

Micro grids for scalable media computing and intelligence on distributed scenarios

Pierfrancesco Bellini, Ivan Bruno, Daniele Cenni, Paolo Nesi

Distributed Systems and Internet Technology Lab, Department of Systems and Informatics
University of Florence, Italy, <http://www.disit.dsi.unifi.it/>, nesi@dsi.unifi.it
tel: 0039-0554796523, fax: 0039-055-4796363

Abstract

The micro grid technology is playing a central role in the semantic evolution of small and medium size services. These flexible applications allow enforcing new functionalities and intelligence into complex and integrated solutions by supporting automated content processing and back office management, controlling content delivering networks, providing services via active and proactive applications on client devices. To this end, they have to provide high flexibility and scalability and a range of features which in the past were unaffordable for small and medium service portal providers, while being available only on large integrated platforms. A solution can be to enforce flexibility and media computing capabilities into micro grid by using a media grid computing language for formalizing processes to produce, post-produce, license and deliver content while meeting a range of scenarios' requirements. The usage of a media language improves the level of automation in services and applications. This article presents the main requirements and architectures of micro grids for media and semantic computing, and for media grid languages. Semantic computing primitives may be also enforced into final user tools to provide local semantic search, recommendations: a real personal assistant. The experience reported herein has been gained in several years of work in the sector.

Introduction

The recent challenges on media services have led software engineers to identify and integrate multiple technologies in a scalable manner. Examples are large scale services such as social networks, content delivering networks, archives, encyclopedia, e-commerce portals and other service networks. In most cases, large-scale media services are managed by large companies and/or institutions, thus allowing a number of simultaneous accesses and millions of delivered content items. Recently, on the spur of final users via high-speed networks, also many small and medium companies/entities are moving towards setting up similar services. This trend is also supported by the decrement of computational and storage costs, making affordable the exploitation of semantic computing and grid-based multimedia computing tools and solutions. Semantic computing includes a set of technologies related to descriptors/profiles; among them: algorithms to extract descriptors, knowledge modeling, semantic search, inferential engines, taking decision systems, and solutions for computing similarities and recommendations. On such grounds, the augmentation of new and more intelligent services and their scaling are becoming accessible for a larger number of actors.

The typical small-scale applications, where new technologies of scalable semantic computing are pervading, are public digital archives, best practice networks, service/content distribution, thematic social networks, e-commerce portals, distributed monitoring systems, content repurposing processes, etc. They are

characterized by both a limited number of users and of content items with respect to the global scale businesses. On the other hand, the insertion of semantic computing can enforce new functionalities and business models. The services can become so attractive to compete with huge service providers. Most of them share common requirements for: media adaptation, repurposing, indexing; workflow management; semantic computing; content management, distribution, quality of services; intelligence on client devices; integration with delivering networks, etc. Therefore, the flexible insertion of media and semantic computing in an automated back office is becoming an urgent need to curb costs for media services.

Most small and medium media services need to be continuously updated to follow current trends, thus increasing, changing and providing new functionalities/services for user communities. Examples on this kind of service reshaping/tuning can be:

- production of additional kinds of recommendations (advertising, content to users, user to users, group to users, content to groups, etc.),
- insertion of additional reasoning on metadata, use data and/or profiles,
- changes in similarity distances and clustering algorithms,
- production of content on demand for new devices,
- insertion of full text indexing in several languages, re-indexing of the database, etc.
- gradual insertion of semantic computing in running service infrastructures,
- insertion and/or tuning of user generated content processing workflow,
- changes in business and transaction models, licensing and trusting,
- integration of additional distributive channels, and thus changed into the content delivering network,
- review of data-model to cope with additional metadata for additional services (insertion of GPS coordinates, insertion of barcode readers, insertion of RFID readers, ..), etc.,
- addition of more detailed analysis on user behavior.

The areas of production and distribution are frequently kept separate in traditional content factories, whereas they have to be integrated in services including processing of User Generated Content, UGC; allowing content enrichment; managing Content Delivering Network, CDN; performing content repurposing/adaptation and delivering for mobile users [InsightReport].

The exploitation of the above-mentioned functionalities in single processes in integrated environments is mandatory when multichannel content production and delivering have to be efficient and harmonized. In such cases, programming and coding are needed at system, business logic and process levels. To this end, old services have to be updated to cope with dynamic changes and allocations, or new services have to be designed; flexible procedures/processes have to be dynamically allocated on computational resources. In most cases, technical solutions are based on setting up custom processes on dedicated computer systems during restructuring and integration. This means, creating solutions that are difficult to be reconfigured to cope with further changes whenever requested. The business logic may be transferred into the processing units in a much faster manner than moving the huge data sized content.

Among technologies supporting flexible management of the media-based back office management we can see the parallel distributed processing, software agents, grid [Foster2002], Service Oriented Architectures (SOAs), cloud computing, semantic computing technologies, and grid processing languages.

The paper is structured as follows. As to macro and micro grid for media computing the most relevant definitions are given in the next section, together with fundamental requirements. The general architecture

of the proposed solution is presented in section “*Micro Grid Architecture for Media Computing*”. In the same section, the state of the art is analyzed considering main functionalities. The proposed solution is based on a media grid computing language which is presented in section “*Media Grid Computing Language*”. Finally, effective scenarios are reported in section “*Micro Grid Usage Scenarios*”, always in comparison with the functionalities identified for analyzing the state of the art. Such comparison highlights the features exploited by five different scenarios coming from: Digichannel, BBC Ax4Home, Mobile Medicine, Variazioni and ANSC Musa.

Requirements for Macro and Micro Grid for Media Computing

Grid based solutions can be exploited for harmonizing distributed services via internet, as well as via local back offices. They can cope with dynamic allocations of services, exploiting flexibility to reduce costs and increase efficiency. In the [Roure2005], the idea of grid was mainly at large and global scales: “*Semantic Grid vision is to achieve a high degree of easy-to-use and seamless automation to facilitate flexible collaborations and computations on a global scale, by means of machine-processable knowledge both on and in the Grid*”. Semantic grids are solutions to cope with user communities, parallel processing, taking decisions and reasoning on descriptors and content usage. Geographical or global grid solutions, typically based on OGSA (Open Grid Service Architecture), such as Globus, are too complex and expensive to be used as flexible micro grids. The grid technologies can be the underlying solutions to manage automated content management and distributed resources such as: industrial computers, dedicated servers, storage systems, playout servers, communication channels, etc. In [Volckaert2008], concepts of macro and micro grid have been recalled in the context of a media grid architecture without addressing the micro grid language for process formalization. Media and multimedia grids try to provide solutions for a number of applications (see the Box on Media Grid). *Micro grids* have to be able to manage local resources such as computational power, storage and functionalities at service of front-end services and larger applications or global/macro grids. Micro grids are the essential operative arms of large-scale grid solutions (see Figure 1), and typically manage the local computational power and receive instructions from the higher-level grid services negotiating the service by means of some Service Level Agreement, SLA, [Leff2003]. Grid solutions may be combined with cloud computing and virtualization technologies to optimize the resource consumption.

