An Engine for Local Communities Sharing Services, Content and Experiences

Pierfrancesco Bellini, Daniele Cenni, Vladimiro Corsi, Paolo Nesi
DISIT-DSI, Distributed Systems and Internet Technology Lab
Dipartimento di Sistemi e Informatica, Università degli Studi di Firenze, Firenze, Italy
http://www.disit.dsi.unifi.it/, nesi@dsi.unifi.it, paolo.nesi@unifi.it

Abstract

The trend of social networking has put in evidence the needs of user in collaborating and sharing information. Most of the business models for those SN are very challenging and may be hardly scalable. Moreover, users prefer to keep under control their content and to have at home personalized views of their preferred services. Thus local community services and content management systems may be of interest to be developed in contrast to service provider that try to monopolize all services for a given user, similarly to the TV. In this paper, we present LOCUSGENS architecture and solution and in particular the features and the capabilities of the intelligent local Autonomous Community Manager which includes a rule based systems to cope with the harmonization and concertation of a number of third party services.

1. Introduction

The state of the art of Social Networks, SN, is grounded on centralized solutions for collection and sharing of user generated content and information. Many different business models have been exploited in social networks, including: subscription, advertising, communities sharing similar interests, access to premium content or services in Pay Per Play/View. The effectiveness of the business models are strictly related to the degree of participation of the members to the community activities. Communities whose members' involvement is characterized by occasional attendance, and short passive participation are generally suitable for business models based on advertising [1]. This model is typically viable in the presence of huge number of users, while the premium VOD (video on demand)/PPV model is mainly associated with solutions for protecting content (e.g., CAS, DRM).

The main problem of SN is the centralized approach of the SN. According to W3C, the future SNs have to be more distributed to share in the network and with the users the workload and the content repositories. Distributed solutions can be viable only if the general ecosystem may guarantee streams of revenues to the actors who are investing money to create the infrastructure and support costs. In distributed solutions costs may be shared among middle level servers and/or with final users' devices (such as decoders, computers, mobiles, game stations, PVR, HD recorders, etc.) or servers: such as media centers, routers, etc. The concept of media centre is less widespread as a commercial device, while it is more common in the open source community where a set of Media Centre software solutions are available. They have been mainly derived from public open source media centers such as XBMC, LinuxMCE, Boxee, MythTV, etc. They mainly integrate capabilities of playing video streams coming from DVB boards and from IPTV streams, receiving EPG, program recording, connecting to some social networks, etc. Some of them may be installed on cheap hardware computer/devices. One example about the standardization of media centre operating system and software model can be the MPEG M3W (supported by PHILIPS and DSI paper authors in the past [2]), which also integrates the MPEG-21 DRM support and concepts [3], [8].

In most cases, the media services are statically connected on receiving devices (e.g., STB) and offered to users. These solutions integrate poor features of content management, advertising management, gateway to manage the owned content in other social networks, automated content publication, and never rules for automating those activities. Furthermore, many service providers are interested in providing social features to the community that visit their websites, allowing them to have a full control of their social information, activities, and relationships throughout the web [4]. Google has created the Google Friend Connect JavaScript APIs which are standards that specifically deal with identity (OpenID), [5], [6] data access rights (OAuth) [7], and social applications (OpenSocial of Open Google). Other technologies needed to create decentralized social networks already exist: FOAF, RDFa, XFN and other microformats.

On the other hand, content services for the community are today divided into two large groups of tools and services: (i) data provider, such as Web Services, RSS and ATOM; (ii) data Aggregator, such as Widgets and Mashups. *Widgets* are interactive virtual tools providing services to the user by gathering data for data sources usually located on the web. A widget provides usually a single service or operation, such as showing the user the latest news, the current weather, the time, a calendar, a dictionary, a map program, a calculator, desktop notes, photo viewers, or even a language translator. Widget usually requires a widget engine that is the core responsible for screen rendering and interacting with the user for all the widgets running on the desktop. Example of widget engines are Dashboard from Apple, Microsoft gadgets in Windows Vista, Yahoo! Widgets, portlets in Google Desktop, etc. A *Mashup* is a web application combining data from different sources into a single integrated tool in an easy way by a third party source such as: API (Web service in general), Web feeds (such as RSS and Atom), Screen scraping (where info is grabbed from the web site directly) [11]; the aim is to produce results that were not planned for single data sources. An example of a mashup is the use of cartographic data from Google Maps to add location information to real estate data, thereby creating a new and distinct Web service that was not originally provided by either source. Many people develop new web application by “mashupping” existing social network or social oriented web portal such as Amazon, eBay, Flickr, Google, Microsoft, Pictometry, Yahoo, and YouTube. Several web oriented companies are creating tools for creating mashups by real mashups editor, among these companies and editors we can cite Yahoo pipes, Google mashup editor, Microsoft popfly, but several others are emerging day by day.

