StreamOnTheFly: A Network for Radio Content Dissemination

László Kovács[1], András Micsik[1], Máté Pataki[1], and Robert Stachel[2]

[1] MTA SZTAKI, DSD,
Lágymányosi utca 11, H-1111 Budapest, Hungary
{laszlo.kovacs, micsik, mate.pataki}@sztaki.hu
[2] Team Teichenberg,
Operngasse 22, A-1050 Vienna, Austria
robert@teichenberg.at

Abstract. A distributed digital library has been designed and implemented for the support of community radios. This framework, developed by the StreamOnTheFly IST project of the EU, provides a common background for preparation, archival, exchange and reuse of radio programs and supports radio personalization. The architecture is based on a decentralized network of software components using automatic metadata replication in a peer-to-peer manner. This approach combines the principles and practice of OAI (Open Archives Initiative) with the peer-to-peer networking paradigm, and extends usual content dissemination with the aggregation of use statistics and user feedback. The network also promotes social self-organization of the community and a new common metadata schema and content exchange format.

1 Introduction

Nowadays many community radio stations (non-profit, independent stations) exist, but there is little cooperation among them. One of the causes for this is the lack of technical support for easy exchange of radio programs. A significant part of the programs produced at community radio stations are of high quality and worth to be broadcast more than once, and listened to by a larger number of people in a wider geographical range.

The StreamOnTheFly IST project [17] was set out to find and demonstrate new ways for archival, management and personalization of audio content. In the project Public Voice Lab (Austria) and MTA SZTAKI DSD (Department of Distributed Systems at the Computer and Automation Research Institute of the Hungarian Academy of Sciences) were development partners and Team Teichenberg (Austria) provided connections to community radio stations and associations.

The project team envisioned and implemented a network of digital archives and helper tools for radio stations. The general aim of this network is to foster alternative, self-organized and independent media by new ways of content distribution, international community based presentation platforms and many added values for local production and organizational work. In the following an overview on the world of radio is

C. Türker et al. (Eds.): P2P, Grid, and Service Orientation ..., LNCS 3664, pp. 188–206, 2005.

given, the concept and the current status of the StreamOnTheFly network are presented, and the architectural design of the network is explained in details.

2 Community Radios in Europe

During the past 30 years community radio stations developed in most countries of Europe, enriching the existing media world with a "third sector", next to public broadcasting as the first and commercial media as the second. Community radios are non-profit-oriented, are open to the general public and have a local or regional range. They provide access to radio production facilities for organizations, groups and individuals who aim to make their own radio shows. There are over 1500 non-profit stations in the whole of Europe, an estimated third of them is provided with a full and permanent 24 hour broadcasting license on their own frequency.

The global umbrella organization AMARC defines community radio as *"which is for, by and about the community, whose ownership and management is representative of the community, which pursues a social development agenda, and which is non-profit."* The success of a community radio cannot be assessed by audience ratings and revenues. The important factor is the effect a broadcast series may have on the community targeted. A radio show in a language spoken by a small minority will be highly appreciated by this group of people, although they would not be considered to be a market big enough to be targeted by other media.

There are a number of projects existing that aim to foster production of programs for an international audience that can be aired all over Europe or even wider. With AMARC's "Radio voix sans frontieres" (Radio without frontiers), more than 30 stations in Europe share a full day of live broadcasting once a year on the UN anti-racism day. Many exchange projects are oriented around themes of global interest, like migration, sustainable development or Human Rights.

What keeps program exchange from becoming a day-to-day routine, is the technical and structural thresholds. The existing program exchange platforms on the Internet are not connected to each other and do not follow a common metadata scheme and user interface, which makes it harder to search and offer content on an international scale, as every program needs to be uploaded and indexed in different standards for every country. The user might not even be able to understand the language a sharing platform's website is written in.

Building on the experiences made at Orange 94.0 in Vienna and in other community radio stations in Austria, we researched the requirements of producers and editors for easy access to online archiving and content sharing. A common metadata scheme and the ability to search in all databases at once are main critera, the other is to provide convenient access to the repositories. Participating in the exchange of audio content should be as easy as sending an email or using a search engine, with program recording and uploading being done automatically.

Only an easy-to-use system that encourages more people to enter their day-to-day programs into an international platform will create a critical mass of content that can be searched and re-used in local production. Such a platform would certainly help to strengthen the position of community radio and foster international cooperation in Europe.

3 Overview of the StreamOnTheFly Network

The project aimed to create a network of radio content that develops in an organic and dynamic way, in decentralized structures that build synergies on an international, community-based level. The content is offered by local producers or stations to the network, but it's the international communities who choose which content is presented and how it is presented. International publishing on the net is not a form of archiving or recycling anymore, but a motivation to communicate with a whole new group of people.

The StreamOnTheFly network consists of the following basic elements: nodes, radio stations and portals (Fig. 1). These components are introduced in more details in the following subsections. Nodes together can be seen as a distributed digital library specialized for audio content. Stations send content to the nodes for archival, and portals present thematic selections of the content of nodes. The StreamOnTheFly network tries to serve the whole lifecycle of radio programs, starting with the creation and ending with various (re)uses and discussions of the programs.

