Università degli Studi di Ferrara

DOTTORATO DI RICERCA IN
MATEMATICA-INFORMATICA

CICLO XXIII

COORDINATORE Prof.ssa Luisa Zanghirati

Multi-sensor Evolution Analysis: an advanced GIS for interactive time series analysis and modelling based on satellite data

Settore Scientifico Disciplinare INF/01

Dottorando:
Dott. Alan Beccati

Tutore:
Prof.ssa Eleonora Luppi

Anni 2008-2010

to my dearly beloved Erminia
to my family

Contents

List of Figures

List of Tables

Listings

Introduction

This thesis proposes an interactive, integrated system for the exploitation of the large and ever growing satellite data archives. Collecting observations of the Earth surface over decades, these archives store large amounts of data that, among other uses, is used to extract meaningful information about the Earth's surface, in a form that humans can directly comprehend. Access to thematic information contained within these vast archives has to be provided in a way that allows its prompt usability by diverse user communities, either interested in using it for research purposes or for decision support. Several catalogue system exist and a continuous effort is ongoing to improve accessibility to these archives, yet few systems provide systematic access to thematic data and the ones doing so provide very specific products (such as, fire events, burned areas, yearly land cover or land use maps).

The evolution over time of primary parameters, measurable from satellite data, is proven to be a practical and viable methodology for characterization of phenomena that influence a given area. Any change occurring in an area that is relevant enough to change one or more of its primary parameters may present a characteristic evolution pattern over time that can be used for its identification. The prompt availability of thematic data, derived from satellite images with high temporal frequency, is thus an improving element for research activities as it can promote an insightful view over study areas and on their dynamic behaviour over time, avoiding to users the time consuming operations usually required to: search for, collect and prepare large amounts of data for multi-temporal analysis.

In the proposed system, thematic-temporal pattern identification (how the thematic classification of an area changes over time) is fostered by visual display both over areas and at single pixel level to permit direct applicability of a valuable research and analysis tool: the human vision. The archive of thematic maps is browsable interactively to display temporal sequences of maps of a selected study area, that can be

filtered dynamically to display only a thematic class of interest, to observe its evolution in the area over time. At pixel level, a complete thematic evolution across the years can be visualized in a compact visualization form fostering identification of patterns along (seasonal variations such as agricultural practices) and across (yearly or long lasting changes) years. Once patterns are identified, either by visual data exploration or by prior knowledge on a given phenomenon, their search over the data archive can be automated so that their occurrences are detected over a given area of interest (at regional or national scale) at the user's request.

Graphical interfaces are provided to model land cover evolution patterns, to perform a spatio-temporal search for these patterns and to display result maps generated by the search process. These result maps provide immediate visual display of the locations presenting the modelled evolution over a given period. The availability of a generic, thematic-temporal pattern definition and matching system, able to provide interactively (within few seconds to few minutes at 1 Km resolution) the results of a query such as: "identify locations over Italy presenting a given thematic evolution across 2010" is a feature unique to the presented system. The availability of such features in an interactive environment with fast response time, loaded with thematic data extracted from 15 years of Remote Sensing (RS) data at 1-km resolution with global coverage, provides an unprecedented dataset to the scientific community to be explored and used in support to their research activities, with the aim to accelerate the acquisition of knowledge both on known and unknown dynamics observable on the Earth's surface from a satellite's point of view.

The presented system addresses also the problem of promptness of usability of outputs coming from research and modelling activities. An access control framework, providing different views on the system for different user typologies is proposed to enable prompt usability of "verified" modelled patterns to end users who are not interested on detailed data analysis but just in utilization of verified models (users of consolidated products available through classical, catalogue based, delivery systems such as burned areas, flooding events assessment, agricultural practices and any other phenomenon that can be characterised by its thematic evolution over time), with the advantage of having an interactive access to those products, directly on the system that produces them.

Interactive analysis of multi temporal data is a valuable tool for determining land

use and detect relevant land cover transition phenomena caused either by human intervention or natural events, the work reported herein provides a starting point toward the realization of an integrated tool for the exploitation of multiple satellite archives, within a framework providing a common platform to different user communities, fostering collaborative contribution based on interactive reporting of data and cross-validation of models.

A complete implementation of the Multi-sensor Evolution Analysis (MEA) system has been realized in the framework of two European Space Agency (ESA) projects: the Classification Application-services and Reference Datasets (CARD) project[1] where I participated in the realization and deployment of the system architecture, described in 4.2, for systematic processing of the entire AATSR and ATSR-2 ((A)ATSR) archives to deliver the classification maps used to build the MEA multi-temporal data stack, that is continuously updated with new data from the AATSR rolling archives. I have also defined the interoperable catalogue interface that permits automated access to the data archive to perform thematic queries over the archived data with the addition of a flexible data order operation.

The implementation of the first MEA prototype over the entire (A)ATSR archives was completed in the framework of the Support by Pre-classification to specific Applications (SPA) project[2], where I participated to all the main aspects in the realization of the system: from requirements analysis, to system design and implementation, with an emphasis on performance oriented design justification and system validation in close cooperation with ESA staff. I also designed and verified the development of a second version of the system (documented herein), which added features oriented toward system usability, such as aggregation of the features domain for configurable level of detail and prepared it for integration of multi-resolution data. A complete instance of the system is now available to the the Earth Observation (EO) community through the Ground Segment Research and Technology Development (RTD) Department at ESA - European Space Research INstitute (ESRIN) (one of the five ESA specialised European centres).

My thesis work was also in close cooperation with Meteorological and Environmental Earth Observation (MEEO), an Italian company located near Ferrara that developed the SOIL MAPPER® (SM) classification system based on spectral signals remotely detected by a satellite's sensor. MEEO was prime contractor of the ESA

projects and owner of SM that has been selected as the classification system to build the classification maps for MEA implementation.

This thesis describes the approach used in building the system, the data processing methodology (conceptual design), details architectural elements and interfaces of the system implementation over 1 Km data and elaborates on results obtained in terms of potential uses and advantages of the features provided, including results of their evaluation by a group of end users participating in validation activities to assess both usability and usefulness of the system. It is organized in chapters as follows:

Chapter 1 provides a short introduction on the idea of using a thematic view over the temporal domain as an enabling tool for insightful exploitation of large satellite data archives. Its founding principles and contextual information is provided;

Chapter 2 defines the goals of the system implementation and outlines the system architecture, elaborating on the main perspectives considered, driving the definition of the system and design choices relating to each of them;

Chapter 3 details the conceptual description of the methodology including the data processing flow to prepare a consistent data set that is comparable in the geographic, thematic and temporal domains, across different sensors. The thematic-temporal pattern matching system is defined along with the proposed layout of the user interfaces for interactive data presentation;

Chapter 4 describes the chosen satellite dataset to populate the system archive and provides an overall view of the implemented systematic data processing chain, built with a minimal, firewall friendly, reusable distributed processing system, to extract thematic data and prepare it for interactive access and analysis;

Chapter 5 describes data access and presentation functions, including key functions of the graphical user interfaces and the model matching engine, implementing a distributed processing platform based on Web server technology;

Chapter 6 provides a critical analysis of results obtained in terms of features provided, performances for interactive analysis and results of the assessment performed in collaboration with a group of end users;

Chapter 7 draws conclusions on the presented material and proposes possible improvements and future work directions;

Chapter 1

A thematic view on the temporal domain

The ever increasing availability of Earth RS data acquired from orbiting satellites calls for the development of tools permitting the EO community to efficiently exploit such vast and growing amount of data. Thematic categorization is one of the possible ways to extract meaningful information from large data archives while reducing data volume to a more manageable size. Such generic thematic information, closer to human semantics, can then be displayed for interactive analysis by a user community interested in gathering this information or in its deeper analysis within a specific thematic field.

This thesis proposes an interactive system that permits to analyse time series of geo-referenced thematic data. The system provides visual data browsing features as well as tools for computer aided modelling of thematic evolution patterns (defining how the thematic data is expected to evolve over time) and their automated matching against very large databases (many years of data over entire continents). Therefore this system extends the normal search in the space and time dimensions with the capability to verify the matching of the modelled thematic evolution patterns and provides derived maps for on-line analysis. A multi temporal thematic evolution search introduces a form of content based information retrieval on the archived data. The interactive system operates on the results of a bulk data processing infrastructure that extracts the thematic data from the huge basic data and stores it in a form suitable for interactive display and analysis. The implementation of the total system (pre-processing and interactive parts) for an entire archive of the data acquired from a moderate resolution RS instrument is also herein described. The remaining of this chapter provides basic foundational concepts to contextualise the presented work.

Visual analysis for data exploration

An interactive environment for data exploration that promotes visual pattern identification is an essential element to ensure direct applicability of the human vision in data analysis. Human vision has been recognized as an important tool in the advancement of science, paired with data visualization tools to foster pattern identification in geographic data applications such as cartography [3]. The way in which data is presented can thus help to provide insight on the causes determining the observations and an exploratory attitude to data can lead to discovery of unknown phenomena or be a tool to provide assessment for further analysis directions; Exploratory Data Analysis (EDA) principles and the importance of such attitude to data are provided in [4].

Data exploration is a concept now permeating several fields of RS, like the observation of the skies through the Virtual Observatory (VO) which provides capabilities to analyse and integrate astronomy data from different providers, as well as to perform interactive computations on elements from its widely distributed digital data archives[5]. Within the Knowledge-based Information Mining (KIM) framework, data mining techniques are also being applied to search collections of EO images for features of interest. KIM provides an interactive environment for spatial data mining, attempting to simplify user interaction with complex multidimensional data [6]. The most recent launch of the Google™Earth Engine project to deliver a platform to browse and access an impressive amount of world-wide raster satellite imagery is also evidence of the increasing availability of on-line data and the focus towards their interactive exploration to increase its exploitation[7].

Graphical EDA tools are powerful instruments to provide an insightful display of data and its interactive visual analysis is a key aspect of the presented work: it provides a Graphical User Interface to interactively browse data, with focus on the temporal domain. Building on top of such vast archives, the data reduction potential of feature extraction can be leveraged to provide access to high level (thematic) data that users can visually explore.

The land cover thematic domain

Among EO applications, Land Use and Land Cover Change (LULCC) topics are becoming more and more critical subjects for the impact they have on global climate. They are in fact linked to climate and weather in complex ways and are fundamen-

tal inputs for modelling greenhouse gas emissions, carbon balance, natural ecosystems and human environment evolution. Both human activity and natural phenomena can affect many of these processes, that are strictly correlated, influence each other and have strong impact and consequences on environmental, social and economic aspects as well as on human health. Land cover refers to everything that covers the land surface, including vegetation, bare soil, buildings and infrastructure, inland bodies of water, and wetlands. Land use refers, instead, to societal arrangements and activities that affect land cover[8].

Many approaches and methodologies exist for land cover change analysis: an extensive survey is provided in [9]. Recent work for multi temporal analysis systems was performed to provide targeted land cover change studies or develop yearly databases of land cover [10]. An interesting bi-temporal approach to land cover change analysis is provided by the Land and Ecosystem Accounts (LEAC) methodology whose main goal is to provide an easy and comprehensive access to land cover data, showing the 'stock' available for each land cover class in the different land cover data, and providing also the changes occurred in the periods between different land cover works, as land cover flows matrices [11]. Besides bi-temporal change reporting, the suitability of modelling the change patterns of derived quantities extracted from multi temporal satellite data to identify relevant phenomena has been confirmed for such fields as agriculture; an example is given by [12] that confirmed the applicability of the Normalized Difference Vegetation Index (NDVI) from moderate resolution satellite data as a cost- and time-efficient mean for large-area crop mapping in the U.S. Central Great Plains.

The availability of large quantities of EO data, with an ever increasing frequency in the temporal domain, especially if considering a combined use of multi-sensor data, creates the opportunity to develop new tools for both interactive analysis and visual exploration for pattern identification over time series of images. Yet no application is available to provide an interactive view on thematic land cover data, allowing interactive exploration of its evolution over time. The proposed system is thus an improving contribution that provides an unprecedented thematic view over data archives.

Feature extraction

A large variety of tools and algorithms are now available to perform categorization over satellite imagery to extract meaningful features associated to understandable semantic

meanings. Many of these tools in the past operated on individual optical images to identify features, such as those obtainable from algorithms working on spatial patterns or spectral signatures. The output of the latter algorithms have the ability to reduce the dimensionality of satellite datasets (in terms of spectral bands) into compressed maps of thematic information (thematic maps). These maps provide increasing abstraction levels towards a higher semantic meanings clear to the end users and close to his application terminology.

The high amount of time required for visual or semi-automated image analysis calls for the use of more automated (unsupervised) pre-processing systems in order to improve satellite data exploitation. In the land cover field, the availability of such systems and the continuous technology improvement allows to obtain land cover maps from huge amounts of data in a relatively small amount of time, as it is the case with the recent global scale land cover maps for 2009 data, produced within the following year by the GlobCover project [13].

On the one hand, virtually every Earth Science (ES) study and application can greatly benefit from the use of the existing archives of long time series of satellite data and the ever increasing availability of new EO images that provide an unprecedented global coverage from different sources at different resolutions. On the other hand, since the size of a single multi-spectral image is in the order of hundreds of Megabytes, the real time utilization of these datasets for on-line analysis is a technological challenge by itself: computer systems play a key role in the EO field to readily process such large data volumes. The use of a fully automated, unsupervised classification system allows to implement clustered and distributed processing, in order to quickly deliver data and extract information for interactive visualization and ultimately pattern search.

Thematic evolution over time

The methodology described in this thesis is based on advanced applications for single image feature extraction to deliver an integrated system with tools for multi-temporal analysis of time series of geographically referenced data. As shown in Figure 1.1, a thematic classification greatly reduces data volume, that becomes manageable to offer fast interactive analysis aimed also at visual pattern identification. Pairing the thematic view with a pattern definition and matching system allows to automate the search for defined patterns. A generic pattern specification that is not focused on

any specific change typology, or phenomena, provides a flexible exploration tool that can be used by scientists to interactively exploit multi temporal data. This interactive search feature can be an improving aid to research activities: the ability to interactively search for a temporal evolution pattern (in the thematic domain) at pixel level gives a new thematic view over existing archives of satellite data and a new way of interacting with them to discover and use information.

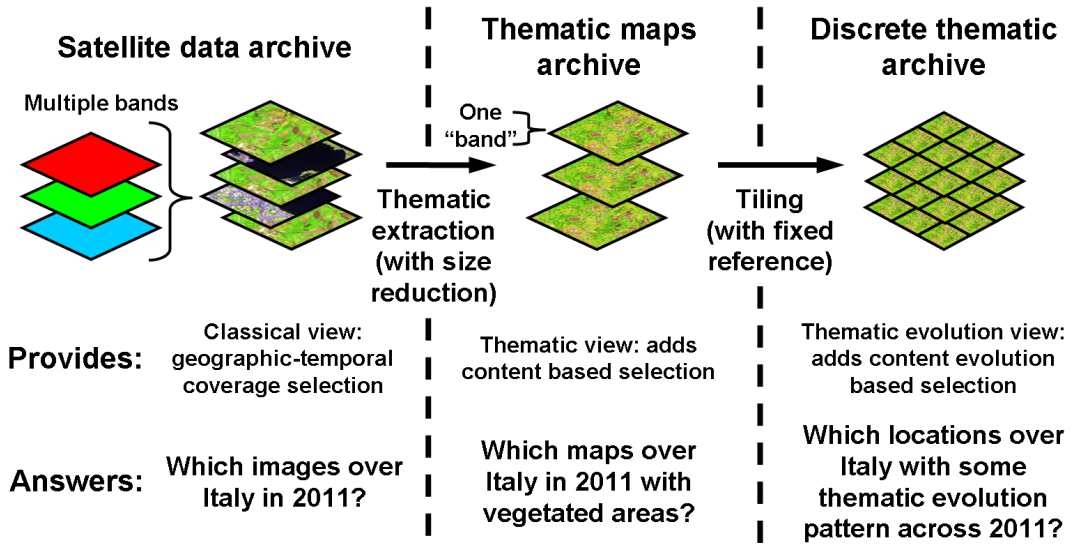


Figure 1.1: From large archives of multi dimensional satellite data, thematic information can be extracted, obtaining dimensionality reduction for each image to three dimensions (two spatial and one thematic). Tiling the thematic maps and linking each tile to a fixed geographic reference system and to the image acquisition time, permits to perform complex modelling and queries on the evolution over time of the thematic content.