In this paper, as to *media grid* we mean a grid architecture where processes may govern integrated functionalities which can be classified into services for content processing, delivering and storage; services for semantic computing. Media grids may be used as dedicated engines to enforce new features and scalability into present small and medium size solutions designed to serve local applications/portals and services. Their flexible exploitation of computational power and micro functionalities allows to cope with the computational needs resulting from the insertion of new functionalities at fair costs. For example, the whole re-indexing of a large database, a global clustering, the global re-classification, a massive video transcoding, a massive calculation of similarity distances among descriptors; all those may be very computationally intensive. On the contrary, the day by day processes can be much lighter since incrementally applied only to daily coming content and information, and thus, the grid may be scaled down dedicating resources to other services. The solution has to be scalable, it has to work in a reasonable manner in terms of reaction and processing time, even when the number of accesses (e.g., requests performed per day, per minute), and the number of items are in the range of millions.

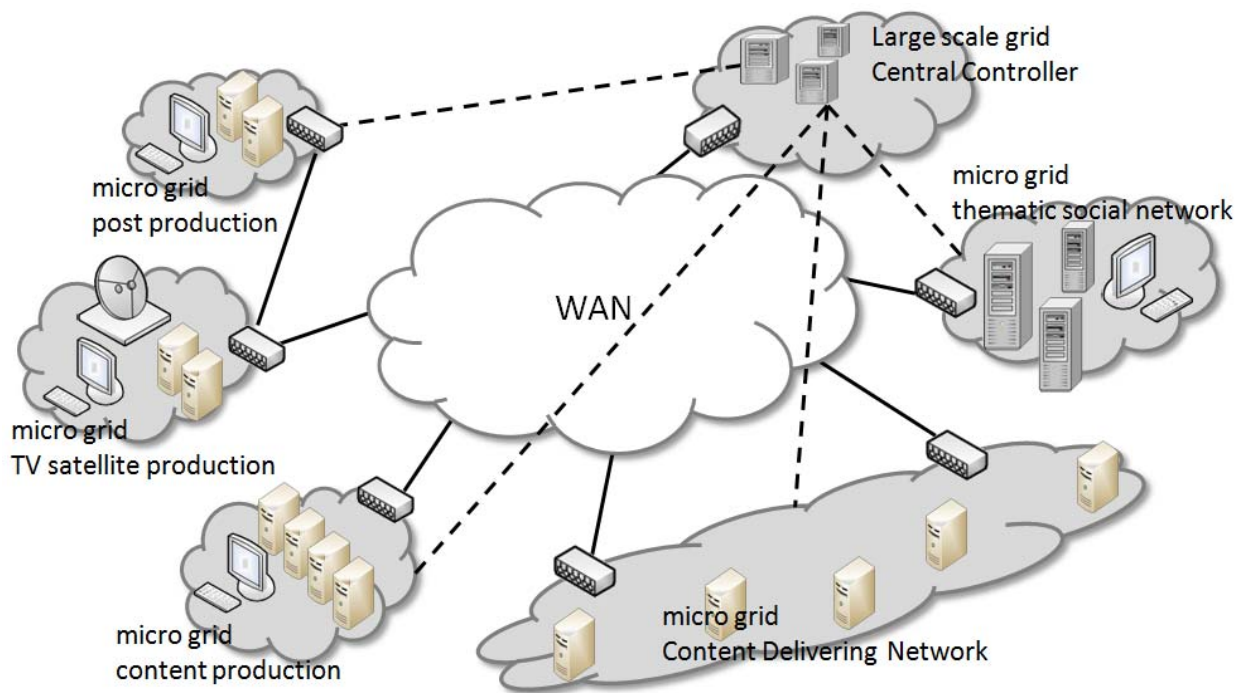


Figure 1 – A Large-scale grid and some related micro grids.
Dashed lines represent logical controls dependencies dynamically established.

Box on Media Grid

MmGrid [Basu2003] has been designed to support interactive applications with graphics, rendering, streaming, and tele-immersion. **MediaGrid.org** [Walsh2005] consortium proposed a grid solution to address computational power to support 3D and virtual reality rendering and distribution. **MediaGrid** [Huang2007] is a P2P solution to share videos for IPTV, monitoring the network conditions and tunes the services maximizing network capacity. It is a solution to set up a CDNs based on P2P progressive streaming. In [Ishaq2009], the grid is used to set up a network of nodes organizing services via a classical DHT (Dynamic Hash Table) P2P models. **GridCast** [Harmer2005] is a service-oriented architecture for broadcasting media via IP. It presents some content management capabilities including those related to scheduling broadcasting and subsequently controlling playout servers for broadcasting. Most of the high-level activities may be passed on a workflow management system, which also plays the role of integrating other tools via programming. The GridCast solution is tuned to cope with broadcasters' content models such as MXF, AAF, etc. GridCast has been developed on Globus Toolkit, and has some capability for setting up a geographical infrastructures rather than for local micro grid architectures on cloud computing. **Omneon MediaGrid** (<http://www.omneon.com/mediagrid>) proposes a grid powered scalable content storage.

In order to curb costs, *micro-grids infrastructures for media* have to provide the needed flexibility to integrate a large set of functionalities, tools and technologies. Moreover, they have to provide basic facilities and support for the following main features.

- **description of grid processes** – Each single grid process has to provide a formal description of the resources needed to execute successfully the process and its scheduling conditions. For example, the list of needed functionalities and tools, CPU capabilities, HD space, networking, execution duration, description of scheduling parameters (e.g., period, deadline), etc.

- **discovering and describing resources** – The grid scheduler has to be able to discover resources via different protocols, and they have to provide their descriptors (e.g., computational and/or data storages). Resources may be located in industrial computers, virtual clusters, dedicated servers, SAN/NAS, as well as portions of desktops.
- **allocation/activation of processes** on physical and/or virtual resources -- The allocations and scheduling of activities on the grid have to provide buffering support to collect execution requests, and reservations. Processes may be programmed as periodic or sporadic to be scheduled by the grid according to the resources available. Sporadic processes are typically activated via web services or other calls. The grid processes may be allocated on other micro grids; thus supporting hierarchical organization of processing.
- **fault tolerant solutions and services** -- Safeness and reliability have to cope with errors and exceptions, managing them at all levels, e.g., on grid nodes, in the network, in the scheduler process execution. The reliability also implies a continuous control by verifying status and liveness of grid resources since they may change profile/capability over time.
- **security and trust** -- The micro grid has to guarantee the security of both processes and data on the grid infrastructure, as well as of data distributed and communicated in/out of the grid. Security aspects have also to be addressed at process level since in most cases the single process may cope with data protection, and with the processing of protected content, such as in Digital Rights Management, DRM, models [Rosenblatt2003].

