Towards Government as a Social Machine

Vanilson Burégio[1,2], Kellyton Brito[1,3], Nelson Rosa[1],
Misael Neto[1], Vinícius Garcia[1,2], Silvio Meira[1,2]
[1]CIn-UFPE, Informatics Center-Federal University of Pernambuco, Recife-PE, Brazil
{vaab, ksb, nsr, mwsn, vcg, srlm}@cin.ufpe.br
[2]INES, National Institute of Science and Technology for Software Engineering, Brazil
[3]DEINFO, Federal Rural University of Pernambuco, Recife-PE, Brazil

ABSTRACT
Government initiatives to open data to the public are becoming increasingly popular every day. The vast amount of data made available by government organizations yields interesting opportunities and challenges - both socially and technically. In this paper, we propose a social machine-oriented architecture as a way to extend the power of open data and create the basis to derive government as a social machine (Gov-SM). The proposed Gov-SM combines principles from existing architectural patterns and provides a platform of specialized APIs to enable the creation of several other social-technical systems on top of it. Based on some implementation experiences, we believe that deriving government as a social machine can, in more than one sense, collaborate to fully integrate users, developers and crowd in order to participate in and solve a multitude of governmental issues and policy.

Categories and Subject Descriptors
D.2.0 [**Software Engineering**]: General-*Standards.*

H.4.0 [**Information Systems Applications**]: General

Keywords
Social Machines; Open Government; Web Platforms

1. INTRODUCTION
The notion of "open government" has been around for a long time. Since the 50s, governments have been concerned about transparency and the idea that citizens must have the "*right to know"* [1][2] the government's workings, policies and administration [3]. Since those years, governments agree that freeing government information has the potential to increase accountability, citizen participation and collaboration, while offering better public services to increase efficiency and effectiveness [4][5][6].

Nowadays, the Web has played a fundamental role in the interaction between government agencies and their citizens. This is because it offers powerful means for enhancing government transparency by providing access to information and services online. In fact, the open approach of the Web is one of the main responsible for fostering the idea that government should also be open to public, and then contribute to the widespread engagement of citizens.

As a practical result, many governments around the world have been making different efforts to benefit from Web technologies as a manner to provide *Open Data* and encourage citizens to get more directly involved in governmental issues and policy. The *Open Government Partnership* (OGP, http://www.opengovpartnership.org) is concrete proof of that. However, despite the existing efforts on open government, some technical issues continue to be a major impediment toward the widespread adoption of open data. These issues include, for example, the existence of a multitude of unstructured and outdated datasets, and the lack of standardized services to facilitate not only the consumption, but also the generation and updating of governmental datasets by citizens.

Motivated by these issues and based on some implementation experiences, this paper proposes a social machine-oriented architecture as a way to extend the power of open data and create the basis to design "governments as Social Machines". Social Machine (SM) is an informational paradigm that blends computational and social processes into Web-enabled systems [7]. It proposes a unified model to deal with the complexity of the emerging web around us, and a practical way to explain each and every entity connected to it. The solution proposed herein combines principles from existing architectural patterns and provides a platform of specialized APIs to enable the creation of several other social-technical systems (kinds of social machines) on top of it.

In this scenario, this paper has two main contributions: *(i)* a concise list of problems of current open government initiatives and (*ii*) a practical guideline for deriving government as a social machine. The remainder of this paper is organized as follows. Section 2 discusses some issues of open government initiatives and introduces the social machine paradigm. Section 3 discusses the process of deriving government as a social machine and outlines the main elements of the proposed Gov-SM. Section 4 outlines our practical experience on Finally, conclusion and future work are given in Section 4.