An implementation is build for LULCC, with moderate resolution data, applicable to identifiable thematic categories (abstract, expressed in semantic terms) and providing an interactive framework for visual exploration of land cover maps, with the immediate benefits of visual assessment of the accuracy of the underlying classification system and the land cover dynamics of any study area. The implementation allows also automated search through the temporal domain for specific patterns, defined using graphical interface. The automated search over the time series can be used to test the extent of occurrence of a pattern over wide areas and different time periods to confirm pattern validity or to identify areas worth studying in more detail. The output of the search can also be exported in computer readable form to be used to drive further processing by other systems requiring more processing resources, hence it is applicable

as a screening tool identifying relevant subsets from the data archive without the need to access and process the datasets.

User community and technology

Another problem tackled by this thesis is to bridge the gap between research and the prompt usability of its results over EO products, a goal that is pursued by: considering different user roles and providing them different views of the system; adopting a web application model for user interaction; defining an access model that promotes collaborative work while allowing access control. With the advent of web 2.0 technologies and the availability of client side programming features in modern web browsers it is now possible to deliver highly interactive interfaces directly over the web, without the need for installation of specific software packages, while providing the opportunity to build central points of references (i.e. web Portals) for collaborative work on specific subjects and leverage on high performance resources close to the data to perform the needed computations.

Summarizing, the MEA system is designed to be a widely applicable, generic tool for interactive analysis of long time series of (raster) geographic information in the temporal domain through the provision of specific tools to model, query and visualize data. It allows not only querying data by single map content but also to search for occurrences of specific content patterns over time at pixel level. MEA is applicable to any data scale while the implemented modelling system is better suited for data at nominal scale of measure, as defined in [14], such as the categories of the land cover thematic maps used for the presented implementation. Thematic categories can be easily displayed for better human understanding; for example they are usually mapped to a meaningful set of colours to produce false colour maps that increase readability. The applied principle is to ease visual analysis of classification.

Although thematic maps are a powerful tool by themselves for the depiction of an area in a readily understandable way, they can also be used in time sequences to effectively depict the dynamic aspects of the area. Two maps at different times are commonly used for the purpose of change detection, however, the prompt availability of a series of these maps can provide better understanding of these changes. An interactive search tool, can then be used to investigate them in greater (temporal) detail. Search features, based on sequential pattern matching in the thematic-temporal domain, have

been defined and implemented in MEA as a starting point toward interactive multi-temporal analysis in an integrated environment.

Chapter 2

Perspectives on an interactive thematic system

The first implementation of the system, described in this work, is targeted to be suitable for fast interactive analysis of land cover maps produced by classification of the entire (A)ATSR archives and to be continuously updated as new data is collected by the sensor. Besides this first dataset, the system is designed to be ready to support further datasets derived from multiple sensors at different resolutions. In this context, the system has been considered from the perspectives presented in the following sections, along with discussion about driving choices, assumptions and key principles followed for each of them in designing the system.

2.1 Users needs and access policy

Clear identification of users of a software system and their needs is the first step and a key factor in providing an effective design, as endorsed by state of the art guidelines in the Space segment [15], while multi-user environments require user authentication and authorization policies to regulate access to their functions. User identification and access policy for MEA have been defined taking into account that it will serve diverse user communities, possibly on different domains that employ thematic maps and can benefit from interactive multi-temporal analysis with the contribution of user provided content.

In particular, with respect to data utilization, the identification and definition of evolution patterns over time and their coding into models that can be automatically searched over time to provide on-demand thematic maps, is a key function, well suited for identification of two broad categories of users. A first category of users is interested

in the exploratory data analysis functions and in the use of evolution models to assist their research activities such as investigations in a specific domain.

A second category of users is interested in the prompt availability of on demand thematic views over user selected geographic areas and temporal ranges: examples are policy makers and public administrators, who can exploit consolidated products resulting from the temporal evolution analysis. The "Expert" role is assigned to the former users, since they have specific knowledge in their domain, while the "Standard" role is assigned to the latter users, because of the simplified system view they require. Users in the Standard role require access to finalised, understandable and valid products obtainable from published models. This user categorization is also an attempt to bridge the gap between research and use, supported by overlapping views on some of the system functions in an integrated environment. Moreover, for any computer system that implements user identification, there are two essential user profiles to consider: "Anonymous", which refers to unidentified users and "Registered", which refers to identified users regardless of their other attributes. Finally one last essential role is considered: "Administrator", which is assigned to users responsible for system and data management functions.

From the perspective of authorisation, in order to keep a simple categorization of functions associated with shared elements, the Unix file access control model is applied to evolution models, to complete the foundation principle for our access policy, it is crossed with the Windows Access Control List (ACL) concept. As detailed in section 3.5, from the Unix model we take the emphasis on ownership and the basic set of permissions: Read, Write and Execute while from the ACL model we add group list to allow assigning permissions for more than one group of users to the same model.

With respect to functionality, different functional needs are associated to each defined role that a registered user can belong to. In our model, access to functionality for user roles is inclusive with respect to the lower role as listed hereafter, in descending order of inclusion:

Administrator an administrator is expected to perform administrative tasks on the MEA system, hence it is granted all the capabilities and permissions granted to the other roles; in addition, specific administrative functions are defined:

- Manage the data archive by means of input rules: a set of rules to drive

automated data ingestion, oriented toward easing the man-machine interaction, have been defined using a generalised set of labels over an underlying work flow as detailed in section 3.6;

- Manage users and monitor system usage and health status.

Expert an Expert user is a domain expert in the thematic covered by the classification maps, with good knowledge of some study areas or interest in modelling some phenomena by its thematic evolution; Expert functional needs are identified as:

- Have a fast interactive query tool to perform multi temporal visual analysis on large (spatial and time coverage) data archives e.g., to have a different/broader perspective of a known area or to search new ones presenting unusual characteristics to study;
- Identify relevant thematic patterns, both at the pixel or area level;
- Define evolution models over some identified or hypothesized pattern;
- Run defined models over the archive to interactively search for its occurrences over some Area Of Interest (AOI) and multiple time periods, to take advantage of the search outputs;
- As the owner of the defined model, the "Expert" user is responsible to define its access policy and its associated meta-data;

Standard an ordinary system user is assumed to be a decision maker interested in obtaining environmental information derived from analysis over time (e.g. specific land use or other phenomena) that has been modelled to be automatically detected over an AOI at any given period in time; identified needs for that role are:

- To have a fast interactive tool delivering high level thematic information to support decision making;
- To have a user friendly interface with a simple and effective interaction model.

The MEA system is thus designed with features aimed at providing an innovative integrated environment to satisfy these functional needs; as an interactive system it is also designed to minimize response time perceived by the users to keep a high level

of responsiveness. As users with different roles access the system with different views on it, i.e. using different interfaces, the contextual view of the reference scenario, manageable with this access control model, is depicted in Figure 2.1. Following the three identified roles, three main views on the system are also defined as interfaces for: Operators/Administrators (OGUI), Expert users (EGUI) and Standard users (SGUI).

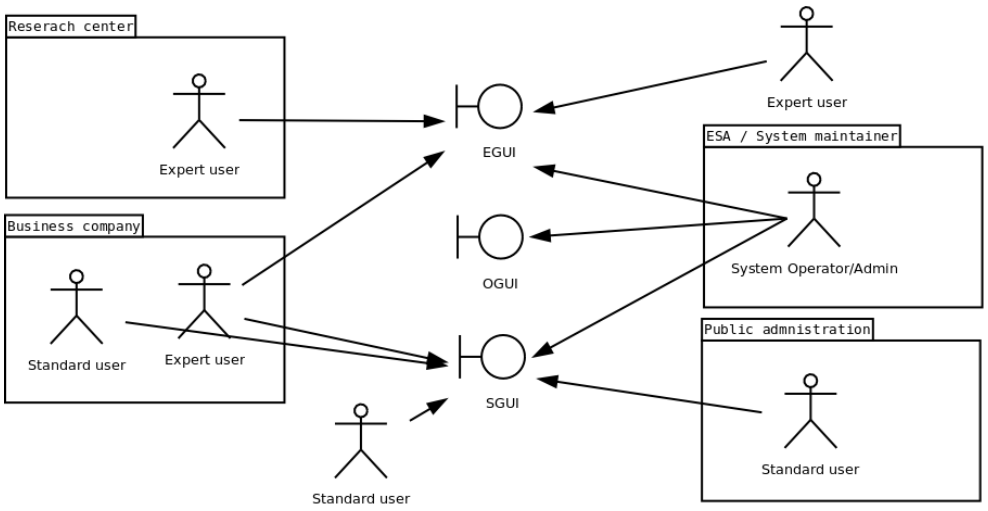


Figure 2.1: User access scenario

The presented policy model is designed to allow collaborative joint efforts for the development of a single evolution model by different expert users, possibly belonging to different organizations or not affiliate with any institution. Moreover, a group based model avoids also the implicit limitation of role based control only to editing features in multi-user environments, as it is the case of systems where a user with access to editing functions, an "Editor", could not only use editing features on its own models but also those belonging to other users without their authorization, unless no collaborative editing is enabled at all.

2.2 Data preparation

As an integrated, general purpose analysis tool, the system is designed to operate on any categorization data that can be mapped on the Earth surface in raster form. The presented implementation, however is focused on the LULCC domain: it is built over land cover maps, obtained by unsupervised classification of satellite data at moderate ground resolution of 1 Km. Use of moderate resolution data brings two advantages: it is well suited for a first system implementation to assess its performances over limited

hardware resources while providing access to a global dataset with high temporal frequency. The potential and applicability of the application can thus be verified providing access to an extensive dataset at global scale. Use of a single data source does not hinder the possibility to define extensions for multi-sensor integration, in fact the selected archive holds data from two very similar sensors with different resolutions, and to build a scalable system that can grow dynamically as further data sources are added. Another advantage from the choice to use (A)ATSR data is to open an unprecedented thematic view over its entire European archive that allows exploring it interactively with a new perspective.

Using two sensors of the same kind, at similar resolutions, the derived maps archive offers also the opportunity to assess across sensor processing performances. The focus on multi-temporal analysis drives the approach at data and dictates how it has to be processed to build a consistent stack that can be used for time series analysis: accurate geolocation and accurate radiometric calibration are the basis for the pre-processing chain in order to attain spatial and radiometric comparability; furthermore, a spectral classification based on prior-knowledge is applied with the aim to provide a sensor independent data layer, consisting of land cover typologies, with a human understandable semantic meaning.

During the development of the system, a strong standardization effort was ongoing in the EO data archives domain toward increased accessibility of data and services; among other events, the Service Support Environment (SSE) reached its operational state providing a generalised web based platform for delivering value adding services to users through a common interface and a defined set of operations with customizable parameters [16]. To promote interoperability and data accessibility, the output maps produced by classification of the entire (A)ATSR archives are also provided as an SSE service, named Advanced Semantic Query system for Large satellite Database (ASQuLD); it enables the provision of search by area and time coverage of the Earthnet OnLine Interactive (EOLI) interface[17] to browse the maps. The extensibility of the standard has been exploited to add thematic content to the search parameters allowing searches on image thematic content. In addition a minimal, yet flexible, custom Order operation that allows full automation in dataset retrieval via File Transfer Protocol (FTP) is added, as defined in section 4.2.2.

The ASQuLD service is the foundational building block of the data processing

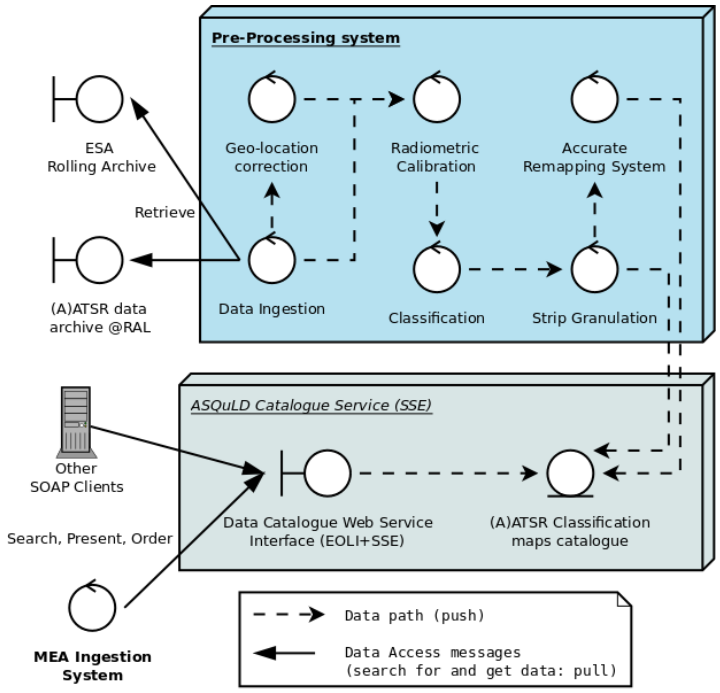


Figure 2.2: Overall diagram of data preparation elements: the ASQuLD service retrieves satellite data elements both from archived data and continuously from an up-to-date data source (Rolling Archive), processes it to extract thematic data and publishes it to be accessible to SOAP clients. Its interface allows querying on the thematic content of the classified maps.

chain of the MEA system, that we call the Classified Maps Provider (CMP) as it can be also seen as the extension point toward other thematic domains. An overall diagram of that component is provided in Figure 2.2. Its implementation design is focused on performance and scalability in computing the classification maps: taking advantage of the unsupervised, pixel based nature of the selected classification system, a distributed computation approach is used to build a processing infrastructure that can process the entire archive in less than six months. The distributed approach allows further acceleration, if needed, by adding processing elements that can also be moved close to the source archive itself to reduce network traffic by transferring only the classified map to the system. The distributed pull model protocol used for data processing is detailed in section 4.2. To ensure best geolocation accuracy, a new geolocation correction module, accounting for the sensor specific geometry and acquisition model was implemented with the support of Rutherford Appleton Laboratory (RAL) scientists that increased the accuracy of the resulting maps.

Building on top of the classification maps, accessible via the ASQuLD service, the

MEA archive, designed for interactive access, is built to enable multi-temporal analysis by remapping all data over a common reference system so that the time dimension can cross a consistent spatial data stack. Besides facilitating multi-temporal analysis, the use of grid reference systems "has been recognized as key point for the integration of heterogeneous sources of data"[18]. This approach, complemented with the use of a variable grid size, allows to build a system that can work at different resolutions, in order to exploit data from different EO sensors. The resulting layered reference system is defined in a way suitable for fast computer processing.

Being a global scale application, a widely used system for depicting global maps has been selected to display thematic maps content to users in a two dimensional representation: the plate carrée or simple cylindrical projection. That projection is the simplest form of association between map points and their coordinates on the Earth surface (actually on its surrogate ellipsoidal representation), that simplicity can be leveraged to build raster data without the need for associated geo-location information since it can be coded directly in the raster definition: by defining a regular sampling mesh over the standard Latitude Longitude coordinate system, the corresponding digital representation as raster maintains that regularity, resulting in the simple cylindrical projection if directly rendered.

This simple approach to sample and represent geographic data has well known limitations, as extensively reported in [19] and [20] where it is confirmed to be suboptimal, especially if compared to several kinds of Geodesic Discrete Global Grid (GDGG) that partition the Earth surface using polyhedra as base grid structure making them particularly suited for three dimensional representations such as virtual globes[21] as these have the same dimensionality of the base grid structure. Nonetheless the geographic coordinate system is still by far the most widely used reference system for satellite imagery and its plain representation well known to the user community. The adopted solution, that is basically to define a regular square partitioning over the simple cylindrical representation of the Earth, is demonstrated to be adequate for the presented implementation at 1-Km resolution, as reported in section 4.3, and to deliver good performances for an assessment of the proposed approach for multi-temporal analysis. Although the efficiency of the adopted solution decreases toward the poles as the distortion produced by its driving representation, it is considered to be still applicable at finer resolutions to assess system performances and set the basis for future research

on multi-resolution usage of the system. The most relevant drawback of that solution is known to be the absence of equal-area cell regions in the derived grid and this is especially evident for applications near polar regions where a different approach to data sampling would be a better choice.