To satisfy the above described requirements within the micro grid infrastructures lays the foundations for creating a media computing micro grid on which the new functionalities can be natively enforced as described in the next section.

Micro Grid Architecture for Media Computing

In AXMEDIS (automating production of cross media content for multi-channel distribution) [Bellini2006], [AXMEDIS], an infrastructure for micro media grid has been designed and developed in the context of a large international integrated project of the European Commission. The framework has been designed with the aim of reducing costs by increasing efficiency and flexibility for content production, protection, and management for multichannel CDNs, via DVB, P2P and IP distributions towards PC, STB and Mobiles [Bellini2007]. To this end, a new level of flexibility has been achieved by defining and implementing an architecture based on a media grid computing language to formalize grid processing rules which are interpreted at run time exploiting a large number of functionalities [InsightReport].

The AXMEDIS architecture for media micro grid presents five main areas (see Figure 2): resources, Grid Nodes pool, Grid Scheduler, Request Manager and the Integrated Development Environment. Resources are: hosts, storages, connections, databases, distribution channels, etc., and Grid Nodes as well.

The **Grid Scheduler** discovers resources on the grid and nodes, and according to the selected optimization algorithm (e.g., taboo search, genetic algorithm, distributed deadline monotonic,..) it finds the best matching to allocate processes on Grid Nodes. The Scheduler has the duty to control the Grid Nodes, to collect errors and set up fault tolerant invocations, to set up strategies of recovering from failure (e.g., reactivation of a failed rule), and to reconnect nodes that for any reasons lost the connection, to collect updates on their profiles. The Scheduler may be the single point of failure of the micro grid and thus failover/hot-spare solutions to set up a chain of multiple Schedulers are needed.

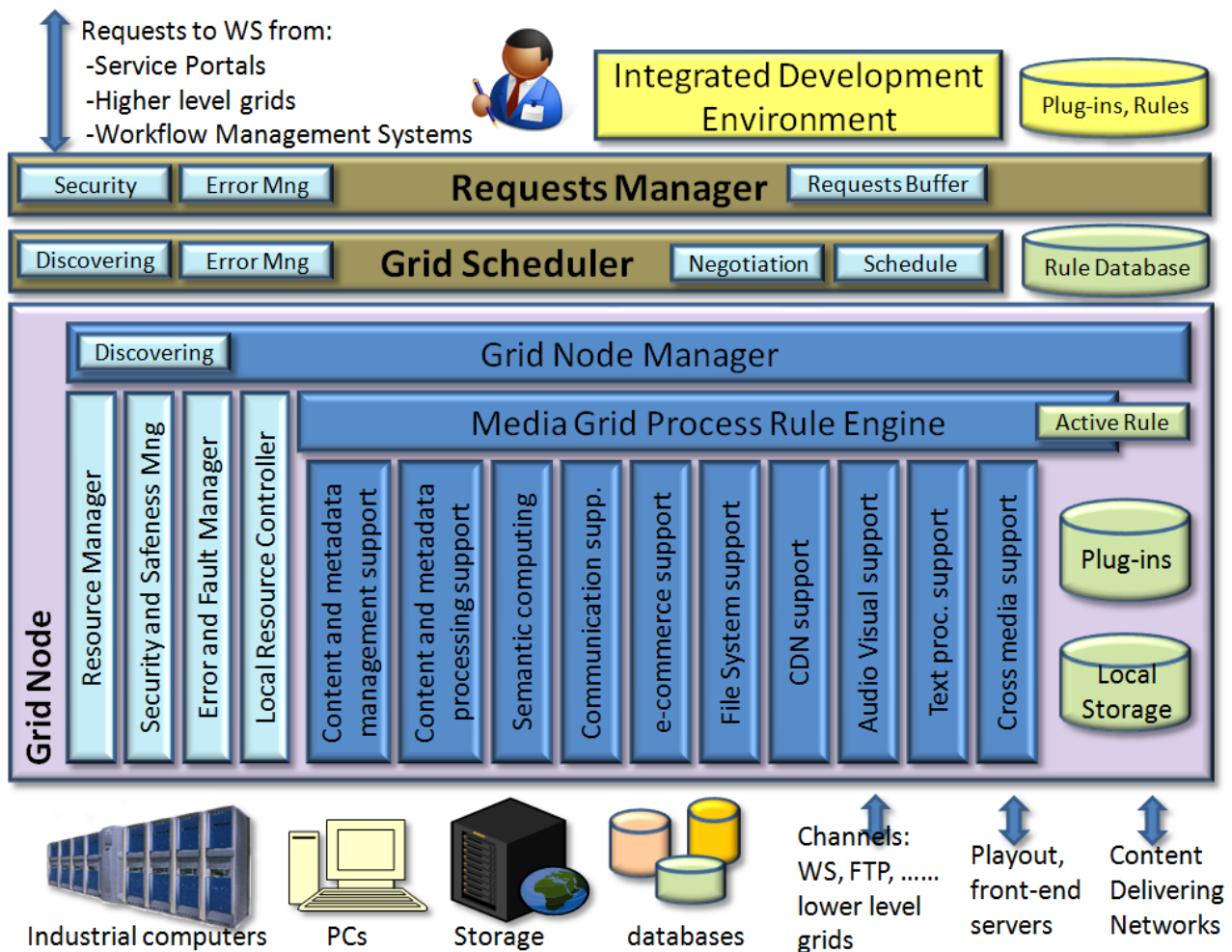


Figure 2 – Micro Grid architecture, five main areas: resources, Grid Node, Grid Scheduler, Request Manager and Integrated Development Environment.

The **Grid Node** may be allocated (put in execution) on general purpose or dedicated industrial computers (virtual or physical computers). The Grid Nodes are the point of access to computational, communication, and/or storage resources. The Grid Node is the minimal grid service element controlled by the Scheduler. The Grid Node has to be able to manage errors and faults in order to recover and/or report back the general control to the Scheduler. The node has to respond to the Scheduler by disclosing processes and providing general status information regarding its profiles and status. In the AXMEDIS solution, a Grid Node process/Rule is coded in a **Media Grid Computing Language**. Thus, a large set of functionalities is made accessible via natively integrated modules and/or plug-ins, which are dynamic libraries with associated XML manifests for the dynamic loading of functional parameters and to support help in the integrated development environment and debugger. According to the architecture reported in Figure 2, Rules are executed on the node by the Media Grid Process Rule Engine which provides support for the exploitation of the local machine resource usage (e.g., CPU clock) according to the planned day by day profile. This capability is enforced via the Local Resource Controller, which allows the exploitation of office desktop computers as grid nodes. The Local Resource Controller allows any controlling of CPU exploitation also when the execution is passed to plug-in code.