The aim of this paper is to present the main engine for setting up scalable local services for community enabling local management of complex services for user and content. The presented solution is part of a larger project called LOCUSGENS: Local Communities Sharing Generated Content, Services and experiences. The components presented in details in this paper refer to the engine that one could host on a media center located into the community premises to provide services to the local community: (1) remapping and reshaping a set of external services and information, (2) providing a set of additional services for integration of the former services and for the fact that the local information and events are locally exploited and managed. The added value services are performed by means of a rule based engine to automatically perform a set of activities for the community such as: download, local recommendation based advertising, privacy control, automated meeting/connection establishment, content filtering, automated content sharing on social networks, dynamic/active query and download, monitoring, notification of events in the community, etc. The definition of a set of rules for the local management of the community may simplify the life to the community users and protect the young elements, centralizing the management of the configuration and abstracting from the several services one could be interested to have in his living room, bed room, study, etc.

The paper is organized as follows. In section 2, a short overview of LOCUSGENS architecture is presented. Section 3 refers to the internal architecture of the Autonomous Community Manager to be hosted into the community structure. In Section 4, some examples are reported regarding the usage of the rule based solution and its expressivity to cope with the community management. Conclusions are drawn in Section 5.

2. LOCUSGENS General Architecture and Scenarios

According to LOCUSGENS, a community is a group of users sharing a set of needs, contents, services and experiences, and may be a small set of locations. LOCUSGENS community can be a family, a group of friends, a public laundry, a restaurant, a golf club, a social group attached to one or several social networks, etc. In Figure 1, the main elements of the LOCUSGENS architecture are reported. The LOCUSGENS architecture includes a set of Logically Accessible Directories (LAD services) that are providing services to the Autonomous Community Manager (**ACOM**). LAD services can be Social Network, content providers, web TV, weather forecast, EPG for TV or other, travel agency, kids schools, golf club, other ACOMs as well, etc.

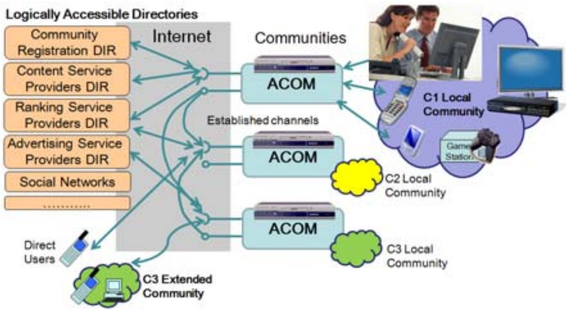


Figure 1 - LOCUSGENS general architecture

The ACOM of a community can be hosted in a dedicated device or in a media center, decoder, set top box, etc. The ACOM can profile local users and according to rules/policies it can manage the community by providing content services to the community devices. In these cases, the content is considered as a set of complex data items where several different kinds of streams of information need to be provided, where multiple paths and possibilities as occurs including classical audio/video (DVB-T, DVB-S..) for IPTV/WebTV, but also educational content, cultural heritage content, games, collections, guides and manuals, cross media reports, play lists, etc., examples are MPEG-21, AXMEDIS, SCORM, MPEG formats, HTML, SMIL, EPUB, etc.

The advantages of the LOCUSGENS architecture are: (i) scalability in terms of socially produced and shared content; (ii) decoupling of content services and advertising; (iii) flexibility in performing for the community repetitive and boring activities from/to other communities and social networks, etc.

