Aggregating information and enforcing awareness across communities with the Dynamo RSS feeds creation engine: preliminary report

F. De Cindio*, G. Fiumara**, M. Marchi^, A. Provetti**, L.A. Ripamonti* and L. Sonnante§

*DICo, *Università degli Studi di Milano*
**Dip. Di Fisica, *Università degli Studi di Messina*
^ DSI, *Università degli Studi di Milano*
§*Fondazione Rete Civica di Milano*

ABSTRACT

In this work we present a prototype system aimed at extracting contents from online communities discussions and publishing them through aggregated RSS feeds.
The major foreseeable impact of this project to the Community Informatics field will be helping people to manage the complexity intrinsic in dealing with the huge amount of dynamic information produced by communities, in particular, keeping up with the evolution of several simultaneous discussions/information sources.
A special version of the Dynamo system, which is described here, was deployed to endow RSS channels to the forum of the Milan Community Network (RCM).

KEYWORDS: *community informatics, on-line community, community network, knowledge management, artificial intelligence, knowledge sharing, RSS, XML.*

1. Introduction: human attention in the modern time

Already in 1971 Herbert Simon was envisioning the advent of the so-called "attention economy", claiming that "*...in an information-rich world, the wealth of information means a dearth of something else: a scarcity of whatever it is that information consumes. What information consumes is rather obvious: it consumes the attention of its recipients. Hence a wealth of information creates a poverty of attention and a need to allocate that attention efficiently among the overabundance of information sources that might consume it*" (Simon 1971, pp. 40-41).

Nowadays his assumptions have proven to be farseeing, since the "attention economy" has become the everyday reality we are living in. Undeniably, one among the most time-consuming human activities in modern times is keeping up to date with a huge amount of continuously evolving and changing information, conveyed to us through a mix of different multimedia source, whose evolution too, is a restless

process. As a consequence of this overexposure, people's attention has become one among the rarest and most valued resources.
Business world (and especially researchers and practitioners involved in the organizational, human resources management or marketing disciplines) are well aware of this phenomenon, and are struggling to define how to appropriately deal with it: let's think of the growing interest around virtual and networked organizations (Lipnack & Stamps, 2000), communities of practice (Wenger, McDermott & Snyder 2002), or tribal marketing (Kozinets, 1999 or Cova, 2003), but it pervades every aspect of our lives, in every moment, being it a working or a leisure one.

The complexity intrinsic in dealing with such information overload grows exponentially for people actively involved in on-line communities, since they must (or want to) keep the pace of discussions that may be very "active" and even split across multiple threads, or – in the worst case – among different (sub)communities. This means not simply adding one (or more) information source, but also being able to manage the emotional involvement and the selection of the highly-prized relevant pieces of information in the sea of chatters that form the humus upon which the collective intelligence that creates knowledge in the social network buds (Polanyi, 1967 Wenger, McDermott & Snyder 2002, Armstrong and Hagel, 1998).

This escalation in the amount of information we [are forced to] process every day underpins the skyrocketing evolution of ICT (Information and Communication Technology), that continuously makes new interaction media available, and that has been paralleled by a similar evolutionary process in the ways people are using (and from time to time also twisting) them to support social interactions through innovative paradigms, enabling more effective informative and emotional exchanges. Perhaps the most outstanding demonstration of this process is the increasingly successful "podcasters[1] movement."

2. RSS, ATOM, aggregators and other "exotic technologies"

One relevant step in the direction of helping people managing such complexity has been made by Really Simple Syndication (RSS) and ATOM readers, that enable users to collect simultaneously news from different selected sources. RSS readers completely revolutionized the paradigm according to which people collects information from the news published on the Internet: it is no longer the user that searches the information she is looking for, but it is the information she values that reaches directly its "consumers", by downloading on their devices (PC, iPod, etc.).

[1] **Podcasting** is the method of distributing multimedia files, such as audio programs or music videos, over the Internet using either the RSS or Atom syndication formats, for playback on mobile devices and personal computers. The term **podcast** like 'radio', can mean both the content and the method of delivery. The host or author of a podcast is often called a **podcaster**. Podcasters' web sites may also offer direct download or streaming of their files; a podcast however is distinguished by its ability to be downloaded automatically using software capable of reading RSS or Atom feeds.

Usually a podcast features one type of 'show', with new episodes released either sporadically or at planned intervals such as daily or weekly. In addition, there are podcast networks that feature multiple shows on the same feed (from Wikipedia – www.wikipedia.org).