2. BACKGROUND

2.1 Open Government Initiatives and Issues
Different initiatives on open government take advantage of current Web technologies to launch portals of publicly available datasets. As an example of that we can highlight the open data portal (http://www.data.gov) launched by the U.S. government, whose the number of publically available datasets increased by more than 60% last year, reaching around 140,000 datasets by Feb 2015. Similarly, United Kingdom government also opened up its own portal (http://data.gov.uk) that offers access to thousands of open datasets and other additional features as, for instance, a map based search tool. This tool provides a way of searching for records of data sets and services that are referenced by geographical coordinates. In comparison to these efforts, other initiatives are only beginning to grow such as the Brazilian government's open data, whose portal (http://dados.gov.br) provides just about 460 datasets by Feb. 2015. Regardless of some

WWW 2015 Companion, May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3473-0/15/05.
http://dx.doi.org/10.1145/2740908.2743976

initiatives have shown that it is possible to take advantage of e-government and open data [4], [5], [8]–[11], the current approaches have presented problems which range from cultural to technological aspects, as can be seen in some reports on the literature about cases of open government in different countries [12]–[15]. Thus, based on these reports and some implementation experiences of building different applications[1] over Brazilian open data, the following issues could be observed:

1. **Overlapped and decentralized data sources:** although governments try to create central repositories, we could observe that some initiatives at local/regional level have been overlapping the ones at national level and vice-versa. New York City and Rio de Janeiro are examples of cities that conduct their own open data initiatives and portals at local level, while other disassociated efforts (dealing with similar datasets) are launched at national level. As a consequence, developers have difficulties in creating new consistent applications, because they need to do an extra effort to analyze, understand and deal with overlapped data extracted from a multitude of distributed sources.
2. **Lack of standards:** in addition to the overlapped and decentralized data sources, there is a lack of standards for data publishing. Each publisher chooses what and how to publish their dataset. Often, there is no common agreement between countries, states, cities or even within one city or a single government agency. As a result the services provided to consume open data as well as the data formats and types vary significantly.
3. **One-way communication channel:** in general, governments tend to publish data in a one-way communication channel, i.e., from government to citizens. Due to that, most of the existing applications are limited to help citizens only to visualize such data, not being possible to get feedback from them.

2.2 The Social Machine Paradigm

Recently, beyond the aforementioned open government initiatives and the open data momentum, there is another phenomenon on the Web called Social Machine (SM). Broadly speaking, Social Machines are solutions that combine both computational and social processes into a socio-technical system that exploit the large-scale interaction of humans with machines [16]. The SM paradigm represents a natural evolution of Web-enabled systems, and companies such as Google, IBM, Microsoft, and others have newly expressed interest for this topic[2]. In practice, we have been using the concept of SMs as a generic and simple manner to describe[7], design[17] and implement[18] social Web applications like Twitter, Facebook and Ushahidi as a combination of "machines" that 1. have a behavior, 2. communicate and 3. obey certain rules or constraints, while acting over internal and external data. Such machines –or their behavior, communication abilities and constraints– can be embedded in a conceptual abstraction model that, in its turn, can be viewed as a basic, "programmable", network building block. These building blocks socially wrap information processing systems to provide a dynamic set of specialized APIs, available under constraints that are determined by, among other things, the relationships with others[17]. They are the *socially connected computing units* we have used in different contexts to derive social machines from individuals[18], businesses[19] and governments - the focus of this work. In this paper, we provide the steps to derive government as a social machine. The obtained result is a preliminary reference architecture which combines the SM and open data paradigms as a way to extend the power of open government initiatives.

3. DERIVING GOVERNMENT AS A SM

Inspired by Tiropanis et al.'s suggestion of considering government itself as a "social machine" [20], this section provides a general overview of our proposed process of deriving government as a SM (Gov-SM). This process is based on the design guidelines defined in [18] and the implementation experiences we have gained on developing different solutions over Brazilian open data. Next, we describe in more details the specific steps we performed in order to achieve the preliminary reference architecture for the Gov-SM.