The solution adopted for the layered, multi-resolution, reference system is basically a Discrete Global Grid System (DGGS) that, according to [19], is congruent and unaligned with respect to the planar representation of the Earth over which it defines an uniform square partitioning, hence well suited for two dimensional display and related operations since it does not require re-projection to the target display and its congruency allows a regular square pixel subdivision at each level. On the other hand the solution is non-uniform with respect to cell areas over the Earth surface, hence not optimal for data storage that is still greatly lightened by the classification process, but still leaves room for optimization, as reported in section A.5. Re-sampling classified maps on the DGGS can be done on any kind of categorical thematic map using the nearest neighbour algorithm, without changing actual class values.

An ingestion system that retrieves maps from the ASQuLD service has been designed to perform the re-sampling operation to provide full process automation as a solution to ease the data management and maintenance tasks that are common for satellite data archives, especially for derived products that are subject to changes in the algorithms that generate them, thus potentially requiring periodic, selective re-ingestion. A rule based ingestion controller is implemented and an associated set of rules is defined to drive its operation. To complete the MEA ingestion system depicted in Figure 2.3, storage elements are defined to archive ingested data for interactive access.

The approach to the data archive is oriented toward exploitation of the natural spatial-domain partitioning that is induced by using a regular sampling over a flat square Earth representation: grid elements are grouped together in uniform Tiles of equal size. Tiles are the basic storage units of the system and their fixed spatial addressing on the grid eases both the implementation of parallel processing solutions and spatial queries efficiency. Efficient storage solutions applicable to data organized in regular grids exists in literature, for example the on disk layout proposed by [22] to accelerate real-time exploration with optimized storage techniques. The advantage of such Out-of-core computing [23] techniques should be investigated as new data layers

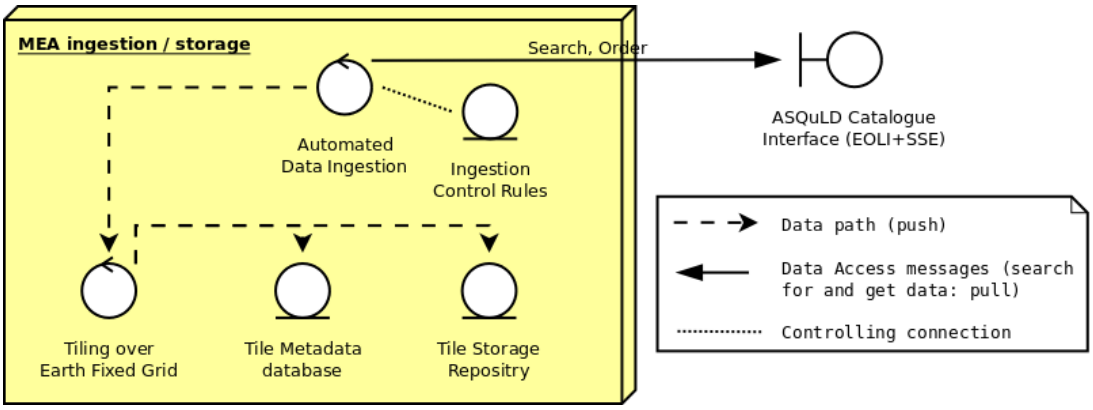


Figure 2.3: Overall diagram of the Automated Data Ingestion system. An automated ingestion component, driven by a set of ingestion rules defined by an administrator, retrieves classification maps from the map provider. Maps are Tiled over the reference grid and stored in the Tile archive for interactive access.

at increasing resolution are added to the system as the advantages obtained with performance oriented data layout restructuring could be applied to the Tile archive. In the presented implementation however, the use of standard files stored on a robust standard file system has been preferred and the spatial partitioning scheme exploited by organizing data into an hierarchical directory tree designed to have leaves (data files) oriented toward the time dimension, as shown in figure 2.4. The geolocation space is partitioned by directories to enable direct mapping between Tile identifiers and file paths. This approach exploits also the operating system cache by organizing directory meta-data content toward the geographic location, accelerating both location of data along the temporal axis and responsiveness to subsequent request related to the same area.

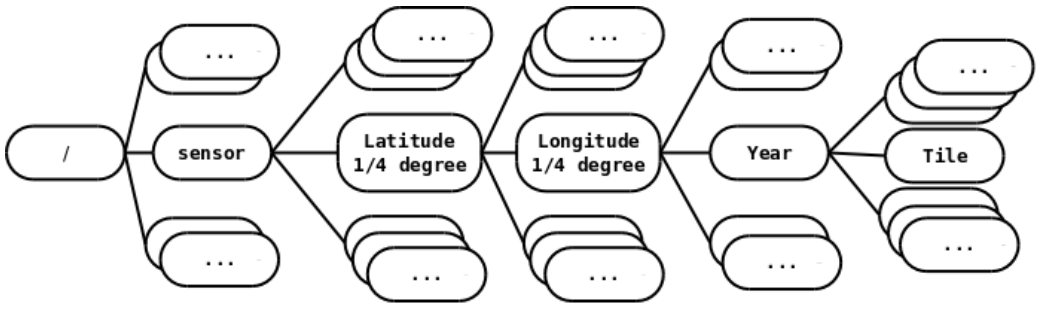


Figure 2.4: Directory tree structure of the Tile storage. The hierarchy is multi-dimensional: sensor, geographic location and finally temporal reference are used to partition the file system into manageable units.

The use of direct file system mapping promotes also storage scalability by allowing

distribution of data across several file systems on different storage arrays without requiring any customization. As shown in Figure 2.5, different storage units with different size and performance characteristics can be used to store a partition of the archived Tiles by making direct use of mount points within the directory tree.

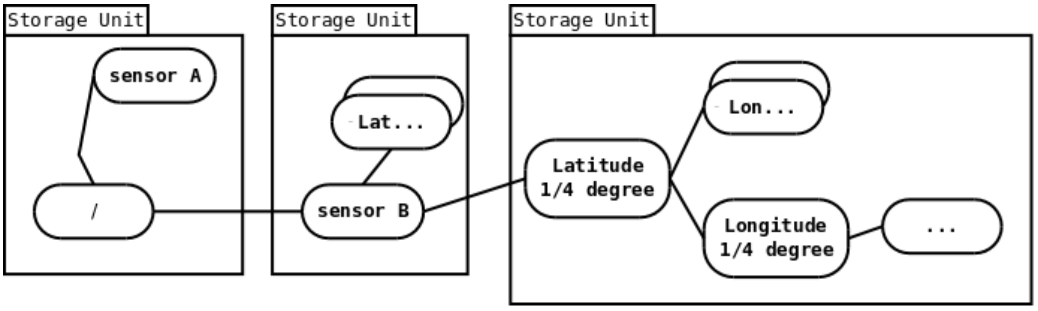


Figure 2.5: Use of different storage units by direct use of mount points. A file system based approach allows linking and mounting of different storage elements transparently to the software elements managing its content.

Besides the file system based Tile storage, Tile meta-data is organized in a customized, performance oriented database to provide fast identification of Tiles as fast interactive browsing of the archive is recognized as one of the key success factors for this system. Fast access to Tile meta-data for interactive queries is thoroughly examined with the goal of performing complex spatio-temporal-semantic(thematic classes) queries within few seconds over the entire temporal domain at national scale areas. Besides removal of the real-valued geographic coordinate system in favour of an integer valued Tile addressing scheme, modern Data Base Management System (DBMS) features such as partitioning and clustering, as well as recent technology such as Solid State Drive (SSD) are also employed and tested as means of performance improvement.

To be noted also that the data duplication issue toward poles has been assessed but not addressed in this implementation, for example the use of an equal-area projection as the basis for grid definition, such as the sinusoidal, would remove the duplication at the expense of greater geometric distortion or the adoption of different base projections on different zones, as is the case with the Goode Homolosine projection, that gives a raster-friendly equal area representation at global scale with a balanced look for land areas. These solutions would also require further customization toward results display and area of interest management and its representation to the users with respect to the simple cylindrical. Application scope and design choices however, help to mitigate that issue. Firstly the focus on land cover and the cloud detection capability of the

classifier allows discarding cloud samples to focus only on valid land cover classes, as clouds are filtered out of the data stack, entirely cloudy tiles are discarded together with those completely over sea. To further mitigate storage issues, temporal resolution is limited to one day, adequate to typical revisit time of moderate resolution satellites, that allows elimination of the multiple acquisitions for a single day approaching the poles, due to the near polar orbit of those satellites.

2.3 Thematic search and evolution modelling

Fast thematic content filtering at single Tile level enabled by the customised meta-data handling is a key element of fast interactive visual display of long time series. Browsing data in a way that aids temporal pattern identification is one of the main goals of the MEA system. An automatic system to search interactively for occurrences of identified patterns across the archive is then added providing a feature unique to this system to further proceed in the analysis process. The provision of a modelling tool to define the identified (or supposed) pattern, that can be readily matched to assess the extent of its occurrence and the visual display of the results again as a thematic map is a simple, yet effective way to provide a versatile tool for interactive visual analysis in the temporal domain. In addition, the option to retrieve search outputs also as plain text files allows external processing systems to make direct use of them.

Automated data analysis techniques, even if very relevant for pattern detection (data mining) in large data archives, require considerable effort in design and validation and can be applied only to specific fields. Our definition of evolution model is instead designed to let the user precisely define each model element with tolerance margins in both thematic and time domains while aiding tools are provided to derive model elements from observed data. All the knowledge for multi temporal analysis is provided by domain experts in the form of evolution models (searchable patterns) that provide immediate thematic understanding of the modelled phenomena. This pattern matching approach has been selected as a first analysis tool to demonstrate the capability of an interactive thematic analysis to explore and exploit satellite data focusing on the temporal domain, it lays also the basis for the introduction of more automated techniques in the integrated environment.

In the MEA context, an evolution model element, or simply Model Element (ME) defines an association between a given set of categorical values (thematic classes) and a

temporal reference, defining "when" that set is expected to be observed. An Evolution Model (EM) is in turn defined as a sequence of model elements, positioned along the temporal line by relative time references. In the LULCC domain, an evolution model defines a sequence of land cover classes that are expected at given times to model the land cover evolution over time. As a sequential pattern, an evolution model can be quickly matched with actual time series data at the pixel level, to determine if that data matches the modelled evolution pattern. The suitability of change patterns over time to identify relevant phenomena has been confirmed in agriculture[12] and its applicability to define also transition phenomena to identify changes of interest directly derives from the temporal attributes of such changes (e.g. burned areas identification imply a change in the land cover of the affected area as do flooding events).

The proposed evolution model matching is a form of change analysis that, according to the topology summarized in [9], falls in the "classification" category and in particular, it is a form of the commonly used post-classification comparison; it is combined with pattern matching to provide comparison at an arbitrary set of time intervals to detect an evolution pattern over time, thus realizing an automated multi-temporal search tool applicable for fast, on-demand time series analysis.

One foundational basis of the system in the LULCC field, the evolution of land cover classification over time, based on spectral analysis of principal measures, can lead to the identification of land use typologies and to the contextual detection of major disturbances (areas of rapid land-use / land-cover variations). The key to the automatic identification of relevant evolution patterns is the definition of a corresponding evolution model that can be systematically used to determine if a given series of observations conforms to the modelled pattern.

Being a post-classification system, its accuracy is strongly dependant on the accuracy of the underlying classification system, in fact it can be at most as accurate. Single image classification depends instead on the resolution of the underlying data, since land cover changes must be discriminated by the sensor for the classification system to detect a different pixel class. The effect of mixed pixel acquisition is also a known source of misclassification for systems based on pure spectral signatures: the higher the resolution detail, the less mixed pixels would occur with respect to identified typologies. The SM classifier detects classes that are suitable for global to local scale applications as it is pixel based and independent from the geometric appearance of

data. It is still affected by the acquisition geometry that leads to mixed pixels and, although an accurate study of this effect at different resolutions is not part of this work, advantage from the availability of MEA can be taken immediately by using it as an investigation tool toward mixed pixel effects, using its multi-level grid system and display features over areas with low distortion.

With respect to sequential pattern matching alone, which leads to a yes/no response from the search for an exact match, evolution models feature two kinds of tolerance parameters that change pattern detection behaviour. Changing tolerance parameters, a high degree of detection flexibility can be obtained:

- To identify relative position of elements in the sequence along the temporal line, their temporal reference is defined with respect to the element preceding them in time so that the pattern is independent of absolute time references, making it applicable at any point in the time line.

- Each element is designed to detect a particular set of values, and a tolerance set is also admitted to let the user evaluate the incidence of selected values over the result and to divide the results into two levels of confidence. To cope with possible misclassification, values to be ignores can also be specified;

- Besides sets of values, a model element defines "when" those sets are to be found by means of its temporal reference. Since it is common to not have daily acquisitions (for polar orbiting satellites, revisit time is in the order of several days) and to have invalid observations that are not applicable to the domain (e.g. clouds for land cover), there is the need to introduce some flexibility in the pattern to let the model accept data not only at the day referred by its temporal reference: the Time Tolerance (TT) of an element defines the radius in days of a temporal interval, centred on the element's temporal reference, as its sampling window over the time dimension.

- Another option considered for model matching is whether an element will evaluate just the closest available value to its temporal reference or search all the values covered by it. A "persistent" element can be useful in defining a limit to accepted variability over some time window during the search for a match or to require a uniform coverage type.

An algorithm to perform the matching has been defined and a distributed processing engine built to allow its use to search over an AOI and one or more temporal intervals. With the aforementioned extensions, a pattern match can deliver four different results for any given pixel and time reference pair: Perfect Match, Match within Tolerance, Not Match and No data (i.e. possible match but not enough data to test all elements). A match is tested by assigning a position in time to the first element of a model, then analysing observed data as detected by all model elements at their respective temporal reference.

2.4 User Interfaces

Any system foreseeing user interaction has to provide some kind of interface to its users; for software systems designed for visual analysis a graphical interface is an effective choice and can be one of the main factors driving its success in being used effectively and with satisfaction by its users. The interfaces designed for MEA are tailored to its specific purpose, context and user typologies while still presenting well known elements for any Geographic Information System (GIS) application. Three interfaces are defined to provide a different view of the data and access to different functions following the three main user roles.

The interface for users in the "Expert" role provides access to all multi-temporal analysis features of the system, with the aim to aid the user detecting relevant patterns in the thematic data over time. A geographic map browser is provided, common to virtually every two dimensional GIS interface, along with interface elements focused toward the temporal domain, such as a complete profile of a given pixel over time and the dynamic display of Tiles time series. Tree dimensional effects were also considered for some interface elements but they have been discarded as a limiting factor due to their tendency to overlap data elements, thus impeding their full visibility to the user. The layout of the expert interface is designed along the three essential steps of the research sequence, as reported in [3]: an exploratory phase where data is explored to formulate an hypothesis (visual browsing of data, pattern detection and modelling), a confirmatory stage where the hypothesis is tested (automated pattern matching and visual analysis of results, numerical analysis versus external data). Once confirmed, the hypothesis is then made public for others to take advantage of it (a model becomes usable also by "Standard" users). As the interface for users in the "Standard"

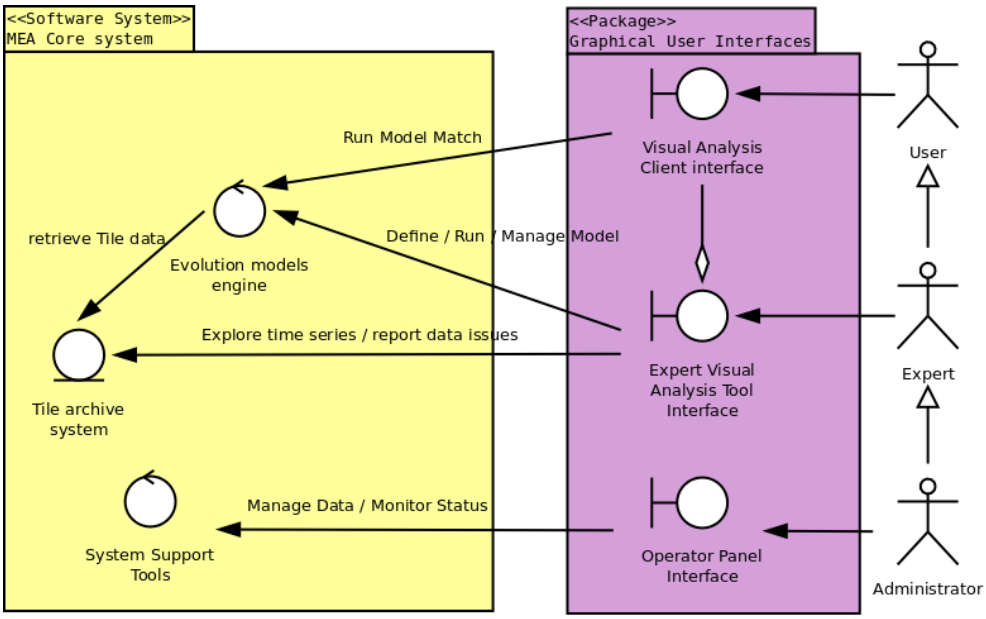


Figure 2.6: Overall diagram of user interfaces with main functions provided. Three different interfaces are defined to provide different views of the system and different sets of functions to users in different roles. These are backed by the Tile archive, the Evolution Models engine and a set of system administration tools.

role coincides with the confirmatory function of the Expert interface (although with reduced options), it becomes seamless to deliver an effective, verified pattern to the user community thanks to the system integration. Meta-data is also associated to each model to define how it has to be used to effectively search through the time series for a modelled phenomena. Finally, management interfaces and cooperative reporting functions are made available to let the user community become a direct resource for system improvement. By reporting issues in archived data its quality improves as its error rate decreases; collaborative effort in pattern validation and modelling is also possible. A top level overview of the MEA user interface system is depicted in Figure 2.6 where main functions of the interfaces are represented as labelled messages arrows.