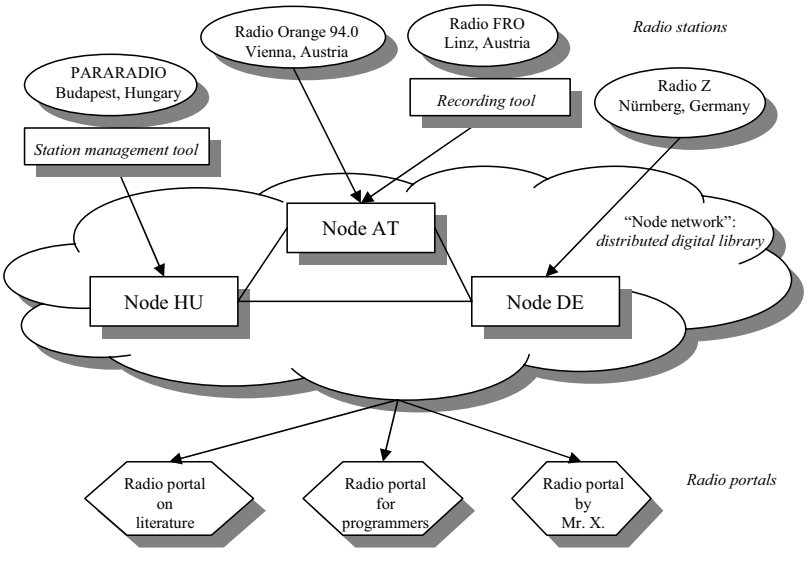

Fig. 1. Example for a StreamOnTheFly network

For example, Orange 94.0, a community radio in Vienna, may send some of its broadcast material on the node running in Austria. These programs are archived at the node and become accessible for the whole StreamOnTheFly network for a given period of time. A German radio may find a program on the Austrian node that they want to rebroadcast locally. They find the contact for this program, in this case Radio Orange and ask for their permission for rebroadcast. If there is a portal in France for the Turkish people, its portal editor may include some of the Turkish programs of Radio Orange in

the recommendation list. Users may also go directly to a node, query/browse the whole content of the network, and listen to selected programs.

3.1 Nodes

Nodes implement the basic digital library infrastructure for the network. Each node hosts a set of radio stations. For each station the node archives radio programs with a rich set of metadata and other associated content. For example, the content may be present in several audio formats, and photos may be attached to the audio content.

Users get transparent access to the content of the whole network at any node. They can browse through radio stations, topic trees or search in the archive. Continuing our previous work [14], an advanced query facility is integrated which enables users to construct and save complex queries. Queries are built step by step by composing atomic expressions (e.g. broadcasted before year 2004) and joining them with logical AND/OR. In atomic expressions the query target can be any field of the metadata scheme. The possible query syntax offered for the user changes according to the type (a string, a date, a number or an enumeration) of selected metadata element in the expression. Query expressions can be saved with a name in the users' profiles, and later reexecuted easily.

Interesting items found in search results or during browsing can be collected into a playlist. Playlists can also be used as a kind of personalized radio. A user may listen to all or some of the programs in her playlist successively. Several audio formats are supported for listening (MP3, Ogg/Vorbis) in different bit rates or quality.

The node gathers and forwards information about radio programs back to their creators. This involves the collection of various statistics (e.g. audio file downloads), the collection of comments, references and rating information.

Editors have a comfortable "console screen" where they get an overview about all of their programs, they can publish new programs and manage existing ones. Access privileges can be controlled on various objects, for example insert, delete and modify privileges can be defined for stations, series or programs.

3.2 Stations

A station management tool which connects radio stations to the network (Fig. 2) has been developed by the project. This tool provides an easy way to feed a node with radio programs. Furthermore, it helps creting the schedule of the station, and leads the creators by the hand before and after the broadcast of a radio program. Authors may add metadata to their programs and publish them on a node.

The station management tool provides editors and other users of a radio with proper access privileges. Programs and series can be scheduled for broadcast by the creators or editors, and station managers control the weekly broadcast plan. This tool provides most of the basic software support required for a radio.

A recording tool can be used when no pre-recorded audio is available for a radio program (e.g. live shows). This software records the broadcasted audio of a station, and automatically cuts the stream into individual radio shows.

The desktop upload tool is under development, it will enable creators to assemble metadata and audio files into a program package and send it directly to the node from their personal computer.

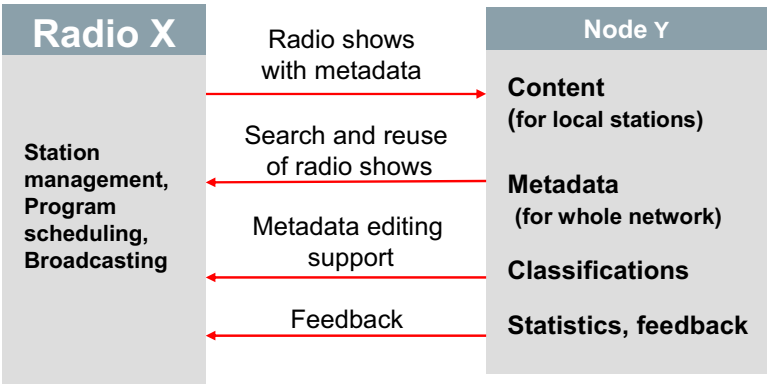

Fig. 2. Roles and cooperation of stations and nodes

3.3 Portals

In the StreamOnTheFly project a radio portal offers a selection of radio programs in a custom design (Fig. 3). Portals may serve communities or geographical regions by selecting relevant content from the whole network. Additionally, a portal can offer reviews, opinions or pictures about the selected programs. Visitors may also rate, comment and discuss programs. Comments and ratings made on portals are forwarded to the nodes, and the comments are delivered to the creators of the program via e-mail (Fig. 4).

The StreamOnTheFly portal is a specialized CMS, thus the main activities are content and design management. Design is controlled by a special HTML based WYSIWYG editor, which can be used from any web browser, without installing any extra programs, components or plugins. The user arranges a table as the skeleton of the portal page, where each cell is filled either with design elements or with content. The editor uses rows as basic screen layout elements, where any number of cells can be inserted in each row. Each cell has its own type; so far the following five have been implemented: Text/HTML, Picture, Spacer, Query, Program list. When the user clicks on a cell in design mode, current cell settings appear in a popup window and can be changed according to the desired layout. This solution requires minimal Javascript capabilities available in all new and most of the old web browsers. Using this method, users without HTML knowledge can also build elegant portals, while trained web editors may insert HTML pieces into the cells and use CSS as well.