Unfortunately this solution is still too basic, since RSS readers offer a very limited interactivity to users, that are enabled uniquely to subscribe the specific "feeds" they are interested into. Clearly they are very helpful, avoiding people to spent an amount of time daily to search for the information they need across scattered sources, but they are unable to select only "really relevant" information or to aggregate them consistently with their semantic meaning.
If this limit is quite tolerable while reading feeds collected from "news" (e.g. an on-line newspaper) – after all I can quite easily skip irrelevant news by reading their title -, the problem assumes different boundaries when applying RSS readers to on-line communities. In this latter case no editing service exists to format information according to a uniform standard; on the contrary, information flows freely in the stream of the discussions, and people posts their opinions and messages according to their mood, not worrying about how their posts will look like if collected by a RSS *aggregator*[2]. According to this paradigm, it is quite impossible for the user to distinguish between two (or more) different replies made by different community members to the same post, unless reading the whole contents. This may add a dramatic overhead, that risks to reduce to zero the advantages offered by the RSS reader (after all, what is the utility of downloading on my PC exactly the whole discussion taking place on my favoured on-line community instead of reading it on-line?).

To address this problem we are now working on the Dynamo project, which is aimed at developing an RSS feeds creation engine based on artificial intelligence techniques – namely the Answer Set Programming (ASP) - in order to provide community members with sophisticated news digests, tailored on their specific information needs. At the moment, a special version of the Dynamo system is undergoing a preliminary phase of testing in several forum of the Milan Community Network (RCM: *Rete Civica di Milano*, cfr. De Cindio et al., 2003).

This article is organized as follows: after a brief survey of news syndication techniques we will give an account of the RCM community network, where the Dynamo content extraction technique, which is described next, has been applied.

3. Starting point: the current RCM technological platform

RCM is a network of more than 20.000 citizens living in the Milan area, forming an community network (CN – cfr. De Cindio & Ripamonti, 2006), whose main focus is "being a citizen of Milan" (in the broadest sense of the term). RCM has been founded in September 1994, as an initiative of the Civic Informatics Laboratory at the University of Milano, and, since then, it has developed several projects aimed at designing and implementing services for citizens, local no-profit associations and local small businesses.

[2] An **aggregator** or **news aggregator** is a type of software that retrieves syndicated Web content that is supplied in the form of a web feed (RSS, Atom and other XML formats), and that are published by weblogs, podcasts, vlogs, and mainstream mass media websites.

From a technological point of view, RCM core community services are currently managed through the *FirstClass* server, produced by OpenText Corp.
FirstClass is a messaging and communications platform, mainly addressed to schools, learning organizations and businesses, but also to on-line community and CNs. FirstClass provides its users with the ability to communicate and share resources and information via email, conferencing, directories, individual and shared calendars and on-line chats. All these features can be accessed both through a Web interface and a proprietary client.

In order to understand RCM technological choice, it's important to bear in mind that, when RCM started in 1994, dialog and communication facilities through the Internet were not user-friendly enough for "everyman" (at that time they were implemented on text-based BBS (Bulletin Board System), and the World-wide Web was in its early stages, still not providing interactivity). In that technological landscape, FirstClass was, and to some extent still is, among the solutions with the best trade-off between costs and requirements. Indeed, FirstClass:

- is highly interactive, hence good at supporting active citizens producing contents of local interest;
- is endowed with an easy-to-use windows-based interface;
- supports a wide range of communication protocols;
- supplies a light "client application", that guarantees reasonable performances even with cheap/out of date personal computers and modems (thus being an affordable mean of communication for a large group of potential users).

At present, RCM is still running its community services using FirstClass. This is mainly due to the fact that changing abruptly the communicative infrastructure would have too relevant an impact on users' habits. However, new services are developed gradually adopting open source technologies (e.g., Web applications integrated with FirstClass-based user authentication and forums).

The central element of a FirstClass system are the *forums* (or the so-called "conferences"): discussion areas where information publishing takes place by sending messages. When a community member is interested in a specific forum, she can subscribe to it, that is, put a link to it into her personal home page. New messages sent in the forum are marked with a red flag icon appearing near the forum icon, in order to enable community subscribers to monitor a forum activity simply by logging into the system and scanning through icons on her home page. Messages are then displayed to community members organized in lists, showing the author, the subject and the creation date, but not the body of the message, that can be accessed only by following an appropriate link.
Obviously this approach requires a certain amount of overhead activities for the community member: actually, for being aware of the real relevance of posts and people interacting in the discussion area, she has to log into the system, scan the forums' icons, open the forums she is interested into, check for new messages, open them in order to access their contents (with no prior indication of their relevance to the discussion or to her interests). Moreover, at the moment, a notification mechanism via email for subscribed forums still does not exists.

