The Algorithmic-Autoregulation essay
a collective and natural focus on self-transparency

Renato Fabbri
IFSC/USP, Participa.br/SG-PR, labMacambira.sf.net
January 21, 2015

Abstract

There are numerous pursues for a lightweight and systematic account of what is done by a group and containing 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 enhancements, and prove dedication. AA is used since June/2011 by dozens of FLOSS and social developers, with the support of different AA software gadgets and for distinct tasks. Intermittence and activity concentration of users activity follows expected natural properties. Social participation and ontological understandings of AA eases comparative analysis and furthers integration.

Resumo

Keywords: distributed development, FLOSS, social participation, OWL, statistics, anthropological physics

Contents

1 AA start

The Algorithmic Autoregulation (AA) is a self-transparency mechanism for sharing processes, proving dedication, and enhance personal or collective self-transparency. Purposes for AA usage are numerous: enable automated and fair compensation for dedications, ease co-working, introduce 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 upkeep. 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 present features are screencasts, peer validation and periodic messaging.

Transparency in this context should be understood as usual organization or State transparency is: a public account of activities [1]; not directly as transparency in self-knowledge, as is the case in some philosophical and political contexts [2]. One should reach [3] for a noteworthy overview of AA as a Global Software Development (GSD).

1.1 Related work

Authors know of no *civil society transparency* platform. There is a number of transparency initiatives for governments [4], for religious parties [5] and for private institutions [6]. Data analysis methods are derived from Natural Language Processing (NLP) and Complex Networks (CN) fields, constituting a hybrid framework of classical [7, 8] and novel [9, 10] approaches.

1.2 Historical note

7th June, 2013, Cleodon Silva [11] died by heart failure. In his memory, the labMacambira.sf.net 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 official repositories of Evince, Firefox, OpenOffice, Puredata and other software [3]. 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 Sections 6 and 7, and in the GSD article about AA [3].

1.3 Essay structure

Section 3 describes AA uses incident and envisioned. Section **??** exposes different software written or used for AA. Section 5 is dedicated to data. Section 6 further develops statistics about AA in terms of vocabulary and networks. Section 7 states results and Section 8 a concludes with further works and acknowledgements. Tables and figures are inplace, kept as simple briefings and illustrations. External resources - mainly documents, data and scripts - are referenced for further inspection.

2 Design features

To understand use practices and software support (Sections 3 and 4), one needs to observe 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 refer to the time lapse and the message, this is context dependent and will be pointed on text if ambiguity occurs.
- Shouts should report the task being tackled and/or a briefing of what was done in the slot.
- Shouts are grouped into "sessions". Each session is ideally linked with a short screencast by the user, with a few dozen seconds of explanation about the AA session.

- Each session is sent by email to a random AA user for validation.

Variants of this features were conceived and practiced. Figure 1 exposes 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 proving dedications and sharing processes.

3 Use practices

Distinct use methods are incident, mostly regarding the design exposed in Section 2. Even those cases which are not standard can be understood in the light of AA paradigm. Deviations from the ideal case is always present (Section 3.6).

3.1 Words and tags

Throughout AA usage, particular words and tags has 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 particular used of tagging within AA, with independence of other systems and easing concurrent use of AA and other social networks.
- Words and abbreviations. Sometimes used in the beginning of shouts, others on the end of them, these also had the purpose of easing categorization of the shouts. These cases sometimes were pointed as tags for entire sessions or for all shouts since tagging, until another tagging shout was sent by the user.

These tagging schemes was 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 Twitter streaming AA shouts to a considered database as is the most prominent ubiquitous AA manifestation. Facebook tagging was also used to indicate posts and comments that were AA shouts. On some extreme cases, tagging was used in any platform, considering ubiquitous AA implemented, but not yet mined.

3.2 Messages

Messages for AA usage can be of various types, as shown in Table 2. Usually, the type was dictated by first word of the message. Start messages started an AA session, while stop messages finished an ongoing session. Push messages sent local AA sessions (or independent shouts) to a shared database. There was only one automatic message, designated to register "lost timeslot" of sessions (see Section 3.3). Additional messages were dedicated to query for tickets attributed to the user, milestones and other traditional software development managements facilities.

3.2.1 Shouts

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 add a natural language description of what is being done. Nevertheless, shouts are used by all AA users as a short natural language description of what current efforts, without classification whatsoever of the message. Example of structured shout proposals are in [12] and [13].