The **Request Manager** is the front-end of the micro grid and has the duty to control the authorization to the service usage/access. Requests may be filtered by a business process to sign a SLA with the grid manager and organization. At technical level, the Request Manager receives and collects the requests (in a

Request Buffer) in order to activate processing Rules on the micro grid according to their descriptors. They may arrive from several sources such as: front-end service portals, higher-level grids, workflow management systems, programmers on the integrated development environments, from the same micro-grid, etc. [Bellini2006]. Thus, the micro grid processes have the capability of activating rules/processes on the micro grid sending invocations to the Grid Scheduler; this feature can be very important to implement recovery from failure procedures or for certain parallel algorithms where the number of processes is not deterministic. In some cases, the requests to the grid may consist in: providing a new version of a processing rule/process, sending updated versions of plug ins/tools, making query on the **Rule Database** (containing the pool of processes that may be activated), posting a rule on the rule database, verification on the available resources and node descriptors, etc.

The **Integrated Development Environment** allows the formalization and validation of Rules/processes to be allocated on Grid Nodes. In many grids, process executions are totally delegated to executable programs developed in native code for the node operating system. That model implies a reduced flexibility and control, since the business logic of the process is enforced into external routines. In the AXMEDIS micro grid, processes are typically directly formalized as process Rules formalized in the Media Computing Language, which is a JavaScript-based language extended with thousands of functionalities to cope with content, metadata, semantic computing, databases, distribution channels, CDN, etc. [Bellini2006]. A grid Rule is coded in XML and contains: a rule description (e.g., resources need, schedule, parameters, metadata) and the functional parts. The **Integrated Development Environment** provides an editor and debugger for Rule prototyping and validation. The solution also provides a visual editor where procedures and processes are created by combining visual elements [Bellini2009].

	AXMEDIS	MMGRID	MediaGrid	GridCast	MediaGrid.org	Omneon MediaGrid
Content Management: storage, UGC, ..	X		(x)	(x)	X	X
Content computing/processing: adaptation, processing conversion, cross media content packaging, ..	X		(x)	(x)	(x)	X
Content Delivery Network Management	X	X	X	X	X	
Metadata enrichment and reasoning	X					
Content Protection Management (CAS/DRM)	X					
Content Indexing and Querying, knowledge base	X			X		X
Semantic Computing Reasoning on user profiling, content descriptors, recommendations	X					
User Interaction Support, rendering, collaboration	X	X			X	
Client player as grid nodes for intelligent content	X					
Global and/or Local grid	L/(G)	G	G	G	G/L	L

Table 1 – Media grids and coverage areas; (x) means partial support, G stands for global, L for local, and X for full support. All the mentioned solutions support the distribution of processes. Most of them may support some level of semantic computing via external services and tools.

Table 1 reports a comparison of the above mentioned media grids and the proposed AXMEDIS solution in terms of main functionalities. The main advantages of AXMEDIS consist in the usage of a flexible architecture based on a Media Grid Language for the formalization of Rules/processes into the micro grid nodes for coding the logic. The solution allows to create dynamically media grid processes to get access to functionalities coming from different areas: content and metadata ingestion, indexing, management, mapping, processing; semantic computing: taking decision engines, inferential engine, descriptor processing, and solutions for computing similarities and recommendations; communication capabilities; database management, query activation; content delivering, managing CDN; workflow integration; extraction of descriptors from media; multilingual translations.

At level of Grid Nodes these functionalities have to be accessible as native API of the Media Grid Language and/or as plugins. The possibility of using interpreted Rules allows to integrate media computing functionalities and to deliver dynamically processes for the grid nodes without re-compiling the process code, while maintaining the full control and monitoring of the logic at a higher level, even controlling integrated plug ins. The activation of external executable tools and batches are an additional possibility as compatibility with other grid approaches. Batch solutions are quite widespread, while they have some limitations due to the limited control on the code activated by batch. In these cases, the called executables have to be recompiled to inject a deeper control in the process which is put under execution into the node, see for example: Condor or Globus grid. This last solution is very expensive since each new functionality has to be accessible at code level and has to be recompiled with some grid library. It is not always an easy going process, and therefore very expensive to enforce new functionalities. On the contrary, AXMEDIS Grid Nodes may execute new and non-planned procedures without stopping the service. It supports changes at run time, such as: changing business logic, adding/removing computational resources, storages, etc.; changing the process code from one execution time to the next in periodic processes; changing binary library and tools.

Media Grid Computing Language

The AXMEDIS micro grid solution provides an open infrastructure compliant with the architecture reported in Figure 2. In AXMEDIS, additional features and/or installations can be easily deployed at run time via plug-ins and tools and made accessible at level of Media Grid Computing Language. Language functionalities can be grouped into a set of categories, which are reported in the following to highlight what is needed into a media grid:

- **enhanced media types/objects** such as: connection (FTP, ODBC, FTP, mail, etc..), cross media (MPEG-21, zip, NewsML, ePub, MXF, mpeg-4,..), CDN (broadcast, P2P, IP, ..), format (HTML, SMIL,..), profiles and descriptors (CCPP, MPEG-21 DIA), license (OMA, MPEG-21 REL,..), metadata (DC, TvAnyTime, XML, ..), etc.
- **content and metadata management.** It implies the access and retrieval from any kind of databases and storage area networks, as well as the access and processing of XML files and schemas. Moreover, metadata access implies also reaching them via communication channels and harvesting via archive protocols such as OAI PMH, Z39.50.
- **content processing** for text, audio, video, image, multimedia, etc., via specific procedures and tools, such as adaptation and transcoding (e.g., from WMV to MPEG, resize, rescale, resample, ..); file processing and interpretation (XML, HTML, RDF, RDFS, etc.); formatting/layouting of presentation layer models (e.g., HTML/CSS, SMIL, MPEG-4); cross media format conversions (among them: MXF, MPEG-21, NewsML, SMIL, HTML/CSS, ePub, XML,..); fingerprinting and watermarking estimation and

recognition/extraction. Among the content processing capabilities: the possibilities of processing XML via DOM and XSLT, the application of regular expressions, the modeling of complex data structures via associative arrays, etc. More complex algorithms may be realized as native code and called via plug-ins.

