Publicaciones en Ciencias y Tecnología
Vol 7, N^{0}1, Ene–Jun 2013, pp.51–72, ISSN:1856-8890, Depósito Legal: pp200702LA2730

SELF-ORGANIZATION AND EMERGENCE PHENOMENA IN WIKIPEDIA AND FREE SOFTWARE DEVELOPMENT USING MASOES

[*] Niriaska Perozo [†] José Aguilar
[‡]Oswaldo Terán [§] Heidy Molina

Recibido: 05/03/2013 Aprobado: 16/06/2013

Abstract

This work models Wikipedia and Free Software Development through a multiagent architecture for self-organizing and emergent systems called MASOES without mathematically representing the system. In that sense, each component, mechanism and process of MASOES is instanced at individual and collective levels by the observed phenomena at the modeled systems. Thus, this paper proposes a methodology to show how to model real systems using MASOES, in order to study their self-organizing and emergent properties and, later on, to facilitate the verification of these properties, mechanisms, components and social interactions for promoting collaborative work and sharing individual and collective knowledge in these systems.

Keywords: Multiagent systems, Self-Organization, Emergent Systems, Wikipedia, Free Software Development.

[*] *Unidad de Inteligencia Artificial, Decanato de Ciencias y Tecnología, Universidad Centroccidental Lisandro Alvarado, Barquisiimeto, Venezuela, nperozo@ucla.edu.ve*

[†] *CEMISID, Facultad de Ingeniería, Universidad de los Andes, Mérida, Venezuela, aguilar@ula.ve*

[‡] *CESIMO, CEMISID, Facultad de Ingeniería, Universidad de los Andes, Mérida, Venezuela, oteran@ula.ve*

[§] *CEMISID, Facultad de Ingeniería, Universidad de los Andes, Mérida, Venezuela, heidym@ula.ve*

AUTO-ORGANIZACIÓN Y EMERGENCIA EN WIKIPEDIA Y EL DESARROLLO DEL SOFTWARE LIBRE A TRAVÉS DE MASOES

Resumen

Este trabajo modela el comportamiento de Wikipedia y el desarrollo de Software Libre, a través de una arquitectura multiagente para sistemas emergentes y auto-organizados llamada MASOES, sin especificar matemáticamente el sistema. En ese sentido, cada componente, mecanismo y proceso de MASOES se instancia a nivel individual y colectivo en cada uno de los sistemas modelados. Así, en este trabajo se propone una metodología para mostrar cómo modelar sistemas reales utilizando MASOES, con el fin de estudiar sus propiedades emergentes y auto-organizadas, y posteriormente, facilitar la verificación de estas propiedades, mecanismos, componentes e interacciones sociales para promover el trabajo colaborativo y el intercambio del conocimiento individual y colectivo en estos sistemas.

Palabras clave: Sistemas Multiagente, Auto-Organización, Sistemas Emergentes, Wikipedia, Desarrollo de Software Libre.

Introduction

The Web has made possible the arising of a new knowledge production model built around a participation architecture [1], exploiting the possibility of obtaining small contributions at a very low cost from a large and diverse group of collaborators, in order to produce information products and services. This production model is based on a form of social self-organizing, and productive activity, very different from the centralized way of the control systems [2]. Its comprehension is important in order to understand how small local contributions can solve very complex problems, and how thousands of collaborators coordinate and organize themselves, among other things. Some projects based on this production model of knowledge are Wikipedia and Free Software Development. Some questions about these projects are: which are the mechanisms that both of these projects possess for promoting self-organization and emergence? How can we model and study them for analyzing these aspects? In order to face these questions, the multiagent modeling approach can be used. Thus, each of these systems would be seen as a society of agents that interact in order to cooperate autonomously, where each agent would have a macroscopic behavior derived of their local interactions at this macro-scale, the collective behavior of the system is what matters. The flexible way in which agents operate and interact (with each other and with the environment) in such projects makes them ideal for modeling dynamic and unpredictable scenarios.

The current multiagent methodologies have focused on handling the microscopic aspects alone, such as how agents interact, their rules, without explicitly handling the macroscopic behavior required or the management

of individual and collective knowledge involved [3]. There are other works that, aware of this, have made contributions to this respect, such as a general methodology consisting of a vocabulary to describe self-organized systems, and a control mechanism based on restrictions of agents behavior, to try to describe the synergy amongst the elements and the friction (conflicts) that could result from competing for limited resources, for instance [4]. In [5], the creation of a methodology to design self-organized systems based on Unified Process is also presented. Here, the development of solutions is carried out by incorporating (macroscopic) variables that describe that macroscopic behavior and the adjustment or calibration of certain key parameters within the system. This paper is only a starting point, there is not a guide for choosing macroscopic variables.

In this sense, we suggested MASOES, a multiagent architecture for designing, modeling and studying emerging and self-organized systems [6]. This architecture describes the elements, relationships and mechanisms, both at the individual and the collective levels, which determine the emerging, self-organizing phenomena in a system, without mathematically modeling the system. MASOES, in relation to the described works, considers both microscopic and macroscopic aspects of a system; that is, it manages generated knowledge either at the collective or at the individual level. Also, it allows each agent to change its behavior, guided by its emotional state. In this paper, MASOES will be used to study self-organization and emergence phenomena in Wikipedia and Free Software Development, with the purpose of showing its usefulness, generality and flexibility in diverse contexts. Besides, these systems are studied due to their capacity to generate emerging behavior such as: formation of communities, division of tasks, definition of collective rules, generation of high quality content, among others. Finally, we propose a methodology to model real systems using MASOES, in order to study their self-organizing and emergent properties.