The two cell types for presentation of content are Query and Program list. The first one is a saved query expression, which is normally constructed using the advanced query interface of the node and then it is sent from the node to the portal. With this method one can make a self updating portal page, where the newest programs are automatically added to the page. The second option is to use a Program list, which is a static collection of selected programs. The number of query and program list boxes on a page is not limited, so the editor can place chosen content on pages rather freely. Program

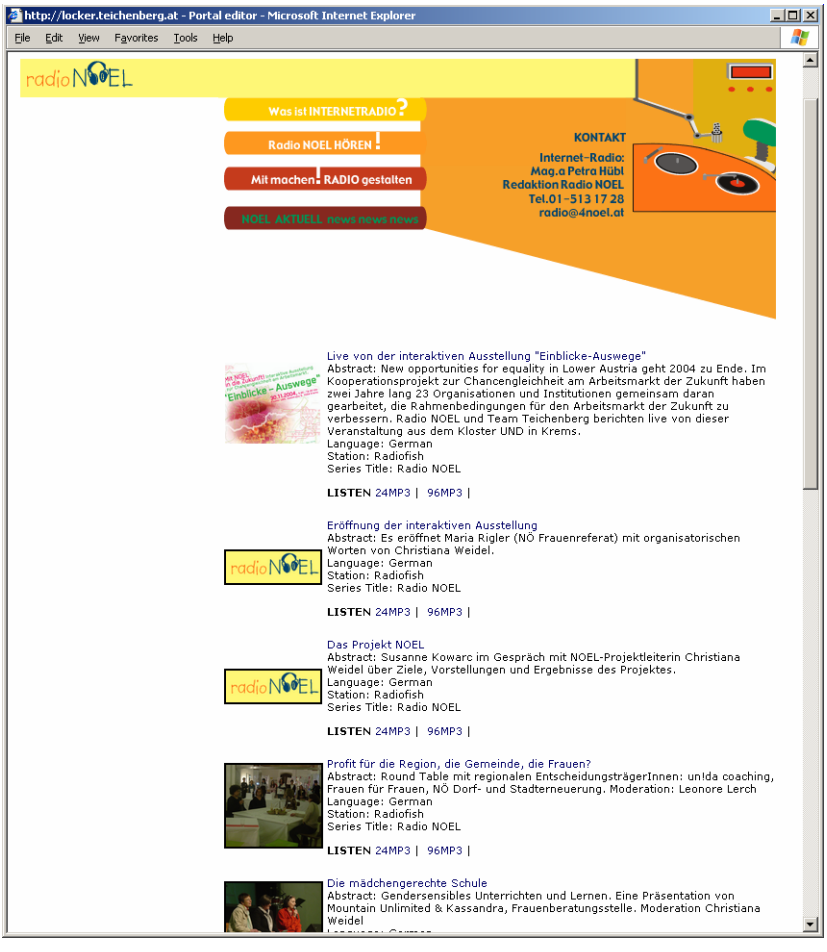

Fig. 3. One of the portals in the StreamOnTheFly network

references used in program lists, similarly to queries, can be sent from the node to the portal.

The portal editor may create a short introductory text for a program (often called a teaser in jargon), which is visible in program lists. A longer review with attached images or documents can also be given for a program. Clicking on a program in a list brings up the program page, where the review and all the other details about the program both from the node and the portal are displayed, including metadata, listen/download option, ratings, statistics and comments by users.

3.4 Content and Metadata Format

At the time of the design phase, the project team found many emerging activities for attaching metadata to multimedia content. These solutions provide schemas for metadata

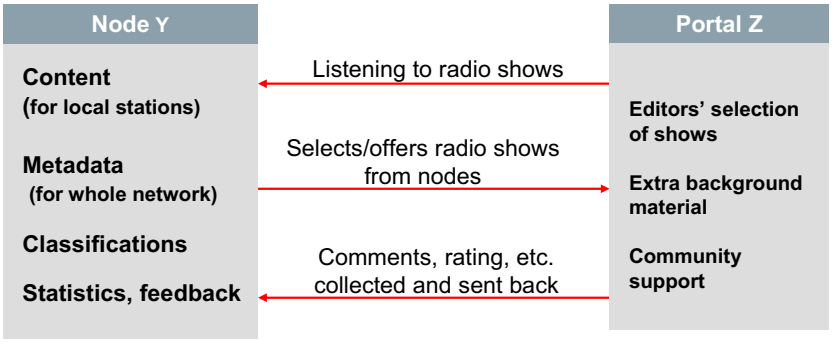

Fig. 4. Roles and cooperation of portals and nodes

and methods to attach or embed metadata into the content (e.g. SMPTE, AAF, MPEG-7 [19,6,12]). Schemas often contain an object hierarchy for metadata description and controlled vocabularies for the values of object properties. We found that these activities were overlapping and they all targeted large radio and television stations with a highly complex format and structure. Community radios expressed on various forums that the use of these metadata formats is neither feasible nor practical for them and they would welcome a simpler and more accessible recommendation for metadata usage.

The European Broadcasting Union (EBU) prepared a recommendation on a core metadata set for radio archives [9] based on Dublin Core [1]. This document defines how to use the Dublin Core elements in the area of radio archives. The StreamOnTheFly project participated in the launch of an initiative called Shared Online Media Archive (SOMA). The SOMA Group (consisting of AMARC, OneWorld, CMA and Panos) published the SOMA Metadata Element Set [3], which builds upon the EBU Core Metadata Set.

The SOMA metadata set became part of the new exchange format called XBMF (Exchange Broadcast Binary and Metadata Format) defined by the project. XBMF is designed for transfer and archival of radio programs coupled with metadata. This is a simple container format, in which metadata (in XML format), content files and other associated files can be packaged together. XBMF may contain the main content in several audio formats and also any other files related to the content (e.g. photos, documents, logos). XBMF can easily be applied for video as well.

Technically, an XBMF file is a compressed archive of a fixed directory structure, where naming conventions and placement of files identify the data they convey. The implementation of software for handling XBMF files is straightforward, and can be based partially on existing open software. The content of an XBMF archive can be examined on every computer platform with standard utilities.

The metadata part of XBMF can be seen as an application profile of Dublin Core for radios. This profile is constructed from Dublin Core elements and the StreamOnTheFly element set according to the rules implemented for metadata schema registries in the CORES project [13].

