User-Oriented Analysis of Interactions in Online Social Networks with Patterns

Rubén Fuentes-Fernández, Jorge J. Gómez-Sanz, Juan Pavón

Abstract—Online social networks are increasingly popular on the Internet. As people gain online expertise, their interactions become more complex, replicating behaviours from the real world and also creating new ones. Analysing and interpreting these interactions from a user perspective requires techniques that go beyond statistics and integrate people and systems. Our work focuses on these issues by borrowing from Social Sciences, specifically from Activity Theory, which identifies patterns of social structures, human behaviours and uses of artefacts in social environments. This work describes these patterns as social properties with a description for automated processing and a human meaning. The analysis looks for correspondences between these properties and design information and observations on online social networks. Where a correspondence is found, the meaning of the property is useful for interpreting observations in terms of human behaviour patterns. A case study on a recommendation service illustrates the approach.

Index Terms— D.2.0.b Software psychology, D.2.1.a Analysis, H.5.3.c Computer-supported cooperative work

I. INTRODUCTION

ONLINE Social Networks (OSNs) are among the most popular sites on the Internet [1]. They adapt social networks from the real world to the web [2]. Members of OSNs construct personal profiles with the information they want others to know, share interests through recommendations, links or documents, and build lists of people with whom they are connected. These mechanisms are not mere replicas of their real-life counterparts. The scale and ability to make information visible to all the members of the OSN allows drawing conclusions and making contacts that could not be otherwise achieved [2]. For instance, people find friends that they lost contact with a long time ago and discover new ones. Researchers observe trends in society and get immediate feedback from their communities; and software engineers gain access to large test populations of real users for their innovations.

The information available about OSNs is very heterogeneous. There is information related to Social Sciences such as motivation, organization, rules and economic aspects. Support systems have a design and generate observations on their use. These support systems include, for instance, the applications and web pages that members use to interact with the OSN, server systems and monitoring tools.

This information is studied for a large number of reasons such as discovering trends in communities, user preferences or conflicts, but also desired features or bugs in support systems.

The techniques applied in these studies can be grouped into two main approaches [3], [4]. On one side, there is a need to analyze large amounts of data such as link graphs, usage time or preferred features. These types of studies are for discovering general trends, connection structures or group profiles. The meaning of such data emerges from statistical distributions and data mining techniques are common in such cases. On the other side, there is a need for analyses focused on specific interactions. These types of studies are useful, for instance, for customization, ethnographic and cognitive studies, interface design and software maintenance. The meaning of these data comes from how individuals perform certain actions. Interpretation usually demands intellectual endeavour and involves multidisciplinary teams, requiring skills from diverse areas such as Software Engineering, Sociology, Psychology or Economy.

The dichotomy between these two lines of research has a negative impact on the study of OSNs [5]. It is difficult to discover new behaviours without a detailed and costly scrutiny. Designers begin studying sets of data. They interpret those data and set up hypotheses about what people are doing and why. Then, they validate their hypotheses gathering new information. This process can be performed for specific situations, but it is not cost effective for large numbers of interactions (see the “Analysis of online social networks and its limitations” sidebar). For instance, finding errors in the design of the user system and tuning it to the growth of the community requires software engineers to continuously collect logs and distil them. Designers then repeat the previous analyses and, if their hypotheses are correct, propose changes to the system.

Our work is intended to bridge this gap with a framework that formalizes some aspects of a user-oriented analysis of OSNs as patterns. These patterns, called social properties, represent knowledge grounded in Social Sciences about motivation, behaviour, organization, interaction and learning. The specification of a pattern consists of a design description
Analysis of online social networks and its limitations

The most difficult part of studying OSNs is discovering relevant groups of data and interpreting them in terms of people and systems features [1]. Trying to partially automate this analysis, some studies set up models that become their theoretical basis. These studies mainly focus on link graphs and message exchange. The analysis of link graphs considers that the topology of networks mirrors human concepts and relationships. These graphs are analyzed with metrics such as the indegree, centrality and closeness [2], which are applied to explicit and implicit relationships. Examples of explicit relationships are those between contents (i.e. hyperlinks in web pages), users (e.g. acquaintances, friends and groups), and between both (e.g. participation in events and recommendations). From these explicit relationships the analysis extracts implicit relationships such as interest groups, leaders of communities or relevant contents for topics [3].