From a more technical point of view, forum and messages are managed in different ways: forums can be nested, forming a tree structure, while messages are all at the same level, but can be grouped into threads. Forums can be both public or private: in particular, in public ones messages can be posted only by authenticated community members, but can be read by anyone visiting the CN Website.
Forum and message list are rendered via Web as dynamic pages generated through templates coded as a mix of HTML (for the fixed part of the page: headers, frame disposition, etc.) and Server-Side-Include-like (SSI) scripting language (for the variable part: the message list, the forum list and so on). These templates can be accessed and customized by the FirstClass administrator. However, customization cannot alter:

- the order in which templates are picked-up to compose the whole page;
- the availability of objects in the rendering phase (e.g. the body of a message is unavailable – due to internal server processing constraints - during the message list composition).

Even though the FirstClass customization features could be exploited to produce an XML/RSS version of the message list, two main drawbacks remained:

- the list of items could not contain the description of the news (i.e. the body of the message), due to the above mentioned processing constraints;
- every forums can have its own RSS feed, but there is no way to integrate in one single RSS feed contents derived from two or more different forums.

These two drawbacks imply that the RSS feeds would reproduce in a “local version” exactly the same structure of the online system, hence also with the same overheads we have described before for the community member accessing the contents.

4. The Dynamo extractor

The Dynamo Project[3] (Bossa, 2005, Bossa et al., 2006) addresses data extraction and channeling over legacy Web sites in plain HTML. Dynamo is intended to benefit two types of users. First, webmasters may employ it to manage the creation of RSS feeds, thus avoiding to do it by hand or by means of proprietary software. Second, users, i.e., consumers of feeds, may use it to overcame limitations such as i) old feeds may not be consulted and usually are deleted from servers and ii) traditional HTML servers cannot execute advanced queries *directly*.
On the contrary, with Dynamo it becomes possible to:

- automatically and dynamically generate RSS feeds starting from HTML Web pages;
- store feeds in chronological order;
- query and aggregate them thanks to Web Services (WS) acting as agents.

[3] Dynamo is an open-source project available, under GPL, from *http://dynamo.dynalias.org/*

It is important to stress that these results were obtained with a lightweight pull algorithm for retrieving HTML documents by Web servers, thus minimizing the required Web traffic for the updates of news sources (Bossa et al., 2006).

HTML documents contain a mixture of information to be published, i.e., meaningful to humans, and of directives, in the form of tags, that are meaningful to the browsers and determine the appearance on the screen. Moreover, since the HTML format is designed for visualization purposes only, its tags do not allow sophisticated machine processing of the information contained therein.

Among other things, one factor that may prevent the spread of the Semantic Web is the complexity of extracting, from existing, heterogeneous HTML documents machine-readable information. Although the Dynamo project addresses only a fraction of the Semantic Web vision, our management of HTML documents needs some technique to locate and extract some valuable and meaningful content. Therefore, a set of annotations, in form of meta-tags, were defined; they are inserted inside HTML in order to highlight informational content that is essential for the creation of a RSS feed. In our application, meta-tags are used as annotations, to describe and mark all interesting information, in order to help in the extraction and so-called *XML-ization* phases. Notice that with pages that are dynamically generated out of some template (which is the case with practically all on-line fora) Dynamo annotation is done, manually but only once and for all, over the page template.

Once HTML documents are processed by Dynamo, annotated semantic structures are extracted and organized into a simple XML format to be stored and used as a starting point for document querying and transformation. The structure of the XML output resembles the structure of meta-tags previously defined and the RSS XML structure, in order to facilitate transformations from the former to the latter.

4.1 Dynamo: Structure and underlying technical choices

In order to allow a full transparency with respect to the action of scripting languages (for example, ASP, PHP, JSP) that produce dynamic [X]HTML pages, Dynamo meta-tags are enclosed inside HTML comment tags, which implies that both Web browsers and scripting language interpreters simply ignore them.

Another important element is the possibility of querying a Dynamo database through Web services which, for sake of simplicity, have been designed for the REST (Fielding, 2000) style of interaction (see further).