4 Architectural Aspects of StreamOnTheFly

The most important requirements to form the base of our implementation decisions were:

- no organized hierarchy in the operation of the network can be expected,
- most of the services in the network should be decentralized and without a single point of access,
- user access to radio shows should be fast and effective (focusing on search facilities),
- the amount of network communication for each node should be controllable.

These properties are easily deducible from the usual characteristics of community radios. These radios act very independently, and they operate on a rather low budget. There is little chance to establish rigid rules or hierarchies of cooperation among them. Based on the above facts, a working solution in case of a very loose cooperation was sought. The criterion of controllable network load originates in the financial status of these radios and radio associations. They need to plan the cost of their Internet connections and hardware equipment well in advance and from a low budget. Despite all these restrictions, searching and browsing should be global and effective, otherwise users will not change to this new platform.

The requirements clearly drove us towards a solution where the paradigms of peer-to-peer communication and component-based architecture are applied.

The architecture can be seen as the evolution of the OAI approach [15] using peer-to-peer communication. The principle of open archives proved to be useful and it is becoming widespread for connecting repositories worldwide. Basically the principle is that valuable content remains under the control of the creator or publisher, while less valuable metadata is made available through a well-defined protocol.

In the StreamOnTheFly network metadata is diffused using peer-to-peer communication in a push manner (Fig. 5). Content metadata for the whole network is available at all nodes of the network, which creates the possibility of fast searching and advanced query interfaces based on metadata. The user interface functionality (including searching, browsing, listening) is also available on all nodes of the network. In this situation the traditional data provider, harvester and service provider components are all present at each node of the network.

The push method used for metadata replication was considered better than the pull method used in OAI, as fast distribution and notification of new radio programs is better achieved this way. The fact that a radio show could be quickly and surely removed from the network, also voted for push-style communication. More details on metadata replication can be found in section 4.2.

Replication of content was considered as not feasible due to the relatively big size of broadcast quality audio material (the storage requirement for one hour audio material was calculated as 65MB) compared to the storage capabilities of node operators. The size of metadata for a single radio program is usually between 2kB and 25kB which presents no problem for replication on all nodes.

Nodes in the StreamOnTheFly network have to meet certain capabilities: they need a permanent, wide bandwidth, reliable connection to the Internet so that they can handle

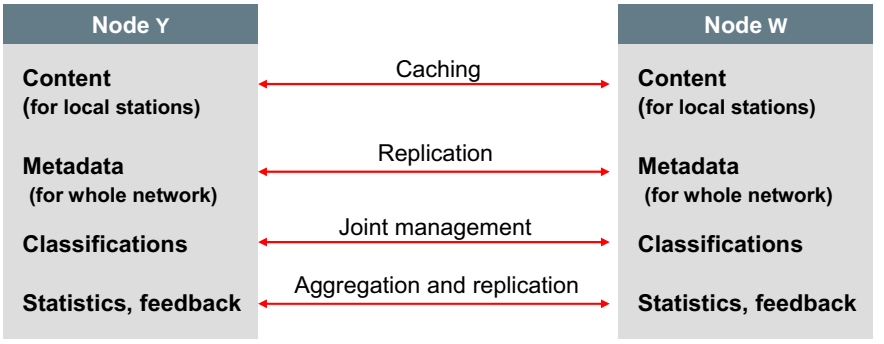

Fig. 5. Cooperation of nodes

many simultaneous audio streams and replication traffic. They also need higher processing power and more storage space than average workstations. Professional maintenance of the software (e.g. periodical backups) is also beneficial in this case. Typically, StreamOnTheFly nodes are provided on a higher granularity than per radio. Nodes offer their services to radios as a national, regional or ad hoc cooperation with or without commercial or other purposes.

4.1 Managing Network Topology

The project is planned to establish some dozens of nodes in Europe. Dynamically changing and dynamically optimized topologies are dispensable with this small number of nodes. On the other hand, a decentralized authentication of peer-to-peer connections is a basic requirement. The project decided on using a static network topology where these static connections provide the way for authentication and a control for expansion as well. Nodes may exist in themselves, without being part of any StreamOnTheFly network. Usually, this is the way how new nodes operate at first times. When a node wants to join a StreamOnTheFly network, personal agreements are made with the operators of the prospective neighbours of that node, and new peer-to-peer communication channels are configured connecting the node to the rest of the network. This method has one more advantage in this situation: personal contacts are established among the operators of nodes, which may enforce some kind of cooperation in the network. Each node has choices for the level of participation in the network: for example, a single connection to the network is less reliable but needs less network traffic, while several connections to various parts in the topology provides more fault-tolerance and content will be faster updated.

If the communication between nodes is configured to be one way, a node gets all content from the network, yet its content remains local. On the contrary, another node might choose to live with its own content exclusively, still it is able to disseminate its content to the whole network. This extreme situation has already been required by one of our partners.

In the example topology of Figure 6 nodes A, C and D form one core and node group G forms another one. These groups are connected with a single connection. Breaking

this connection results in two separate StreamOnTheFly networks. Node E uses the network for the propagation of its own content, thus makes it available at any other node in the network. It does not receive replicated data, so its user interface shows only local content. Node B uses the network to get more content. All network content is available at node B, yet its own content remains hidden for other nodes.

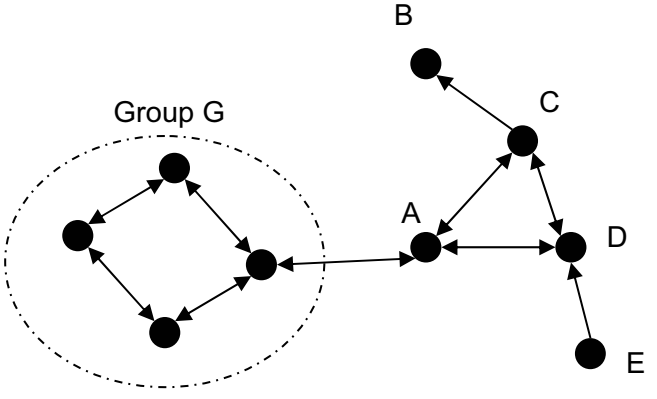

Fig. 6. Example for network topology

Standard operations with the network topology are examined in the following. When a new node joins the network by establishing a new connection, all replicable contents are exchanged through that connection until both ends of that connection have equal metadata database. Then new content from the new node begins to spread over the network. This method works also in the case when two node groups get connected.