Other studies try to get further insights on interactions through the exchanged messages [4]. Message contents describe OSN topics and their evolution, and can be considered a kind of implicit link. The difficulties of analyzing free text lead to focussing on messages between users and systems. For instance, instead of analyzing chat between two users, requests to the system interface, their types and the timestamp of the messages are analysed.

These studies show two main limitations regarding human elements. Firstly, they make assumptions that are poorly applicable when dealing with people in real situations. For instance, they assume perfect knowledge about interactions or unambiguous interpretations of messages. Secondly, they usually defer all data interpretation to human experts, as processing is independent of the actual semantics of the observations. Enabling automated reasoning on these facts requires formalizing knowledge from Social Sciences as well as the observations. Efforts in this direction, for instance with ontologies and logics, face difficulties in capturing the non formal and holistic perspectives of social studies.

The exploitation of contents is also limited, as most studies rely basically on structured information. For instance, [2] and [3] use tags of posts, links, emails or paper co-authorship. The analysis of information such as textual references in blogs or papers presents challenges to natural language processing [5]. Moreover, such research also needs help to interpret its results from a user perspective.

References:

Activity Theory

Activity Theory (AT) [1] is a paradigm for the analysis of human groups with a focus on acts. It assumes that people’s behaviour depends on the physical and socio-cultural context in which they are embedded [2]. At the same time, people interact with the environment, changing it. These interactive acts are called activities and their contexts activity systems.

The activity is a transformation process driven by people’s needs. This process transforms objects into the outcomes satisfying those needs. The active component carrying out the activity is the subject. Any other element used in the transformation is a tool. Tools mediate the interaction of the subject with the environment. The social dimension of activities is organized around communities. These are groups of subjects sharing social meanings and artefacts. Two social artefacts mediate the relationships within the community: rules with the subject, and the division of labour with the object. Rules include laws, social conventions and norms external to the activity but affecting it. In the transformation process, the division of labour establishes the role of the actors in the community, the power that they hold and the tasks they are responsible for. All the elements of these types related to an activity constitute the activity system.

From the AT point of view, objects, tools and outcomes can be both mental and physical elements. So, the paradigm

Contradictions [3] are conflicts between elements, both within and between activity systems. Subjects try to remove contradictions by changing the activity systems but these changes carry the seed of new conflicts. Thus, contradictions are the driver for the evolution of social systems.

AT has been applied to analyze complex human settings in software development [4]. Most of these studies retain the original AT practices, which are hard to capture in formal languages and difficult to automate.

The authors of this paper have contributed to improving this situation by defining a UML profile called UML-AT [5], which captures the AT conceptual framework for software development. Studies on the integration of UML-AT with the agent paradigm support the application of AT to software systems analysis [4]. These applications include the definition of social properties to elicit and interpret knowledge about a target software system. Some of these properties [5] can be reused in the context of OSNs, e.g. those related to user motivation, external social structures or knowledge sharing.

References:
Given these elements, AT describes social systems in terms of networks of activity systems interconnected by shared elements. Subjects simultaneously execute activities in these networks following their own rationality. This approach needs a suitable theoretical framework that provides the concepts for the description of the properties and also potential patterns. For this purpose, our work adopts Activity Theory (AT) (see the “Activity Theory” sidebar). AT is a social research paradigm with a holistic approach to the study of societies. It includes social movements to individual cognition in the frame of historical development. AT constitutes our primary source of knowledge and metaphors that can be adapted for the study of OSNs.

AT has been already applied in software design (see the “Activity Theory” sidebar). Our work makes use of three main results from previous research. First, it uses the modelling language UML-AT, which formalizes the conceptual framework of AT. This language is the basis for the description of social properties as models. Second, it adapts some of the AT patterns identified in other software development fields for the analysis of OSNs. Third, it applies the results of studies on the integration between AT and the agent software paradigm [6] to provide a description of the systems supporting OSNs in terms of AT concepts. This description facilitates the integrated analysis of people and their software.

II. SOCIAL PROPERTIES

A social property can be modelled as a network of AT concepts with a given intentional and organizational meaning. These properties describe different kinds of aspects of OSNs and their environments, such as the motivation driving users, the kind of social structure a group of connected users has, or their patterns of interaction.

The definition of these social properties is the result of a collaborative effort: AT practitioners determine the relevant social features and help in their interpretation; domain experts determine suitable formulation of properties in their areas; software engineers decide what data from the support systems can provide evidence of the kind of information required. To facilitate the work of these experts and the application of properties in analysis, they are represented with the structure exemplified in Fig. 1.