ACOM policies can be for example:

- content services identified or matching a certain profile can be accessed only at a given time window during the day or from a specific device of the Community, for example the main TV set and, thus, not by the TV in the kids room.
- locally produced video has also to be accessible on the community mobiles and on a set of other identified communities "Carl's fast food", "Wells Golf Circle", "Robinson Family" and "Smith Family".
- use MPEG-4 video formats when the network connection is strong enough, in alternative use YYY formats at low bitrate.
- videos tagged with "ski" can be made accessible by the "Rocky Ski Club" community and to uncle Tom.
- At 20:30 the ACOM has to request a comment on Political news from Philip's friend Carl belonging to the Community "Smith family". (the news can be identified on the WEB, or recorded by the ACOM and be accessible, while the comment can be taken and protected for Philip only).
- When Tom connects show him last Amazon books on fishing.
- Philip's video of last week-end can be accessed/played from external users of the ACOM belonging to the following list of associated communities: Smith Family, Uncle Bob, Pub Happy Boat, or by people from 35-45 year old.
- Please access and record the weather information on the web during the day and make forecasts accessible at 19:00, together with the recorded news of the 14:00 on DVB-T Channel RAI1, and those of BBC.
- At 20:00 the ACOM WEBCAM is activated to establish a contact with the grandmother's ACOM, so as to have dinner together, automatic connection on both ACOMs, making some automatic adaptation, storing the videos in the ACOM.
- Allow reception of Advertising in Push if it matches with certain descriptors provided for or it is associated to specific content services such as DVB-T, SKY, WEBTV, Social Network, other ACOMs, etc.
- Philip's service about his blog and news on his experiences on gravity lifters; is suggested to be accessed by people older than 15 years, interested in science bricolage as hobby.
- At the 15:00 please activate a query on YouTube to get the first 15 videos on "ski Rocky" at high resolution, do not take duplicates with respect to what has already been taken in the past. Philips has imposed this policy to avoid waiting for download and see videos at dinner-time with his friends.
- Go periodically on the connected social networks and communities to download statistics and reports about the usage of our content in other communities.

3. ACOM Architecture

The ACOM architecture is presented in Figure 2.

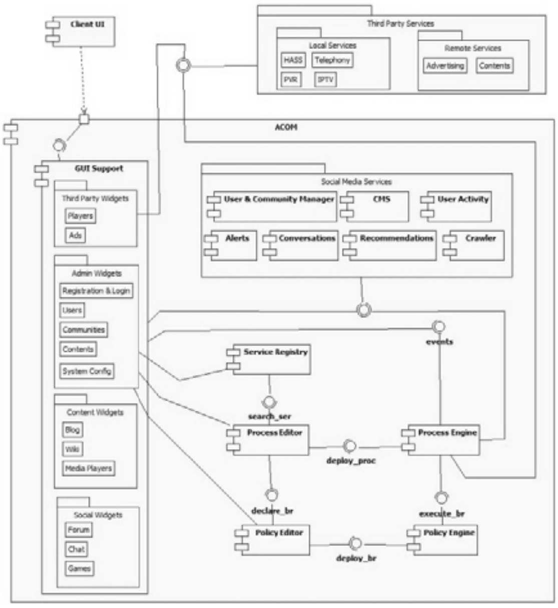


Figure 2 – ACOM architecture

The ACOM architecture includes:

- **GUI support**: the front end interface for the devices and applications that are used in the local community and virtually from the Internet. This interface allows to mount the selected content services from the LOCUSGENS network of ACOMs communities depicted in the previous figure and to make them available for the community devices and users. As widget manager the Liferay solution has been adopted [9]. In that environment a number of widget are available as open source and may be enhanced to interact with the Process Engine and Policy Engine to make them manageable in the ACOM.