Leaving the network is equivalent to the elimination of all connections to other nodes. Upon discontinuing the connections, the metadata of the network could remain available on the node that left the network, but this may lead to data inconsistency over time. Normally, after the elimination of the connection, all the nodes in the local registry are checked if they are reachable. This can be calculated using the table of active connections which is also replicated at all nodes. For unreachable nodes a deletion message is propagated through the network, which results in a cascading deletion of all data belonging to those nodes. This operation also works when two node groups get disconnected.

Finally, optimization for better connectedness can be done manually by configuring more connections between nodes.

It can be seen that node operators govern the network. They decide on accepting new radio stations, on establishing and eliminating network connections. Those unsatisfied with the administration of a node may change to another node.

In case of an editor or a radio station abuses the network, the hierarchical permission system of nodes makes it possible for station managers or node operators to remove unwanted content within their station or node, respectively. What happens when a node abuses the network? Banning a node from the network has to be based on a consensus

of node operators, so that all connections to that node are ceased. The misbehaving node should not be able to negotiate new connections with any other nodes in the network. Personal agreements and exchange of opinions have an important role in the life of a StreamOnTheFly network.

Manual maintenance of peer-to-peer links in the network balances the evident drawbacks of less efficient communication topologies with social management and control, which is in some sense forced by the implementation of the network. Elaboration of this aspect is continued in section 4.5.

4.2 Replication Between Nodes

The software of StreamOnTheFly nodes is built upon an object repository, containing three main types of objects. The first type is local object which does not contain information of global interest, and therefore is not replicated in the network. The other two types of objects are replicated over the network, the difference is that there are simpler objects containing basic data, and complex objects which aggregate simple objects and refer to other complex objects.

Examples for local objects include user data, as it was decided not to replicate the user identities and their preferences in the network. The reasons for not doing so were partly fear from having the network flood with data of one-time users and partly to entitle the nodes to implement their own usage policies. User authentication is implemented as open API and sample wrappers are provided to connect various user management solutions to the node. One example wrapper provides basic user management in itself, while other wrapper delegates user management to the Typo3 portal engine.

Examples for replicated simple objects include a single audio file of a radio show, or a metadata field and value describing the radio show. Complex replicated objects are for example radio shows and series. The object storage is implemented using two parallel storage methods, one in PostgreSQL database and one in plain file system.

Each replicated object has a globally unique identifier in the network. This identifier is composed of the node identifier where the object was created, an object type designator and a local identifier part. Additionally a current home node identifier is known for each object, this is the only node (the owner) which is allowed to modify the content of that object. An object stamp which is incremented at every modification of the object is also stored. Replication is implemented based on these administrative data. When an object is modified it is placed on the replication queue with the list of the current neighbour nodes as addressees. Each neighbour node is contacted periodically and the objects on the replication queue are sent to there. When a node receives new or modified objects from one of its neighbours, it places these objects also on the replication queue, and forwards them to all other neighbours, except the one who sent it. Modifications may arrive to a node from more than one of its neighbours. Based on the stamps it can be decided which object has a newer state than the other, so this kind of redundancy in replication cannot cause problems in data integrity, rather it makes replication safer and helps to detect network problems.

Deleting an object on the node triggers the creation of a new object which contains the fact of deletion. This so called deletion object is replicated over the network, and whichever node it reaches, the replica of the given object is deleted there as well.

All network communication between nodes is based on XML-RPC calls, a very simple way of remote procedure call with XML data as input and output. XML-RPC is an ancestor of SOAP, and migration to SOAP is very easy. This migration is planned in the near future as SOAP is now widely accepted.

4.3 Extensions to Data Replication

The StreamOnTheFly network uses controlled vocabularies (or simple thesauri with other words) for classification purposes. These vocabularies can be used to define the genre of the show (e.g. interview, comedy, news), the roles of creators and contributors (e.g. speaker, producer, composer) and the topic of the show (in this case the controlled vocabulary is in the form of a subject tree). These subject trees have to be identical on all nodes, and translated to all languages used. The replication mechanism was extended to deal with changes in controlled vocabularies as well. In this way, there is no central authority controlling the vocabularies, rather these can be enhanced and translated in a distributed way.

The StreamOnTheFly network collects usage events about the stored content. This is another important addition compared to traditional open archives. While a passive repository disseminates data only from the creator to the user, in our case creators get feedback on how their content was used.

For the implementation of this scenario we selected the collection of statistics, ratings and comments as examples. Statistics are collected on number of page impressions, listens and downloads of each radio show. Users may rate any show simply on a 1-5 scale. Users may give comments or participate in discussions about radio shows on portals. All the information is collected, processed and presented to creators. It is also presented to normal users, but in a limited way.

Calculation of rating information and statistics from usage data is most effective when usage data are concentrated on a single node. Gossip algorithms [18] are also known for calculating the mean (or other expression) of values distributed in a peer-to-peer network, but in this case the large number of aggregate values that needs to be computed in the whole network discourages the use of such algorithms.

It was decided to collect all usage information about a radio show on its home node (where the audio content is stored). The direct messaging protocol was introduced as an extension to the replication mechanism. While normal replication (called data sync) goes through the neighbour nodes, direct messages go straight to the addressed node. Elementary rating and usage data are wrapped into direct messages and sent to the home node for each object. A node receiving direct messages stores incoming elementary data as local objects and puts the subject radio show on its update queue. An internal update process periodically visits objects in the update queue and recalculates their aggregate statistics and rating data. As these recalculated data are stored in replicated objects, they travel on the usual way to all other peers of the network.

This solution ensures that usage data is collected quickly on the home node of the object. Update periodicity can be tuned to the computational capacity of the node. It can happen that overall statistics about a radio show on a remote node in the network is some days older, but the home node, where usually the creator of the show is a user, always contains the most up-to-date usage information.