3.1 Wrap datasets as individual SMs

The first step to design government as a social machine is to define which representative sources of data should be wrapped as individual social machines, and also how these SMs should be designed. In practice, by using our SM abstraction model, as per detailed in [17], any provider of open data can be considered a kind of an *IPS* (i.e., *Information Processing System*) to be involved by a *Wrapper Interface* (*WI*). Thus, in this case, we designed the whole Gov-SM as a composite social machine internally formed by the combination of multiple sources of data that are socially wrapped as independent and autonomous social machines as well. Each of these SMs has its own identification URL, used to access its provided services. In this way, it is possible to independently deploy each SM and offer its services on different providers. Table 1 lists some internal SMs considered by the governmental social machine proposed herein, including their base identifiers (URL), the wrapped sources of data and their provided data formats, as well as a brief description of each SM. In the base URL, *{host}* represents the service provider on which the SM is deployed. It is worth noting that the SMs listed in Table 1 do not represent a final or complete set of SMs that must be present in all Gov-SMs, since they depend on the context of each country or region, including, for example, the types of data publicly available. However, the identified SMs can serve as a basis for constructing an initial Gov-SM that provide services in different domains.

3.2 Design Data Extraction Mechanisms

As the proposed Gov-SM deals with data from multiple sources, there is an evident need to provide ways of integrating such heterogeneous data. To start with, in the Gov-SM context, we categorize the wrapped datasets into different abstract data types to be handled by the designed SMs. These abstract data types include, for example, Deputy, Senator, Company, HealthUnit, School, TouristicPlace and others. As per presented in Table 1, each social machine manipulates one or more of these abstract data types. However, the wrapped government portals and other websites do not directly provide such abstract data types.

[1] Applications available at http://meucongressonacional.com, http//rio.cidadaointeligente.com, http://cidadaorecifense.com

[2] http://www.sociam.org/partners

Table 1 List of some internal SMs considered to compose our governmental social machine

Social Machine	Base URL	Wrapped Source	Data Format	Description
Deputy-SM	*http://{host}/deputies*	*Chamber of Deputies open data portal*	*json, xml*	*Provides services to manipulate data related to federal deputies.*
Senator-SM	*http://{host}/senatores*	Federal Senate open data portal	*json, csv*	*Provides services to manipulate data related to senators.*
Company-SM	*http://{host}/companies*	*Federal Revenue Service webpage*	*html**	*Provides services to manipulate data related to companies.*
Health-SM	*http://{host}/health*	*National register of health facilities*	*html**	*Provides services to manipulate data related to health units.*
Crime-SM	*http://{host}/crime*	*Brazil Open Data portal and Municipal open data portals*	*csv, xml*	*Provides services to manipulate data related to crime report.*
Education-SM	*http://{host}/education*	*Brazil Open Data portal and Municipal open data portals*	*html*, json, csv*	*Provides services to manipulate data related to schools and colleges.*
Tourism-SM	*http://{host}/tourism*	*Tourism Ministry website and Municipal open data portals*	*html*, json, csv*	*Provides services to manipulate data related to touristic places.*

*Not open data: Human readable web pages

[1] *http://www2.camara.leg.br/transparencia/dados-abertos*
[2] *http://dadosabertos.senado.gov.br/*
[3] *http://www.receita.fazenda.gov.br/*
[4] *http://cnes.datasus.gov.br*
[5] *http://dados.gov.br*
[6] *http://turismo.gov.br*

Instead of that, a variety of different data formats, e.g., csv, xml, html, pdf, json and xls, are available. Thus, it is necessary to have mechanisms to retrieve data out of such publicly available datasets for further data processing and use. In this case, the element *Wrapper Interface* of our SM model was used as the responsible for extracting and converting data from each specific *IPS* (i.e., datasets from dados.gov.br, Federal Revenue Service, cnes.datasus.gov.br and other websites). An example of an individual SM wrapping a dataset, from one of the public sources of data listed by Table 1, is depicted in Fig. 1 (some components are omitted for the sake of simplicity).

As it can be seen in Fig. 1, the element *WI* of the illustrated SM has an *extractor* component that uses *pipes and filters* as an integration pattern to create the logic for collecting and filtering the flows of data from wrapped datasets and/or converting data into common and consistent abstract data types for SM manipulation. In this case, each WI defines a set of interconnected components that performs specific tasks such as loading data from the wrapped dataset, filtering unnecessary data, formatting them to the desired abstract data type and storing them in an appropriate way.