Interactive analysis, the main function of MEA interfaces, requires high responsiveness to user actions and interface accessibility is another key factor to obtain user satisfaction. Recent developments in Internet technologies make the World Wide Web the ideal application distribution channel, especially for multi-user applications that can take advantage of the proximity of the computing elements engine to the data archives. With the added advantage of immediate application update dissemination by design as the application accessed over the web is always presented to the users in its

most up-to-date version. The Asynchronous JavaScript and XML (AJAX) approach to web applications[24], coupling rich features with high interface responsiveness, offers the opportunity to deliver rich features for interactive exploration of time series directly over the web making them readily accessible with a modern web browser, with very limited resource requirements to the user. The functional diagram provided in Figure 2.6 omits the container component serving the user interfaces and allowing such delivery model: a web server supporting server side programming for the AJAX asynchronous functions. The interfaces are also assessed for their Usability[25] as perceived by a limited set of test users.

2.5 Hardware and performances

Hardware used for prototype implementation of MEA has been sized to deliver a proof of concept system, yet efficient and adequate for use by an initial small community of users in an operational environment. It has been also tailored towards Tile meta-data access performance to ensure high interactivity while browsing data and it is designed to run on commodity hardware with no specialized components.

The reported implementation of MEA has been built in two steps: a semantically searchable catalogue of land cover maps over the entire (A)ATSR archive has been built as a Web Service: at this point we provided basic "image content" search capability allowing queries based on land cover type percentage in single images. Then the evolution analysis system to interactively browse through the temporal domain and search for occurrences of a given sequential pattern in time at pixel level was added; the addition included a second batch of hardware to store the Tile archive and provide a fast meta-data catalogue. Thus, the system hardware comprised two batches of different systems that, paired with reuse and share resource strategy, led to the final configuration of:

- One central storage and controller unit, that has also been reused to host the core web server, hosting both the MEA application and the ASQuLD maps catalogue service. A Direct-Attached Storage (DAS) unit was added with the second batch to provide Tile storage;
- Two Processing Nodes (first batch); devoted to systematic ingestion of satellite data to build the classified maps catalogue, that are also used to host model

matching processing;

- Two Database Nodes (second batch); dedicated to fast meta-data access (employing also SSD technology), model matching processing and also systematic ingestion in addition to the processing nodes.

Performance

The target implementation over limited, heterogeneous hardware resources promoted the design of distributed processing components and a policy to share that hardware among all software components to deliver high performance for burst-load operations addressed to user interaction and continuous processing for data ingestion and background operations. The Linux kernel scheduler offers process niceness level[26] that has been used to give adequate prioritization to the system functions to provide adaptive (prioritized) hardware utilization, according to the following performance priority requirement of each function:

1. Data selection queries for Tile identification have top priority as they provide high responsiveness to interactive browsing and data exploration on the user interface: database functions run at slightly above normal niceness level;

2. Model matching computation must deliver high performance as it provides results to the multi-temporal thematic search, it however depends on the database function and is thus run at normal niceness level;

3. Systematic data ingestion has the lowest priority with respect to the other interactive functions hence, it is given background priority.

During development, all components underwent performance monitoring and analysis aimed at detection of anomalies and system improvement where feasible.

Chapter 3

Multi-temporal analysis system

This chapter provides a conceptual description of the MEA system aimed at realizing a generalised, integrated solution for multi-temporal analysis over geographic sets of thematic raster data. Without loss of generality, the specific case of land cover classification is considered since the system implementation is based on land cover maps. The methodology is aimed at building a comparable set of data in the thematic-temporal domain that, consisting of thematic data (semantically identifiable), is readily understandable by humans so that interactive visual browsing can be effectively employed as an analysis tool. The visual presentation of data to the user is aimed at easing the identification of relevant patterns in the classification over the temporal domain and aiding tools are provided to formally define such patterns as Evolution Model that define an expected thematic behaviour over time at pixel level. The formal definition of patterns enables the definition of a multi-temporal search engine that can be used to test for occurrence of a given pattern in a subset of the data archive. The integration of thematic maps, graphical interfaces with tools aimed at easing pattern detection and a modelling tool for pattern formalisation, allows the execution of interactive thematic searches in the temporal domain that can be performed on the thematic evolution over time of each image pixel in time series of satellite data. The result of such search can itself be effectively represented as a thematic map to be readily accessible for an assessment of the occurrence extent of the modelled evolution.

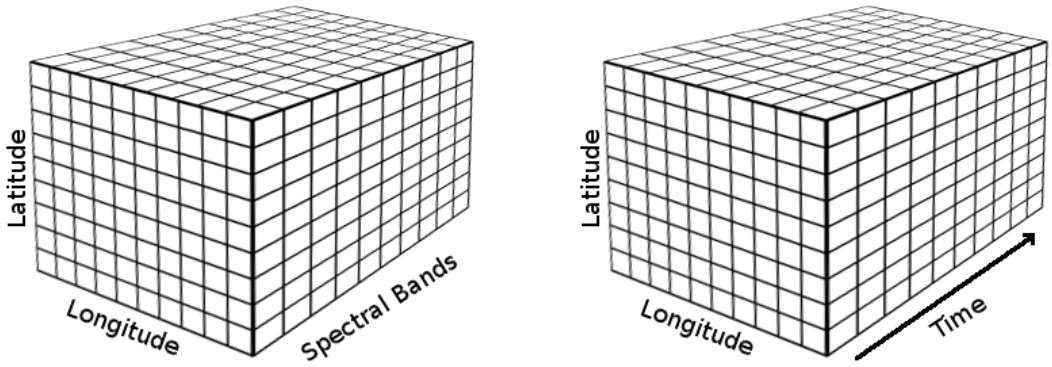
The practical use of the presented approach becomes evident when considering that a given evolution of thematic data over time (such as land cover class) can be used to describe a given phenomenon (such as an agricultural practice, that can be identified by its seasonal cycle or any transitional event such as flooding or deforestation); the particularity of a formalised pattern is to be clearly understandable to the user and its

codification in a machine readable format, combined with the condensed information of a thematic data allows fast searches for occurrences to be performed on-line at the user request. Moreover, the preparation of a thematic archive of satellite data can be prepared for multi-sensor applicability if the data archive is built on a fixed reference system permitting multiple resolution levels.

3.1 Comparable data over time and across sensor

The ideal dataset for multi temporal detection of patterns over time, applicable across sensors, would consists of a thematic dataset that is consistently derived from any EO data type in order to always provide the same class value for the same observed surface feature, independently from sensor specific parameters and atmospheric conditions at the time of observation. Since mixed pixel effect cannot be avoided across different resolution levels, the dataset should also be scale invariant for pixels that present no mixed content at several scales: while the contours of a given feature would present some changes, the core of the feature should be effectively detectable at different resolution scales with the characterization of its thematic-temporal behaviour. With such consistent and robust thematic classification, the outputs of different sensors at similar resolution levels could be merged together to increase data coverage in both spatial and temporal domains. Post classification analysis allows to operate across sensors, provided the classification system is able to detect the same class for the same observed object at different times.

For multi-spectral optical sensors, each image is a co-registered data cube in the geographic-spectral dimensions where, as shown in Figure 3.1(a), each cell/pixel has a specific value in each spectral band. Extraction of a thematic classification map flattens the spectral dimension to a single layer with extracted semantic meaning (e.g. land cover type). Provided that pixels are remapped or co-registered across a series of images of the same area taken at different times, a new data cube can be built in the geographic-temporal dimensions where each cell/pixel value, as shown in Figure 3.1(b), is a feature value understandable by the user at thematic level. Moreover, since different optical sensors may have different spectral bands (in number and position vs. the frequency domain), the classification can be used as an effective system to flatten such differences by providing a single value, representing the thematic meaning of the spectral data. It is thus a possible method to reach sensor independence. Several



(a) Multi-spectral image: a geographic-spectral data cube with observed values in spectral bands

(b) After classification and time series collocation: a geographic-Temporal data cube with thematic classes over time

Figure 3.1: Spectral classification flattens the spectral dimension, collocated time series provide a thematic cube over time.

examples of hyper-spectral image classification algorithms, implemented to provide fast processing of images with high spectral dimensionality over hardware clusters are presented in [27]. With adequate hardware resources, it is then possible to perform systematic classification in a reasonable amount of time over a large volume of datasets. The implementation herein presented is not based on hyper-spectral data and makes use of a single pixel based classification system that requires limited resources to deliver adequate performances for systematic classification. Moreover, as reported in section 6.1.1, faster processing would lead to data locality issues.

A system providing a coherent classification represented by a finite set of (semantic) labels can be the basis for change detection, since variation in the output label for a pixel would indicate a corresponding variation in the observed surface area. To be directly comparable the output set has to be the same for each data source, at a minimum for sensors with a similar resolution level. Such classification provides comparable data in the temporal domain within a given margin, which decreases as adherence to the common classification increases.

It must be recalled that radiometric calibration, i.e. digital numbers to radiance or surface reflectance conversion, is a key factor for quantitative analyses of multi-temporal images [9] and that primary parameters, like the spectral surface reflectance can be used to produce a standardised characterization of soil and vegetation[28].

Therefore a spectral classification system working at pixel level with calibrated observations can be used to build the thematic characterization and its output maps processed for comparison in the temporal dimension.

One of the disadvantages of post-classification methods is the considerable amount of time and expertise usually required to produce the thematic classification. The availability of classifiers that consolidate the ever increasing knowledge about observed spectral information enables systematic production of thematic maps within relatively small time frames that can build thematic layers of information. Applicability to multiple sensors allows increasing the temporal density of data, providing an improved data base. However the strong dependency of the final accuracy on the quality of the single maps remains an intrinsic disadvantage of post-classification methods.

Among other spectral classification systems, SM characterizes the vegetation index and other derived properties as tone and brightness and provides as output a set of classes. According to its product specification[29], SM is an unsupervised classifier that performs multi-spectral analysis on calibrated Top Of Atmosphere (TOA) physical values to generate a preliminary classification map over the main categories of Vegetation, Bare soil / Built-up, Water / Shadows and Snow / Ice. Furthermore it has built-in detection of clouds and provides indication of outliers that do not fall within its classification scheme. Each main category is provided in a discrete set of intensity scale that can be directly mapped to a quantization of the vegetation index crossed with other spectrally derived properties such as brightness and relevant signature characteristics, for a total of 56 classes. SM classification technique takes inspiration from the decision tree classifier published in [30] that uses prior knowledge on spectral response to directly provide a semantic category for a given observation. SM software implementation is named Enhanced SOIL MAPPER (ESM) that extends the supported set of input sensors providing an improved classification system and cloud detection technique; further details are provided in an internal report of the providing company[31]. One key aspect of ESM is its standardised output set allowing direct classification comparison among thematic maps derived from images of different sensors.

There is evidence of the suitability of features extracted by quantities derived from spectral signature analysis like the NDVI for effective multi temporal characterization of agricultural phenomena [12]. We assume that any phenomenon changing the spec-

tral signature can be detected from comparison of the output of a spectral classifier, provided its intensity is enough to be detected in a given time frame, hence by characterization of the output classes (either observed or expected) over time provides a model that can be matched to the geo-temporal feature cube to detect the modelled phenomenon. The implemented MEA system builds on time series of homogeneous thematic data obtained by processing satellite images using the SOIL MAPPER® classification system that processes data coming from different sensors in a consistent way, generating maps of land cover classes with consistent semantic meaning. That classification system is thus believed suitable to permit multi-temporal and multi-sensor applications.

3.2 Comparable data locations across sensors

The availability of a consistent dataset with respect to sensor and time is not enough to achieve good accuracy in multi-temporal analysis. The geographic reference of the observed surface has to be collocated in the time series of images: the effect of discrepancies in the pixel locations is more evident toward high resolution images where a slight shift in the observed features would produce highly spurious comparison outputs[9]; geometrical rectification, topographic correction over mountain areas and image registration are major steps in any change analysis project.

One of the most common problems in multi sensor and multi temporal analysis is the registration (re-sampling) of all images not just one against another one in a limited time series, but placing all of them into a common reference projection. While this problem is far to be effectively solved by the approach adopted by MEA, the only methodology adequate for re-sampling categorical data is adopted, that is the nearest-neighbour algorithm which does not alter the pixel values, and an Earth fixed grid is defined to support information geocoding.

The Earth Fixed Reference defined in MEA derives from the definition of a regular, uniform angle sampling mesh over the geographic coordinates system (Lat. / Lon.), an example of the resulting grid mesh over the Earth surface is shown in Figure 3.2. The first layer of the multi-level DGGS sets the sampling angle at 1/256th degree, that is grid level 0. Each further level doubles the sampling rate in both dimensions (e.g. 1/512th degree at level 1 and so on). Cell values are provided by re-sampling the thematic maps with the nearest-neighbour algorithm. To give all pixels of the original

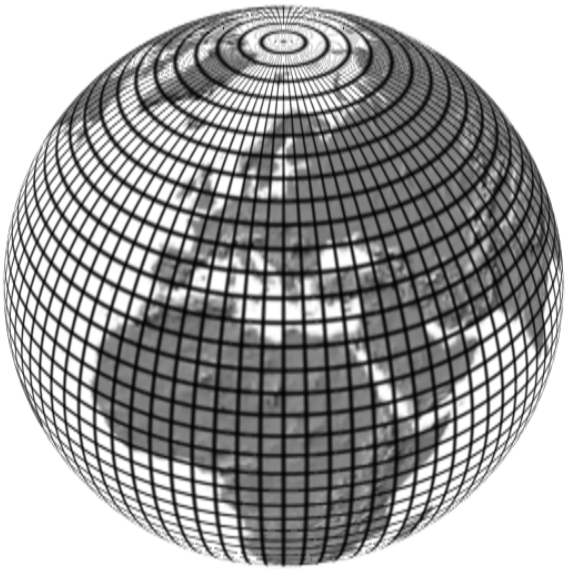


Figure 3.2: Discrete Grid Tiles on the Earth surface. Image background is an inverted color Shaded-Relief image from NOAA's National Geophysical data center[32].

image a chance of being remapped into a cell, the sampling rate is choosen such that a cell size is at most half of the Ground Sampling Distance (GSD) of the sensor; this is obtained by considering the equatorial ground pixel resolution that results from the sampling rate, which is the maximum area represented by a pixel in the chosen grid. By doubling the sampling rate at each subsequent grid level a congruent grid system is built, with respect to the coordinate system, that allows re-sampling data from any given sensor by selecting the closest level to its reference GSD, as shown in Table 3.1.