MASOES

Our architecture is divided into two levels: *individual and collective* (see figure 1). **Collective cognitive emergence** arises from three interaction levels: **Local Interaction Level**, which might be direct or indirect (via the environment);**Group Interaction Level**, involving social networks or structured groups; and, **General Interaction Level**, which includes the entire community of agents. With respect **to individual cognitive emergence,** the idea is to produce cognitive emergence imitating the way in which human beings go from unconscious to conscious; handling the agents behavior at 3 different levels and establishing a hierarchy of behav-

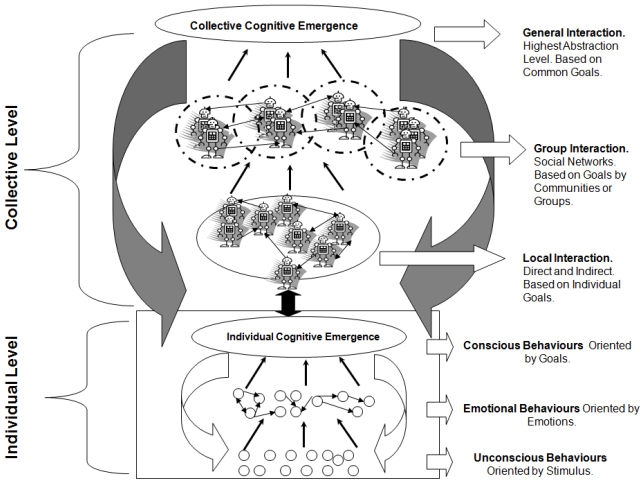

Figure 1: General Architecture of MASOES.

iors: **Unconscious Behavior** or reactive; **Emotional Behavior** guided by emotions; and **Conscious Behavior.** Each agent changes its behavior (behavior-switching) dynamically, guided by its emotional state in a given moment. We propose an emotional model which will be used in the decision-making process to choose a type of behavior in particular [9].

MASOES proposes the following components at the collective level (see figure 2): a) **Agents**; b)**Feedback Mechanisms** (Positive and Negative); c)**Aggregation Mechanism**; and d)**Direct and Indirect Interactions.** At the individual level, the architecture has four components and other general elements (see figure 3). These main components are: a) **Reactive Component**; b)**Cognitive Component**; c)**Behavior Component;** d)**Social Component**; and e)**Type of Emotion** (more details about components and mechanism can be found in [6]).

The affective space for MASOES is represented through positive and negative emotions oriented by the achievement of personal goals or by the actions of other agents or changes in the environment [9]. The behavior component will prioritize and select one type of behavior over the others dynamically, depending on the agents emotional state, according to the following rules established in [9]:

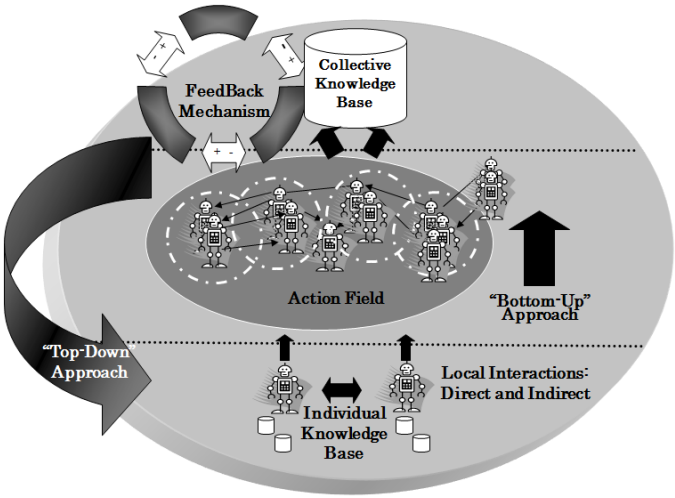

Figure 2: Components of MASOES at Collective Level.

Rule 1: If `< Emotional State >` is Positive then
`<Priority_Imitative_Behavior>`

Rule 2: ElseIf `<Emotional State>` is Slightly Negative then
`<Priority_Cognitive_Behavior>`

Rule 3: ElseIf `<Emotional State>` is Highly Negative then
`<Priority_Reactive_Behavior>`

Phases in the knowledge management in MASOES

The phases for the knowledge management in MASOES consist of a circular cause-effect process that reflects the process of creation, conversion, integration and diffusion of knowledge at individual and collective level: a) **Socialization**; consists of sharing experiences through local interactions, and requires**turning implicit knowledge into explicit transferable concepts.** b) **Aggregation**; the agent**creates trustworthy explicit knowledge** through exchange of points of view, meetings, etc. c) **Appropriation**; consists in **translating explicit knowledge into implicit one.**

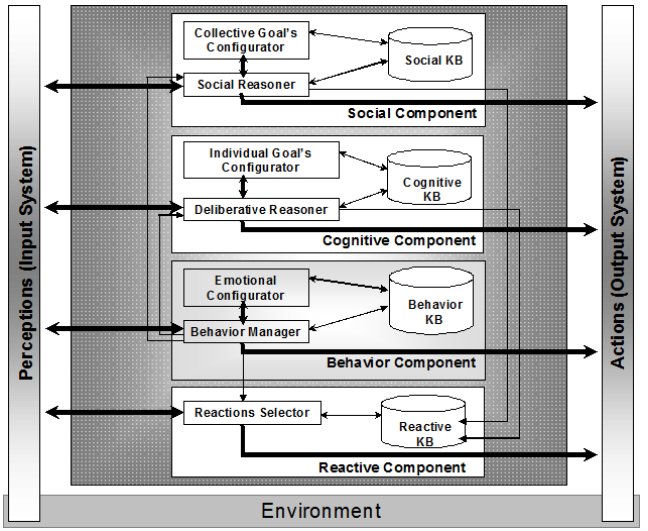

Figure 3: Components of MASOES at Individual Level.

Instantiation of real systems using MASOES

For modeling a real system using MASOES, it is necessary to describe the involved elements, processes and mechanisms at the individual and collective levels. Thus, we propose a methodology consisting of four steps:*analysis, design, integration and verification.*

I)**Analysis Stage:** The basic characteristics of the modeled system, such as agents, tasks and interaction levels are described:

I.1. To identify and to describe the types of agents, and the tasks these agents assume in the modeled system.