The *extractor* is responsible for converting the wrapped datasets to the format required by the component that encapsulates the core logic to be provided by the SM, such component is called *Business Controller* (BC). The BC is used to support the SM's provided services by retrieving and updating the specific abstract data types to be handled by the SM.

3.3 Specify a common set of specialized APIs

After wrapping representative datasets (step 1) and designing the data extraction mechanisms (step 2), this step concerns the design of the services provided by each SM. Normally, these services are designed as endpoints of a REST API, then a set of common specialized APIs is specified for each social machine. This includes, for example, services like *search, list, get details, report abuse, rank, recommend* and *subscribe*. As an example, the Table 2 shows how some services of the Deputy-SM are specified. The most part of these services are published by the composite social machine (i.e., Gov-SM) as a way to minimize the complexity of third-party applications to consume and handle the existing public datasets. Additional details of that are given in Section 4.4.

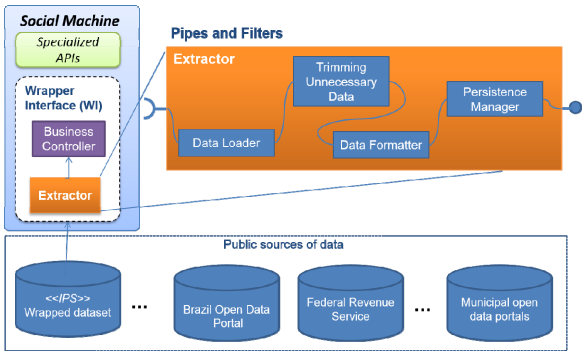

Figure 1. An individual SM wrapping a data source

Two-way communication channel. In brief, the steps presented so far help us to overcome some of the issues presented in Section 2.1, namely *lack of standards* and *one-way communication channel*. The former is reduced through the definition of abstract data types and the common set of specialized APIs as well; and the latter (i.e., *one-way communication channel*) is minimized by some services listed in Table 2, like "*comment"* and "*rate*". In fact, the concept of SM can improve open data through providing dynamic sets of APIs that allow users to not only consume data but also give their feedback. For example, considering the Health-SM, it is important to provide APIs that allow users to not only find the nearest health unit (based on his/her location), but also make recommendations on public hospitals as well as rank their facilities and health services. Other APIs defined by our proposed SMs also enable the implementation of two-way communication between government and citizens, like "*complaint registration"*. The Deputy-SM's service called "*report abuse"*, for example, allows citizens to give a feedback on inappropriate or abusive things related to a specified deputy. Abuse complaints should be stored on the Gov-SM and possibly be redirected to social media as, for example, be posted on the facebook deputy's wall. Further, "investigative reporting" on public accountability has been used as a mechanism to promote user engagement as well as citizens' complaint against corrupt practices.

Table 2. Some Deputy-SM's specialized APIs grouped into common functionalities

Name	HTTP Request	Description
URIs relative to http://*{host}*/deputies		
link	POST /link	Allows third-party apps to connect to this SM and stablish a relationship with it.
list	GET /	Returns a list containing all deputies and their basic info.
	GET /expenses	Returns a list containing current aggregated expenses of all deputies.
	GET /expenses/commisions	Returns a list containing current aggregated data on commissions.
	GET /expenses/laws	Returns a list containing current aggregated data about project laws.
search	GET /search*?q=query*	Return deputies, expenses or laws related to the search query.
get details	GET /{id}	Return basic data of deputy specified by {id}.
	GET /{id}/expenses	Return expenses data of deputy specified by {id}.
	GET /{id}/commisions	Return commisions data of deputy specified by {id}.
	GET /{id}/laws	Return proposed laws of deputy specified by {id}.
comment	POST /comment/{id}	Provide opinion about the deputy specified by {id}.
rate	POST /rate/{id}	Rate the deputy specified by {id}.
report abuse	POST /abuse/*{id}*	Report inappropriate or abusive things related to the deputy specified by *{id}*.
subscribe	POST /subscribe/*{topic}*	Subscribes to a topic of interest specified by *{topic}*.