StreamOnTheFly portals use the content of the whole network and they need access to the open APIs of nodes to obtain the content and metadata. Following earlier design decisions about security and controllable network traffic each portal is connected to a single node. This makes it easier for administrators to protect node servers or portal engines, and a software felt insecure is hardly adapted in this part of the content industry. A portal issues queries and downloads metadata from its so called parent node it is connected to. Complemented with caching, this should not present bigger load for a node than a couple of normal users. Portals are important for collecting consumer feedback. Collected data are sent periodically to the parent nodes, where these usage data are routed towards the home nodes of programs referred in the data as direct messages. The collected information states when

- a portal editor adds or removes a show on the portal,
- a show is visited/downloaded/listened by a user,
- a show is rated by a user,
- a show is commented by a user,
- the portal is visited or updated.

Streaming is provided by the parent node of the portal. Audio files are downloaded directly from the node. This is in harmony with previous design principles and ease the maintenance of the portal engine.

Nodes may choose to cache the audio content. A radio show which is selected to be listened is downloaded from its home node, stored in the cache and streamed to the user in parallel. Subsequent requests are served from the cache until the content gets outdated. Object replication signals the modification of the audio file, and this also triggers the deletion of that content from the cache. Caching is a fair solution for the community of StreamOnTheFly, as instead of being proportional to the number of popular radio shows the streaming load of a node will be proportional to the local users of node. Therefore, smaller nodes are not discouraged from publishing popular content.

4.4 Adding New Content to the Network

Station management software is the last piece in the StreamOnTheFly software suite. Its main purpose is to provide easy ways for uploading new content to nodes, as part of daily work of radio people. It also offers a comfortable way to insert metadata for the radio shows. In order to provide topic or genre for the show, it needs the up-to-date controlled vocabularies from the StreamOnTheFly network. These are periodically downloaded through open APIs of the node.

The station management software is also considered as a demonstrator for the use of station-node communication channels. As such, we wanted to apply very simple communication patterns. XBMF was designed to contain all information stored at a node. It was natural to implement content upload as upload of XBMF files. Nodes have a special directory for incoming new or modified content as XBMF. It depends on the node administrator how the XBMF files actually get into this directory. The sample software solution uses rsync for this purpose, a reliable and secure file transfer utility. Incoming XBMF files are periodically processed by the node software, the radio show is created/updated and audio files are automatically converted to recommended formats.

Some conversion of the metadata is also needed during this process. The station software assigns identifiers to objects within its scope and sends these identifiers along with metadata to the node. The node records these identifiers, but creates its own globally unique identifier for the object. The mapping between show identifiers at the station and identifiers at the node helps to detect if an older version of the radio show has to be modified.

Technically, the StreamOnTheFly network is growing into a heterogeneous component-based network where several methods coexist for the injection and reuse of content (Fig. 7).

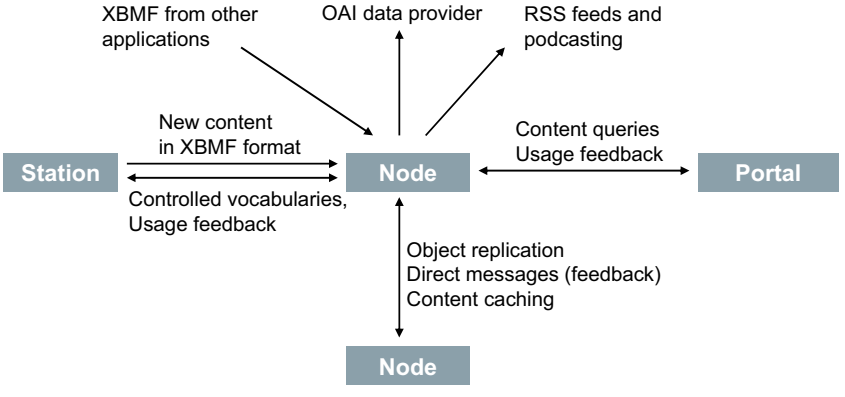

Fig. 7. Overview of communication within a StreamOnTheFly network

RSS feeds of the node not only disseminate the latest content from radios, series or people, but can also provide all important information for a single radio show including metadata, listening URLs and statistics. This latter option is a simple but efficient way of content export into other portal engines. Nodes can also act as OAI-PMH data providers, which is another way of opening up StreamOnTheFly archives.

4.5 Social Considerations

The architecture of the StreamOnTheFly network is based on the principles of self-organization. In system theory, the field of self-organization seeks general rules about the growth and evolution of systemic structure, the forms it might take and finally, methods that predict the future organization that will result from changes made to the underlying components [2]. Self-organization in community life (such as within the community of radio stations in Europe) means that the (social) rules governing the activities within the community evolve in time in order to adapt the community to the new challenges appearing within the sector. Self-organization in social systems can be identified as a continuous loop of emerging new structures and the reflection/response of these new structures in personal actions [11].

In this social meaning of the word, StreamOnTheFly has several properties that enhance the function of self-organization. The network consists of several modifiable and replacable components, and several gateways for communication with the rest of the world. Data moves in and out of the network very easily, providing practical ways for symbiosis with other information or content services.

Distributed data repositories, based on the principles of self-organization, should reflect the community (social) structure and adapt to its evolution. Self-organizing networks encourage users to contribute information. The StreamOnTheFly network helps editors to contribute radio shows. The network can thus be considered as a virtual platform for collaborative, community-based radio content production, distribution, and evaluation. Of equal importance, self-organizing networks aggregate and qualify the most relevant information in (semi-) automated fashion. The results are twofold: users can find quality information with confidence, while administrators and developers need less work to manually sift through large amounts of information. The StreamOnTheFly network provides services for content rating (and other statistical information about the actual use of audio content) and backward propagation of this information to the authors. Self-organizing networks create a community-like environment, and attempt to emphasize and visualize relationships between users thus increase user participation [5]. StreamOnTheFly network users are clustered according to their roles (e.g. station manager, editor, series editor, metadata classification scheme editor, general listener) and system is structured according to their feature requirements. The lack of central control within StreamOnTheFly network reflects the actual rather chaotic community structure of European community radios and the easy way of joining and leaving the network mimics the dynamics of movements in this sector.