I.2. To describe the interaction levels in the society of agents of the modeled system, including:

a) Local Interaction.

b) Group Interaction.

c) Global Interaction.

II) Design Stage: In this step, the individual and collective levels of the modeled system are designed, as follows:

II.1. Individual Level.

a) To describe and represent the individual components to consider in the modeled system: there are four components at the individual level: reactive, cognitive, social and behavioral, according to MASOES.

b) To associate a behavior or a set of behaviors (reactive, cognitive, or imitative) to each agent type defined in the modeled system.

II.2. Collective Level.

a) To describe the collective components of the system, this includes:

- Set of interaction rules.
- Action Field.
- Knowledge bases.
- Collective goals.

b) To describe the collective processes, including:

- Social networks formation.
- Feedback mechanisms.

III) **Integration Stage:** In this step, the individual and collective knowledge of the society of agents is integrated, as it was defined in step II. This integration is modeled following the phases of knowledge management, described in section 3. This step is fundamental, since knowledge management in MASOES permits visualizing the synergy between the individual and the collective knowledge, which will confer adaptative capacities to the system and the arising of some properties such as: *new collective policies and norms, cooperation between agents, and creation of groups or communities* as final product of the emergence and self-organization in the system.

IV) **Examination Stage:** The model based on MASOES about the real system must be validated. That is, for the cases where we know that the real systems have emergent and/or self-organizing properties, we need to verify if their models based on MASOES can determine their emergent and/or self-organizing properties, as those observed in these real systems. A verification method for carrying out this step has been proposed and evaluated in [10], based on the use of the Wisdom of Crowds Paradigm (WCP) [11] and Fuzzy Cognitive Maps (FCM) [12], which is not presented in this work. However, for using this verification method is mandatory to model the real system via the methodology proposed in this work, i.e. accomplish the stages of analysis, design and integration before as shown later with two case studies.

First case study: Wikipedia

Wikipedia results of a collective work, where each component, each article, arises from multiple contributions, which usually are improvements and extensions occurring from a first draft. In this section, Wikipedia is described using MASOES,

AGENT	DESCRIPTION	SOME TASKS
Developer	It is an agent involved in the maintenance of the servers and/or the development of Wikipedias software. In addition, they grant system privileges to the "administrators" and "bureaucrats".	To create portals on specific topics. To develop code for improving MediaWiki. To make tutorial or documentation pages. To create templates and algorithms. To maintain servers. To grant privileges to administrators and bureaucrats. To block and to unblock IP's. To participate in voting for candidates for outstanding articles, country of the week, candidacy to administrator, etc.
Steward	These agents possess the same responsibilities as Bureaucrat agents, and are additionally capable of changing any given agents role. Besides, they are the ultimate arbiters on the content of Wikipedia.	To name and eliminate other sysops and bureaucrats. To arbitrate in serious conflicts on the content of Wikipedia.
Bureaucrat	It is a special class of sysop who is able to name or to eliminate other sysops and bureaucrats. The existence of the bureaucrats is for alleviating the tasks of the developers.	To name and to eliminate other sysops and bureaucrats.
Administrator or Sysop (System Operator)	It is a Wikipedian that can access some of Wikipedias softwares restricted functions. The sysops control themselves and each other; almost all the powers of the sysops are completely reversible by any other sysop (including deleting and blocked IP directions).	To erase pages and images. To see and to recover erased pages and images. To block and to unblock anonymous users IPs. To block and to unblock registered users. To protect or to block a page, as well as the inverse functions. To edit protected or blocked pages. To revert pages quickly. To edit the space of names of MediaWiki. To mediate in conflicts. To close debates for deleting. To fight vandalism. To participate in voting for candidates for outstanding articles, country of the week, candidacy to sysop.
Registered User	It is an agent who has created his login (or nickname) and a password. He can have a list with his contributions. Also he can give information on himself through a page, facilitate a contact email address and have a discussion page" from where other users can comment or establish dialogues. The contributions of a registered user are identified with his nickname in the historic file of articles.	To acquire experience in the use of techniques for syntax and edition, in the use of templates. To maintain his personal page. To interact with other users through his discussion page. To personalize aspects of appearance of Wikipedia and the article edition environment. "To watch" certain articles (that are brought up to his own list of follow-up?) in order to check the changes introduced in them and take part when he considers it necessary. To transfer an article (necessary for fusing pages). To edit articles or discussion pages. To request article deletion. To fight vandalism. To demonstrate his good faith, through useful contributions during a period of time. To participate in voting for candidates for outstanding articles, country of the week, candidacy to sysop, consultations of deletion. To verify Copyright.
Bot User	These agents are like software robots which run both autonomously and manually in order to perform repetitive tasks. In addition, these are users that have been created by any Registered or Administrative User in Wikipedia.	To update and to improve many of the topic pages by reducing link redundancy. Creating new pages based on composite information, and to spell check entries.
Anonymous User	These agents have not registered with a login and a password in the system. They can edit almost any article or discussion page but they do not have certain functionalities. Their interventions are identified in the historic file of the article through their IP of access.	To acquire experience in the use of techniques for syntax and edition, in the use of templates. To edit articles or discussion pages. To ask for article deletion. To fight vandalism. To verify Copyright.

Table 1: Agents with some of their tasks in Wikipedia

following the methodology proposed in the previous section, particularly, the first three stages:*analysis, design and integration.*

I) Analysis Stage

I.1 Agents and their Tasks in Wikipedia

Modeling Wikipedia through MASOES means to consider some agents as actors with diverse roles and tasks at individual or collective level. Specifically, from the point of view of the "privileges of system" there are seven types of users in Wikipedia with a defined hierarchy:*anonymous, bots, registered, bureaucrat, steward, administrator and developer*. Wikipedians interact in a common space (Web environment), and each participant uses the same editor and obeys the same set of rules (see table 1).