- **semantic computing** is grounded on the production and exploitation of semantic information. Semantic descriptors may be directly accessible as collected information: user profiles and preferences, network capabilities, device capabilities, context descriptor, use data, advertising, resources, .. The access to knowledge may be obtained by exploiting tradition as well as RDF stores. Semantic extraction implies the application of algorithms to produce descriptors -- e.g., video and image analysis, audio analysis, doc analysis, for multilingual text processing for indexing (taking into account semantics, ontologies, dictionaries and metadata) and obtaining integrated indexes where one may perform general and semantic queries in any language and form. Semantic processing is used to perform reasoning on knowledge to: take decisions on content layout formatting, provide recommendations, perform media file transcoding (i.e., via standards such as MPEG-DIA, CCPP), offer/propose advertising or suggestions, provide support to intelligent fuzzy search, reasoning on DRM licenses, etc. On the grounds of semantic computing, a number of low-level algorithms and tools are needed. For example, to estimate similarity distances among heterogeneous symbolic descriptors, so as to use them in direct matching and in clustering (e.g., K-Means, K-Medoids, hierarchical clustering); and to take decisions such as logic and inferential engines based on rules, Horn clauses, and First Order Logic, FOL. More powerful, inferential engines are exploited via RDF stores, for example by using SESAME and OWLIM, or via direct access of Jena via scripting.
- **communication** via several different channels (FTP/SFTP, HTTP/HTTPS, WSDL, ODBC, WebDav, etc.) and for content and information movement and distribution on file systems, databases and SAN/NAS. The communication may lead to perform the integration with portals, by migrating content, users and user profiles, automated publication for dissemination and promotion; and also by accessing the content of those portals for repurposing and/or serving the users propagating queries in those large networks.
- **content protection:** functionalities to protect content and license users, which are needed to enable e-commerce and IPR management. Content protection may be implemented by content packaging and encryption (e.g., according to some standards such as MPEG-21, DVB, OMA). In order to set up effective e-commerce solutions based on DRM and/or CAS (Conditional Access Systems), many other aspects have to be managed. For example, the user registration with certification production, the unique ID assignment, the verification and validation of devices and players, the digital signature of content, etc. Once the protected content is produced on the micro grid, the content can be delivered to final users, who can be authorized by means of a licensing mechanism. In some cases, the licensing may involve massive production of codes and licenses for each content and user. Micro grid may also be used to manage the customer relationships in the e-commerce applications and portals for billing, advertising, reporting, profiling services, etc. Among these services, the user profiling for supporting the on-demand business with personalized advertising is computationally intensive.
- **managing protected content.** Grid Nodes may be used to manage protected content, for example when it comes from content providers and it has to be repurposed and/or redistributed (for example in MPEG-21). In this case, the node has to be authorized/licensed and the technical platform certified to guarantee that the rights are correctly enforced according to their semantics. For example, a video adaptation can be performed, only if the grid itself is authorized via a corresponding license, the same for other rights such as those for extraction, packaging, synchronization, etc.
- **content distribution.** Media and content distribution is realized by preparing the selected content and data from the back office to the distribution servers and channels. At present, the distribution is

typically multichannel and thus a media factory may have to provide content towards a plethora of front-ends: playout streamers, web servers, mobile servers, P2P networks, social networks, satellite carousel, DVB Mux, SMS/MMS services, etc. For each of them, specific formats and information packages have to be produced and updated. The media grid is needed in the management of the distribution channel controlling and programming distribution servers to optimize: P2P networks, CDN, playout streamers, customer relationship management systems, etc., and for managing Quality of services, QoS.

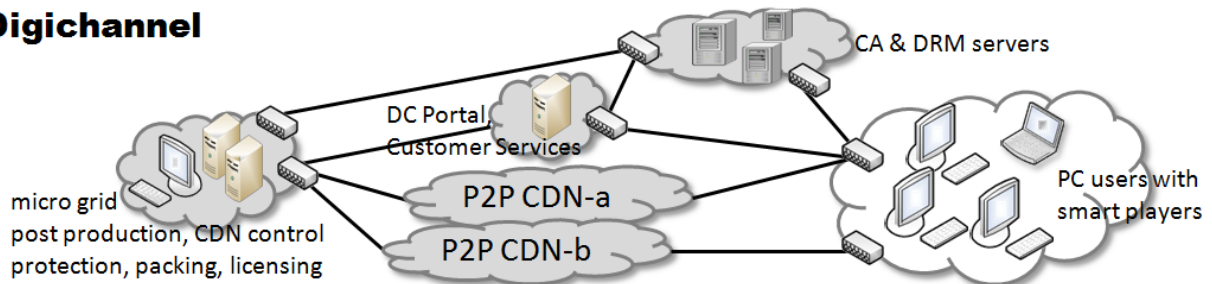
	Digichannel	Mobile Medicine	AX4HOME	VARIAZIONI	Museum MUSA
Content Management: storage, UGC	X	X			X
Content processing: adaptation, cross media content packaging,...	X	X	X	X	X
Intelligent Content: MPEG-21, XML	X	X	X		X
Content Delivery Network Management: P2P, streaming, http, ..	X	X	X	X	
Metadata enrichment and reasoning: EPG, TvAnyTime, DC, ..		X	X		X
Content Protection Management: MPEG-21 DRM	X		X	X	X
Content Indexing and Querying: knowledge base, lucene/solr, ..		X			
Semantic Computing Reasoning: user profiling, content descriptors, recommendations, use data, ..		X	X		
User Interaction Support: collaboration, groups, social networking		X			
Client player as grid nodes, intelligent content: PC, Mobile	X	X	X	X	X

Table 2 – Micro grid usage scenarios coverage areas.

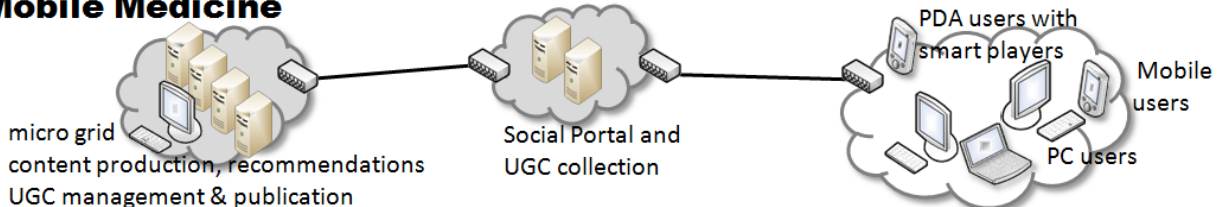
Micro Grid Usage Scenarios

In this section, details regarding some scenarios implemented by using AXMEDIS micro grid solution for media are reported. Among the trials: back office management for IPTV; multichannel management for mobiles with OMA and MPEG-21 DRM; automated production and posting of content for satellite carousel; automated production and distribution of content towards local area kiosks distributed on several locations. Some examples of micro grid usage as back office and/or CDN engines are reported in Figure 3 and described hereafter with some details. The AXMEDIS micro grid solution is called AXCP (AXMEDIS Content Processing) [Bellini2006], it is distributed free of charge, and has been used in several applications and trials as listed on: <http://www.axmedis.org>. Table 2 highlights the exploitation of the identified features mentioned in Table 1 in the cases presented in the following. This table allows to stress the advantages of the proposed solution. All the mentioned activities are implemented as distributed sporadic tasks, while other processes are run periodically such as translation, service monitoring, cache cleaning, clustering, etc.