Grid Level	Reference GSD	Equatorial pixel resolution	Samples per degree	Supported sensors
	m	m	#	
0	1000	434,84	256	(A)ATSR, MODIS
1	500	217,42	512	MODIS HKM
2	250	108,71	1024	MODIS QKM, MERIS
3	125	54,36	2048	Landsat TM TIR
4	60	27.18	4096	Landsat ETM+ TIR
5	30	13,59	8192	Landsat TM/ETM+ MS
6	15	6,79	16384	Landsat ETM+ Pan, SPOT5, AVNIR-2

Table 3.1: Discrete Grid System levels and supported sensors

Having defined a congruent, regular square partition of the simple cylindrical representation of the Earth allows grouping grid elements together easily in fixed size tiles of 64 by 64 cells called Tiles. At level 0 each Tile covers one quarter of degree in

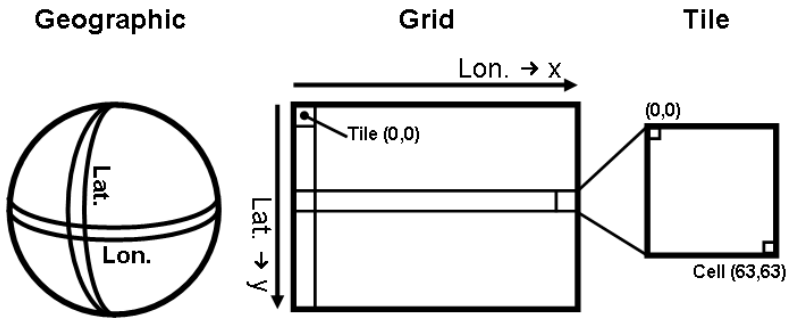


Figure 3.3: Addressing of cells: from geographic Lat/Lon to Grid Tiles. Inside Tiles cells are addressed as in a matrix.

both Latitude and Longitude. Tiles can be stored in raster format without requiring geo-location information as it is directly encoded in their definition and can use an addressing scheme based on integer grid coordinates, instead of the real-valued geographic coordinates. Grid coordinates are defined by numbering Tiles locations in two dimensions with x mapping to Longitude and y mapping to Latitude; origin of the grid coordinates is set with Tile zone (0,0) at Lat/Lon (90,-180). Within each Tile, cells are in turn addressed by their position in the two dimensional space defined by Tile coordinates with origin in the upper (northern) left (western) cell; the addressing scheme is shown in Figure 3.3.

Multiple grid levels of the congruent grid partition Tiles in square sub-tiles, as shown in Figure 3.4. Each level halves the area covered by its cells on the Earth cylindrical projection, with respect to the previous level, squaring the number of Tiles at each level.

3.3 Evolution Model concepts

Many change detection techniques for specific change detection projects have been adopted, as summarized in [9]: their suitability however depends on the specific project and the selection of the best one is not easy in practice. The approach proposed for MEA, being based on a post-classification dataset, provides a generalized system for change detection, suitable for interactive definition and application of search patterns over considerable amounts of data. The evolution model aim is to permit the definition of a pattern of changes in the temporal domain that removes the requirement for similar phenological states across images as it variation over time is one of the factors enabling

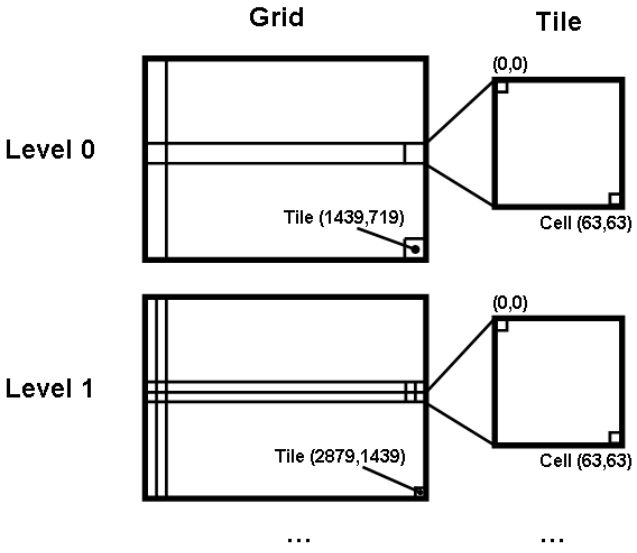
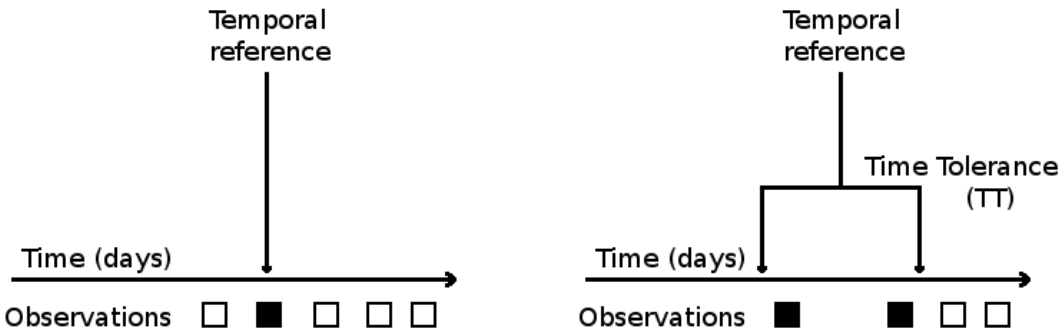


Figure 3.4: Multiple grid levels: each level halves cell size on the cylindrical projection, with respect to the previous level. The number Tiles squares at each level wile the number of cells in each tile does not vary.

the distinction of features for their identification with multi-temporal analysis. This section extends and details the concept of Evolution Model defined in [33].

In the MEA context, an Evolution Model is used to define the thematic behaviour of a feature, at pixel level, over time in a form that is easily understandable by the user and that can be readily used to quickly search time series for occurrences of the modelled evolution. It allows defining a sequence of expected sets of land cover classes along the temporal line to characterize the modelled phenomena. An Evolution Model Element, or simply Model Element, defines an association between a given set of class and a temporal reference thus defining "when" that set is expected to be observed, as shown in Figure 3.5(a). Since the time dimension of the data cube is likely to be irregularly filled due to satellite revisit time and possible cloudy acquisitions, a single day is not suitable as a sampling reference to allow finding a valid data to match the element's set. The TT of an element defines a temporal extent centred on the temporal reference: observations falling within the time tolerance are covered by the element as shown in Figure 3.5(b). The temporal coverage of an element is thus twice its TT plus one day (the temporal reference itself).

An Evolution Model is in turn defined as a sequence of Model Elements, positioned along the temporal line by specifying the distance from their predecessors. To connect the model elements in a sequence that composes an evolution model, a temporal parameter is associated to all elements except for the first one: the Time Since Previous (TSP) defines the number of days a given element follows its preceding one.



(a) Temporal reference of a Model Element: defines the point in the time line where to search for thematic data.

(b) Output message element

Figure 3.5: Temporal parameters of a model element
The temporal reference of a model element defines its sampling point along the temporal line, the Time Tolerance defines a sampling interval.

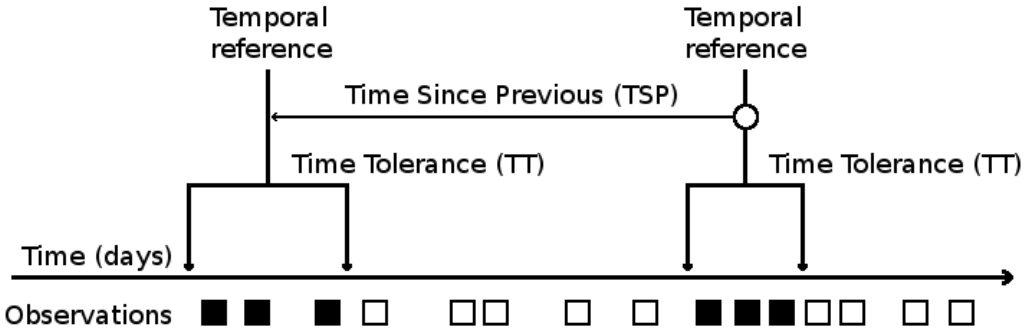


Figure 3.6: An Evolution Model is a sequence over time of Model Elements, with the TSP parameter specifying of how many days it follows the previous one.

In that way the evolution model defines a sequence of expected classes in defined moments thus allowing the definition of the expected classes at different times.

In the LULCC domain, an evolution model defines a sequence of land cover classes that are expected at given times to model the land cover evolution over time. As a sequential pattern, an evolution model can be quickly matched with actual time series data at the pixel level, to determine if that data matches the modelled evolution pattern as depicted in Figure 3.6.

The suitability of change patterns over time to identify relevant phenomena has been confirmed in agriculture where the evolution of the NDVI allows identification of different kinds of crop lands, two of them are shown in Figure 3.7. By defining a sequence of elements that samples the time line at given intervals, representing the

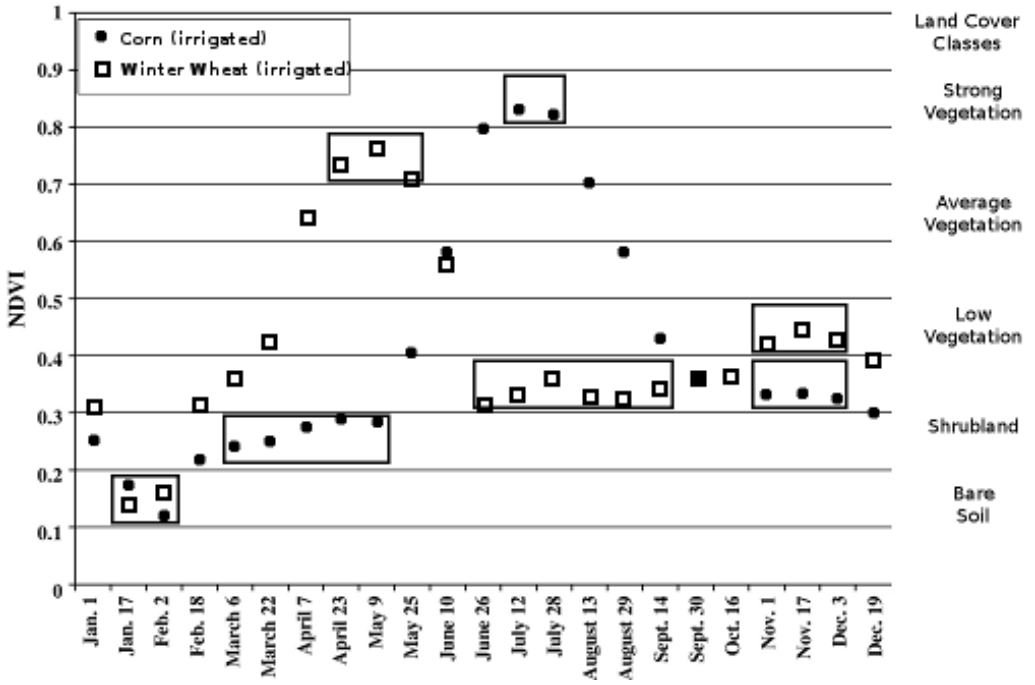


Figure 3.7: NDVI profiles of two specific agricultural crops as observed in Kansas by MODIS with candidate temporal intervals for model elements highlighted. Adapted from [12]

most relevant state of the crop field expected at a given time (with respect to the previous relevant state) an evolution model characterizing crop lands can be easily defined, given the set of land cover types is sufficiently diverse to allow specification of all the relevant states over time. Rectangular blocks in Figure 3.7 show possible candidate time periods for the definition of related model elements, since they present stable classification (assuming that the classification on the right side is assigned, depending on the NDVI, including intermediate levels).

Besides the definition of well defined seasonal phenomena, EM can be used to model transitional phenomena, by defining a model with only two elements (one before and one after the transitory event to be modelled) and even single element models can be used to search information as they can serve to find stable coverage or assess frequency of occurrence of a given class. Metadata is associated to evolution models (i.e. model name, model type, model description, associated keywords, area of applicability, applicability to grid levels and category) to provide information about the model and the modelled information.

Having defined the temporal side of a model element and its parameters, we now define the thematic side of the pair: each model element defines three sets to which

thematic values (classes) can be assigned: Main set, Tolerance set and Ignore set. These sets define the behaviour of the element when tested with time series data falling in its temporal coverage. Any number of classes can be assigned to each set provided the sets are disjoint (no class is allowed to appear in any two different sets).

A Model Element can be tested for a match with actual data across the time dimension; for any given sampling location (Grid Element), four possible outcomes can result from the test:

Match if all observations considered within the temporal coverage belong to the Main set;

Match within tolerance if all the observations considered within the temporal coverage belong to the Tolerance set or the Main set and there is at least one observation belonging to the Tolerance set;

Not Match if any of the observations considered within the temporal coverage does not belong to any of the element's sets, that is an unexpected observation;

No Data if there are no observation within the temporal coverage or all observations considered within the temporal coverage belong to the Ignore set[1].

In other words, when testing an element for a match: data in the Ignore set is ignored; the Main and Tolerance sets define expected classes to give a match (possibly within tolerance) while any unexpected data results in a failed match. An element can also be marked "persistent" to influence how observations within its temporal coverage are considered: if persistence is set, all observations are considered as described above, otherwise only the closest element to the temporal reference is considered to determine the outcome.

Since an EM is a sequential pattern of elements, the outcome of a test on the model can also result in four outcomes, depending on the outcome of its elements, as follows:

Match if all elements returned a match;

Match within tolerance if one or more elements returned a match within tolerance while all others returned a match;

[1]The availability of the ignore set may lead to a change in the intuitive meaning of the "No Data" outcome: if many valid classes are assigned to that set (e.g. to search only for a specific class) the "No Data" outcome will be returned if the expected data has not been observed to confirm the match, while many observations could have been ignored that would have led to a "Not Match" outcome.

Not Match if any element returned a not match;

No Data if at least one element returned no data, while no other element returned a not match.

In other words, when testing a model for a match: if one element fails the match, the entire model fails, otherwise, if there is no data to test an element, the whole model cannot be tested. Finally, the kind of match returned will be within tolerance if any element so resulted or perfect match if all elements resulted in Match.

To summarize, we define an Evolution Model as a sequence of Model Elements connected together over time by the TSP, that defines the number of days an element follows its preceding one in the time line. The temporal reference of each element defines a day along the time line where its Main class set is expected. The temporal reference can be extended to become a temporal interval by means of the Time Tolerance that allows both to cope with the possibility of missing data along the time line (such as cloudy acquisition or gaps due to satellite revisit time) and to extend an element's class expectancy over arbitrarily long periods by setting element persistence. Besides the Main class set, yielding to a perfect match when testing a model, a Tolerance set is also defined to provide two confidence levels for a matching outcome. Finally an Ignore set is defined to specify classes that are not seen by the element during a test for a match.

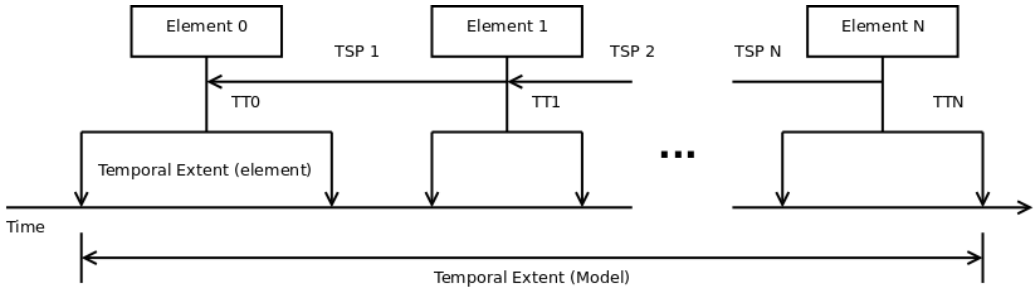
The proposed evolution model matching derives from a form of change analysis that, according to the topology summarized in [9], falls in the "classification" category, in particular, it is a form of the commonly used post-classification comparison, combined with pattern matching to take advantage of an integrated GIS environment to provide interactive comparison at an arbitrary set of time intervals to detect a pattern of changes over time. This integration allows realizing an automated multi-temporal search tool applicable for fast, on-demand time series analysis. The pattern definition allows flexibility in both temporal and thematic domains with tolerance parameters; it is however limited by the rigid temporal bounds between adjacent elements set by a fixed temporal constraint. The TSP allows defining rigid temporal distances between elements which has the advantage of being well suited for modelling phenomena presenting low temporal variability but is less effective when searching for evolution patterns with high variability in the relative duration of their characterising elements.

The applicability of the defined models to a user readable categorization has the immediate advantage of allowing effective intuitive pattern definition and the direct assessment of the result of a test for match as the information is readily readable by expert users but can be also understood by end users, depending on the semantic level of the thematic categories. The thematic categories can be seen as words and the time of their occurrence as their context so that pattern matching can be effectively employed to query data at the same semantic level for known contexts and can also be searched for unknown patterns. The aim of the defined evolution models is also to allow defining sequences in time that are readily understandable by the user and can effectively define derived phenomena.