I.2Interaction Levels

There are 3 levels of interaction:

- **Local**. Wikipedians interact with each other, contributing with their knowledge and abilities. Each agent acts in agreement with the local information and objectives. These local interactions can be indirect via the article, discussion and policies pages, portals and boards; and direct via email or chat channels (IRC, "*Internet Relay Chat*"), among others.
- **Group**. Wikipedians interact following the norms and objectives of the community they belong to, i.e., Wikipedians of the English-speaking community follow the norms and objectives of Wikipedia in the English language. This group is variable, not predefined and maintained by its own members. Furthermore, Wikipedians group themselves within the community by topic or area of interest, or in committees to solve specific problems.
- **General**. It represents the highest interaction level where the interactions are among the existing multilingual communities (English, French, Spanish and Portuguese, among others) and with other Wikimedia Foundation projects, such as Wikibooks, Wiktionary and Wikiversity, among others. The foundations projects are coordinated in order to reach their general objectives. With this interaction, the existing communities and projects will be able, for instance, to emulate policies and actions which have been successful for others in the foundation.

II) Design Stage

II.1.Wikipedias Components and Processes at the Individual Level

In this section, we show the Wikipedias components and processes at the individual level. In Wikipedia, the agents must be defined as having neutral attitude, as established at the Wikipedia pillars [13]. This also can be made from MASOES, since MASOES emotional model considers three attitudes: neutral, positive, and negative with respect to others. In this case, neutrality will help to reduce friction, to increment the individual satisfaction,

INDIVIDUAL COMPONENT	REPRESENTATION IN WIKIPEDIA
Behavior	The agents in Wikipedia activate their behavior depending on the situation they face and the emotional state being handled at a given time. The emotional state (which measures the degree of motivation and commitment of each Wikipedian) will allow the behavioral component to dynamically carry out changes of the agents behavior. An example of this could be an edition war (defined by Wikipedia as 3 text editions by a particular user in a given article within 24 hours, amid other users editions) which could provoke a conflict among those, and thus, different types of emotions and behaviors: an emotional state highly negative, if the problem leads to a verbal fight (a reactive behavior associated); an emotional state slightly negative, if the agent prefers evasion (a cognitive behavior associated), and finally, an emotional state positive if the agent tries reconciliation (a imitative behavior associated).
Reactive	There are monitoring mechanisms in Wikipedia to ensure that a page or a set of pages maintain their quality. A person willing to maintain these pages will be notified in case of changes, allowing him or her to react in cases of vandalism, demonstrating reactive behavior in case of a negative emotional state such as ire is reached in agreement with the MASOES affective model. Also when a wikipedian makes a mistake such as copyright infringement, and it is punished by the community, reactive behavior is triggered and guided by a negative emotional state such as depression.
Cognitive	It is represented by the cognitive mechanism of each agent and its individual objectives. Whenever there are changes in the published content, the wikipedian makes use of their knowledge and experience to edit, evaluate, discuss and improve the content.
Social	It will store the important knowledge about guidelines, objectives, actions and results of the collective activities. Besides, it will store through a set of rules, the necessary knowledge to manage the editions, the portals, and creation of articles, in other words, knowledge about how to use Wiki technology, among other things. For example, the guidelines that are established by the community to fight vandalism, to avoid conflicts, and to use the discussion boards. Thus, Wikipedians try not to violate community standards as are described in the policies and terms, persuade those incurring in faults, and collaborate in the learning process of new participants that try to imitate more experienced Wikipedians.

Table 2: Individual Components of MASOES in Wikipedia.

COLLECTIVE COMPONENT	REPRESENTATION IN WIKIPEDIA
Set of Rules	This set of rules is made up of all the rules collectively established such as: article-editing rules (e.g. no deletion of useful material, use of discussion pages) and the rules for social interaction with other wikipedians (e.g. wikitags, collaborating with new users, making useful, polite comments), among others. According to [18] by the year 2007, Wikipedia had 20 general rules, 21 points of etiquette (about how to work with others) and 42 policies (developed by the community to describe best practices, clarify principles and resolve conflicts, among others).
Action Field	It is thanks to the set of direct and indirect interactions that the Wikipedians delimit their action field within the Wikipedia environment. In fact, this action field is made up of the pages of its community where each member participates, contributes and shares his knowledge with the rest.
Collective Knowledge Base	Here is the content of articles generated by the collective. Besides, the set of common or collective rules for the edition, communication, promotion and administration are also stored.
Collective Objective	To generate reliable, open, free, verifiable content in a specific language (English, Spanish and French, among others), following the norms and policies of the project.

Table 3: MASOES Collective Components in Wikipedia

and to intensify in the same magnitude, both, the negative and the positive emotions. Additionally, in accordance with the wikipedians developed tasks, it is convenient that the agents develop the three types of behavior proposed by MASOES (reactive, cognitive and imitative), and that they can switch from one to another dynamically, in accordance with their emotional state, as it happens in reality. Thus, each Wikipedian agent will have the 4 individual components of MASOES (see table 2).

II.2.Wikipedias Components and Processes at the Collective Level

The involved components and processes can be seen in tables 3 and 4 respectively. In the case of collective processes we shall describe the formation process of social networks and some of the mechanisms used in Wikipedia for this purpose. III) Integration Stage:

III.1.Phases for Knowledge General Management in Wikipedia The three phases of our architecture for the knowledge management are instanced (*socialization, aggregation and appropriation*) in table 5.

Instantiating the analysis, design and integration stages in Wikipedia, it can be affirmed according to MASOES that the modeled system has the key components and processes, at the individual and collective levels, in order to generate emergent and self-organizing behavior at the macro level.

