HAL
open science

Demo Spontaneous and Ephemeral Social Networks: an Event-based Framework

Stéphane Frénot, Amine Ghorbali, Frédérique Laforest, Pascale Launay, Nicolas Le Sommer, Damien Reimert

To cite this version:

Stéphane Frénot, Amine Ghorbali, Frédérique Laforest, Pascale Launay, Nicolas Le Sommer, et al.. Demo Spontaneous and Ephemeral Social Networks: an Event-based Framework. debs 2015, Jun 2015, Oslo, Norway. 10.1145/2675743.2776772 . hal-01154460

HAL Id: hal-01154460

https://hal.science/hal-01154460v1

Submitted on 22 May 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

Demo
Spontaneous and Ephemeral Social Networks: an Event-based Framework

Stéphane Frénot
Université de Lyon - INRIA
stephane.frenot@insa-lyon.fr

Amine Ghorbali
IRISA, Université de Bretagne Sud
mohamed-amine.ghorbali@univ-ubs.fr

Frédérique Laforest
Hubert Curien Lab., Université de Saint Etienne
frederique.laforest@univ-st-etienne.fr

Pascale Launay
IRISA, Université de Bretagne Sud
pascale.launay@univ-ubs.fr

Nicolas Le Sommer
IRISA, Université de Bretagne Sud
nicolas.le-sommer@univ-ubs.fr

Damien Reimert
INRIA
damien.reimert@inria.fr

ABSTRACT

The C3PO project promotes the development of a new kind of spontaneous and ephemeral social networks dedicated to happenings. They rely both on opportunistic communication networks formed dynamically by mobile devices, and on an event-based communication model. In this paper, we present the C3PO framework. It is based on two parts plugged on an event flow. In our demonstration, we will invite conference attendees to use the Android mobile application we have developped using the C3PO framework to cover DEBS'15 by exchanging digital contents during the conference.

Keywords

Social Networks, Opportunistic Computing, Event-based data flow, in-browser architecture.

1. INTRODUCTION

Online social networks have been adopted by a large part of the population, and have become in few years essential communication means and source of information. Nevertheless, as they rely on a centralized and proprietary architecture, these social networks present some drawbacks. Among these, users rely on a remote authority for information management, filtering and dissemination. All data management algorithms are centralized and determined by the authority; this authority defines solely its terms of service. Secondly, an Internet connection is mandatory to access them. In circumstances such as during peak times in crowded environments, people may experience long latencies and even network outages. Finally, they are not specifically designed and are not well suited for allowing geographically close users to exchange contents in a collaborative way.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
DEBS '15 June 29 - July 03, 2015, Oslo, Norway
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-3286-6/15/06 ...$15.00.
http://dx.doi.org/10.1145/2675743.2776772.

The C3PO project[1] proposes to investigate multimedia content exchange in a new type of social networks, so called Spontaneous and Ephemeral Social Networks (SESN). At the network level, SESNs rely on a peer-to-peer distributed architecture spontaneously formed by mobile devices. Opportunistic communication techniques are used in order to support the connectivity disruptions resulting from the mobility of devices and the short communication range of wireless interfaces. A SESN is created by any user and advertised to other users in the vicinity. As a SESN is geographically constrained by the happening location, the set of users collaborating in the SESN is limited. When the happening stops, peers forming the SESN separate, devices are disconnected and the SESN vanishes. Multimedia contents are stored locally on each device, so users still have access to these contents after the end of the happening. These contents are massively exchanged within the SESN: all peers receive all contents. At the application level, each peer filters contents according to a selection of plugins. Each plugin processes specific events, produces some results and handles the user interactions on the results. Some plugins may create new events thrown in the flow.

Due to their spontaneous and ephemeral nature, SESNs are suited for happenings, such as conferences, cultural or sports events. For instance, the spectators of a marathon can take photos, write comments about the marathon and publish them in the dedicated SESN. The C3PO project objectives are detailed in [2].

Our proposal claims for the following differentiating characteristics:

1. a location-constrained social network, where users only may interact with others using short range communications.

2. a short-lived pseudo, where each user is identified only for the happening lifetime.

[1] http://www.c3po-anr.fr/

3. a highly configurable application based on a set of plugins that can select the events they must process. Plugins may be added or removed on the fly.

4. an infrastructure-less architecture that does not depend on a central authority, each device managing his own view of the SESN.

This paper is organized as follows. Section 2 provides an overview of the C3PO framework. Sections 3 and 4 present respectively the inter-devices communication and the intra-device communication parts of the C3PO framework. In section 5, we propose a demonstration for DEBS'15 of an Android application based on the C3PO framework, and that supports the above-described SESN architecture.