Dynamo is a Java application with a modular structure, for maximizing the flexibility and the extensibility of configuration. Three levels can be distinguished:

- the *Physical Data Storage Level*. It is the lowermost level, which stores resources, and provides a means for retrieving and querying them. It can be implemented over either relational or XML database Management Systems (DBMSs);
- the *Core Level*, which holds the core part of the entire architecture, including the software components that implement the logic of information management and processing. Each component can be implemented using different strategies or

algorithms, and plugged into the system without affecting other components, i.e., by simply tuning the application configuration files;

- the *Service Level*, which is the highest level, interacting with Web clients by means of REST Web services.

A more detailed description follows.

The Physical Data Storage Level can be implemented using various techniques. Currently, Dynamo uses the open-source native XML database Exists (Bourret). This choice allows to store and manage XML documents produced by the Wrapper software component in their native format, and to use the powerful XQuery language (Xquery, 2005) for advanced content querying and aggregation. The native XML database is organized as a set of collections of XML resources, where the nesting of collections is allowed. In our application, we store XML resources as provided by the Wrapper software component, one collection for each resource. Each collection holds the various chronological versions of the resource: so, each collection effectively contains the history of the resource, all its informational content and a changelog.

When a new resource has to be stored, the DataManager component performs checks to avoid duplicate resources. Two resources are considered different when their informational content changes. More precisely, they are different if changes to titles, links or descriptions of the resource channel or items are detected. Once stored, the resource is chronologically archived and ready for later retrieving and querying.

The Core Level consists of several components which define how Dynamo extracts relevant information from Web sources, processes them in order to extract semantic information and finally formats them for clients' consumption:

- the *Poller* monitors changes in a set of HTML sources (defined in a particular configuration file) using a flat polling policy, that is at regular time intervals or, even better, a “smart” polling policy. The latter consists in estimating the time of the next publication of the news on the basis of previously published news;
- the *Retriever*, invoked by the Poller, captures the HTML files and passes them to other components for further computation and storing;
- the *Wrapper*, which extracts the semantically relevant information from HTML files and creates an XML file containing the desired informational content;
- the *DataManager*, a gateway to the Physical Data Storage Level. It takes care of managing information in the form of the new XML documents previously created, storing them and permitting client components to query their contents;
- the *Transformer*, which transforms the XML file into any of the desired RSS formats. It uses suitable XSLT transformations in order to do this;
- the *Engine*, which coordinates all previously described components.

The Service Level lets Web clients access the RSS feeds through the use of REST Web Services.

REST is the acronym of *Representational State Transfer.* It is an architectural style which conceives everything as a resource identified by a URI. In particular, it

imposes a restriction about of the URL defining the page info, that, in the REST view, are considered resources. Each resource on the Web, such as a particular part specification file, must have a unique URL (without GET fields after it), that totally represents it.
This allows the client (and all proxy/firewall systems in the middle) to define the next state change only by inspect the URL of current available forwarding links (for the proxy/firewall systems, only by inspect the header of the HTTP request).

With respect to the well-known SOAP architecture (SOAP, 2003), in REST we never access a method on a service, but rather a resource on the Web, directly using the standard HTTP protocol and its methods, i.e., GET, POST, PUT and DELETE.
This feature of REST allows greater simplicity and maximum inter-operability with any Web client, either *thick,* like a desktop application, or *thin,* like a Web browser.

5. Applying Dynamo to the RCM community

As shown above, the template generation model adopted by the FirstClass system, although widely flexible, cannot be used directly for generating a *useful* RSS index file. However, the template model easily permits to put the special mark tags inside Web pages and to carefully choose where to place tags inside the text. This option, however, is an all-or-nothing type: either all generated pages are marked or none will. As a results, experimentations can take place according to an incremental schema, since single subgroups of forums can be processed separately, selecting them accordingly whichever criterion we feel adequate for testing a specific aspect or functionality of Dynamo (e.g. semantic similarity, frequency of posts, presence of attachments/pictures, etc.). Presently we have tested the system for collecting the *title* and the *URL* of the forum, that are used to populate the channel description part of the RSS feed *index*, and the *author*, *title*, *date* and *body* of the messages posted in the forum, that are needed for filling the *item* part of the feed. The tags used to guide Dynamo during the extraction are described in the following table:

TAG	Description
<!-- <channel:title> --> <!-- </channel:title> -->	the forum title
<!-- <item:author> **-->**<!-- </item:author> -->	marks the sender of the message
<!-- <item:link index="*n*"> --> <!-- </item:link> -->	marks the link of the *n-th* message
<!-- <item:title index="*n*"> --><!-- </item:title> -->	marks the subject of the *n-th* message
<!-- <item:extension localName="date"> --> <!-- </item:extension> -->	marks the date of the message

Thanks to the insertion of the tags, the Dynamo servlet can collect the marked-up information by periodically polling the RCM forums[4], and store the parsed information into the Exists database. All the RSS generation process, polling, parsing and deploy RSS feeds, are performed by a distinct host respect to that used for RCM framework, running a Tomcat server for the Dynamo engine and a standalone instance on the *Exist* open-source, native-XML DBMS.