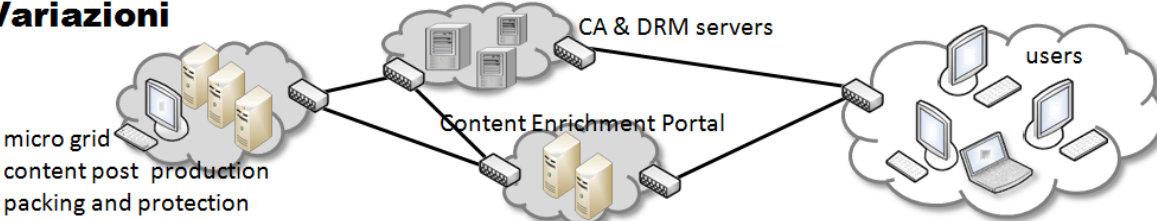
Digichannel



Mobile Medicine



Variazioni



Musa ANSC

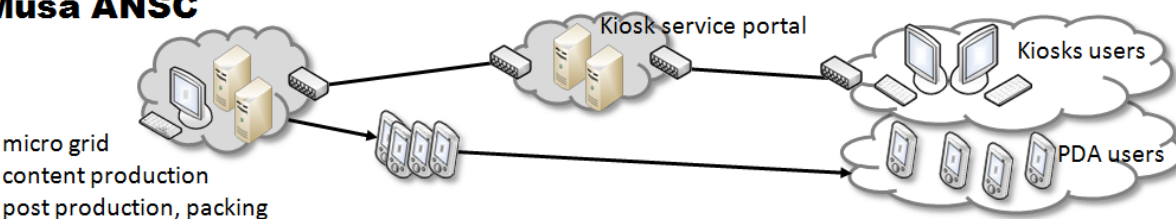


Figure 3 – Examples of micro grid usage patterns.

Digichannel audio visual content portal (<http://digichannel.net/>) distributes audio, video, photo, books and much more in high quality level by distributing original sources. Large size content (and collections) are packed into MPEG-21 intelligent objects which are protected by AXMEDIS DRM and delivered via Bittorrent based P2P networks such as Pirate-bay, Mininova, etc. The MPEG-21 content package may contain multiple and different items such as a CD ISO image plus cover and other info; or the DVD ISO image plus a booklet and images. In this case, the AXCP micro grid supports both the content production and distribution. On the context of production, the grid is used to automatically ingest, pack, protect and publish the content, according to the user and content descriptors. Once published on the Digichannel distribution portal, the same file is published on one or more P2P networks to guaranteeing a fast and spread seeding. The effective distribution and diffusion of the package on bittorrent based P2P networks can be periodically monitored by using AXMEDIS AXEPTool P2P control nodes. The control nodes can be activated on specific torrent files to measure the seeding capabilities and thus the distribution level on the P2P. Once the user has downloaded the content, the user can play the package by using a specific *cross media player* (called Digichannel P2P/player tool) which brings the user to e-commerce portal to buy a license according to DRM and its profile. The latter is used by the grid to produce the license. The DRM tool includes support for the verification and validation of client tools, detecting any changes in the client device/descriptor (player+HW), and using blacklists in the case of infringement detection. The acquisition of the license allows the user to unprotect and unpack the downloaded object obtaining the content in the original media

format. The described business model has been agreed with the content providers distributed by Digichannel. In this case, a reduced version of the AXCP Grid Node (including the Media Grid Processing Rule Engine with a collection of its main functionalities) has been also enforced into a specific *cross media player* to execute the MPEG-21/AXMEDIS content. The *cross media player* presents an extended version of MPEG-21 DIP and DIM capabilities, thus allowing a certain degree of intelligence into the cross media content player; it also includes DRM/CAS supports. Therefore, in this case the semantic computing is enforced into the content production and distribution, but also in the P2P CDN control and user licensing.

Another application of the above mentioned *cross media player* including intelligence capabilities and grid connection has been realized enforcing client side collaboration of both P2P CDN with broadcasting servers (DVB-T plus program guide). This application is a BBC show case called AX4HOME [AX4HOME]. In the **AX4HOME** case, the cross media player tool is activated by clicking on TVAnyTime EPG (Electronic Program Guide) to get information for programming the recording of a free on air transmission in DVB-T, and to activate/program the downloads of additional content from the P2P. The downloaded and recorded contents are used by the cross media player to create an intelligent content in MPEG-21/AXMEDIS format with interactive and proactive parts, which may include advertising. The produced content provides nice shape and presentation interface (similarly to a DVD), for example, it provides links to other related episodes and content, games, biographies, back stages. Moreover, the produced interactive content can be generated in such a way as to be only used into the domain/home of the user who created it [AX4HOME].

Mobile Medicine social network (<http://mobmed.axmedis.org>) uses a range of multimedia players in the browser to cope with different file formats: audio, video, images, documents, complex intelligent content, cross media, collections of videos, sequences of images, animations, medical tools for dosages and procedures, educational content, for PC, iPhone/iPad and PDA [Bellini2009b]. In this case, the micro grid has allowed to automate a range of semantic computing activities such as: (i) ingestion and management of UGC, with the related automated repurposing/adaptation for PC, Windows Mobile, iPhone, iPad; (ii) estimation of content and user recommendations, both based on estimating similarities among descriptors including static and dynamic descriptors continuously updated with the new data about user and content; (iii) automated multilingual translation of textual descriptors and metadata, corresponding indexing for fuzzy and semantic queries; (iv) managing metadata enrichment/update and content versioning that may include reformat and re-adaptation. On the grid, the activities of estimating recommendations, performing translations and metadata enrichment/update, the verification of consistency on the whole database and front-end portal, are periodically activated with different time periods. Other activities are asynchronously activated on demand, such as those related to the management of user generated content (ingestion and adaptation for mobile devices), content publication on other portals, creation of play lists, etc. These activities performed on demand are typically more requested during specific hours of the day and in some days of the week, as it can be observed from Figure 4. This allows to optimize the exploitation of resources in the micro grid, and it suggests to perform the most intensive periodic activities during other hours, for example those carried out on Node 4, namely clustering, a part of content adaptation, etc.