- **Service Registry**: It is the entity to which each new service has to be registered to provide connection with the Process Engine to put the connected services at the disposal of the ACOM.
- **Social Media Services**: These services have been directly developed as native tools to provide a minimum number of basic capabilities in the ACOM middleware such as: managing users, content management system, crawling in the local area network for collecting resources and media, direct connection and chat, events and asynchronous messages. All these services are at disposal of the Process Engine and thus also at disposal of the Policy Engine.
- **Process Engine and Editor:** the Process Engine is an intelligent engine of the ACOM devoted to the user administration with user access control, and manages the actions in response of the widgets which are mounted on the user profile on the ACOM interface. The Process Engine is also responsible of considering semantic aspects of content services: (i) user actions, preference, favorites, etc.; (ii) content descriptors; (iii) local and external events; (iv) actions performed on provided and exported services; etc., to provide local personal recommendations and advertising support (static and dynamic user profile). In this case, the recommendations provided by external content services may be taken into account as well via events and external information.
- **Policy Engine and Editor**: The Policy Engine is the "intelligent" engine of the ACOM to process and execute ACOM policies as defined and downloaded by the ACOM and associated with Content Services and other services. The ACOM Policy Engine can take decisions to serve the local community and has been realized in Drools [10]. The capabilities of the engine are enforced by using semantic processing functions. Policies are formalized according to the following EBNF format (in which not all constructs are reported), while at the user a simpler interface to formalize the rules via a wizard is provided as shown in the rest of the paper.

```
<Rule> = WHEN <PredicativePart> DO <ActivePart>
<PredicativePart> = <FiringCondition>,
            {<MatchExpression>}
<FiringCondition> = <InstantCondition> |
            <EventCondition>
<EventCondition> = ON EVENT WITH TYPE
            <EventLabel> FROM <DomainCondition>
            [FOR <DomainCondition>]
<DomainCondition> = <ObjectType> <Identifier>
            ([ <Constraint> {AND <Constraint>}] )
<Constraint> = [NOT] <Atom>
<Atom> = <Attribute> <OP> <Parameter> |
            <Predicate>({<Parameter>})
<MatchExpression> = MATCH <DomainCondition>
<ActivePart> = <SyncStatement>,
            {<AsyncStatement>}
<SyncStatement> = Block action | Allow action
<AsyncStatement> = <StatementName>
            ({<Parameter>} )
```

It can be noted that some of the constructs of the Policy Rules directly refer to services provided in the Social Media Services (such as: sending events, getting status, etc.), while other are generically referred to dynamically mounted services coming from the Widgets connected by the users for creating their working user interface. The user interface for setting rules has been realized in Drools Guvnor.

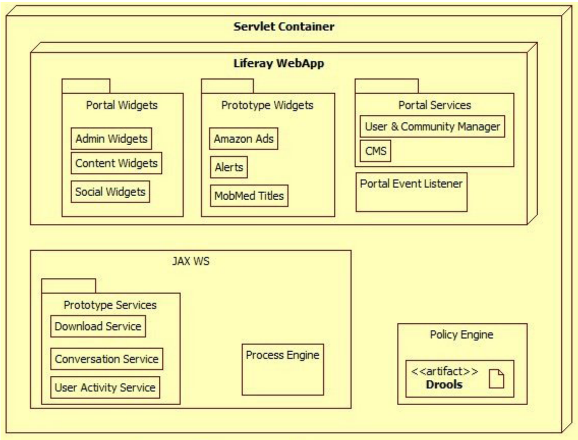


Figure 3 – Details of ACOM architecture

The LOCUGENS widgets are those accessible as Liferay on the web plus those specifically developed to be controlled by the Process and Policies Engines. All the widgets are hosted on Liferay support, but only the LOCUGENS widgets can be managed by the Process and Policy Engines. In order to validate the ACOM solution a number of Widgets have been implemented according to the ACOM communication model, among them:

- Amazon Ads Widget realized to interact with ACOM engines to provide queries and get the published books matching with the a user profile,

- Alerts Widget has been realized to interact with ACOM engines to receive information and events, for example the reaching of a message, the occurrence of a certain condition, the login, the creation of a connection, etc.,
- Content Widget can be instantiated in three different manners as: content player, content list viewer, and as interface to set preferences. It has been realized to be installed as a Liferay portlet and to interact with ACOM engines. In Figure 4, the class diagram showing Content Widget and related classes is presented.