6. Open problems and future work

This article described the deployment of the Dynamo data extraction tool to the RCM community Web site.

The deployment of Dynamo-based RSS services to RCM forums is very recent and meaningful statistical data over adoption by the community is not yet available. Hence, the impact of the introduction of Dynamo RSSs over the community could not be assessed at this stage

The most visible result of this partnership is the solution of the *legacy barrier* that prevented RCM, locked into proprietary and perhaps a little outdated software, to offer to its users the now-standard RSS feed service (this result could be also beneficial to other communities who are using FirstClass or similar software products). Also, the availability of RSSs may somewhat facilitate (and simplify) community research over RCM.

Less visible but very interesting, in view of future developments, is the possibility to conceive and deploy sophisticated aggregation policies for the contents extracted from the community's forums. Since all RCM forums are now marked up with Dynamo meta-tags, users may effectively customize their feed channels to suit their interests and perception associated to each forum. To do so, we are developing a Dynamo customization tool to be offered on the Web to RCM users who want to decide a personal *information mix* over the several forums she may want to consult at once.

Acknowledgements

Sergio Bossa conceived and implemented the first version of Dynamo.

References

Armstrong, A.G. and J.III Hagel (1998). Net Gain – creare nuovi mercati con Internet. Etas (in Italian).

Atom Enabled (2005). Atom Syndication Format (RFC4287). http://www.atomenabled.org/developers/syndication, http://tools.ietf.org/html/4287.

[4] Currently the polling period is set to one minute. See Bossa et al. (2006) for a discussion of the tuning of polling policies.

Baumgartner R. et al. (2005). Web Data Extraction for Business Intelligence: the LiXto Approach. Proc. of BTW Workshop.

Bossa, S. (2005). Gradation Project in Informatics. University of Messina (in Italian).

Bossa, S. Fiumara, G. and Provetti, A. (2006). A Lightweight Architecture for RSS Polling of Arbitrary Web sources. Proc. of WOA conference. Available from *http://mag.dsi.unimi.it/*

Bourret, R. P., XML and Databases. *http://www.rpbourret.com/xml/XMLAndDatabases.htm*

Cova, B. (2003). "Il marketing tribale: legame, comunità, autenticità come valori del Marketing Mediterraneo.." Il Sole 24 ORE (in Italian).

De Cindio, F., Gentile, O., Grew, P. and Redolfi, D. (2003). Community Networks: Rules of Behavior and Social Structure in The Information Society, special Issue "ICTs and Community Networking", Sawhney H. (ed.), vol. 19, n. 5, pp. 395-406, November-December 2003.

De Cindio, F. and Ripamonti, L..A. (2006). Natures and Roles for Community Networks in the Information Society. Invited paper in P.Day (Ed.), AI & Society special issue on "Community Informatics."

Fielding, R. T. (2000). Architectural Styles and the Design of Network-based Software

Gottlob, G. and Koch, C. (2004). Monadic Datalog and the Expressive Power of Languages for Web Information Extraction. Journal of the ACM 51.

Kozinets, R.V. (1999). E-tribalized marketing? The strategic implications of virtual communities of consumption. European Management Journal, Vol.17, N.3, pp.252-264.

Lipnack, J. and Stamps, J. (2000). Virtual Teams: People Working Across Boundaries with Technology. Wiley.

Polanyi, M. (1967). The Tacit Dimension. Routledge and Kegan Paul, London.

Simon, H. A. (1971). Designing Organizations for an Information-Rich World. In Martin Greenberger, ed., Computers, Communication, and the Public Interest, The Johns Hopkins Press, Baltimore, MD, ISBN 080181135X.

SOAP (2003), SOAP v. 1.2. Par 0: *http://www.w3.org/TR/2003/REC-soap12-part0-20030624/*

UserLand (2005). RSS 2.0 Specifications. *http://blogs.law.harvard.edu/tech/rss*

Wenger, E., R. McDermott, and W.M. Snyder (2002). Cultivating communities of practice - A guide to managing knowledge. Harvard Business School Press, Boston, MA, USA.

Xquery (2005), XQuery 1.0 : An XML Query Language, *http://www.w3c.org/TR/xquery*