3.3 Sessions

3.4 Developments

3.5 Suplimentary commands

3.6 Deviations from AA paradigm

4 Software support

There are different software support for AA (Section **??**). Also, This section exposes this diversity and their integration, as linked data, both within AA variants and within participatory instances.

There are mainly three software pieces written to support AA activity. Two of them are a server and client suite each (see Sections 4.0.2 and 4.0.3). The third is a fancy dashboard. Automated conversational agents (software [ro]bots) were used as alternative User Interfaces (UIs), with a highlight for the Lalenia bot (see Section 4.0.4), and an initiative to make AA available in all chat networks (see Section 4.0.5).

All AA software apparatus is contextualized in Table 1.

4.0.1 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. Client functionalities are:

-

Server functionalities are:

-

Core HTML skin functionalities are:

-

Further information of this and other versions of AA are contextualized in Table 1.

4.0.2 PAAinel

4.0.3 AA 01

4.0.4 Lalenia interface

4.0.5 Ubiquitous AA

Table 1: All considered AA versions and their databases. References marked with † are not operational anymore.

version name	main language	user interface	database	git	available at

4.0.6 The #labmacambira@Freenode IRC channel log

4.0.7 Auxiliary scripts

Python script at [?] outputs RDF from a MySQL database, mostly from first AA version. Python script at [?] transcribes a MongoDB database, mostly from first AA version, to RDF data.

5 Data

5.1 The OntologiAA OWL ontology

5.2 RDF data

5.3 Linkage to other participatory data

6 Statistics

6.1 Occurrent activity

Table 2: Registered AAmessages. Operational messages, for signaling session start, stop and publishing local logs (push) are the least abundant. Usage messages with quasi-null semantic content delivers indicative that the user is connected to AA, but no more than that. Messages registering user processes were found to be 34770+1654 = 36424. There were 7504 IRC AA messages, of which 1654 were not registered in databases, probably because of software failures. Automated messages of 'lost timeslot' are the most numerous, with almost half of all messages.

message content	**count**	**type**
push	1718	operational messages = 3936
start	1169	
stop	1049	
empty shouts	92	void messages = 17125
empty alerts	83	
notify	16950	
message shouts	34770	messages about ongoing tasks = 36424
IRC message shouts	1654	
lost timeslot	59863	client automated message
total	115694	all messages are textual

Table 3: Registered AAsessions.

description	value
number of sessions	7288
number of shouts in sessions	20299
number sessions with more than 1 shout	905
number of shouts in sessions with more than 1 shout	13916
number of users in sessions	14
number of checkers in sessions	36
number of screencasts in sessions	295
number of scored sessions	191
average session score	3.18
standard deviation of score	0.74
average number of shouts per session	15.38
standard deviation of number of shouts	19.82
first session from	2011-07-06T03:23:05
last session from	2014-04-01T09:11:36