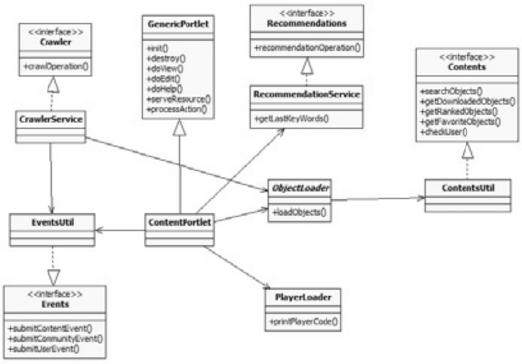


Figure 4 – Class diagram of Content Widget

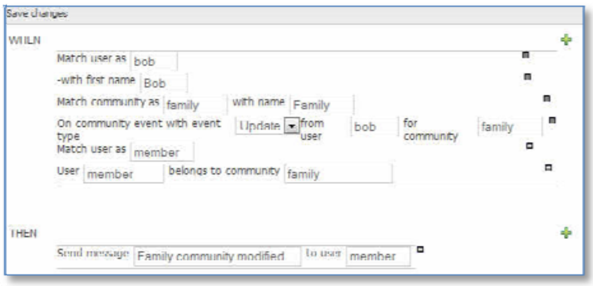
- Mobile Medicine (http://mobmed.axmedis.org) is a best practice network for medicine. In this case, a widget has been created to provide into the ACOM special services from Mobile Medicine: getting content and providing content from/to the social network, etc. The Mobile Medicine social network presents a web service to provide those services towards its widget counterpart for the ACOM.

Other services are available as well such as Portal Event Listener, Conversation Service e Download Service, User Activity Service (UAS), etc. The UAS service is focused on managing the dynamic profile of the users collecting their use data.

4. Some Examples

The solution proposed has been tested by setting up a number of ACOM services into some communities. These where comprised of few people, typically families and their friends. On these bases, a number of rules have been created by using a graphic user interface as depicted in Figure 5. The creation of a new rule is guided by a sort of wizard and should be improved to make it realistically understandable by any final user. An example of rule to prevent access to video content for Junior users belonging to the "Family" community is reported:

```
WHEN
   ON EVENT WITH TYPE "Play"
   FROM User u(age < 18,
         belongsToCommunity("Family"))
   FOR Content c(format == "Video")
DO
   Block action
```



Match user as bob

-with first name Bob

Match community as family with name Family

On community event with event type Update from user bob for community family

Match user as member

User member belongs to community family

Send message Family community modified to user member

Figure 5 – Example of Rules on ACOM

The following rule activates a download for a given content when it matches a certain raking level:

```
WHEN
   ON EVENT WITH TYPE "Content Available"
   FROM User u(firstName == "Alice")
   FOR Content c(ranking > 3)
DO Download c
```

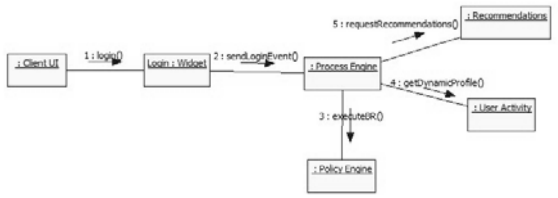
The next rule states that on December, 25 at 22:00, any user receive recommendations for the keyword "Christmas".

```
WHEN
  ON Time(date == 25,
    month == "December", hour = 22)
  MATCH User anyUser()
DO Ask for recommendations
      for anyUser with tags "Christmas"
```

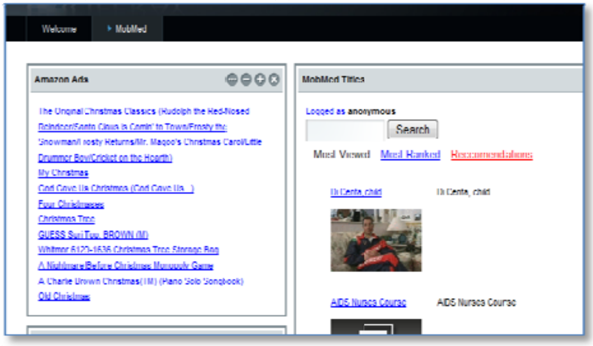
The following rule activates recommendations requests based on the users' dynamic profile.

```
WHEN
  ON EVENT WITH TYPE "Login"
  FROM User anyUser()
DO  Ask for profiled
  recommendations for anyUser
```

Figure 6a represents the collaboration diagram describing the relationships among the entities involved into the execution of the previous rule. While Figure 6b shows a segment of the screen in which the advertising produced by querying on Amazon portal on the basis of user profile (static plus dynamic) is presented. So that exploiting the preferences manifested during the accesses on Mobile Medicine to provide personal local advertising of Amazon content.