Push Notification. Beyond two-way communication, the proposed SMs services also allow the establishment of asynchronous communication through push notification. The service *"subscribe"* is an example of that. It allows requesting a subscription on a specific topic of interest, and then the SM notifies the subscriber when the event of interest occurs. Fig. 2 shows an example of a HTTP request to the Deputy-SM's *"subscribe"* service (see its syntax in Table 2). The example is a request for subscribing on a specific topic of interest, i.e., a deputy's monthly expenditure on fuel. The set of parameters (lines 6-10 of Fig. 2) is passed via HTTP post and specifies a notification *constraint* on the Deputy-SM. Such *constraint* indicates that when the specified deputy's monthly expenditure on fuel exceeds 6,000 BRL the callback URL (line 10) should be called by the Deputy-SM, as part of an asynchronous notification process. Other kinds of notification can also be considered such as SMS and email.

```
1  POST  /subscribe/expenditure HTTP/1.1
2  Content-type: application/json
3  Authorization: OAuth
4  ...
5
6  deputyId:71636
7  expenditureOn:FUEL
8  period:MONTHLY
9  whenExceed:6000BRL
10 callbackURL: http://<url_provided_by_the_subscriber>
```

Figure 2. Example of a HTTP request for subscribing on a specific topic of interest

3.4 The "relationship-aware" Gov-SM

Inspired by the reference architecture defined in [18], we also designed the Gov-SM as a combination of different architectural patterns (i.e., *pipes and filters, data federation* and *MVC*) to aggregate and relate data and services from various publicly available sources. The overall obtained architecture is depicted in Fig. 4. It is worth noting that for a better understanding, some details of our SM model were hidden away from Fig. 4, and only the provided services and wrapper interface elements were explored in the high level architecture diagram. Essentially, the Gov-SM defines a unified model to wrap and deal with both structured and unstructured data from multiple disparate public sources of data. Besides, the Gov-SM platform comprises a set of internal SMs (Fig.4) that together provide dynamic sets of specialized APIs in order to support the development of *third-party* applications build on top of Gov-SM's services. The whole system is therefore a "relationship-aware" social machine [17]. That is why it does represent an enabler for creating an ecosystem of possibly related and interacting applications and services. In such ecosystem the relationships between third-party apps and Gov-SM should be established according to the model described in [17]. It is the component *Relationship Manager* the responsible for mediating the establishments of such relationships. Hence, prior to access Gov-SM's services, developers need to perform a registration process in order to create the desired relationship between Gov-SM and his/her application. The steps of this registration process are illustrated by the sequence diagram in Fig.3.

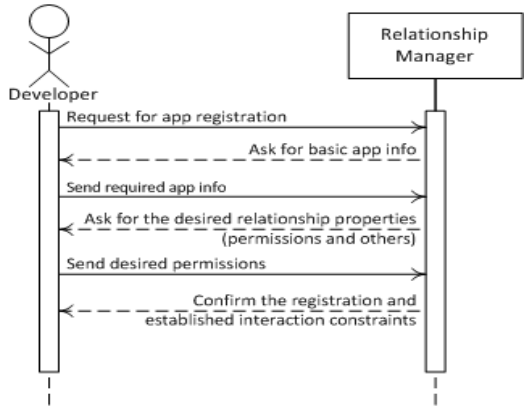

Figure 3. Process of relationship establishment

During the registration process, developers should fill out a form provided by the Gov-SM's *Relationship Manager* which asks for basic information about the application, such as its name, domain, category and so on. The next step, after the developer sends the required data, is to inform the desired relationship properties. In this step, the developer should choose, among other things, the permission his/her application will need. Finally, the confirmation is sent and the established *constraints* (e.g., rate limiting) of the relationship between the registered app and Gov-SM is approved. In this environment, the possibilities of interactions among related parties (i.e., end-users, developers and applications) might potentialize the creation of large-scale social initiatives by combining the existing loosely-coupled SMs in a crowd-powered effort on the Web.