COLLECTIVE PROCESS	REPRESENTATION IN WIKIPEDIA
Formation of Social Networks	We shall consider a social network as an open, horizontal system, grouping a number of people identified with similar needs and problems and which, additionally, work together with intense social interaction in order to maximize resources and contribute to the solution of problems [19]. Social interaction in Wikipedia takes place in a spontaneous and autonomous way, and is formalized in social networks through the creation of communities (*groups of Wikipedians by language*) and committees (*small groups of Wikipedians based on collective objectives, created to make specifics functions within a given community*) with the goal of establishing common interests and aims. This formation takes place day to day since Wikipedians are conscious of the fact that they are who organize the functioning dynamics, who decide what the daily work is, and who evaluate the results. From this point of view, social networks in Wikipedia are self-organized, self-diagnosed and self-evaluated. These social networks allow Wikipedians to act in a grouped manner in order to reach their collective objectives, and to contribute with general objectives of the project, as for example, to have a multilingual encyclopedia.
Feedback Mechanisms	With respect to content generation, there are mechanisms for promoting the generation and depuration of content. The involved feedback mechanisms in Wikipedia are: • **Collaborative learning mechanism:** This collective explicit knowledge is generated by the contributions made by members through a*not directed mechanism of collaborative learning* (without instructor), that is, under the responsibility of each one. In that way, Wikipedians decide what to do, to learn, to discuss, to accept or to reject, but following a set of norms established also by them through the editor, portals and boards. • **Mechanism for getting quality and diffusion:** mechanism for aggregation, filtration and refining of contributions, e.g. reviews, improvements, discussions, deleting and publication of outstanding contributions. • **Mechanism for Rewards:** Mechanism to motivate and reward wikipediansoutstanding contributions, e.g. recognition of outstanding contributions. • **Mechanism of Punishment:** Mechanism to punish disobedience of established rules, e.g., blockade of pages, expulsion or blockade of members, among others.

Table 4: MASOES Collective Processes in Wikipedia

Second case study: free software development

In this section we shall characterize, at an individual and a collective level, the components and the processes involved in Free Software Development (FSD), following the first three phases of the methodology proposed in section 4:*analysis, design and integration* of MASOES. The Linux Kernel Development Community (LKDC) is without a doubt the product most widely used by researchers in different disciplines, to exemplify Free Software community processes and behavior, because of its high level of stability, quality, maturity and good organization, among other features [14].

I) Analysis Stage:

PHASE	REPRESENTATION IN WIKIPEDIA
Socialization	Wikipedians must make explicit their knowledge to the rest of their community through the creation, modification and elimination of articles, among other contributions. When a Wikipedian decides to share his knowledge, he participates in the article edition through the editor of Wikipedia, following the established norms. Wikipedias software facilitates the storage of the contributions, the IP address or names of the contributor and other additional data such as: date, hour, and version. In addition, it allows each Wikipedian to edit following a pre-defined structure (template), which facilitates the transformation from implicit knowledge to structured and transferable forms. To increase the security of acting members, Wikipedia has a space called "Test Zone" where Wikipedias beginners (anonymous or registered users) have the possibility of practicing without causing any damage.
Aggregation	It involves collective processes for revision, depuration, deletion and publication of contributions in relation to generated articles and policies or norms. A Wikipedian action or participation frequently generates reactions from other Wikipedians. Thus, creating an article, for example, causes other members to read it, review it, accept it or reject it. When an article is reviewed, it can be selected and nominated as an outstanding article, erased, or, its author asked to extend it. The collective revisions support: classification of articles by area, their connection to other related articles, the print request of the best articles in WikiPress, the generation of new policies or categories, the delivery of awards to the best authors and the increased reputation of some Wikipedians within the project.
Appropriation	Wikipedians must register in a community, read articles, and learn about: policies, how to use Wikipedias editor, and how to interact or to communicate with other members by means of Wikipedias resources such as wikis pages, mail and chat, among others. This learning process takes place by trial and error (that means the more they practice, the more skills or abilities they acquire) and, at the same time, it stimulates the participation of Wikipedians due to acquisition of self-assurance for participating.

Table 5: Wikipedia through the Knowledge Management Phases

I.1.Agents and their Tasks in FSD Every Free Software community is started by a person or a small group of developers, who make available on the web, to the public in general, the source code of their application or library. The person who started the community is called Project Leader. Free software developers and users are represented by diverse agents that interact and have reactive, emotional and cognitive behavior when acting in the community. When the software product has a version, it is evaluated for different users/programmers; this evaluation allows it to evolve (e.g., include new improvements), making it more likely for other internet users to choose to use it. There are other activities such as: documenting, translating and administrating contributions. In table 6, each agent with its description and some tasks are displayed.

I.2. Interaction Levels Free Software communities, unlike traditional software development teams are seldom physically together in a common geographic location, thus, we can say that there are different levels of interaction:

AGENT	DESCRIPTION	TASKS PERFORMED
Project Leader	Developer of the nucleus and responsible for coordinating activities and functionalities that might be added to the software. Additionally, he supervises the changing team of developers.	Coherently group the functionalities that need to be developed for the release of a new version of the project. Coordinate and collaborates in prioritizing the failures that need to be attacked. Be the voice of the community. Finally, accept the code being released, and notify the release of official versions.
Maintainer or Administrator	A developer with high privileges since he has reading and writing permission in the projects code repositories. He exercises emerging leadership which rises from daily tasks in a community. This type of developer is a regular resource coordinated by the project leader.	Regularly participate in the development of new functionalities and solutions of complex failures. Participate in architectonic design decisions carried out in relation to the project. Decide the orientation to follow in developments in collaboration with other developers. Answer technical questions in forums. See critical errors the software might have. Coordinate the development of one or more Kernel modules or sub-systems. Grant writing and/or reading permission to developers over the repositories.
Developer	User with technical knowledge about the project, that participates voluntarily and partially in the solution of failures or aggregation of new functionalities.	Determine new features or failures that the software might have. Solve basic code problems. Send solution proposals to be evaluated by the team of developers.
Error Notifier	User that through experience from using the software detects errors and notifies them to the community for solution. It also leaves a record in the community of the need for new software functionalities.	Pick up errors. Identify the need for new functionalities. Record and characterize errors as well as update the list of errors appearing in the error management system.
User	Internet user that out of curiosity or a simple wish to participate in the community downloads the project code (source, or binary) and starts to use it in its regular activities.	Download the source or binary software code. Study its documentation. Use the product to satisfy its needs. It could turn into an Error Notifier or a Developer at any given time.