2. GENERAL ARCHITECTURE OF THE C3PO FRAMEWORK

C3PO targets the mobile user, where the panel of devices is large and heterogeneous: genericity of the C3PO application is mandatory. C3PO aims at being independent from the operating system as much as possible, and therefore a large part of the C3PO framework relies on the today's in-browser technologies.

Using an event-based paradigm, the C3PO framework general architecture can be represented as on figure 1. We structured this framework in two parts: the inter-devices communication part, that manages the data exchange between devices, and the intra-device communication part enabling the propagation of events within the local device.

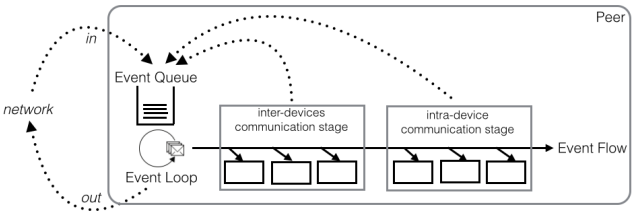

Figure 1: C3PO general event-based architecture

To implement the event-based architecture, we have used a turn-based processing approach [2], that consists in processing an event through all the parts of the architecture before processing the next one. This approach follows three principles: no blocking, no wait, finish fast. Moreover, it allows to manage only one queue of events that corresponds to a single network interface connection, as most mobile devices have.

The architecture uses a set of components plugged on the event flow. Components use a specified API to be notified of events. Current components are developed in Java or JavaScript, to cope with low level hardware APIs and high level HTML5 in-browser development.

Sections 3 and 4 explain the inter-devices and intra-device communication parts.

3. INTER-DEVICES COMMUNICATION

The C3PO inter-devices communication part is built upon four blocks of functionalities and an event registry (Figure 2). The Connectivity and Network Management block makes the device discovery and the connection with neighboring devices. It supports different connectivity types, i.e. Bluetooth, Wi-Fi Legacy, Wi-Fi Direct and 3G/4G.

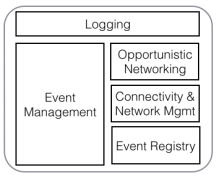

Figure 2: C3PO Inter-devices communication

The Opportunistic Networking block provides two communication paradigms, namely a pub/sub topic-based, and a send/rec channel-based. It defines a generic mechanism to forward messages in mobile ad hoc networks, and allows to devise different forwarding strategies. Finally, the inter-devices communication part uses two other blocks of functionalities: Events Management and Logging.

In the C3PO framework, this inter-devices communication part is developed in Java, so as to be implemented on Android mobile devices and Wi-Fi access points based on Raspberry-Pi devices. The latter can be deployed during a happening in order to help form the network.

3.1 Opportunistic Networking

The short communication range of Bluetooth and Wi-Fi interfaces, the mobility of people and the radio interferences can yield unpredictable disruptions in communication links. Maintaining end-to-end connectivity between all devices then becomes quite a challenge, as solutions relying on dynamic routing protocols designed for MANETs prove unable to ensure message forwarding in such conditions. They assume that the network is dense enough, so it can be viewed as a fully connected graph. It proves unrealistic in many real situations. Opportunistic communications rely on the "store, carry and forward" principle. The basic idea is to take advantage of radio contacts between devices to exchange messages, while exploiting the mobility of these devices to carry messages between different parts of the network. In opportunistic networks, different message forwarding strategies can be devised, like epidemic and network flooding techniques, or more sophisticated techniques [3]. In the C3PO framework, message forwarding strategies use reactive and event strategies. Four main types of events are currently considered: (1) the emission of a message by a local component, (2) the reception of a message from the network, (3) the discovery and connection of a new neighbor and (4) the disappearance and disconnection of a neighbor.

Simple flooding message forwarding

The flooding strategy is the most elementary one. When a message is received from a local application, it is stored in the local cache and forwarded to the connected neighbors. When a message is received from the network, it is stored in the local cache and transmitted to the intra-device communication part. When a new neighbor connects, all the messages in local cache are forwarded.

Gossiping-based message forwarding

The gossiping strategy relies on a prior exchange of catalogues of messages IDs. Each device maintains an up-to-

[2] `http://www.youtube.com/watch?v=dkZFtimgAcM`

date vision of the cache of its neighbors. Based on these catalogues, each device asks the new neighbor to forward the unknown messages (see Figure 3) and broadcasts its new catalogue to other neighbors.