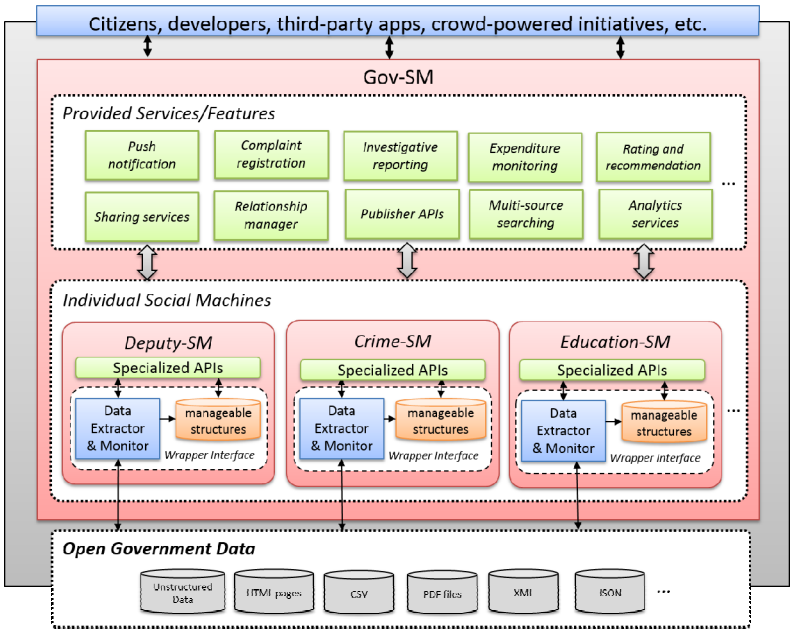

Figure 4. Gov-SM: architecture overview

4. IMPLEMENTATION EXPERIENCE

The proposed approach herein takes into account the practical experience our research group has gained when developing and deploying solutions over Brazilian open data.

Among these solutions we can highlight *Meu Congresso Nacional* (My National Congress, http://meucongressonacional.com) which collects data from several sources to create a dashboard about the Brazilian Federal Parliament, including parliamentarians' expenses, frequency, law propositions, and so on. This project was the winner of the First Brazilian Parliament Hackathon and, in 2014, extended to consider data from the Brazilian Superior Electoral Court. This data included information of all candidates for Brazilian Elections 2014, such as, identification data, properties, campaign donations and received votes. This new feature allows citizens to cross-check the relation between parliamentarians campaign donations, their properties and company donations. Citizens broadly used this application during the Brazilian elections of 2014, being indicated as service that influenced their vote decision. Fig. 5 shows information provided by "Meu Congresso Nacional", relating the amount of money a construction company funded for the main parties competing for 2014 Brazilian presidential elections.

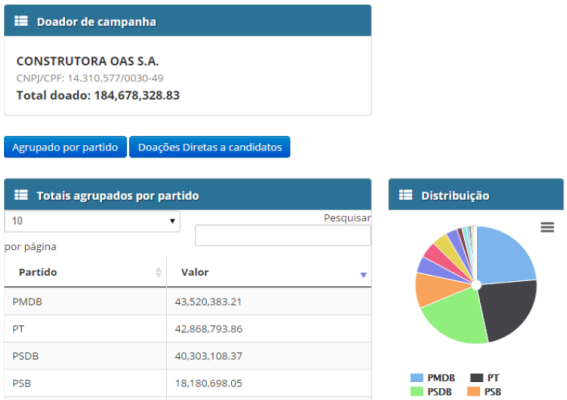

Figure 5. Data provided by "Meu Congresso Nacional"

Rio Inteligente (Smart Rio, http://rio.cidadaointeligente.com) application is another initiative of our group that deals with of Rio de Janeiro's open data on health assistance. This app helps citizens and tourists to find the most appropriated and closest health unit. It was the top 2 application within its domain on the RioApps contest, promoted by Rio de Janeiro City Hall in 2013.