Table 6: Agents and tasks involved in the LKDC.

- **Local:** Local interactions among agents can be at least two ways: directly via emails and instant messaging, among others. Indirectly, as they occur via the source code between whoever wrote a portion of code and someone interpreting it in order to learn it or solve an existing problem with it.

- **Group:** The concept of modularity in the design of Free Software promotes the grouping of developers by sub-system or project module, which also allows the specialization of developers in tasks (e.g., design and implementation of the user graphic interface, data structures and documentation) or functionalities (e.g., memory management, processing and storage).

- **General**: Global interactions can take place when messages are sent to the projects mailing list, with information on most of the announcements, discussions and debates occurring during the kernel development. Another way of interacting is by publishing web content aimed at every member of the community. Lastly, it is possible to find interactions through the use of a versions control system that allows the distribution, revision and control of contributions to the code made by the community (http://www.kernel.org).

II) Design Stage:

II.1.FSDs Components and Processes at Individual Level The individual level components defined by MASOES for the LKDC (reactive, cognitive, social and behavioral) are shown in table 7. In general in the LKDC, it is convenient that all agents manage the three types of behavior suggested by MASOES (i.e., reactive, cognitive and imitative), and can change it dynamically in accordance to their emotional state in a certain moment.

II.2.FSDs Components and Processes at Collective Level The components and processes involved at collective level are described via MASOES in tables 8 and 9.

III) Integration Stage:

III.1.Phases for Knowledge Management in FSD Table 10 shows how Socialization, Aggregation and Appropriation phases are represented in Free Software Communities, specifically in the LKDC.

Like the Wikipedia case, with the instantiation of the analysis, design, and integration stages in the FSD, specifically in the LKDC, we can affirm that the modeled system has the key components and processes, both at the individual and at the collective level, in order to show emergent and self-organizing behavior at the macro level according to MASOES. Particularly, these two cases have some similarities, such as: different agent types with task specialization among them, meritocratic mechanisms for decision making, norms of behavior established by the participants of the project themselves, stigmergic features (this aspect is addressed in the conclusions) used for temporally tracking the data and attracting the attention of the participants. These similarities probably appear because the two systems rest on internet as the production platform. Among the differences between the two systems we can basically say that they are reflected by the hierarchy established among the different participants, and in the mechanisms of aggregation, diffusion and learning used, among others

Conclusions

With the models presented of Wikipedia and LKDC using MASOES, we can tell, among other things, that MASOES is independent to the application domain, and

it is easily usable in order to describe social systems as multiagent systems, aiming at studying/ determining its emerging and self-organizing properties/components. MASOES is a generic architecture with the aim of providing the designer of systems with a user-friendly, fine grained process description, a tool that can be applied in the modeling of social systems in different contexts in order to determine if the system presents self-organization and emergence, and, additionally, to characterize the key components and processes to generate such an emergence and self-organization at the macro level.

In accordance to the modeling done through MASOES, the two systems have the necessary components and mechanisms to generate emergent and self-organizing behavior. Particularly, both Wikipedia and the LKDC have stigmergic features [15, 16, 17], which mimic the behavior of insect societies. These stigmergic features are present in both systems because they have a high number of agents (and indirect interactions) who stimulate the others to participate through work carried out, as a result, these agents are limited to react and to imitate the actions of the group. As has been shown, these features can be modeled without problem using MASOES. Another aspect of interest in MASOES is the possibility of considering the emotional state of the agent to dynamically change its behavior, what represents an important difference in the modeling of these types of systems, comparing with other approaches.

The modeling of these systems, through MASOES, requires the verification of them in order to confirm their quality but each one represents a good example for showing how to use MASOES via the methodology proposed in this work. Besides, with these models we can study the emergent and self-organized behavior in real systems (last phase of the methodology). For this purpose, a verification method has been proposed in [10], based on the use of the Wisdom of Crowds Paradigm (WCP) [11] and of Fuzzy Cognitive Maps (FCM) [12]. In future works we will model through MASOES the collective behavior of the pedestrians, in an attempt to capture the characteristics and properties of another type of system that, despite of displaying self-organized and emerging properties, is not based on a collaborative architecture, as is the case of Wikipedia and FSD. Moreover, it will interesting use this methodology for modeling other complex systems like an aquatic system [20] and comparing the final results and interactions between collective and individual components based on MASOES.

References

[1] T. O'reilly, Open source paradigm shift, http://tim.oreilly.com/archives/ParadigmShift.pdf, 2009.

[2] E. Raymond, The Cathedral & the Bazaar, Musings on Linux and Open Source by an Accidental Revolutionary, Revised Ed. OReilly Media, ISBN: 0596001088, 2001.

[3] F. Zambonelli & A. Omicini, Challenges and Research Directions in Agent-Oriented Software Engineering, Autonomous Agents and Multiagent Systems, Vol. 9, p.p. 253-283, 2004.

[4] C. Gershenson, A General Methodology for Designing Self-Organizing Systems. Multiagent Systems session of the CSS'05 Conference in Liverpool, 2005.

[5] T. Wolf & T. Holvoet, Towards a Methodology for Engineering Self-Organising Emergent Systems, SOAS, p.p. 18-34, 2005.

[6] N. Perozo, J. Aguilar & O. Terán, Proposal for a Multiagent Architecture for Self-Organizing Systems (MA-SOS), Lecture Notes in Computer Science, vol. 5075, p.p. 434-439, 2008.