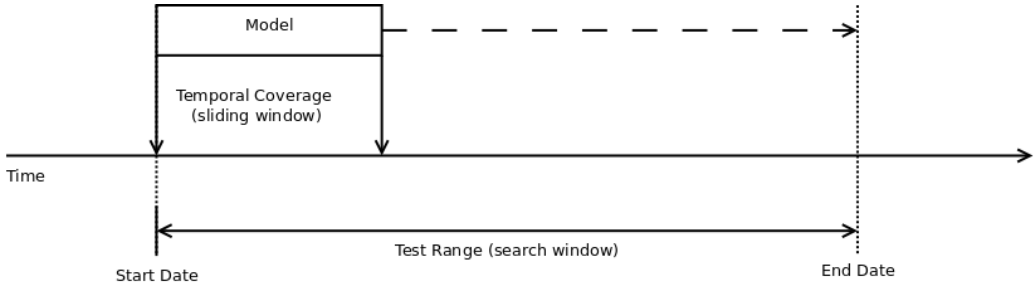
The strong dependence of the achievable accuracy from the accuracy of images at each date cannot be removed from a post-classification methodology; it can however be mitigated by the availability of many observations (theoretically a daily coverage could be obtained, depending on the atmospheric conditions). Two model features help mitigate the problem. The class Tolerance set provides for a different level of confidence allowing inclusion of values that are likely due to misclassification or to accept some margin in pattern definition. The Ignore class set allows to specify classes that are not relevant to the detection of a given pattern and can be also used to "remove" noise by not letting a model element consider classes likely subject to misclassification in the model context. Conversely, models targeted at detection of possible misclassification in the underlying classification to search for single image quality issues can be defined.

Another strong dependence of the matching system from the underlying data is on the expressive power that can be obtained with the models with respect to the underlying thematic set: the richer the set of values, the more detailed the models can be. For example, a thematic classification over the NDVI would allow detection of phenomena that can be entirely defined by variation of a vegetation index over time. The inclusion of water as a possible value, lets the model define phenomena depending on both vegetation level and presence of water, such as rice fields or flooding events.

An EM test for a match is done by setting the temporal reference of its first element, thus setting all other references consequently (as defined by their element's TSP). We can call this a daily test since it is done referencing the first element to a single date. The temporal extent of the model is equal to the sum of all its element's TSP plus the time tolerances of its first and last elements, as shown in Figure 3.8(a).



(a) The temporal coverage of an evolution model is given by the sum of its element's TSP values + TT 0 + TT N



(b) When testing for matching over wide temporal interval, the model is tested daily as a sliding window over the test range

Figure 3.8: An evolution model defines a temporal extent where data is tested

When searching for a given evolution pattern in a wide temporal range (defined by start and end dates), the evolution model is tested iteratively for each day in the test range, starting from the start date and ending when the sliding window defined by the temporal extent of the model reaches the end date, as shown in Figure 3.8(b). As the model is tested over several days, the final output of a search for a given grid element over the test range can give one of the four values:

Match if at least one daily test matched over the test range. Matches are counted as a modelled evolution can occur several times within the test range. Moreover, when a match is found, the next test day is moved forward in time of an entire sliding window to avoid redundant matches (otherwise, a match, once detected, will persist until the sliding window passes over it leading to redundant count);

Match within tolerance no daily test resulted in a match and at least one daily test resulted in a match within tolerance (once a match within tolerance is found, the search continues on the next day and other matches within tolerance are not counted);

No Data if no daily test resulted in any kind of match and at least one daily test

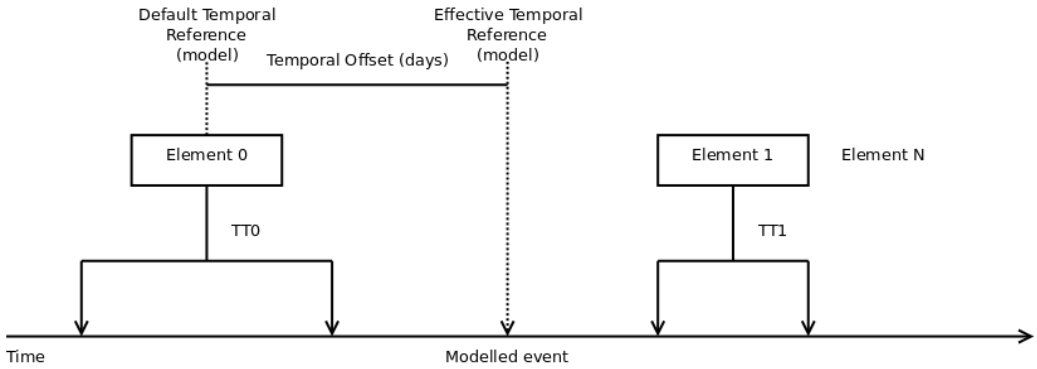


Figure 3.9: An evolution model designed to model a specific event occurring within its temporal extent may specify a temporal offset to provide information of the event location in the time dimension, with respect to its first element temporal reference.

resulted in no data (this avoids false negatives due to missing data in given intervals of the test range that would have led to a match if present).

Not Match if all daily tests resulted in not match for the entire test range: in that case there is certainty of a negative match;

Finally, the evolution models can be defined to detect some specific event characterised by a change that is not necessarily covered by the entire model or practical to be referred as the temporal reference of the first element when the matched test occurred (default behaviour). To allow specification of the precise date of the modelled event with respect to the temporal reference of the first element, an optional temporal offset can be defined for a model that can be used to refer temporal parameters to when interacting with the user (interested in when the modelled event occurred more than in when a model test matched). The concept of the temporal offset is shown in Figure 3.9.

3.4 User interaction

The evolution models permit to query the data stored in the MEA system in terms of the evolution of its thematic content over time as it is observed at specific sampling frames. To run a model search, user input is required and results must be made accessible to users. Moreover, functions for basic browsing of the data archive and visualization tools for time oriented data analysis are needed to provide an integrated GIS platform. This section presents the overall architectural model of the user interfaces for multi-temporal analysis that provides user interaction with the system.

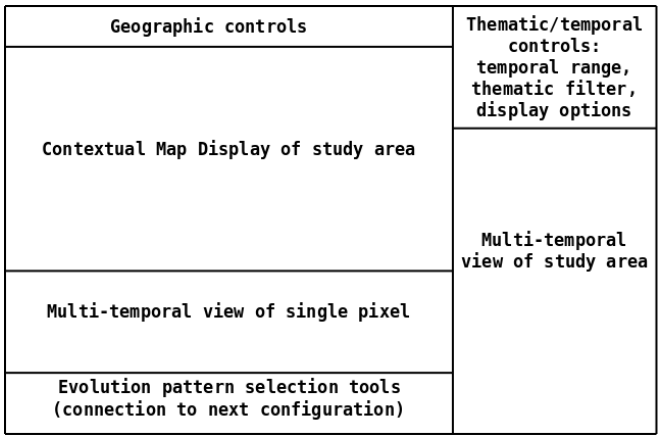


Figure 3.10: GUI outline for exploratory analysis.

With respect to data analysis, the user interface is designed following an underlying work flow to assist the user in the steps commonly associated with the research process; three essential phases are reported in [3] and to each one is associated a specific configuration of the user interface:

- In the exploratory phase, where data is explored to formulate an hypothesis, the interface provides interactive visualization tools to explore the data archive, focusing on the temporal domain and providing also spatial context to data visualization (world map background) to allow spatial pattern identification. In the exploratory configuration the interface aims to promote spatio-temporal visual pattern identification and should also aid in definition of a model corresponding to an identified temporal evolution. Interface configuration for this phase is outlined in Figure 3.10;

- The formulated hypothesis takes the form of an evolution model to be tested over the data archive, the modelling part of the exploratory phase is associated to a specific interface that graphically aids the user in the definition of the model. In this configuration the focus is on model definition and both reference to observed data and complete freedom in model definition are provided. A pattern can be modelled from scratch to define an expected thematic evolution for a known feature. Interface configuration for model editing is outlined in Figure 3.11;

- The confirmatory phase follows the definition of the model and consists on its use to run a search over subsections of the data archive to automatically detect occurrences that can be compared with expected results. Results themselves are

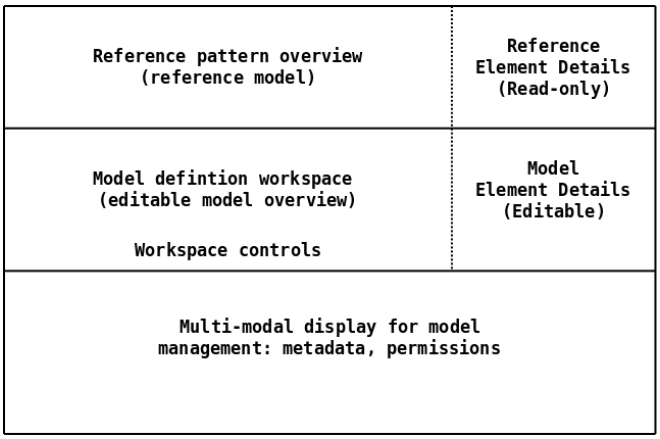


Figure 3.11: GUI outline for model editing.

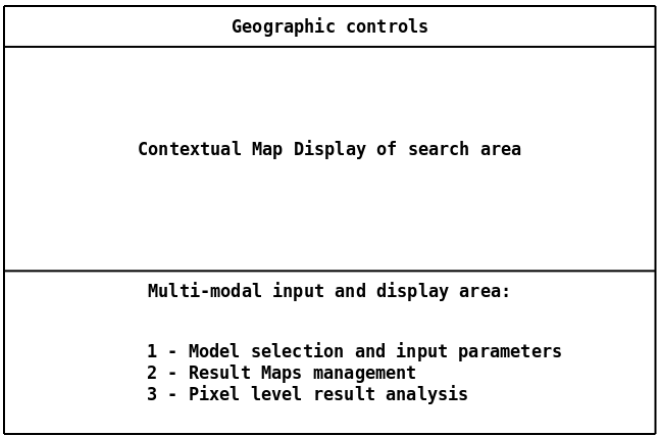


Figure 3.12: GUI outline for model matching (thematic evolution search).

displayed as thematic maps to provide immediate readability and to foster spatial pattern identification in the occurrences. In this configuration the interface provides results analysis tools and an option to connect external data sources or processing system by exporting the results for further analysis. Interface configuration for this phase is outlined in Figure 3.12;

- The publication phase, consisting in making confirmed hypotheses public, is provided by the access control model that allows publication of verified models. In the context of this system, publication means to grant access to a wide range of users that are not interested in the model contents or definition but in its use as an effective search tool to examine the data set at a higher semantic level (such as a new thematic class, derived from temporal evolution of the base dataset). As the interface for users in the "Standard" role coincides with the confirmatory

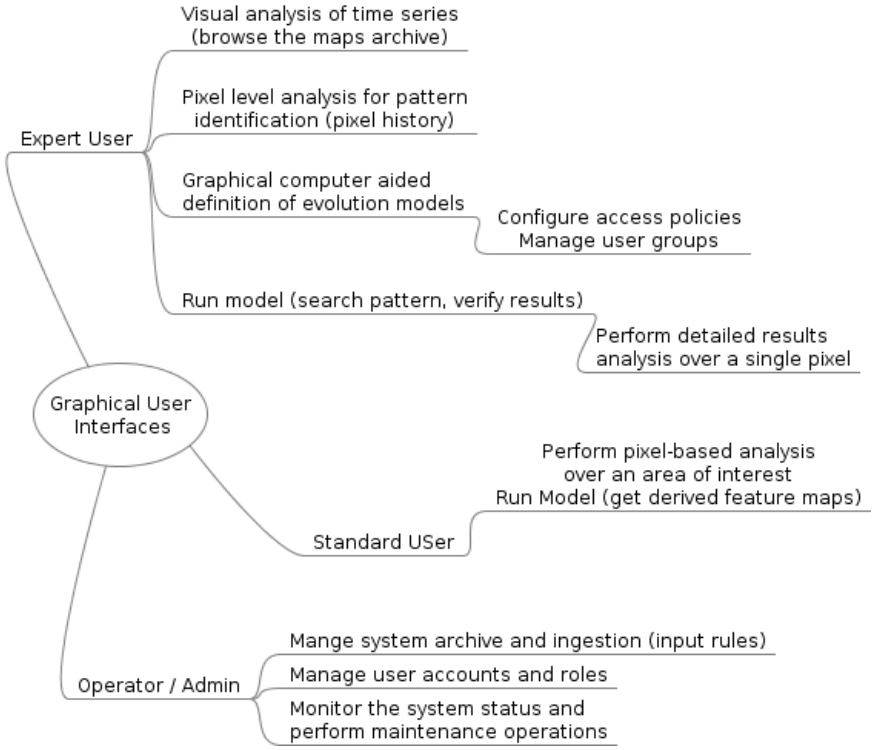


Figure 3.13: Tree view of the interfaces with associated functionality and user need.

function of the Expert interface (although with reduced options), it becomes seamless to deliver an effective, verified pattern to the user community thanks to the system integration. Meta-data is also associated to each model to define how it has to be used to effectively search through the time series for a modelled phenomenon;

The integrated approach allows to perform change detection and feature / event identification putting together post-classification, GIS and visual analysis principles to give a potentially insightful thematic view on satellite data archives and provide a thematic search engine that can be used to query time series for given pattern over time in the data set. A complete overview diagram of the features provided to users of the MEA system through its interfaces is provided in Figure 3.13.

To provide pixel level multi-temporal display, an hybrid graphical method has been defined, taking advantage of the intuitive mapping that can be done between thematic information and colours, commonly used to obtain false colour images. Two of the graphical methods for EDA, defined in [4], have been adopted to build what we call the Pixel History graph (PXH) which is basically a stem plot like representation of

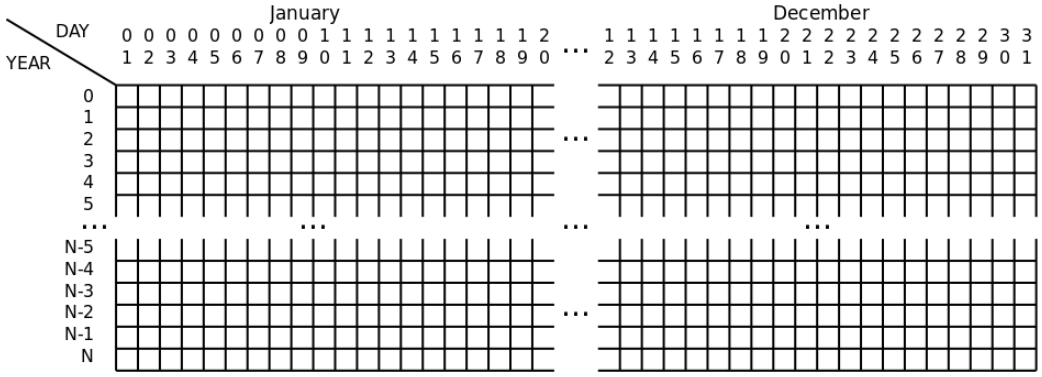


Figure 3.14: The pixel history graph: the profile of the thematic pixel evolution over time is displayed in a matrix that eases seasonal and yearly pattern identification. Each cell is filled with a colour representing the thematic class observed for a the same area at the time identified by year-day crossing.

(land cover) observations where the stems are the years for which some data may be available in the archives (it is potentially covered by a data source) and the leaves are uniformly sized boxes, coloured according to the observed value and the absence of data is considered a valid observation for plotting purposes that is assigned the plot background colour. The result is a matrix that displays the class observed at any given day in rows, each row representing a year and each year presented below the preceding one; year stems are chosen to ease identification of phenomena following yearly cycles that become clearly identifiable along rows while the identification of patterns across year is also eased by consistency of day of year along along column, to ensure alignment to calendar days, a background coloured box is also added corresponding to February 29 for non leap years. That design makes the PXH also comparable to multiple run charts for categorical variables over a year, since it is a multi-row plot on the temporal dimension. The PXH visualization matrix is outlined in Figure 3.14. The Pixel History graph can also be used to select identified patterns by selection of cell ranges to be used as a basis for evolution model definition. The other elements of the interfaces are also oriented toward the specific goal of the system and are defined in detail in chapter 5, where examples of Pixel History graph display of actual values are also presented.