A broader version of Rio Inteligente application is the *Cidadão Recifense* (Recife Citizen, http://cidadaorecifense.com). This application - winner of the contest promoted by Recife City Hall in 2013 - extracts data on several domains, such as financial, education, health and culture in order to support citizens as well as make them aware about public expenditure. Fig. 6 shows a comparative budget in which users can track actual Recife's investment expenditures over time and in different areas.

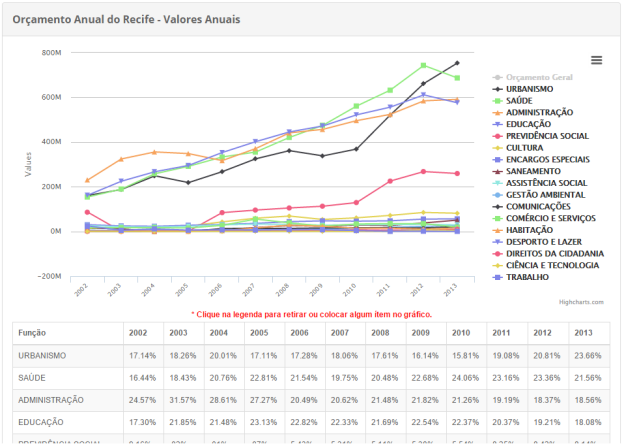

Figure 6. Recife's investment expenditures over time

5. CONCLUSION REMARKS AND FUTURE WORK

There are no doubts that open governments practices need to be revisited in preparation for building a unified platform that indeed

promote transparency, citizen participation, and collaboration. In this paper, we discussed some issues on existing open government initiatives and then used the social machine paradigm to support the process of deriving government as a social machine. By combining computational and social processes into a composite and possibly crowd-powered platform, the SM paradigm can significantly extend the power of open government initiatives, while requiring only a proper combination of existing technologies and patterns. In conclusion, we believe that the proposed approach for open government helps converging the different visions of social machines presented in [7], referred to as "social software", "people as computational units" and "software as sociable entities". The "social software" vision is achieved by implementing applications on top of the governmental social machine with the aim of providing two-way communication channels between governments and their citizens (as *users*), leading to different levels of social interactions between them. The *"*people as computational units*"* vision is achieved by using the resultant crowd-powered platform as the basis to launch different kinds of initiatives that encourage the *crowd* to solve numerous governmental issues and policy. Last but not least, the "software as sociable entities" vision is achieved by providing dynamic sets of specialized APIs that naturally conduct to the establishment of an ecosystem of possibly related and interacting applications and services, built by *developers* with a passionate interest in a more effective public oversight. Thus, in more than one sense, the architecture proposed herein can offer different avenues of possibilities that converge to the fully integration of *users*, *developers* and *crowd* in order to participate in and solve a multitude of current and future governmental problems in diverse areas. The Gov-SM platform is an important step not only for, in the present, [re]thinking government as an "administrative machinery", but also for creating a solid technological foundation that, in the near future, guides us **towards government as a [true] "Social Machine"…**

6. ACKNOWLEDGMENTS

This work was partially supported by the National Institute of Science and Technology for Software Engineering (INES, www.ines.org.br), funded by CNPq and FACEPE, grants 573964/2008-4 and APQ-1037- 1.03/08.

7. REFERENCES

[1] W. Parks, "Open Government Principle: Applying the Right to Know Under the Constitution," *Georg. Wawhingt. Law Rev.*, vol. 26, no. 1, 1957.

[2] D. Mitchell, "The constitutional right to know," *Hastings Constit. Law Q.*, vol. 4, p. 109, 1977.

[3] J. Little and T. Tompkins, "Open Government Laws: An Insider's View," *North Carol. Law Rev.*, vol. 53, p. 451, 1974.