[7] S. Camazine, J. Deneubourg, N. Franks, J. Sneyd, G. Theraulaz, & E. Bonabeau, Self-Organisation in Biological Systems, Princeton University Press, 2001.

[8] R. Sun, Cognition and Multiagent Interaction, From Cognitive Modeling to Social Simulation, Edited by Ron Sun, Rensselaer Polytechnic Institute,Cambridge U. Press, 2005.

[9] N. Perozo, J. Aguilar & O. Terán, Un Modelo Afectivo para una Arquitectura Multiagente para Sistemas Emergentes y Auto-Organizados (MASOES). Technical Journal of the Faculty of Engineering University of Zulia, Vol. 35, Nro. 1, p.p. 1-11, 2012.

[10] N. Perozo, J. Aguilar, O. Terán & H. Molina, A Verification Method for MASOES. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, Vol. 43, pp. 64-76, 2013.

[11] J. Surowiecki, Wisdom of Crowds, New York: Random House, 2005.

[12] J. Aguilar & J. Contreras, The FCM Designer Tool. Fuzzy Cognitive Maps: Advances in Theory, Methodologies, Tools and Applications, Ed. Glikas Mixalis, Springer, pp. 71-88, 2010.

[13] Wikipedia Project, The Five Pillars. http://en.wikipedia.org/wiki/Wikipedia:5P, (October, 2012).

[14] J. Gonzalez & G. Robles, Introducción al software libre, Technical Report, ESCET, Universidad Rey Juan Carlos de Madrid, http://www.uoc.edu/masters/oficiales/img/693.pdf, 2003.

[15] C. Xiaohui, J. Beaver, L. Pullum, J. Treadwell & T. Potok, A Stigmergy Approach for Open Source Software Developer Community Simulation, In Proceedings of Symposium on Social Computing Applications, Canada, 2009.

[16] A. Ricci, A. Omicini, M. Virola, L. Gardelli & E. Oliva, Cognitive Stigmergy: Towards a Framework Based on Agents and Artifacts, Lecture Notes in Computer Science, LNAI Vol.4389 p.p. 124-140, 2007.

[17] G. Robles, J. Merelo & J. Gonzalez-Barahona, Self-organized Development in Free Software: A Model Based on the Stigmergy Concept, In Proceedings of the 6th International Workshop on Software Process Simulation and Modeling, USA, 2005.

[18] C. Goldspink, Normative self-regulation in the emergence of global network institutions: the case of Wikipedia. ANZSYS Conference, New Zealand, 2007.

[19] M. Rizo, Redes. Una aproximación al Concepto. Universidad Autónoma de México. CONACULTA, UNESCO, 2003. Disponible [http://sic.conaculta.gob.mx/centrodoc_documentos/62.pdf].

[20] N. Fernandez & C. Gershenson, Measuring Complexity in an Aquatic Ecosystem. Proceedings of the CCBCOL 2013, 2nd Colombian Computational Biology Congress, Disponible [http://arxiv.org/pdf/1305.5413v1.pdf].

INDIVIDUAL COMPONENT	REPRESENTATION IN THE LKDC
Behavioral	Agents within the LKDC display one type of behavior or another depending on the situation they are facing and the emotional state they might be in at a given time. The emotional state will allow the behavioral component to dynamically perform behavior change in the agent and to measure the degree of motivation and commitment of each agent in the project. There are situations that best illustrate this, for example the presence of discussions in the mailing list, for instance, some with elevated tones, provokes the activation of a negative emotional state in those involved, who could execute cognitive or reactive behavior according to the intensity of the emotion. Other example is the inclusion of a developer within the credits of a software version, due to the high quantity and quality of the contributions made to it, will result in the activation of a positive emotional state in the developer that would trigger imitative behavior in order to continue to reproduce the actions that have a high degree of satisfaction.
Reactive	Rejection to contributions could cause depression and triggering reactive behavior according to the MASOES Affective Model. In the LKDC, social reputation plays an important role for developers. That reputation is gained because they contribute in the module where they participate, so they can attain peer recognition and thus be promoted within the community. These are examples of reactive behaviors.
Cognitive	The cognitive component lets us carry out the process of collective knowledge appropriation for the community. It is represented by each agents learning mechanism. An agent uses its individual knowledge when there is an error present in the code and it evaluates the way to correct it within the LKDC, or when making decisions regarding the architectonic and functional design of the software product, decisions which require a process for evaluating and choosing the best alternative. These are examples of cognitive behaviors.
Social	The social component allows observing the successes and mistakes of other members of the community and learning from them to imitate their behavior and use their experiences. For example, the announcement of a new kernel version trigger the need of testing the code (the error notificators), and then send to the developers the errors found. On the other hand, agents avoid conflicting behavior such as self-promotion within a group, or assuming leadership of a module, without first providing contributions evidencing knowledge and ability to take leadership.

Table 7: Individual MASOES Components in the LKDC

COLLECTIVE COMPONENT	REPRESENTATION IN THE LKDC
Set of rules	Members of the LKDC have norms that guide the interactions in the community. There are rules for sending mail to the lists, error notification, code changes, among others. For example: before asking a technical question over email, on a news group or on a web sites forum page, it is necessary to proceed as follows: • Try to find the answer by reading the manual, the FAQ (Frequently Ask Questions), doing a web search or asking a more experienced friend. • Carefully choose the forum, avoiding posting a question in the wrong forum or posting a very basic question in a forum where more complex technical questions are expected (or vice versa), or simultaneously posting the same message in very different news groups. • Write clearly paying attention to spelling and grammar. • Send questions in an easy format. • Use specific, meaningful titles. • Carefully and clearly describe the problem or the error symptoms, as well as the setting in which they take place. The code must also be complete and thoroughly documented. • Send contributions in diffs (a command that allows listing of differences between two files).
Action Field	The code developed in the LKDC is the area of greatest aggregation in the field, where agents, through their interactions affect the environment and vice versa, in order to motivate their participation in the Project.
Collective Knowledge Base	It is made up of the code repositories, the mailing list and forums files, the existing documentation, the FAQ list, among others.
Collective Objective	To successfully achieve software development with a high level of quality and functionality in open code, following a decentralized and distributed methodology under its own rules, which dictate the generation of code as well as collective and individual participation.

