ontology and a mapping routine from relational and NoSQL databases to RDF data.

5.1 The OntologiAA OWL ontology

An ontology, in linked data contexts, is a formalized conceptualization. The Web Ontology Language (OWL) is a family of languages designed for authoring ontologies. Core uses of ontologies include: 1) reasoning by means of ontological specifications; 2) linking data from different sources; and 3) organization of domain knowledge for coherent consideration. For further ontologies in contexts pertinent to AA, the reader should visit Fabbri et al. (2015); Fabbri (2014).

For AA, an ontology facilitates key usage aspects:

- The conceptualization of AA is not unique or steady nor always easy to grasp. The OntologiAA delivers a formal paradigm with which the community can communicate and develop. Figure 2 illustrates the conceptualization formalized in OntologiAA.

- There is AA related data in different databases, and even in social networks (see Section 4). Standard classes to which relate data is a sound method for integration.

- AA data is integrated to other social participation instances, such as Brazilian Presidency (2012-7); Instituto Seva (2012-7), by super-classes and super-properties of the OntologiAA.

XX ENMC e VIII ECTM
16 a 19 de Outubro de 2017
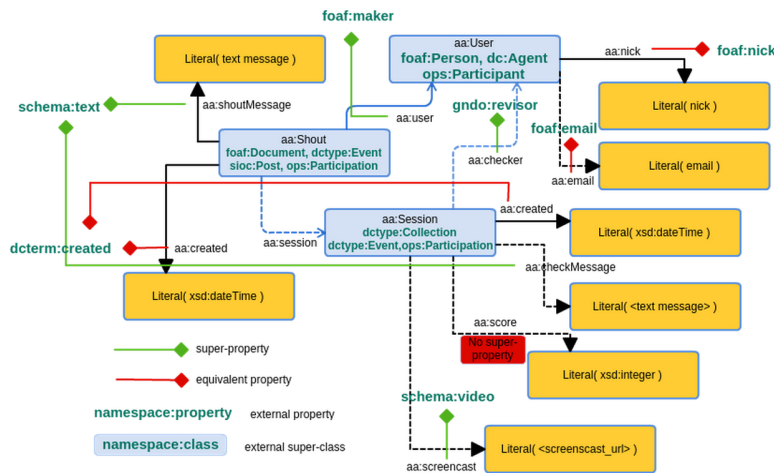*Instituto Politécnico Universidade do Estado do Rio de Janeiro – Nova Friburgo - RJ*

Figure 2: The OntologiAA: an OWL ontology of AA. Classes are concepts related by properties. Classes are also related to upper ontologies (FOAF, Dublin Core, Schema, SIOC, GNDO, OPS), as are the properties. All properties are functional, except for `aa:nick` and `aa:email`. Properties with a full line yield existential restrictions to the subjects of the triples suggested by the diagram: `aa:user`, `aa:nick`, `aa:shoutMessage`, `aa:created`. Blue lines mark object properties, while black lines are for data properties. For further information, see Section 5.1.

.

5.2 RDF data

The AA data is currently in relational and NoSQL databases, and social networks to be mined (Section 4). By using the same ontological background (see OntologiAA in Section 5.1), this data can be translated to RDF for integration and linkage.

The scripts at Fabbri (2015) output RDF from a MySQL database (mostly from first AA version), from a MongoDB database (mostly from AA 0.1), and from IRC logs.

5.3 Linkage to external data

The usage of RDF and OWL protocols enables linked data facilities. The OntologiAA (and thus all AA data) is integrated by two means of special interest:

- Through OPS, AA data is linked to OPa, OBS, VBS, and OCD. These are ontologies with a social participation focus that have been used for data representation and conceptual studies (Fabbri, 2014).

- Through all other ontologies (e.g. FOAC, Dublin Core, Schema.org, SIOC), AA data is linked to the Giant Global Graph (GGG) of Linked Open Data (LOD) (Bizer et al., 2011; Fabbri & Oliveira, 2016).

This linkage gives meaning to data while facilitating data discovery and comparative analysis, to point just a few of the benefits of such approach (Heath & Bizer, 2011).