The StreamOnTheFly system is a special example that is also based on the general principles of computational self-reflection [8] but the self-reflected system architecture (in which the self description of the actual system is an active part of the system itself) is substituted by a live human community and its social structure. An active (social) community (the community radios in Europe) is thus in the loop, the system is structured according to their (social) relationships. The community mimed system (service) structure emphasizes the openness, thus the system is rather an open infrastructure than a closed and static network, in which users can change and adapt the system itself to their own patterns of usage. Adaptation is possible on both syntactic and semantic level via the portal and the metadata related service components. Dynamics within the community can also be easily reflected. New StreamOnTheFly nodes can join or leave according to the actual changes in the community. A new classification scheme or evolution of the current classification scheme can immediately be embedded in an automatic way.

Summarizing the needs for a self-organizing collaborative media-sharing network, the following principles were used:

- no central role, server or governance, rather provision of an open, evolving infrastructure,
- easy to join and leave, but with the possibility to limit or exclude partners,
- free and unified access and content publishing,
- support for personalization on several levels (e.g. content, user interface, community),
- support for information (e.g. awareness, rating) propagation back to the authors.

5 Use Cases

The StreamOnTheFly network has been in operation since October 2003. The framework was built by using open source and free software components and the code base was published as open source on SourceForge. Since that time StreamOnTheFly has proved its flexibility and extensibility in several cases. On the other hand, the take-up of new technology by the radio community is rather slow. The core of the network (excluding experimental and stand-alone nodes) is accessible in 4 languages, and contains more than 500 hours of audio content from 20 radio stations[1].

The first real use case of the system is at the Technical Museum in Vienna. Medien.welten is the permanent exhibition on the development of media at the museum. To illustrate the convergence of radio and computers, StreamOnTheFly is used to record the complete Ö1 Programme, segment and categorize it automatically and publish it as a 30 day archive in the exhibition. Due to copyright issues this node is not available outside the museum.

The Volkswirtschaftliche Gesellschaft uses StreamOnTheFly technology to run a webradio station[2]. The project started late summer 2004, and will include up to 10 involved schools in the lower Austrian region.

For the third time, Christian Berger's team had broadcasted live from the Frankfurt Book Fair. In 2004, the whole webcast and archive was realized with StreamOnTheFly technology. The Frankfurt Book Fair is Europe's largest come-together of the european literature scene. Literadio is a joint project of German and Austrian community radios, and covers lectures, round tables, speeches and interviews in a daily webcast from 10am to 17pm live from the studio on Literadio's stand. The shows were stored at a node, and special integration with other portals was done using the RSS feed of the node. Another similar project based on StreamOnTheFly was the live radio broadcast at the 2002 IST conference in Copenhagen.

RSS 2.0 gave rise to a new phenomenon called podcasting. With a special software one can download not only the news like in traditional RSS, but also audio files for radio shows. Users select series they like and download them directly to their iPod or other MP3 listening device. After that they can listen their favourite programs anytime anywhere. With a small software update StreamOnTheFly is now capable of serving content for podcasters.

Another interesting approach to use StreamOnTheFly was developed to proof-of-concept stage at the Vorarlberg Universtity of Applied Sciences, where the node was used as a backend for a music publication site. In this case, the user interface was completely developed in Typo3, a php-based Content Management System. The interaction with the node was designed completely by using the XML-RPC API and the RSS distribution methods, thus validating the approach used in the design.

Radioswap.net uses this platform for content collection for Belgian community radio stations. At the moment their node cannot join the network because of legal and copyright issues. They are also working on a desktop tool facilitating easy upload of radio shows to StreamOnTheFly nodes.

[1] http://radio.sztaki.hu

[2] http://www.schulradio.at

Trials were implemented also to use StreamOnTheFly as a commercial audio service, an e-learning platform, and to add support for handling video content. The project officially finished in June 2004, but voluntary work keeps going on implementing various extensions.

6 Related Work

The StreamOnTheFly network builds upon the successful principles of the OAI-PMH protocol, although it applies its own special protocol for communication because of additional features. StreamOnTheFly replicates metadata in a peer-to-peer network in a push manner. As a result, the service provider functionality is uniformly available at all nodes in the network. Furthermore, the communication is extended with collection of feedback and statistics about the access and use of content.

There are only a few references to extend OAI-PMH into the peer-to-peer direction. OAI-P2P is a suggestion to organize a peer-to-peer network from nodes which are both OAI data providers and service providers [4]. This approach is based on Edutella, a network for storage, query and exchange of metadata [20]. In these networks queries are automatically routed to nodes which are able to provide answers, and the query is executed in the usual peer-to-peer way, parallelly on several nodes. In StreamOnTheFly, such solution was unacceptable because of two reasons: distributed searching may cause incomplete results and thus information loss (if some of the nodes are temporarily unavailable), and the response time of the query may be fluctuating. Distributed search is also a setback when experimenting with novel search methods and interfaces, as it was the case in our project as well.

Lagoze and Van de Sompel are considering peer-to-peer concepts in the OAI world [16]. They realize that OAI harvesting chains have increasing complexity, and thus they are thinking of peer-to-peer networks behaving as OAI data providers as an evolutional next step. Ding and Solvberg suggest a framework for harvesting metadata with heterogeneous schemes in a super-peer based P2P network [7].

Cooperation in archival and program exchange is an old demand of the radio community, and there are some services supporting this, for example the Open Meta Archive[3] and the Cultural Broadcasting Archive[4]. These services are all centralized and lacking the full support of the radio "supply chain" from production to listening.