6.1.1 Time activity

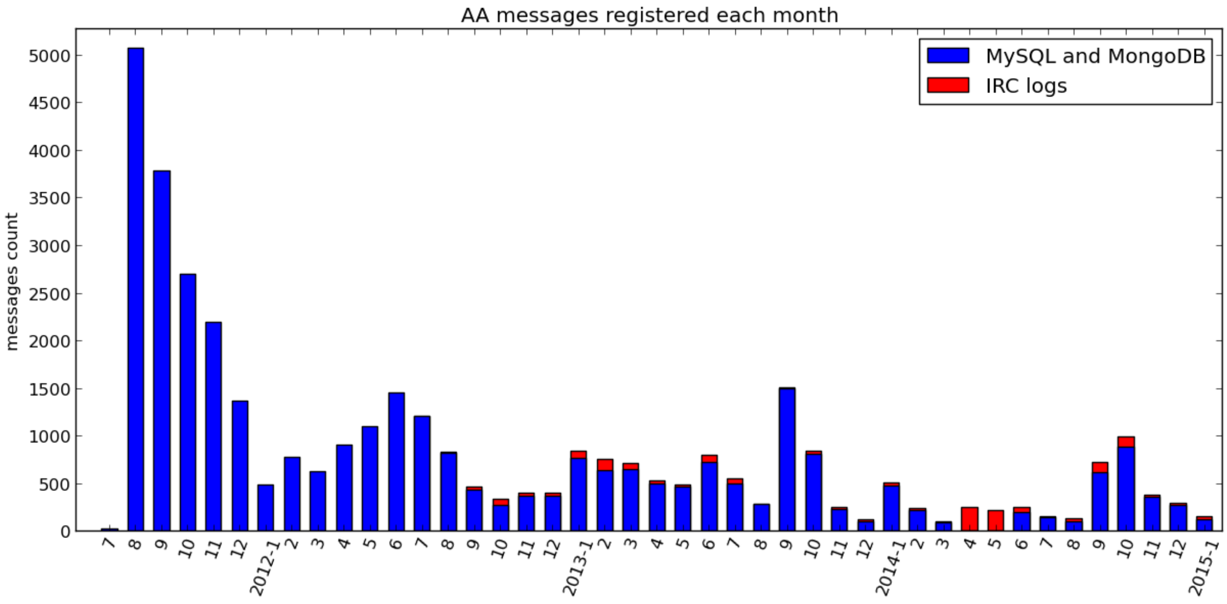

Figure 2: The average number of messages each month is $\mu = 507.1$, with a standard deviation of $\sigma = 336.63$.