3.5 Access Control

Along the access control model introduced in 2.1, which is centred on object ownership, use groups and simple level of authorization paired with user roles, the following access control policy have been defined for the MEA system. By definition any unregistered

user accessing the system, if allowed to do so, is assigned the standard user role for the duration of his session and some additional limitations may be applied to him as an unregistered user. Unregistered users are also by definition members of the system group "Everyone" while registered ones are members of the "Registered" group: those system level groups are defined to allow assignment of permissions to a broad category of users without the need to list them in a user group. Furthermore, each user can belong to one and only one of the defined user roles: Expert, Standard and Administrator.

Having defined system level user roles (used to present different views of the system and its functions, thus regulating access to the system interfaces), we can now detail the authorization policy with respect to user provided access to controlled entities (i.e. Evolution Models): that model is based on the two essential concepts of ownership and user groups, according to the following principles:

- Three permissions are defined: read, edit and run (taking from the Unix read, write, execute flag set) and are granted to user groups (not to individual users);
- A set of operations available on models is assigned to each permission label as listed in 3.2;
- Users are grouped into user groups where each group has an owner (by default its creator) who is the only user (besides administrators) authorized to add or remove users from that group;
- Each Evolution Model has an owner (by default its creator) who is the only user (besides administrators) authorized to manage the model and to grant or revoke access permissions for it to user groups.

Following the aforementioned principles, any user group may have different permissions for different EM so that a project may have its own groups to manage who can edit models related to that project or thematic collections of models may be made accessible according to thematic user groups to ensure their prompt accessibility. An approach to allow administrative delegation over user groups is also proposed by granting Expert users the permission to create new user groups without requiring action by an administrator. This approach has been chosen to ensure autonomy to (potential) model owners in deciding who is going to access their models and how. This approach

Operation	Ownership		Group Permission			User role	
	Model	Group	Read	Edit	Run	Expert	Admin
Create model						V	V
Change model ownership	V*						V
View/copy model	V		V				V
Edit model	V			V			V
Apply (run) model	V				V		V
Publish model	V*						V
Restore published to editable							V
Delete Model	V**						V
Withdraw published model							V
Create group						V*	V
Delete group		V					V
Edit group		V					V
Assign group permission	V						V
Change group ownership		V*					V
Change group membership		V					V
Add users to the system							V
Delete users							V
Manage users							V

(*) revocable semi-administrative operation;
(**) actually a "hide element" operation.

Table 3.2: Permissions reference matrix

of granting semi-admin privileges to Expert users also allows to ease management of permissions at project level. Suppose some project requires creation and use of experimental models for a given study; permissions can be easily managed by Experts involved in the project by creating a user group for that project to effectively grant user permissions, without requiring an administrator to intervene.

Besides user defined groups, the applicability of that same model to manage permissions for the entire user community and the general public, two default system level groups are also foreseen: "Everyone", that includes any user accessing the system and "Registered", comprising only registered users: by granting specific permissions to these groups a model owner can make his model publicly accessible.

Figure 3.15 illustrates and example of group assignments and permissions for models, clarifying a possible scenario.

Supposing that User 1 is the owner of all the three models, the following applies [2]:

- Owning them, User 1 has full permissions on model 1, 2 and 3, regardless of group permissions;
- User 2 can: Read and Edit models 1 and 2 (group permissions and is also member

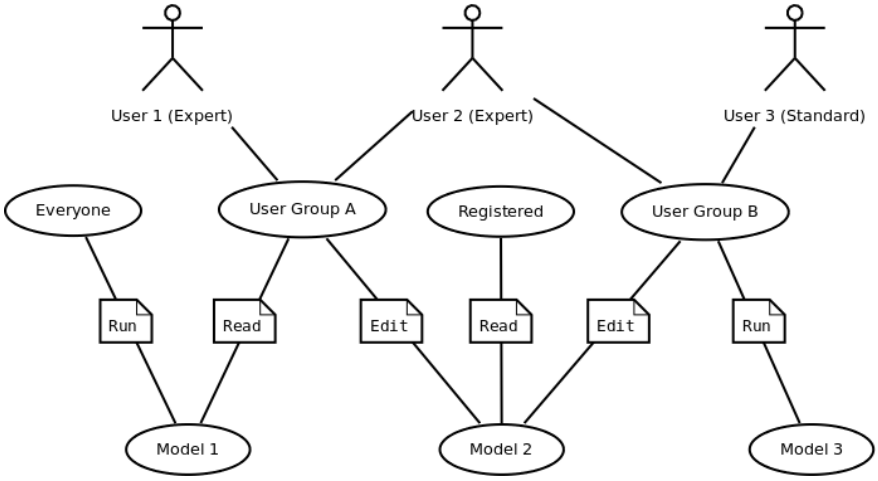


Figure 3.15: Authorization policy example

of the Registered group); Run models 1 and 3 (due to Everyone system group);

- User 3 can: Read model 2 (as a Registered user) and Run model 1 (as member of the Everyone group) and model 3 (for group permission).

The expert user who defines a model becomes its owner and is the main referee for that model. The owner grants permissions to other users (with respect to that model) and is the only user that can decide whether or not to mark its model as stable or make it accessible to the user community. During its development stage, an EM can be accessed only by expert users that are developing it; when model has reached a stable point where it can be effectively used by the user community it can be published: by publishing a model, the model owner creates an immutable copy of that model that can be applied by any authorised user. Once a model is published, only an administrator can remove it from the available models.

3.6 Data flow automation

Any system for data processing that is designed to build an integrated archive by processing large data sets should provide a good degree of automation in its data ingestion process such as standardised access to its data source and provide the possibility to

[2]To be noted that the different role-based system views still apply. Therefore, even if User 3 has group permission to Edit model 2, he cannot do so since editing features are accessible only to "Expert" users; an administrator has to assign User 3 to the "Expert" role in order to actually allow editing of model 2. Similarly, an anonymous user accessing the system (if allowed to do so) can only apply model 1, since he is considered a Standard user and assigned by default policy to the "Everyone" group.

define control rules for conditional processing. As a complex system it should also provide an administrative view allowing to monitor its status and to perform maintenance operations in a safe environment, that is less error prone compared to direct administration not constrained by an interface view and executed manually instead of through the use of automated functions. For the MEA system, the most relevant custom solutions relate to data ingestion control and to the definition of a data provider interface that permits interoperability and some degree of independence from the data source. Such independence provides readiness for multi-source support and applicability to different thematic domains by allowing to connect the ingestion system to any other data provider that implements the defined interface.

To drive the automation of the data ingestion, a set of control rules has been defined and a work flow with corresponding states and state transitions / actions defined for the ingestion activities, making the controller a finite state automata. Since the focus of the system is on a large temporal domain, any input archive is conceptually divided into processing units of one month, hence input rules can be defined for each month. The geographic extent of the operations is set to cover the whole globe: even if the interface defined with the data provider allows for sub-region definition no interface element is defined for inputting this parameter that can be manually set in a specific implementation to reduce the geographic coverage of the system. The set of input rules correspond to the background filled states in Figure 3.16 and can be set by the operator. A month is locked when the system performs operations on it and its status is shown along with statuses where the system awaits user input that cannot be set by the user (Processed and Finalised).

With the proposed set of rules the data archive can be conveniently managed, particularly in case of updates to the source data set or changes in the ingestion policy (e.g. filters applied to thematic data) when automated re-processing can prove useful. Moreover, in order to avoid the system marking months as processed when not enough data is still available from the maps provider, a threshold value is associated to each data source. If the number of maps available from the provider is under the threshold, the month will not be marked as processed, regardless of new data availability. New data is intended as data that has not yet been ingested into the system (without reference to the acquisition or processing time). To provide an overview of the data archive that is immediate to understand and practical to control with a graphical

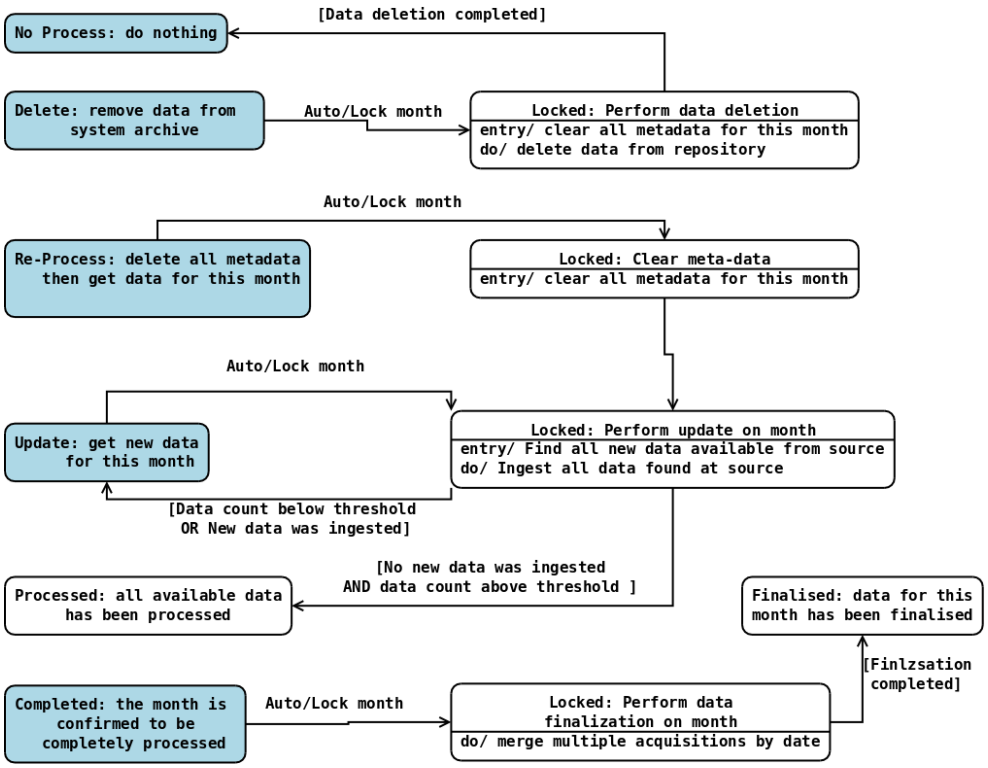


Figure 3.16: Automated data ingestion rules and control flow: the normal flow would be to set a month to be updated, then an operator must confirm its completeness from the processed state: then data finalisation operations are performed. Deletion and Re-processing operations are also automated. Background filled states can be set directly by the operator. When a month is locked its state cannot be changed by the operator.

interfaces, a matrix display is defined with years as rows and months as columns so that information about processing status of each month can be displayed in a compact form. Once again the use of thematic colouring can be effectively employed by associating each status to a colour, providing a compact and immediate overview.

To obtain automation in data flow when different systems are involved, the key requirement is to ensure the ability of such system to exchange information in a way that allows them to effectively use that information. For our implementation, the automated data ingestion system has been designed to interoperate with any system implementing the Classified Maps Provider interface, detailed in section 4.2.2. That interface has been designed and implemented to update the ASQuLD service, that provides the building block of our moderate resolution implementation: multi-sensor and multi-product support has been added to the basic catalogue interface, resulting in the interface described in [34]. Provision of an equi-rectangular projected product, identified by standard nomenclature in the interface, enables direct map ingestion from the system (direct Tiling of the source map). This allows management of geo-location at data provider level: an obvious advantage considering that it is a highly sensor dependant operation. The definition of an interface at map provider level fosters readiness for multi-domain applicability, obtained connecting a different thematic map provider; it also enables multi-sensor support to the administrative tools as each sensor can be equally managed in terms of its processing rules and its data accessed in an interoperable way.

Chapter 4

Implementation: building the data stack

This chapter describes the data processing system implemented to process moderate resolution data. It has been designed and implemented already with multi sensor and multi resolution extensibility options for components allowing such design. A distributed computation system has been implemented to process an entire satellite repository, holding around fifteen years of data at the time of writing (early 2011). Moreover, as new data is sent by the satellite it can be automatically processed and added to the system data archive. This chapter does not provide a complete detailed description of the system, it is focused on systems integration and performance aspects.

4.1 Along Track Scanning Radiometer

Along Track Scanning Radiometer (ATSR) is a typology of instruments that collects observation of the Earth surface at one kilometre spatial resolution from a satellite platform. The first of these instruments is ATSR-1, launched in July 1991, that we do not use for this implementation due to a lack in visible spectral bands (required by the classification system). Following the first instrument, two enhanced versions were launched that sound also the visible spectrum required for vegetation classification over land: Along Track Scanning Radiometer 2 (ATSR-2), launched in April 1995 and Advanced Along Track Scanning Radiometer (AATSR), launched in March 2002[35].

An instrument on-board a satellite platform, in a near-polar orbit, scans a portion of the Earth surface underneath it, the visible portion that is observed by the instrument on the surface as it moves along its orbit is called swath, as shown in Figure 4.1. The projection on the surface of the orbit is called the track. As the Earth revolves, the

satellite eventually flights over its entire surface, revisiting a certain spot after a period dependant on its altitude. Satellites hosting the ATSR instruments have a revisit time of about three days, hence they are able to provide an observation of any given area once in three days. This is the case at the Equator, where the distance between the Earth's revolution axis and the satellite orbit are the farther away while the frequency of observation raises to several in a single day toward the poles. Also the width of the sensed area under the satellite (across its flying direction) contributes to the actual frequency of observations as it tends to overlap to the area covered by the previous orbit approaching the poles.

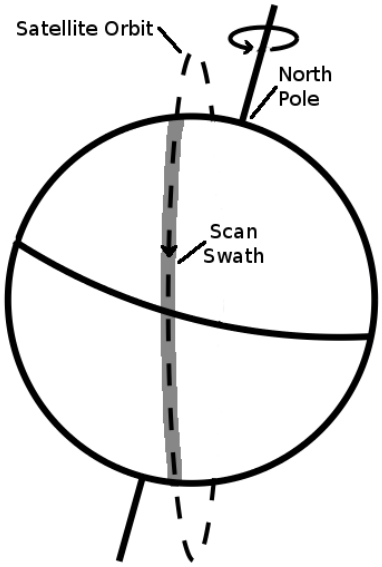


Figure 4.1: Near Polar Orbit

The along track scanning technique is a particular acquisition geometry that allows sensing the same location from two different points of view at a close interval. As shown in Figure 4.2, the actual field of view of the instrument consists of two 500 km-wide curved swaths, one crossing the track right below the instrument (nadir swath) and the other crossing the track ahead of it (forward swath)[36]. For our purposes we consider only the nadir view of the instrument that produces 500-km-wide images of the Earth's surface.

The spectral resolution of the instrument is of seven bands in the visible and infrared frequencies, as shown in Table 4.1 that can be used to retrieve geophysical parameters for the land cover classification.

To ensure the best performance of the ATSR datasets, several quality assessments

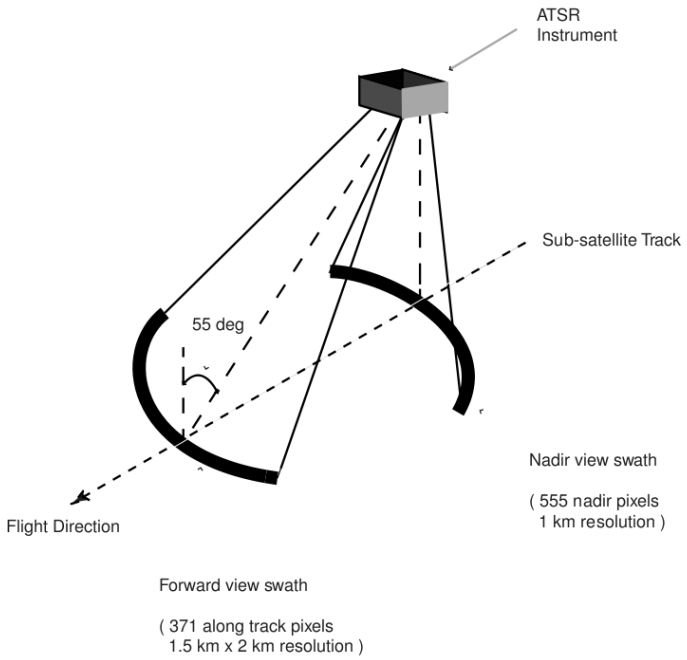


Figure 4.2: Along track scanning technique: the conic acquisition geometry provides a view at 55° along the track acquiring data ahead of the satellite (Forward view), then a second acquisition is made over the same area when the satellite flights straight over it (Nadir view). Source [36]

Target feature	Band type	Wavelength	Bandwidth
Chlorophyll	Visible	0.55μm	20nm
Vegetation	Visible	0.67μm	20nm
Vegetation	Near infrared	0.87μm	20nm
Cloud	Short wave infrared	1.6μm	0.3μm
Sea surface temperature	Medium wave infrared	3.7μm	0.3μm
Sea surface temperature	Thermal infrared	10.8μm	1.0μm
Sea surface temperature	Thermal infrared	12.0μm	1.0μm

Table 4.1: Spectral resolution of ATSR-2 and AATSR instruments.

have been done and identified issues corrected, in particular: the long term drift in the calibration parameters of the visible AATSR bands has been corrected by application of drift tables[37], while images from the ATSR-2 instrument from January 2001 to July 2001 present quality issues that make them not adequate for pixel based comparison and were excluded from processing[38].