Personalisation of radio or television program is getting more and more interest nowadays. An art project and experiment called Frequency Clock[5] provided a broadcasted TV channel where the program schedule could be modified during broadcast through a web server. Last.fm is a personalized online radio station where users create their profiles by selecting songs to listen, and then they get recommendations of new songs. The idea of flycasting [10] is to adapt the playlist according to the musical tastes of the people currently listening to an online radio station. StreamOnTheFly creates a unified platform for these ideas of personalisation, including rating, collaborative filtering, personalized radio program, personalized queries and portals.

[3] http://oma.sourceforge.net
[4] http://cba.fro.at
[5] http://www.frequencyclock.net

7 Conclusions

StreamOnTheFly is an operating example for a distributed network of heterogeneous digital library services. The core of the network combines peer-to-peer networking and open archive principles, while the rest of the services are realized as separate network components communicating through open APIs.

The network fulfills the initial goal to establish a common format and software support for the archival, exchange and reuse of radio programs. It also proceeds towards experimentation with latest networking paradigms and social self-organization. Our experience shows that the software components are easily adapted by various user groups and new emerging phenomena, such as RSS feeds or podcasting, are simple to integrate with the network.

Instead of automatically optimized network topologies, this network chooses the manual maintenance of peer-to-peer connections, which solves the problem of node authentication and fosters social self-organization of the network. We presented various tasks, problems and their solutions in such a network. The number of trials and applications supports the design and implementation of StreamOnTheFly. In the future, we plan to examine and further support emerging synergies catalysed by this service platform.

With the upcoming multimedia services via broadband and G3, audio archives get an additional boost. As seen in the US, streaming audio in general became an important aspect of daily business [21]. It is time to support this new technical potential with recent service infrastructure.

Acknowledgements. Authors would like to thank to the participants of the project for their support and help in the design and development of StreamOnTheFly software: Roland Alton-Scheidl, Thomas Hassan and Alexey Kulikov from Public Voice Lab, Jaromil, an independent audio and streaming expert, Thomas Thurner, Wolfgang Fuchs and Roland Jankowski from Team Teichenberg, Tamás Kézdi and Gergő Kiss from MTA SZTAKI DSD. This work was partially supported by the EU under contract IST-2001-32226.

References

1. Dublin core metadata initiative. `http://dublincore.org/`.
2. Self-organizing systems (sos) frequently asked questions. `http://www.calresco.org/sos/sosfaq.htm`.
3. Shared online media archive (soma) metadata element set version 1.0, September 2002. `http://soma-dev.sourceforge.net/`.
4. Benjamin Ahlborn, Wolfgang Nejdl, and Wolf Siberski. Oai-p2p: A peer-to-peer network for open archives. In *31st International Conference on Parallel Processing Workshops (ICPP 2002 Workshops)*, pages 462–468. IEEE Computer Society, 2002.
5. Hisham Alam. Self-organizing sites. `http://www.newarchitectmag.com/archives/2002/01/alam/`.
6. AAF Association. Advanced authoring format (aaf) object specification v1.1, 2005. `http://www.aafassociation.org/`.

7. Hao Ding and Ingeborg Solvberg. Metadata harvesting framework in p2p-based digital libraries. In *International Conference on Dublin Core and Metadata Applications (DC-2004), 11-14 Oct 2004, Shanghai, China*, Oct 2004.
8. Paul Dourish. Developing a reflective model of collaborative systems. *ACM Transactions on Computer-Human Interaction*, 2(1):40–63, 1995.
9. European Broadcasting Union (EBU). Core metadata set for radio archives (tech3293), 2001. http://www.ebu.ch/CMSimages/en/tec_doc_t3293_tcm6-10494.pdf.
10. James C. French and David B. Hauver. Flycasting: On the fly broadcasting. In *DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries*, 2001.
11. Christian Fuchs. Science as a self-organizing meta-information system, 2004. http://ssrn.com/abstract=504244.
12. MPEG Group. Mpeg-7 (multimedia content description interface). http://www.chiariglione.org/mpeg/index.htm.
13. Rachel Heery, Pete Johnston, Csaba Fülöp, and András Micsik. Metadata schema registries in the partially semantic web: the cores experience. In *Dublin Core Conference, 2003, Seattle, Washington USA*, Oct 2003.
14. László Kovács, András Micsik, Balázs Pataki, and István Zsámboki. Aqua (advanced query user interface architecture). In José Luis Borbinha and Thomas Baker, editors, *Research and Advanced Technology for Digital Libraries, 4th European Conference, ECDL 2000*, volume 1923 of *Lecture Notes in Computer Science*, pages 372–375. Springer, 2000.
15. Carl Lagoze and Herbert Van de Sompel. The open archives initiative: building a low-barrier interoperability framework. In *ACM/IEEE Joint Conference on Digital Libraries, JCDL 2001, Roanoke, Virginia, USA*, pages 54–62. ACM, 2001.
16. Carl Lagoze and Herbert Van de Sompel. The oai and oai-pmh: where to go from here?, February 2004. Presentation at the CERN Workshop Series on Innovations in Scholarly Communication: Implementing the benefits of OAI (OAI3). http://agenda.cern.ch/fullAgenda.php?ida=a035925.
17. András Micsik, Thomas Hassan, and László Kovács. Distributed archive and web services for radio stations. In *12th International World Wide Web Conference, 20-24 May 2003, Budapest, Hungary*, May 2003.
18. Alberto Montresor, Márk Jelasity, and Özalp Babaoglu. Robust aggregation protocols for large-scale overlay networks. In *2004 International Conference on Dependable Systems and Networks (DSN 2004)*, pages 19–28. IEEE Computer Society, 2004.
19. Oliver Morgan. Metadata systems architecture. In *International Broadcasting Convention (IBC 2002)*, September 2002. http://www.broadcastpapers.com/ibc2002/ibc2002.htm.
20. Wolfgang Nejdl, Boris Wolf, Changtao Qu, Stefan Decker, Michael Sintek, Ambjörn Naeve, Mikael Nilsson, Matthias Palmér, and Tore Risch. Edutella: a p2p networking infrastructure based on rdf. In *11th International World Wide Web Conference*, pages 604–615, 2002.
21. Arbitron/Edison Media Research. The value of internet broadcast advertising. *Internet and Multimedia*, 12, 2004.