Each social property has a unique identifier and a generic description. This description contains a general discussion on aspects such as goals, applicability or benefits. Besides these common and general aspects, a property is considered in particular contexts with specific interpretations. For instance, a property can be related to the identification of selfish users in a community. Such users provide contents of low quality or devote little time to their editing; in both cases, the sequences of actions are very similar, but there are differences in the aspects to measure that identify different settings. The description of a setting comprises UML-AT diagrams, a textual explanation, related social properties, examples of its use, and bindings between variables to indicate what information is related between the different diagrams.

UML-AT diagrams model the situation represented by the setting. They have slots for the name, type and value of their entities, relationships and properties. These slots can contain fixed values or variables. A setting appears in some OSN if there is a correspondence between its UML-AT representation and the OSN information. This is essentially a pattern matching process: both groups of information have the same fixed values and the setting variables can be instantiated with information from the OSN. When this happens, the descriptions of the property and the setting help to build a social interpretation of that OSN information.

The examples of a setting show its application in previous projects. The related properties are references to settings of other properties that can complement or be a consequence of the setting under consideration. For instance, given a property describing a type of group organization, related properties can show typical workflows in it; given a conflict, a related property can point out a possible solution, such as in Fig. 1.

Although settings allow describing detailed contexts of application of properties, they are not enough to cover the different perspectives that experts have on properties. In order to enable active involvement of people with different backgrounds, setting descriptions have several perspectives aimed at different audiences and purposes: the AT perspective is intended for social practitioners and describes the intentional and social interpretation of the setting; the domain perspective is for specific domains, translating abstract AT concepts to the organizations, laws, workflows or artefacts relevant in the domain; the system perspective helps to map the previous elements to data and functionality present in the support systems. For instance, the AT practitioner talks about the activity that satisfies user needs, but for the economist the user is making a transaction in order to make a profit, while the software engineer sees actions that end with the validation of a payment in the system. There can be an instance of the structure in Fig. 1 for each perspective of a setting.

---


Identifier: Exchange Value

Perspective: ☑ AT ☐ Domain ☐ System

Generic description
This contradiction emerges when a subject has to generate an outcome to be consumed by other members of the community. However, none of the generator subject’s relevant objectives are satisfied by that outcome. Consequently, the subject who is able to create the outcome and give it to the community, may not have sufficient motivation to do it.

Setting 1
- Specification

The Generator Subject is able to produce a Generator Product with the Generator Activity. Doing that, it satisfies a Generator Objective. However, this is not its main objective, which is Individual Objective. Hence, depending on the circumstances, the subject may not be interested in generating the product. The Subject in the Community needs this product to carry out its Activity in the Community.

- Related properties: 1. Exchange Value solution

The Subject in the Community can overcome the Exchange Value contradiction by encouraging the Generator Subject to execute the task with a reward or making the benefits of its execution explicit. In the first alternative, they should provide the generator with products that satisfy one of his/her relevant needs. The diagram represents this solution with a new Reward Activity for the Subject in the Community. It generates an outcome Reward that directly contributes positively to the main objective of the Generator Subject.

Note: Dashed lines enclose the elements to add in the solution.

- Slot bindings. For perspectives

<table>
<thead>
<tr>
<th>AT</th>
<th>Domain</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Individual objective</td>
<td>Person goal</td>
<td>Person/agent goal</td>
</tr>
</tbody>
</table>

- Slot bindings. For related properties

<table>
<thead>
<tr>
<th>Exchange Value</th>
<th>Exchange Value Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any variable</td>
<td>Same value that in the original contradiction</td>
</tr>
</tbody>
</table>

Fig. 1. Example of the use of the structure for social properties.

The glue among the previous elements is slot bindings. These bindings are pairs of names of elements (i.e. entities, relationships and properties) in different diagrams. They specify the correspondences between the slots in different perspectives of a setting, or in a setting and its examples and related properties. The first are necessary to grasp the relations between the different perspectives, facilitating their interpretation and discussion by the different specialist groups involved in their use. The second help to understand the potential applications of related properties in the contexts where the current setting is applied. For instance, considering the AT perspective, they can indicate that the outcome of an activity system in a setting can become the tool or subject of another activity system in a setting of a related property. The third group shows the instantiation of the setting in the particular context that an example describes.