The Mobile Medicine solution integrates a Content Organizer for mobiles, called AxObjectFinder, for Windows Mobile PDA and iPhone/iPad (see Mobile Medicine application on Apple Store). It is an assistant for medical personnel and it is strictly connected to the media grid to get descriptors and content. The content with its semantic associated information is transmitted from server to client in terms of enriched MPEG-21 packages and/or XML files in the case of iPhone/iPad. The PDA Content Organizer includes a reduced version of the Grid Node engine for interpreting the same Media Computing Language and

provides personal recommendations to users by using local semantic computing based on locally collected user behavior data and content descriptors.

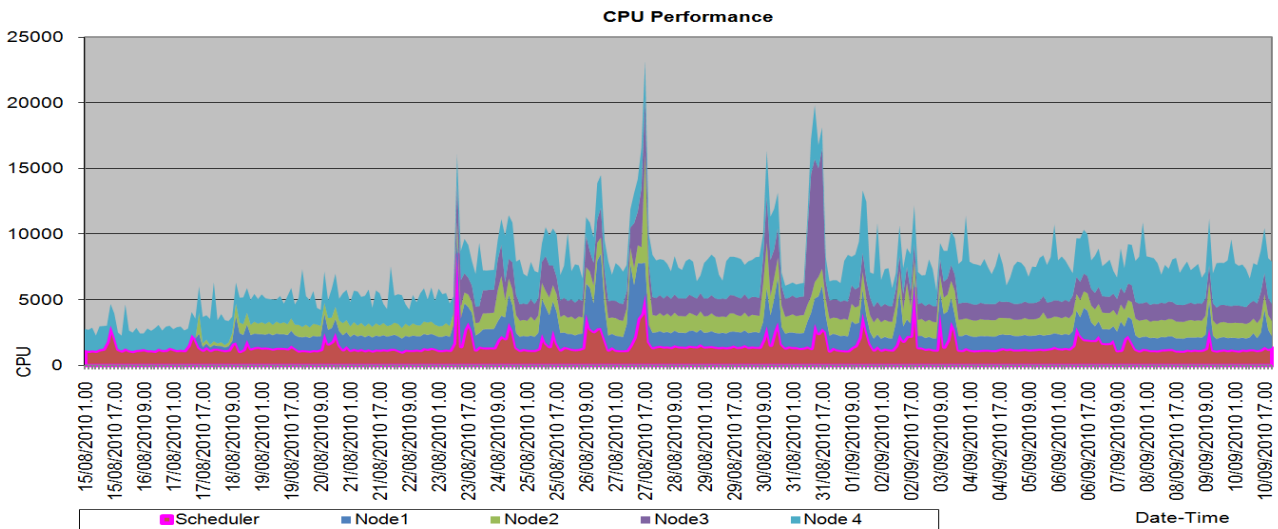


Figure 4 – Workload graph for the Mobile Medicine back office. Execution of periodic and aperiodic tasks are evident. The values are combined in terms of CPU units in the cloud in terms of normalized Mhz. Nodes are referring those dynamically allocated in the micro grid. The graph reports the cumulated values of the workload of the grid nodes used for the Mobile medicine application (nodes 1, 2, and 3 started later with respect to the main application server/Scheduler).

Variazioni Content Enrichment Portal (<http://cep.variazioniproject.org>) performs enrichment on music educational content. Among the distributed content items, some of them need to be adapted and protected according to some specific licensing conditions. For such reasons, a DRM solution has been tuned according to the IPR policies of the content providers. In this case, a media micro grid has been used as an external remote service for the Variazioni service portal. The Variazioni portal invokes the AXCP micro grid every time certain content has to be updated and re-published. The micro grid performs automatically some semantic computing activities: it gets the new content via SFTP, adapt the content according to some descriptor, packages and protect the content according to AXMEDIS MPEG-21. Then, a set of licenses are issued according to the profiles of registered users. In this case, the micro grid is also used to produce client certificates (at the first registration) and licenses on demand whenever users access to the protected content according to a pay per play model. According to several business models, that may be activated, the licenses may provide different kinds of rights to: play, enrich, and print the content. The protected content (audio, video and documents) may be delivered via progressive download (if audio visual) or via P2P bittorrent, or downloaded.

Musical Instruments Museum of the Accademia Nazionale di Santa Cecilia (MUSA) adopted the micro grid solution to automate the production/update of both Windows Mobile PDA and kiosks with interactive cross media content. This interactive content depicts one of the foremost collections of instruments in Italy (http://museo.santacecilia.it/museo/cms.view?munu_str=0_3_1_1&numDoc=148&l=EN). The AXCP micro grid automatically gets access to the MUSA archive and composes the audiovisual guides according to the current arrangement of the gallery and purposes, and thus by using the device descriptors. Once the content is produced, it is automatically posted on the related computers for distribution and usage. The advantage consists in the velocity of services can be updated with, thus offering new content for the users.

Further Research and Conclusions

The media languages and micro grid technologies are suitable tools in the evolution of small and medium size services that are rapidly moving towards media and semantic computing. The usage of distributed systems and their pervasion are blurring the differences from what can be provided by servers and what can be privately computed on clients. In the above-described scenarios, the media and semantic computing enforced in the AXMEDIS flexible micro grid via its media language for formalizing the processes allowed to automatically produce, post-produce, license, and deliver content by exploiting a large range of features, while at the same time managing all the symbolic information needed to cope with users profiling, such as recommendations and suggestions. Therefore, the proposed solution allows any setting up of flexible and scalable solutions to cope with a range of different applications and back office scenarios as reported in the article. Moreover, most of the proposed solutions enabled business models and they are today accessible for small and medium enterprises at low cost.

In some cases, the computing primitives included semantic computing and have been also enforced into intelligent tools and players (see the cases of player for unpacking content, and the content organizer and player). This means to bring into the hands of the final users more features and capabilities, such as local semantic search, recommendations, and therefore, a real personal assistant. In most cases, those clients are not only passive tools since they play a role in the general media grid architecture. The described scenarios stand for many other cases which may be realized by using micro grid models and tools.