XX ENMC e VIII ECTM
16 a 19 de Outubro de 2017
Instituto Politécnico Universidade do Estado do Rio de Janeiro – Nova Friburgo - RJ

6. Statistics

From all the AA data, many statistics were obtained: number of messages by type, of sessions, of users, screescants, sessions checked, scores, etc.; activity along time in different scales from seconds to years; most incident words, radicals and sizes of different sets of tokens. All these and some other measures are available in Fabbri (2015). Most importantly, in the histograms it became clear that even in the months when no software were being used to log the shouts, AA kept being used in IRC within the Ubiquitous AA paradigm described in Section 4.5.

7. CONCLUSIONS AND FURTHER WORK

This article describes effectively the AA design goals, the software implementations it received and the usage entailed. A preliminary thorough exposition of this content together with the statistical measures reached 29 pages (Fabbri, 2015) and is better presented herein where the thorough analysis is an auxiliary document.

Potential next steps are:

- implementing mining routines or chat bots to better support the Ubiquitous AA as described in Section 4.5.

- Making the data related to AA available in the linked open data cloud (Bizer et al., 2011; Fabbri & Oliveira, 2016), e.g. through uploading OntologiAA and data to Data Wold or DataHub.

- Deepening our understandings of related paradigms, such as Pomodoro (Gobbo & Vaccari, 2008).

- Relating the data analysis in Fabbri (2015) to data from other online social platforms.

Acknowledgements

REFERENCES

Begalli, M., Fabbri, R. Soares, G. (2011). After all, who is Yupana Kernel. Mutirão da Gambiarra. From http://mutgamb.org/blog/Afinal-quem-e-Yupana-Kernel

Bizer, C., Jentzsch, A. & Cyganiak, R. (2011). State of the LOD Cloud.

Brazilian Presidency (2012-7). ParticipaBR: a Brazilian federal social participation portal. From www.participa.br

Costa, L. D. F., Oliveira Jr, O. N., Travieso, G., Rodrigues, F. A., Villas Boas, P. R., Antiqueira, L., & Correa Rocha, L. E. (2011). Analyzing and modeling real-world phenomena with complex networks: a survey of applications. Advances in Physics, 60(3), 329-412.

Das Sarma, A., Das Sarma, A., Gollapudi, S., & Panigrahy, R. (2010, February). Ranking mechanisms in twitter-like forums. In Proceedings of the third ACM international conference on Web search and data mining (pp. 21-30). ACM.

XX ENMC e VIII ECTM
16 a 19 de Outubro de 2017
Instituto Politécnico Universidade do Estado do Rio de Janeiro – Nova Friburgo - RJ

Fabbri, R. (2013). Nuvens cognitivas e a unificação da espécie humana. Ciberium Revista Eletrônica. From http://wiki.nosdigitais.teia.org.br/Cyberiun and http://cyberiun.tumblr.com/post/64607669758/nuvens-cognitivas

Fabbri, R., Fabbri, R., Vieira, V., Penalva, D., Shiga, D., Mendonça, M., Negrao, A., Zambianchi, L., & Thumé, G. (2013). AA: The Algorithmic Autoregulation (Distributed Software Development) Methodology. RESI. From https://arxiv.org/abs/1604.08255

Fabbri, R. (2014). Social participation ontologies and rules to fuel a social participation linked data cloud. Technical report for the United Nations Development Program. From https://github.com/ttm/pnud5/raw/master/latex/produto.pdf

Fabbri, R.. (2015). The Algorithmic-Autoregulation essay: a collective and natural focus on self-transparency. Github repositories. From https://github.com/ttm/ensaaio/raw/master/ensaio.pdf

Fabbri, R. (2016). The Participation Python package to translate AA, ParticipaBR and Cidade Democrática data to RDF linked data. Github repositories. From http://github.com/ttm/participation/

Fabbri, R., de Luna, R. B., Martins, R. A. P., Amanqui, F. K. M., & Moreira, D. D. A. (2015). Social Participation Ontology: community documentation, enhancements and use examples. arXiv preprint arXiv:1501.02662.