The Exchange Value contradiction is an example of a social property. Fig. 1 shows a partial description of a perspective in one of its settings. As this property corresponds to a conflict, AT studies have described potential solutions for it. Fig. 1 models one of them as a related social property. This kind of related property is optional and not supposed to appear in every setting. Note that the description of the related property should appear in its own structure, though Fig. 1 integrates its UML-AT representation to facilitate the presentation.

The use of these properties is partly supported by functionality developed for the Activity Theory Assistant (ATA) [7]. ATA is a software tool able to check the occurrence of social properties in UML-AT specifications.
III. Recommendation service

To illustrate the use of social properties, consider the case of a collaborative recommendation service. In this example, the OSN is intended to build up a virtual community where users can share their leisure interests. Users create a profile along with their account in the community. This profile allows the system to contact people with similar interests. To make the community attractive, it is expected that users actively participate in it by keeping their profiles updated, setting up and taking part in activities with other members, and reporting potential misuses. Users who adopt a passive role impoverish the life of the community.

In order to perform and analyze these activities, the OSN has several support systems: user applications, monitoring tools and systems on the servers. Users interact with the OSN using desktop software and web pages. In addition, the service monitors user behaviour creating logs of requests and responses to the central services. The overall goal of these activities is to be able to advise users on how to improve their interactions with the systems and the community. Practitioners and software engineers use social properties and ATA for this analysis.

The first step is initializing the repository with the laws and initial elements of the OSN. In this case, the laws include: the types of profiles and expected motivations of users; several interactions in which users can engage in the real world with influence in the OSN; the design of support systems, as they determine the user-system interactions, and how the systems should behave. The initial facts include the existence of a default administrator.

The workflows include two types of users: the member provides contents and mainly interacts with peers, the administrator manages the members and contents of the community. Since both types of users are active elements in the OSN, they are subjects in UML-AT. A member can perform the activities: consult information, update information and notify misuse. The last two activities improve the contents of the service by generating modified contents as outcome. An active member should regularly perform these activities in order to contribute to the OSN. The administrator performs the activity support members, which supervises contents and notifications for the community. This user also performs offline management of the contents in the activity support community. For instance, it moves contents to historical data or performs data mining to discover trends.

After monitoring the system for some time, practitioners want to check the behaviour of the OSN. The first step is to update the repository according to observations. In this case, all the facts are initial laws, planned facts, certain observations, or false and non-valid information. Foreseen facts may appear in the future when the observation of the external activities of users reveals new interaction patterns. Other sources of foreseen facts are not applicable here: the support systems do not allow new forms of interaction, unless software engineers change their design, and no analysis of contents is performed. So, the update activity generates the planned information only from the laws of the system. These planned facts along with laws indicate the expected behaviour of the OSN. Then, the activity changes the state of those planned facts already observed to certain. For instance, members have performed the activity consult information to check the suggestions.

Then, engineers and practitioner try to detect social properties in the repository using the ATA. A correspondence with the setting of the Exchange Value contradiction (see Fig. 1) appears in the OSN as instantiated on the left of Fig. 2. The original explanation of the setting interprets it. The member should carry out the activity notify misuse to provide the outcome modified contents to the administrator. The administrator, through the activity support members, processes this information to improve the contents of the service. The contradiction occurs because the member is not performing the activity despite all its prerequisites (subjects and objectives) appearing in the repository as certain information. After a given time, no exchange of messages in the support system indicates the execution of the activity notify misuse and the subsequent generation of the outcome modified contents, so this planned information becomes non-valid. Thus, the activity detects the contradiction whose interpretation is that the member (i.e. the generator subject) is not sufficiently motivated by the community to carry out the action. Consequently, the administrator (i.e. the subject in the community) cannot perform the related activity.

The diagram on the right of Fig. 2 shows a possible solution to the Exchange Value contradiction according to AT (see the related property in Fig. 1). The subjects in the community who benefit from the product of the generator subject should motivate him/her. They can provide products that satisfy relevant needs of that member or make explicit the contribution of the generator’s activity to those needs. For this case, practitioners adopt the second alternative: when the member consults the OSN, the system will show information about how his/her activities contribute to information in the community. The execution of the activity support members can trigger a new activity generate statistics, which gathers information for reports about members’ activities. When the member demands information through the activity consult information, the system uses the reports to display relevant statistics about the member’s contributions. In this way, it is expected that members will get a clear perception of the influence of their collaboration in keeping the community alive. The diagram shows this modification as a contribute positively relation from the outcome report to the main objective of the member. Alternatively, practitioners and software engineers can use the reports to set up a mechanism of rewards to members according to their activities, so their involvement no longer depends on subjective perceptions.
Fig. 2. Exchange Value contradiction with users: on the left the match of the contradiction and on the right a possible solution. The patterns include planned facts (within dashed lines), the added elements for the solution (within dotted lines), and certain facts.