(a)



Amazon Ads

The Original Christmas Classics (Rudolph the Red-Nosed
Reindeer/Santa Claus is Comin' to Town/Frosty the
Snowman/Frosty Returns/Mr. Magoo's Christmas Carol/Little
Drummer Boy/Cricket on the Hearth)
My Christmas
God Gave Us Christmas (God Gave Us...)
Four Christmases
Christmas Tree
GUESS Suit Top, BROWN (M)
Whitmor 6120-1636 Christmas Tree Storage Bag
A Nightmare Before Christmas Monopoly Game
A Charlie Brown Christmas(TM) (Piano Solo Songbook)
Old Christmas

MobMed Titles

Logged as anonymous

Search

Most Viewed   Most Ranked   Recommendations

Di Centa, child   Di Centa, child

AIDS Nurses Course   AIDS Nurses Course

(b)

Figure 6 – Personal local advertising produced dynamically on the basis of Amazon content and local user profiling.

5. Conclusions

In this paper, LOCUSGENS architecture and solution and in particular the features and the capabilities of the intelligent local Autonomous Community Manager which includes a rule based system to cope with the harmonization and concertation of a number of third party services have been presented. The solution proposed goes in the direction of locally managing services for communities. Among them content services but also news, connection, advertising, control and supervising, automating some boring activities of content collection and sharing among friends and social networks. The components presented in details in this paper refer to the engine that one could host on a media center located into the community premises to provide services to the local community. The added value of the solution resides on the possibility of setting up a set of "smart" rules to automatically perform synchronous and asynchronous activities for the community such as: download, local recommendation based advertising, privacy control, automated meeting/connection establishment, content filtering, automated content sharing on social networks, dynamic/active query and download, monitoring, notification of events in the community, etc. From the experiments we have verified that they can simplify the life of the community reducing boring activities and creating more stable connection among other communities, the larger families, the houses of the close friends, etc. The work has to continue in the direction of making more formal validation.

Acknowledgments

The authors would like to express their thanks to all the project partners and to Mobile Medicine project partners.

References

[1] Bourdoncle, F. (2008). Future Internet: Multi-Media Search Engines and Scalable Infrastructures. Conference Session Report on The Future of the Internet, ICT Event.

[2] A. Colzi, T. Martini, P. Nesi, D. Rogai, ``A Component-Based Solution and Architecture for Dynamic Service-Based Applications", The Nineteenth International Conference on Software Engineering and Knowledge Engineering, SEKE'2007, Hyatt Harborside at Logan Int'l Airport, Boston, USA, July 9-11, 2007, Organized by Knowledge Systems Institute.

[3] P. Bellini, P. Nesi, D. Rogai, "Exploiting MPEG-21 File Format for cross media content", Proc. Of the International Conference on Distributed Multimedia Systems, DMS 2007, September 6-8, 2007, San Francisco Bay, USA, Organised by Knowledge Systems Institute.

[4] R. Geambasu, C. Cheung, A. Moshchuk, S.D. Gribble, H. M. Levy. Organizing and Sharing Distributed Personal Web-Service Data.

[5] D. Recordon, D. Reed. OpenID 2.0: A Platform for User-Centric Identity Management. DIM'06, November 3, 2006, Alexandria, Virginia, USA.

[6] D. Reed, L. Chasen, W.Tan. OpenID Identity Discovery with XRI and XRDS. IDtrust '08, March 4-6, 2008, Gaithersburg, MD, USA.

[7] M. Hatakeyama, S. Shima. Privilege Federation between Different User Profiles for Service Federation. DIM'08, October 31, 2008, Fairfax, Virginia, USA.

[8] MPEG Group, "Introducing MPEG-21 DID", http://www.chiariglione.org/mpeg/technologies/mp21-id/index.html

[9] Yuan, J. X. (2009). Liferay Portal 5.2 Systems Development. Packt Publishing.

[10] Drools: http://www.jboss.org/drools

[11] Charfi, A., & Mezini, M. (2004). Hybrid web service composition: business processes meet business rules. Proceedings of the 2nd international conference on Service oriented computing, (p. 30 - 38).