Fabbri, R., & de Oliveira, O. N. (2016). Linked Open Social Database. Github repositories, from https://github.com/ttm/linkedOpenSocialDatabase/raw/master/paper.pdf

Gobbo, F., & Vaccari, M. (2008, June). The pomodoro technique for sustainable pace in extreme programming teams. In International Conference on Agile Processes and Extreme Programming in Software Engineering (pp. 180-184). Springer Berlin Heidelberg.

Habermas, J. (2006). Religion in the public sphere. European journal of philosophy, 14(1), 1-25.

Heath, T. & Bizer, C. (2011). Linked Data: Evolving the Web into a Global Data Space (1st edition). Synthesis Lectures on the Semantic Web: Theory and Technology, 1:1, 1-136. Morgan & Claypool.

Instituto Seva (2012-7). Cidade Democrática: a Brazilian civil society social participation portal. From www.cidadedemocratica.com.br

Kim, S., & Lee, J. (2012). E-participation, transparency, and trust in local government. Public Administration Review, 72(6), 819-828.

Jensen, H.H., & Hayes, D.J. (2006, May). Private sector approaches to secure traceability, transparency and quality assurance in food chains. In Int. Agric. Trade Research Consortium Symp., Bonn, Germany.

labMacambira.sf.net. (2011). Git repository of first AA's server. Gitourious repositories. From https://gitorious.org/macambira_aa/macambira_aa

labMacambira.sf.net. (2011). Lalenia IRC bot. Sourceforge repositories. From http://sourceforge.net/p/labmacambira/lalenia2/ci/master/tree/

labMacambira.sf.net. (2011-5). #labmacambira@Freenode IRC channel. Freenode. From http://webchat.freenode.net/?channels=#labmacambira

labMacambira.sf.net. (2012). Git repository of first AA's client. Sourceforge repositories. From http://sourceforge.net/p/labmacambira/aa/ci/master/tree/

labMacambira.sf.net. (2012). Wiki page for AA. Wiki Nós Digitais. From http://wiki.nosdigitais.teia.org.br/AA#Tags_Propostas

labMacambira.sf.net. (2012). Ubiquitous AA. Sourceforge repositories. From http://sourceforge.net/p/labmacambira/aa/ci/master/tree/omniscientAA/

labMacambira.sf.net. (2012). What is AA (dialog). Sourceforge repositories. From http://sourceforge.net/p/labmacambira/aa/ci/master/tree/aa-what-is-aa

labMacambira.sf.net. (2014). Client of AA01. Heroku. From http://aaserver.herokuapp.com/minimumClient/

labMacambira.sf.net. (2014). Core code repository of AA01. Github repositories. From https://github.com/ttm/aa01/

labMacambira.sf.net. (2014). Readme of current AA. Github repositories. From https://github.com/ttm/aa01/blob/master/README.md

labMacambira.sf.net. (2014). Server of AA01. Heroku. From http://aaserver.herokuapp.com/shout?nick=oNickOuAnonimo&shout=a

McBride, C. (2003). Self-transparency and the possibility of deliberative politics. Journal of Political Ideologies, 8(3), 289-310.

Nanotube. (2015). Supybot: a powerful ITC bot. Sourceforge repositories. From http://sourceforge.net/projects/supybot/

Perkins, J. (2014). Python 3 Text Processing with NLTK 3 Cookbook. Packt Publishing Ltd.

Cleodon Silva. (2017, janeiro 29). Wikipédia, a enciclopédia livre. Retrieved 15:07, janeiro 29, 2017 from https://pt.wikipedia.org/w/index.php?title=Cleodon_Silva&oldid=47863134

XX ENMC e VIII ECTM
16 a 19 de Outubro de 2017
Instituto Politécnico Universidade do Estado do Rio de Janeiro – Nova Friburgo - RJ

Transparency (behavior). (2017, March 19). In Wikipedia, The Free Encyclopedia. Retrieved 18:22, August 23, 2017 , from https://en.wikipedia.org/w/index.php?title=Transparency_(behavior)&oldid=771122692