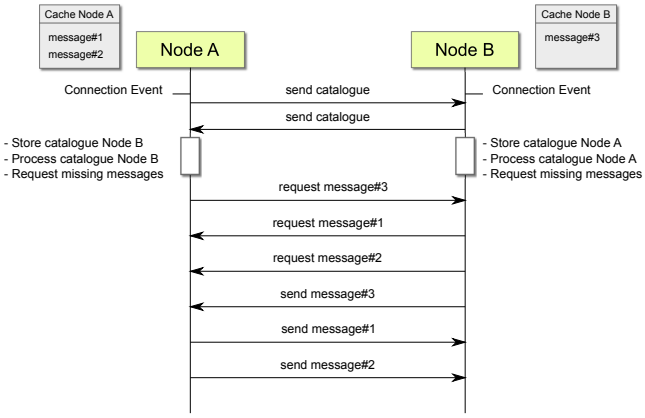

Figure 3: Exchange of messages after a connection event using gossiping-based message forwarding

The simple flooding strategy is dedicated to SESNs with few neighbours. The gossiping-based message forwarding strategy should be used when the number of neighbors increases: catalogue exchange permits to effectively exchange only necessary contents.

3.2 Communication Paradigms

The C3PO framework provides two distinct application-level communication paradigm.

The pub/sub topic-based communication model makes it possible to develop applications that can publish multimedia contents on specific topics, and subscribe in order to receive the contents related to given topics. This model relies on a purely peer-to-peer decentralized approach. Subscription and publication are local to each device. Due to the spontaneous nature of the opportunistic network, it is not possible nor suitable to rely on a centralized registry. Thanks to the store, carry and forward principle, contents published on a topic by publishers (P in Figure 4) are disseminated opportunistically in the communication network by mobile devices, being either devices hosting subscribers for this topic or any intermediate device (resp. S and I in Figure 4).

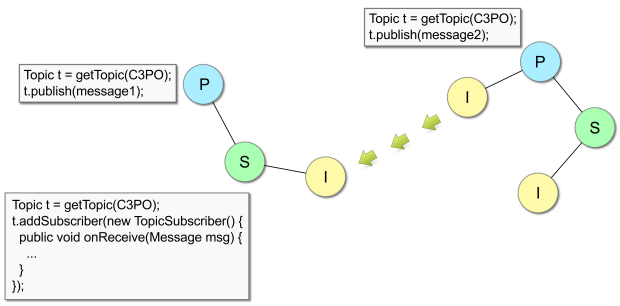

Figure 4: Communication between mobile devices using the topic-based publish/subscribe paradigm

The point-to-point communication paradigm based on the concept of channel is intended for applications that allow users to communicate with each other by sending messages addressed to specific recipients. In the framework, a channel between two devices is identified by the addresses of the devices and a port number. Messages sent through a channel are opportunistically forwarded by intermediate devices towards their destination according to one of the message forwarding strategies implemented in the framework. Thus, two devices can exchange data even if they are not within mutual radio range.

4. INTRA-DEVICE COMMUNICATION

The intra-device communication part manages the events for the end-user. It gets events from the flow of events, processes them and presents the results to the end-user. This intra-device communication part is also responsible for the creation of new events reflecting some end-user actions. For example, the end-user can create new contents to be exchanged with other end-users; he can also promote a received content to stress his special interest.

Depending on the SESN happening type, the events may carry very different contents and thus event processing requires extensibility. Devices in the same SESN have a set of identical processing units, but some more powerful devices may also have additional specific ones.

The global architecture of the Intra-device communication part is illustrated on figure 5. Plugins are components responsible for the processing of incoming events and for the interaction with the end-user [1]. Each plugin has a specific event processing unit, dedicated to a kind of event. For each incoming event, the event dispatcher selects in a registry of plugins the corresponding plugins and calls their processing unit with a reference to the event. Plugins can store their processing results in their own context.

When a device connects to a SESN, the SESN Manager registers the corresponding plugins in the registry. A SESN Canvas organizes plugins in the graphical user interface.

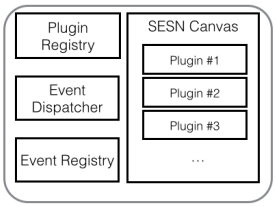

Figure 5: C3PO Intra-Device Communication

4.1 Plugins

A plugin is defined with three elements: (1) the grammar of the events contents it is interested in, (2) the processing modalities of the corresponding events, (3) the management of end-user interactions with the results of its process. For example, an image plugin can declare in its grammar that it requires events containing references to images, process the events by uploading the images, interacting with the end-user by showing the list of images and proposing a "promote" button for each image.

Plugins must conform to the C3PO plugins API, that contains one standardized function invoked by the event dis-

patcher. This method has one parameter: a reference to an event in the event registry. Plugins are developed using the AngularJS framework. User interactions in a plugin are developed as an AngularJS directive. This Directive may also include an event creation. This new event is thrown into the global events queue, as shown in the general architecture on figure 1.