[4] W. Wong and E. Welch, "Does E-Government Promote Accountability? A Comparative Analysis of Website Openness and Government Accountability," *Governance*, vol. 17, no. 2, pp. 275–297, Apr. 2004.

[5] D. S. Sayogo and T. Harrison, "Effects of the internet and sociocultural factors on budget transparency and accountability," *Proceedings of the 13th Annual International Conference on Digital Government Research*. ACM, College Park, Maryland, pp. 11–20, 2012.

[6] T. Harrison and D. Sayogo, "Open budgets and open government: beyond disclosure in pursuit of transparency, participation and accountability," in *International Conference on Digital Government Research*, 2013, pp. 235–244.

[7] V. Burégio, S. Meira, N. Rosa, and V. A. A. Buregio, "Social Machines: a Unified Paradigm to Describe Social Web-oriented Systems," *22nd Int. World Wide Web Conf. (WWW 2013 Companion)*, pp. 885–890, May 2013.

[8] J. C. Bertot, P. T. Jaeger, and J. M. Grimes, "Using ICTs to create a culture of transparency: E-government and social media as openness and anti-corruption tools for societies," *Gov. Inf. Q.*, vol. 27, no. 3, pp. 264–271, 2010.

[9] T. B. Andersen, "E-Government as an anti-corruption strategy," *Inf. Econ. Policy*, vol. 21, no. 3, pp. 201–210, 2009.

[10] S. Kim, H. J. Kim, and H. Lee, "An institutional analysis of an e-government system for anti-corruption: The case of OPEN," *Gov. Inf. Q.*, vol. 26, no. 1, pp. 42–50, 2009.

[11] S. Picazo-Vela, M. Fernandez-Haddad, and L. F. Luna-Reyes, "IT's alive!!: social media to promote public health," *Proceedings of the 14th Annual International Conference on Digital Government Research*. ACM, Quebec, Canada, pp. 111–119, 2013.

[12] W. Jho, "Challenges for e-governance: protests from civil society on the protection of privacy in e-government in Korea," *Int. Rev. Adm. Sci.*, vol. 71, no. 1, pp. 151–166, 2005.

[13] D. Dada, "The Failure of E-Government in Developing Countries: A literature review," *Electron. J. Inf. Syst. Dev. Ctries.*, vol. 26, no. 7, pp. 1–10, 2006.

[14] S.-Y. Hung, C.-M. Chang, and T.-J. Yu, "Determinants of user acceptance of the e-Government services: The case of online tax filing and payment system," *Gov. Inf. Q.*, vol. 23, no. 1, pp. 97–122, 2006.

[15] S. J. Piotrowski and G. G. Van Ryzin, "Citizen Attitudes Toward Transparency in Local Government," *Am. Rev. Public Adm.*, vol. 37, no. 3, pp. 306–323, Sep. 2007.

[16] N. R. Shadbolt, D. A. Smith, E. Simperl, M. Van Kleek, Y. Yang, and W. Hall, "Towards a classification framework for social machines," pp. 905–912, May 2013.

[17] V. A. Burégio, S. L. R. Meira, N. S. Rosa, V. A. Buregio, and V. C. Garcia, "Moving Towards ' Relationship-aware ' Applications and Services: A Social Machine-oriented Approach," in *17th IEEE International EDOC Conference (EDOCW 2013)*, 2013, pp. 43–52.

[18] V. Burégio, L. Nascimento, N. S. Rosa, and S. Meira, "Personal APIs as an Enabler for Designing and Implementing People as Social Machines," in *23rd International World Wide Web Conference (WWW 2014 Companion)*, 2014.

[19] V. Burégio, Z. Maamar, and S. Meira, "An Architecture and Guiding Framework for the Social Enterprise," *IEEE Internet Comput.*, vol. 19, no. 1, pp. 64–68, 2015.

[20] T. Tiropanis, A. Rowland-Campbell, and W. Hall, "Government as a social machine in an ecosystem," pp. 903–904, Apr. 2014.