Table 8: MASOES Collective Components in LKDC

COLLECTIVE PROCESS	REPRESENTATION IN THE LKCD
Social Network Formation	Social interaction in the development of Free Software takes place spontaneously and autonomously through the creation of communities (groups of developers and users of a specific software product), with the aim of establishing common interests and objectives. This formation starts out with a product's publication and then, as it is improved, it gets the interest of new people who need to cover a necessity, are simply curious, or are willing to participate in the community. Participants in the LKCD are grouped according to the Kernel's functionalities such as: input and output devices, network connections, storage and memory, among others.

Table 9: MASOES Collective Processes in the LKCD.

Feedback Mechanisms	The generation of code involves a series of mechanisms for its production, learning, diffusion and documentation. At the same time, it involves a series of reward and punishment mechanisms in order to influence the behavior of each agent involved. In the LKDC, these mechanisms are: • **Mechanism for Production of Contributions:** The way to produce contributions to the code is by improving the software based on the needs expressed by users (new features, errors found), and also by detecting and correcting errors or new software surfacing in the market, which would require the corresponding drivers in open code. Another way of contributing is through the documentation of the generated code. • **Mechanism of collaborative learning:** The learning mechanism takes place through self-explained code, documentation for configuring and using the software, and mailing lists as a means of interaction for the entire community. The aim of mailing lists in the LKDC is to allow every member to intervene or, at least, remain informed about what takes place inside the community regarding discussions and decisions taken in the community. On the other hand, the idea is to constantly share information on solutions to problems brought up through these lists, this is a way to share knowledge and achieve understanding from member collaborations. Currently, the main LKDC list (kernel@vger.kernel.org) keeps a file that can be found at the lkml.org site. Even though the general discussion on Linux kernel development takes place in the LKML, there are literally dozens of other popular mailing lists whose discussions address Linuxs functionalities. • **Code Diffusion Mechanism:** The code is released through Linuxs central kernel repository, which is kept at kernel.org, and through mailing lists, in order to be revised and corrected by developers and users. From Linuxs central kernel repository anyone can freely download the Linux code. Some developers, however, dont download the source files from kernel.org, and instead use a tool called Git (Linux kernel code administration utility). There is also an Error Notifier System where these are classified according to importance and dependence. It is also possible to monitor and see whether theyve been solved or not. • **Quality Attaining Mechanism:** The evaluation of the quality of software contribution is determined by a number of factors related by the dynamics of free software development, such as the use of distributed collaborative tools, version control systems, mailing lists, among others, that enable participation and evaluation of contributions from the entire community. Another form of quality evaluation derives from code merging and depurating processes that require a detailed revision; and finally, existing documentation helps evaluate and improve the software that has been created. In the LKDC, quality is derived from voluntary collaboration from a large number of people that contribute in a parallel manner, reporting and solving code errors or providing new functionalities. The acceptance of contributions that solve errors or provide functionalities will be largely dependent on how well they function. The use of a reduced number of lines, thorough documentation and the acceptance of signoffs are aspects for qualifying the proposed code as acceptable. Lastly, the accepted code must be approved by the leader of the corresponding module and Linus Torvalds, so it can be taken in consideration for the next version release.

Table 9: MASOES Collective Processes in the LKCD (Continuation).

	• **Reward Mechanism:** Among the motivations to participate in the LKDC we find: learning and developing new skills, sharing of knowledge, gaining prestige through the start of a successful Project, carrying out tedious work that normally not many people are willing to do (such as writing up documentation or adding an innovative feature), among others. In this way, participants that make outstanding contributions in programming, documenting, translating or administrative tasks are recognized in order to promote their participation. The recognition is done through a series of metrics that measure the degree of each agent's participation in the Project such as: – -Developers with the largest number of modified files (it is possible to determine these figures through records known as change logs, in charge of saving the changes made to the software, dates, comments and authors of such changes). – -Developers with the largest number of lines changed. – -Developers that have deleted the largest number of code lines. – -Developers with the largest number of signoffs (signed patches for each developer). • Punishment Mechanism: The project's participant could be subject of public ridicule or be ignored and rejected by the other members of the community, with the intention of punishing disobedience to the established rules, in order to minimize the appearance of this bad behavior, sanctioned by the community.

Table 9: MASOES Collective Processes in the LKCD (Continuation).

KNWOLEGE MANAGEMENT PHASE	REPRESENTATION IN THE LKDC
Socialization	At this stage there is a process of conversion of implicit knowledge to explicit knowledge easily communicable deriving from the same basic interactions of the actors, such as reporting existing errors, or the explicit demand for new features, or new device drivers for available hardware. Also, the self-explanatory source code contributions are important (behavior encouraged by the importance of quality and elegance of design in Free Software), which becomes the representation of individual knowledge about a particular problem, that is, the proposed solution that a member of the community has, and that will be made available to the community in the versions control repository of the source code. Incorporation of such knowledge (solutions, errors and needs) through the tools available in the community setting, are the primary goals of socializing to stimulate the generation and sharing of knowledge.
Aggregation	Debugging explicit knowledge generated in the process of socialization comes from the deployment of generic solutions or design patterns promoted by existing ones. The good programming practices, the documentation, the optimization of the code, the creation of generic routines that facilitate code re-use by other members (components), consolidate a collective learning on how to attack certain specific recurring problems. Furthermore, these design patterns eventually become part of the rules and standards.
Appropriation	The appropriation process occurs thanks to the mechanism of individual learning. For example when users download and understand the project's source code, they are able to modify it according to their own criteria and style. In this way, they are incorporating collective knowledge generated by the community into their individual knowledge base.

Table 10: FSD through Knowledge Management Phases