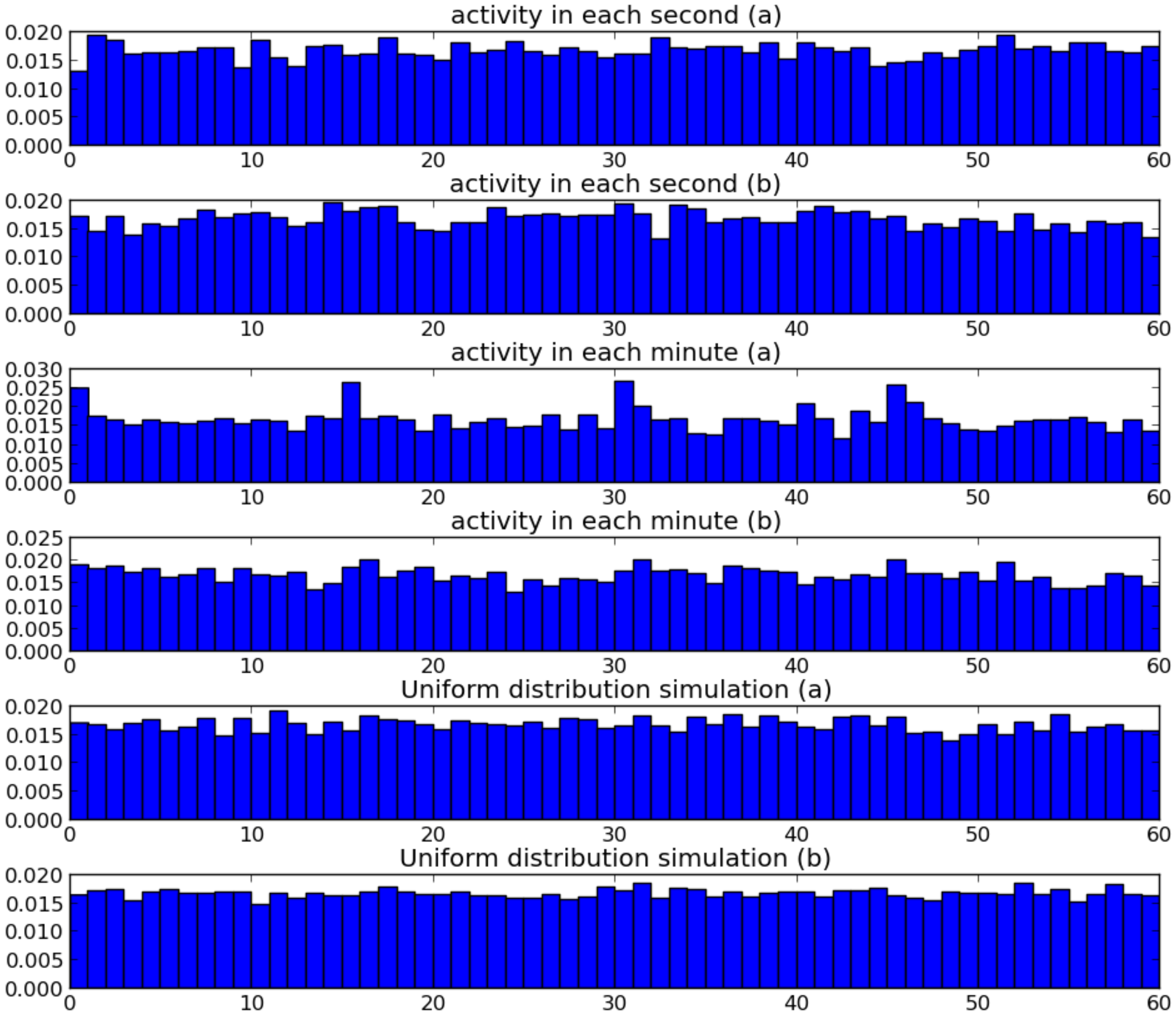

Figure 3: Histogram of activity in AA along seconds and minutes. A strong 15 minutes pattern is visible in (a). The same pattern is apparent in all present histograms, although less incisive. This pattern was not found in mailing lists [**?**], where distribution of activity along seconds and minutes was more homogeneous than Numpy uniform distribution simulator. In AA the scene is the opposite: while simulations delivers $\tau = \frac{max[count(i)]}{min[count(i)]} \approx 1.38$(a) and ≈ 1.26 (b), shouts present $\tau = 1.47$ and $\tau = 1.48$, seconds for (a) and (b), and$\tau = 2.32$ and $\tau = 1.58$, minutes for (a) and (b). Means are considerably bellow 29.5, which might indicate a tendency to shout messages in the beginning of minutes and hours. As these fluctuations among seconds and minutes were not observed in email lists (or other networks, as far as authors know), a hypothesis arises: the tasks at hand and the culture and socioeconomic factors makes timing more prominent. AA itself is time-focused.

6.1.2 User activity

6.2 Dependent activity

6.2.1 Character and token incidence

6.2.2 Time and user dependent activity

6.2.3 String and user dependent activity

6.2.4 String and time dependent activity

6.2.5 Morphosyntactic incidence

6.2.6 Time-related stability

6.3 Network activity

6.3.1 Time user networks

6.3.2 Lexical user networks

6.3.3 Network measures

6.3.4 Network primitive sectioning

6.4 Principal components formation

6.5 Immediate clustering

6.5.1 Users clustering

6.5.2 Words clustering

6.6 Timeslot clustering

6.7 Comparative analysis

6.8 AA, OCD, and Participa.br

7 Results

8 Conclusions

8.1 Further work

8.2 Acknowledgments

References

1. Wikipedia, "Transparency (behavior) — wikipedia, the free encyclopedia." http://en.wikipedia.org/w/index.php?title=Transparency_

(behavior), 2015. [Online; accessed 13-January-2015].

2. C. McBride, "Self-transparency and the possibility of deliberative politics," *Journal of Political Ideologies*, vol. 8, no. 3, pp. 289–310, 2003.

3. R. Fabbri, R. Fabbri, V. Vieira, A. Negrao, L. Zambianchi, M. Mendonça, D. Penalva, and D. Shiga, "Aa: The algorithmic autoregulation (distributed software development) methodology," 2013.

4. S. Kim and J. Lee, "E-participation, transparency, and trust in local government," *Public Administration Review*, vol. 72, no. 6, pp. 819–828, 2012.

5. J. Habermas, "Religion in the public sphere," *European journal of philosophy*, vol. 14, no. 1, pp. 1–25, 2006.

6. H. H. Jensen and D. J. Hayes, "Private sector approaches to secure traceability, transparency and quality assurance in food chains," in *Int. Agric. Trade Research Consortium Symp., Bonn, Germany*, 2006.

7. A. Das Sarma, A. Das Sarma, S. Gollapudi, and R. Panigrahy, "Ranking mechanisms in twitter-like forums," in *Proceedings of the third ACM international conference on Web search and data mining*, pp. 21–30, ACM, 2010.

8. A. Das Sarma, A. Das Sarma, S. Gollapudi, and R. Panigrahy, "Ranking mechanisms in twitter-like forums," in *Proceedings of the third ACM international conference on Web search and data mining*, pp. 21–30, ACM, 2010.

9. J. Perkins, *Python 3 Text Processing with NLTK 3 Cookbook*. Packt Publishing Ltd, 2014.

10. L. d. F. Costa, O. N. Oliveira Jr, G. Travieso, F. A. Rodrigues, P. R. Villas Boas, L. Antiqueira, M. P. Viana, and L. E. Correa Rocha, "Analyzing and modeling real-world phenomena with complex networks: a survey of applications," *Advances in Physics*, vol. 60, no. 3, pp. 329–412, 2011.

11. Wikipedia, "Cleodon silva — wikipedia, the free encyclopedia," 2015. [Online; accessed 13-January-2015].

12. labMacambira.sf.net, "AA wiki." http://wiki.nosdigitais.teia.org.br/AA#Tags_Propostas, 2012. [Online; accessed 21-January-2015].

13. labMacambira.sf.net, "Readme file of current AA." https://github.com/ttm/aa01/blob/master/README.md, 2014. [Online; accessed 21-January-2015].