Acknowledgements

Our experience has been gained in several years of work in the sector and in a large integrated research and development project, AXMEDIS, by the European Commission with more than 40 partners, which has created a framework currently distributed free of charge. The authors would like to express their thanks to all the AXMEDIS partners and affiliated members for their contribution and collaboration (namely: BBC, TISCALI, EUTELSAT, TIM, TEO, ELION, GIUNTI, HP, SIAE, SDAE, University of Leeds, FHG, University of Reading, UPC, EPFL, etc.), and to the European Commission. A warm thank-you to all the AXMEDIS people who have worked and sorry if they have not been mentioned by name. A specific acknowledgment to institutions and people behind the mentioned running services: Digichannel, ANSC MUSA, Mobile Medicine and Variazioni for their collaboration and stimulus.

Bibliography

[AX4HOME] AX4HOME DVB-T Recorder and Broadcast Enhancer,

http://www.axmedis.org/documenti/view_documenti.php?doc_id=4005,

http://www.axmedis.org/com/index.php?option=com_content&task=view&id=113&Itemid=80

[AXMEDIS] <http://www.axmedis.org> , AXMEDIS portal web site.

[Basu2003] S. Basu et al., "mmGrid: Distributed Resource Management Infrastructure for Multimedia Applications," Proc. 17th Int'l Parallel and Distributed Processing Symp., IEEE CS Press, 2003, p.88.

[Bellini2006] P. Bellini, I. Bruno, P. Nesi, "A GRID based Framework and tools for automating production of cross media content", Proc. of the 11th IEEE International Conference on Engineering of Complex Computer Systems (IEEE ICECCS2006), California, USA, 14-18 August 2006.

- [Bellini2007] P. Bellini, I. Bruno, D. Cenni, P. Nesi, D. Rogai, "P2P Architecture for Automated B2B Cross Media Content Distribution", Automated Production of Cross Media Content for Multi-Channel Distribution, 2007. AXMEDIS'07. Third International Conference on, AXMEDIS 2007, IEEE press, 28-30 Nov. 2007, pp.105-112.
- [Bellini2009] P. Bellini, I. Bruno, P. Nesi, "Visual Programming of Content Processing Grid", The 15th International Conference on Distributed Multimedia Systems, DMS2009, Redwood City, San Francisco Bay, USA, 10-12 September, 2009.
- [Bellini2009b] P. Bellini, I. Bruno, D. Cenni, P. Nesi, M. Paolucci, "[Personal Content Management on PDA for Health Care Applications](#)", 2009 IEEE International Workshop on Semantic Computing and Multimedia Systems ([IEEE-SCMS 2009](#)), Berkeley, CA, USA - September 14-16, 2009.
- [Foster2002] I. Foster, C. Kesselman, J. M. Nick, S. Tuecke, "Grid Services for Distributed System Integration", IEEE Computer, June 2002.
- [Harmer2005] T.J. Harmer, J. McCabe, P. Donachy, R.H. Perrott, C. Chambers, S. Craig, R. Lewis, B. Mallon, L. Sluman, "Gridcast: A Grid and Web Service Broadcast Infrastructure," Proc. IEEE Int'l Conf. Services Computing, vol. 1, pp.35-42, IEEE CS Press, 2005.
- [Huang2007] Y. Huang, Y.-F. Chen, R. Jana, H. Jiang, M. Rabinovich, A. Reibman, B. Wei, and Z. Xiao, "Capacity Analysis of MediaGrid: A P2P IPTV Platform for Fiber to the Node (FTTN) Networks", IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, vol.25, n.1, Jan 2007.
- [InsightReport] The Insight Research Corporation Report "Content Management for Wireless Networks, 2008-2013", August 2008, New Jersey, USA, <http://www.insight-corp.com/reports/content08.asp>
- [Ishaq2009] A. Moiz Ishaq, S. Iftikhar, H. Farooq Ahmed, "Semantic Grid Service Registry", 2nd International Conference on Computer, Control and Communication, pp.1-6, 17-18 Feb. 2009.
- [Leff2003] A. Leff, J.T. Rayfield, and D.M. Dias, "Service-Level Agreement and Commercial Grids," IEEE
- [Rosenblatt2003] B. Rosenblatt, B. Trippe, S. Mooney, "Digital Rights Management - Business and Technology", M&T Books, 2003.
- [Roure2005] D. De Roure, N.R. Jennings, N.R. Shadbolt, "The Semantic Grid: Past, Present, and Future", PROCEEDINGS OF THE IEEE, vol.93, n.3, pp.669-681, March 2005.
- [Volckaert2008] B. Volckaert, T. Wauters, M. De Leenheer, P. Thysebaert, F. De Turck, B. Dhoedt, P. Demeester, "Gridification of collaborative audiovisual organizations through the MediaGrid framework", Future Generation Computer Systems, vol. 24, pp.371-389, Elsevier, 2008.
- [Walsh2005] A. E. Walsh, "The Media Grid: A Public Utility for Digital Media," Dr. Dobb's J., Distributed Computing, pp.16-23. <http://mediagrid.org/>, Nov. 2005.

Biographies

Pierfrancesco Bellini is a contract Professor at the University of Florence, Department of Systems and Informatics. His research interests include object-oriented technology, real-time systems, formal languages, computer music. Bellini received a PhD in electronic and informatics engineering from the University of Florence, and has worked on projects funded by the European Commission such as: ECLAP, AXMEDIS, MOODS, WEDELMUSIC, IMUTUS, MUSICNETWORK, VARIAZIONI and many others. He has been co-editor of MPEG SMR.

Ivan Bruno is a contract Professor at the University of Florence. His research interests include grid computing, image processing, semantic computing, and audio processing. Bruno received a PhD in electronic and informatics engineering from the University of Florence, and has worked on projects funded by the European Commission such as: ECLAP, MOODS, WEDELMUSIC, IMUTUS, MUSICNETWORK, AXMEDIS and many others.

Daniele Cenni is with Department of Systems and Informatics of the University of Florence. His research interests include social networks, semantic database, semantic computing, and clustering. Cenni received a degree in informatics engineering from the University of Florence, and has worked on projects funded by the European Commission such as: ECLAP, AXMEDIS and VARIAZIONI.

Paolo Nesi is a full professor at the University of Florence, Department of Systems and Informatics, chief of the Distributed Systems and Internet Technology lab and research group, and vice-director of the department. His research interests include massive parallel and distributed systems, physical models, semantic computing, object-oriented, real-time systems, formal languages, and computer music. He has been the general Chair of IEEE ICSM, IEEE ICECCS, WEDELMUSIC, AXMEDIS international conferences and program chair of several others. He is general co-chair with Kia Ng (University of Leeds) of the international conference DMS 2011 of Software knowledge Institution. He is and has been the coordinator of several R&D multipartner international R&D projects of the European Commission such as ECLAP, AXMEDIS, WEDELMUSIC, MUSICNETWORK, MOODS and he has been involved in many other projects. He has been co-editor of MPEG SMR. Contact Paolo Nesi at nesi@dsi.unifi.it