An event can be processed by many plugins. Each plugin declares regular expressions to be matched on the event content in order to be notified. The plugin registry gathers all regular expressions in a way that optimizes plugins selection. The event dispatcher uses this registry to trigger processing units in plugins. Regular expressions are flexible enough to accept any kind of identifier within events. In the current implementation, regular expressions provide a list of hashtags that must be present or absent.

4.2 Plugins examples

In this section, we describe a subset of the plugins that have already been developed in the C3PO project.

- The "Flow" plugin takes all the received and emitted events and displays them as a list ordered with the last received first. Its regular expression is empty. The user can browse the list, and can click a "promote" icon that indicates that he believes this event is important. Each promote done by the end-user generates and throws a new Event including a #promote hashtag and the source event id. A screenshot is given on figure 6.

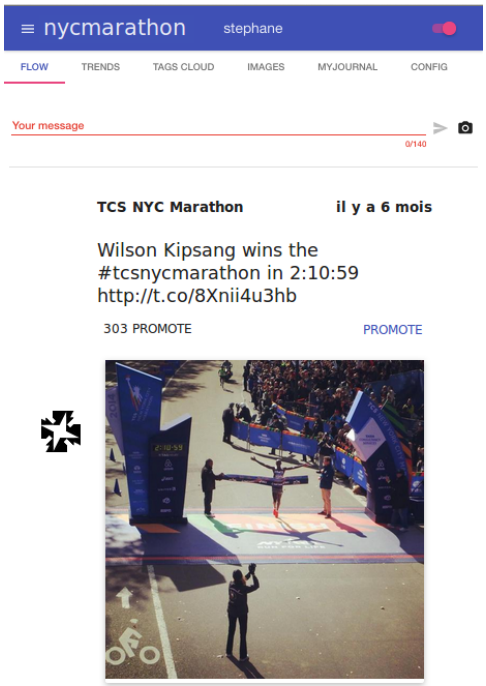

Figure 6: Event flow display

- The "Trends" plugin selects events containing a #promote hashtag. The promote event can come from any node of the SESN. Its regular expression is `#promote`. It consolidates in its context a promotion value associated with the event. When the promotion value reaches a given threshold, it displays it, creates and throws a new Event including a #trend hashtag with the consolidated promotion value and the source event id.
- The "Tags Cloud" plugin takes all the incoming events and builds a tags cloud that reflects the frequency of each hashtag in the events set. Its regular expression is empty. Its user interface displays the tags cloud.
- The "Image" plugin selects events containing a reference to an image (url) and displays them as a list ordered with the last received first. Its regular expression and its user interface are similar to the "Flow" plugin.
- The "My Journal" plugin displays all trending events interleaved with all events manipulated by the user. The user can manipulate the displayed list to raise, diminish or remove events. At the SESN termination it provides a summary of the happening.
- The "Config" plugin helps user to activate or to deactivate each other plugin. When deactivated, the plugin is removed from the user interface but keeps its local context. If the user reactivates a plugin, it starts processing events with the previous saved context.

5. DEMO PROPOSAL

The demonstration will be used by the conference attendees. They will enter the DEBS'15 SESN instance using the C3PO application with their own Android device. They will publish messages and photos, promote contents and get trending events during the conference. A voting plugin will be provided to elect the best paper from the SESN community.

Our C3PO application can be installed on Android smartphones or tablets, provided these devices are equipped at least with Android version 4.4. This application could be installed thanks to a flash code or a URL indicated on a poster. Even if our application has a friendly and simple to use interface, a short introduction to the possibilities offered by the application will be provided online and on site.

6. ACKNOWLEDGMENTS

This work is done in the C3PO project, supported by the French ANR (Agence Nationale de la Recherche) under contract ANR-13-CORD-0005. `http://www.c3po-anr.fr/`

7. REFERENCES

[1] S. Frénot and S. Grumbach. An in-browser microblog ranking engine. In *Advances in Conceptual Modeling - ER ECDM-NoCoDA workshop, Florence, Italy, October 2012*, volume 7518 of *LNCS*, pages 78–88.

[2] F. Laforest, N. Le Sommer, S. Frénot, F. De Corbière, and Y. Mahéo et al. C3PO: a Spontaneous and Ephemeral Social Networking Framework for a collaborative Creation and Publishing of Multimedia Contents. In *MoWNet 2014*, pages 1–6, Rome, Italy, 2014. Elsevier.

[3] V. F. S. Mota, F. D. Cunha, D. F. Macedo, J. M. S. Nogueira, and A. A. F. Loureiro. Protocols, Mobility Models and Tools in Opportunistic Networks: A Survey. *Computer Communications*, March 2014.