Note that the AT catalogue of properties and solutions for human conflicts has been a useful guide for building the OSN: experts get advice on how to elicit information and deal with people behaviour regarding their objectives, and engineers on how to develop sound and robust support systems.

IV. CONCLUSIONS AND FUTURE WORK

This paper introduces a framework for the runtime analysis of OSNs. It shows the feasibility of using highly abstract concepts to describe and understand OSNs without getting stuck in low-level details. The use of these types of concepts combined with analysis using the agent paradigm for software systems also has the advantage of allowing an integrated analysis of the support system and its human environment. This integration is a prerequisite to studying certain failures of support systems to meet expectations of practitioners and users that can be better diagnosed during runtime.

From the point of view of its application, the proposed framework has several important features. Firstly, it offers a simple and semi-automated analysis method for OSNs, as it is simplified to pattern matching of social properties with information about OSN. This approach has the advantage of providing a preliminary interpretation of the data, though practitioners still need to analyze the results carefully. Other semi-automated methods offer less digested results, as data interpretation is an exclusive responsibility of practitioners. Secondly, it includes a catalogue of ready to use social properties based on AT to understand OSNs with multiple interacting elements. This catalogue relieves the practitioners' workload, as they do not need to define all the properties of interest and have additional information to guide the analysis. Thirdly, it serves in understanding what happens inside a support system that generates long information logs during its execution. By using social properties and agent concepts, the framework can summarise complete logs to a list of understandable interpretations. Fourthly, it can deal with incomplete models of OSN. This aspect is essential when dealing with people, as there is no way to get exact models of their behaviour. The qualification of the information with certainty degrees allows reasoning with uncertainty and presenting to practitioners and engineers those interpretations that are more likely to be valid. Finally, the use of non-valid facts also takes into account the evolution of the OSN in case of, for instance, adaptation to users, change in the focus of the network, addition of new features to the support system or automated learning. Moreover, it provides a means to discard obsolete information. Nevertheless, this kind of information obsolescence is already considered in other studies.

The experimentation carried out so far, of which the case study provides an example, shows that the process currently faces two main limitations. Firstly, UML-AT needs extensions to include concepts of AT that are currently not being addressed and support expressing semantic rules about facts such as possible transitions in the information states, timeouts or triggering conditions of activities. Secondly, the framework needs the information about the OSN to be described with UML-AT. This is feasible for the design of the system, as ATA can perform a semi-automated translation of specifications to UML-AT [7], but is a far more difficult task when it involves text analysis. This latter case requires natural language processing and practitioners supporting the modelling process. This is not only an issue for this work, but also still an open problem for all the research on OSNs. Our ongoing work will partially address the extraction of information from microblogging sites using templates. A final line of future work is the application of AT contradictions and their solutions not just for the detection of conflicts, but also as laws governing OSN behaviour.

REFERENCES


**BIOGRAPHIES**

**Rubén Fuentes-Fernández** holds a Ph.D. degree in computer science from the Universidad Complutense de Madrid, Madrid, Spain. He is currently an Associate Professor in that university and member of the GRASIA research group. Previously, he was a Consultant in database systems for four years. His main research interests are concerned with the application of social sciences to the development of software systems regarding group activities.

**Jorge J. Gómez-Sanz** holds a Ph.D. degree in computer science from the Universidad Complutense of Madrid. He is currently an Associate Professor in that university. His research is centred on agent-oriented software engineering and its application. A main result of this research is the INGENIAS methodology and its support tools.

**Juan Pavón** holds a Ph.D. degree in computer science from the Universidad Politécnica de Madrid, Madrid, Spain. He worked for ten years in research centres of Alcatel in Spain, France, and Belgium and in Bellcore (USA) in the development of component-based architectures for distributed systems. In 1997, he moved to the Universidad Complutense de Madrid where he is currently a Full Professor and leads the Agent Research Group (GRASIA), with participation in projects of model-driven engineering and multiagent systems.