4.2 A land cover maps provider service

ATSR data comes already calibrated in TOA reflectance and temperature values that can be effectively used to generate a standard categorization of the Earth's surface. An implementation of the Enhanced SOIL MAPPER (ESM) software is used to extract classification maps in the same geographic reference as the gridded input dataset. Moreover, to compensate the geographic misplacement of up to one pixel, on average, present in the AATSR Level 1b re-gridding process, an experimental correction method[39] was implemented to reduce uncertainty in the pixel absolute geo-referencing of both ATSR-2 and AATSR data below half the pixel size [40]. The corrected geolocation is used to generate a re-mapped version of the classification map in the simple cylindrical projection.

The system architecture presented in [41], that provided the first thematic maps archive over 15 years of ATSR data, with an interface suitable to perform content based queries, has been extended. Besides the capability to systematically process the archives and new incoming AATSR data, an interoperable interface has been defined. This new interface allows to browse archived data, to know which typologies of products are available in the archive and to obtain classification maps via FTP in an automated way. These improvements led to the realization of the ASQuLD catalogue service, that provides a standardised infrastructure for advanced database queries based on land cover types[42].

The ASQuLD catalogue service is designed to run over two kinds of hardware nodes, listed in Table 4.2. A central controller and storage node is devoted to provide the catalogue service itself while processing nodes are employed to compute land cover maps from the satellite data archive. Two relevant aspects of this solution are its scalability, allowing load distribution among several processing units, and its interoperability, providing an interface suited for automated access; both aspects are detailed in the following sections.

Node Name	Type id	Purpose
Storage controller	(STOR)	Data storage, service interface and processing control
Processing Node	(PN)	Processing power

Table 4.2: Typologies of hardware nodes for the ASQuLD catalogue: a central storage and control unit (STOR) provides the service and coordinates work for one or more processing nodes (PN).

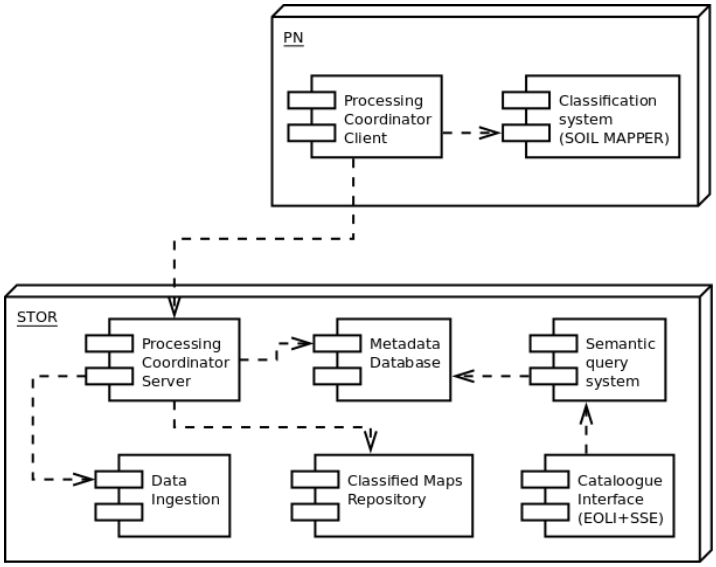


Figure 4.3: ASQuLD service architectural design. The "Processing Coordinator Server" component maintains a workload queue to be distributed to one or more "Processing Coordinator Client" instances. The client executes data processing steps, using also the SOIL MAPPER® classification software. On the server side, products are loaded into the queue by the ingestion component and processing results stored into the repository and meta-data database components, that are used by the catalogue interface and query system.

Several software components, distributed on the hardware nodes, cooperate to the data preparation and service provision for the ASQuLD service, as shown by the deployment diagram in Figure 4.3. The following two sections detail the processing coordination system and the service interface for data access.

Input of the catalogue system are strips of 40000 pixels in length by 512 pixel in width, radiometrically calibrated top of atmosphere values. The length of such datasets make them not well suited for applications at regional scale, since they cover an entire orbit around the globe. Output of the catalogue system are classification maps provided in Granules, that are 2000 pixels long cuts of the input strip. The resulting 20 granules per strip are more manageable to be provided for regional or even national scales as they do not require a full orbit of data to be retrieved. Moreover, since the

classification based on spectral bands in the visible spectra is not possible over night time acquisitions, granules covering such acquisitions can be entirely discarded, further reducing the resulting dataset.

4.2.1 An automated pluggable distributed processing system

Systematic processing of data archives is a task well suited for automation, especially if re-processing may be required with different processing software versions, or entirely different processing software. A solution has been designed for this automation that is based on three assumptions:

- Data to be processed is already available over the network with a well known transfer protocol (e.g. FTP) or can be found in an "Incoming" directory on the controller system;
- Data is provided in units that can be independently processed and are uniquely identified, for the ATSR archives the units are strips of observations covering an orbit and are identified uniquely by the acquisition start date and time (also from their file name that is a unique identifier);
- A workload queue management system exists to execute the processing to which work units can be queued for execution.

With the aforementioned assumptions the automation can be obtained either by monitoring data sources for new data to process (as in the case of Rolling Archives, where new data is made available for a limited period of time before its long term archiving) or by transferring units of archived data in the "Incoming" directory for later processing. The unique identification allows to keep a list of processed data to avoid undesired re-processing (as may happen when data already processed from the Rolling Archive is made available also to long term archives that are being processed). The final element enabling automation is the execution of a concurrent loop process to monitor the Incoming directory, that adds data to the processing queue.

The workload queue management system has been designed and implemented to allow automation and the queue model allows to manage distributed computation in a firewall-friendly pull model that allows distribution from any location enabled to establish incoming connections to the central system. A completely pull model allows

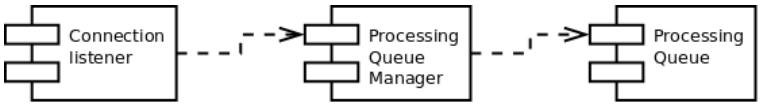


Figure 4.4: ASQuLD service: processing coordinator architecture. A network interface is provided to a queue manager that manages an execution queue.

also to secure the processing nodes behind an additional firewall layer making it not accessible from the service front end or the controller: any proprietary processing system can thus benefit from being secured from network intrusion since no incoming connection to the processing node is required or the processing can be done directly at the provider premises. Besides protection of the processing software another advantage of this model is that processing nodes can be located close to the data to be processed. Given the data compression provided by a classification process, moving the processing close to the data is the most advantageous approach since it saves on network transfer that can rapidly become the bottleneck of the process. The implemented solution, described hereafter, consists of a simple solution built as a Java client application paired with the processing software to be executed, hence requiring very low installation effort and providing a high degree of re-usability.

The work queue is managed by the *processing coordinator server*, that provides a network interface to receive and handle requests by its client component to provide redundant, distributed and balanced data processing. The coordinator maintains a processing queue to coordinate work assignment to the processing nodes. The component architecture is depicted in Figure 4.4. A network interface is provided where instances of the *processing coordinator client* can connect remotely to communicate with the server.

The processing queue is implemented by a database table, holding work units and assignments, paired with a table holding registered client instances. Both tables are shown in Figure 4.5. The minimal set of information to manage the processing is used and a basic authentication mechanism implemented by registration of clients with a security token.

An interface consisting of four methods is provided by the controller to allow access to the queue, as depicted in Figure 4.6:

addToQueue is called to add a new queueEntry to the queue and is called from the local thread responsible for Rolling Archive and remote data sources monitoring;

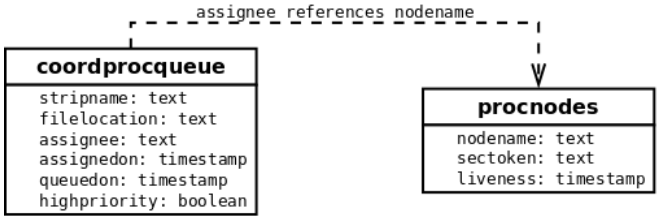


Figure 4.5: ASQuLD service: the processing coordinator queue is implemented as a single database table that holds information about each queued work unit and its assignment. High priority can be assigned to queued entries to ensure they are processed before other entries.

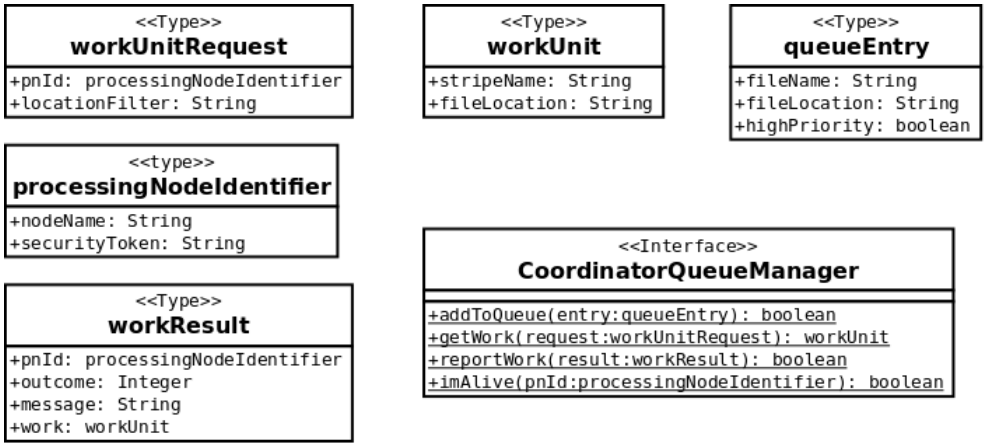


Figure 4.6: ASQuLD service: the processing coordinator interface provides four methods that can be invoked remotely to add a new item to the queue (addToQueue), ask for the next dataset to be processed (getWork), report a completed processing task (reportWork) and to report processing node liveness (imAlive).

getWork is called by the clients on processing nodes to request a new work unit;

reportWork is called by the processing nodes to report a completed work unit;

imAlive is called by the processing nodes to notify its activity status or automatically by the server upon receiving a work request message from a client.

The overall activities executed by the coordinator server upon receiving a message are shown in Figure 4.7. Actions related to reported work are implemented by calling an external script to allow complete customization of data content management. Results are packaged by the client in a single file (results archive) and uploaded to the controller via its FTP server to allow local archival of the produced classified map and management of possible meta-data. Contents of the transferred archive are generated by the processing element on client side, that is also an external component with respect to the processing coordinator. The system is hence completely pluggable in the

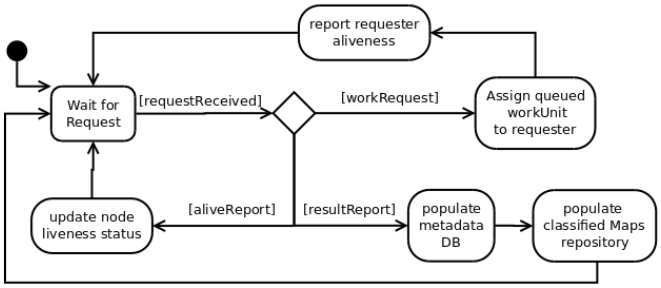


Figure 4.7: ASQuLD service: the processing coordinator server performs actions corresponding to the received message type.

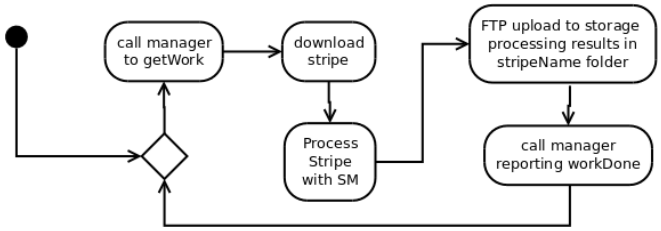


Figure 4.8: ASQuLD service: the processing coordinator client loops in a request, process report work cycle.

sense that it provides a distribution model that is applicable to any data archive and implements extension points where specific components can be plugged to support a new source or new thematic archives. It will suffice to add a processing core to the client that prepares the result archive in the defined structure to support classification of a new data source. By updating accordingly the reported work processing script, new thematic data can be managed in the result archive.

The coordinator client continuously performs the activities depicted in Figure 4.8. Any number of clients can be instantiated, depending on the available hardware nodes to add computational power to the system. The number of nodes can also be dynamically increased or reduced without further operations, provided each client is registered with the controller. Finally the queue is monitored for stale assignments to cope with possible long term failures of any client.

As security policies of the data provider may forbid data processing elements inside their network premises, the only option remains to download the entire data archive to the remote processing system. In the case of the ATSR archive, the processing system was remote and all data needed to be downloaded off site for processing. Wide area network links, even the broad band ones, have considerably less bandwidth than

those attainable in local area networks, hence the download link can easily become the bottleneck of data processing, especially when other network transfers are required over the same link. In such situation the use of a download buffer (feasible thanks to the Incoming folder approach) has proven useful to keep the processing system always busy. A continuous download to the buffer allowed to store enough data locally during periods of high bandwidth availability to compensate for the reduced bandwidth periods, thus avoiding the network bottleneck. This also confirmed the need for processing elements available at data provider premises to further reduce the network bandwidth requirements for systematic data processing, especially (actually only) for processes that provide reduction in the amount of data output.

The coordinator interface is ready both for remote processing nodes and for multiple data sources: the *locationFilter* parameter of the *workUnitRequest* is the key for these features. This filter provides a substring that must appear in the *fileLocation* attribute of queued work units for them to be assigned to the requester. Since the file location is its full network address, the server name could be used to request processing of data hosted only at specific servers (e.g. those close to the processing node). The same approach can also be used to process different sensors with different clients by filtering file locations on the portion of the file name identifying its source. This approach assumes that a new data source has its clear identification in the file names (as is usually the case with satellite data where a part of the file name identifies the sensor or processor that created it).

4.2.2 An interoperable interface for data provision

One of the goals of the MEA system design is to be applicable to multiple sensors at different resolutions. From data pre-processing perspective, the availability of a classified maps catalogue gives the opportunity to use it as a connection point between the multi temporal system and different data archives. This approach allows also to define an interoperable interface that is proven suitable for data access automation by its actual integration into the system. The interface presented hereafter is used to automatically search and retrieve classification maps by the automated data ingestion component that builds the MEA archive.

The ASQuLD SOAP interface, defined in [34] is based on four synchronous operations and a customised type to define data collections consisting of files accessible over

the network. Two of the four operations, "Search" and "Present" are based on the EOLI standard and provide functions to browse the catalogue. As defined in [17] these functions allow to query the catalogue by date and geographic area. An extension point is also provided in the standard that can be used to add the thematic content to the query parameters. Even if the extension is defined as *Satellite domain conditions* it is basically an array of key-value pairs that is also well suited for thematic content. Practically the implemented catalogue recognized specific keys added to the Search operation that correspond to aggregation sets of its output classes paired with a pair of values to bound its occurrence. This solution allows queries that identify all classified maps over *a given area of interest* (a square in the geographic coordinate system), that is produced from data acquired *between two dates* with a given *thematic content*. The thematic content is expressed by triples in the form of (class, min, max) where: class is one of the aggregate set while min and max are minimum and maximum percentage of the given class to be present in the image, with respect to its entire content (e.g Bare soil between 10 and 40 % of the granule). The identified granules keep reference to the source dataset that originate them, hence the catalogue provides also a thematic view over the underlying data archive and can be used as a query system to search the original data archive based on its thematic content.

The EOLI standard uses the concept of *CollectionsId* to identify groups of products and it is assumed that each product is uniquely identified within the same collection by a *resTitle* . Under these assumptions, each product (be it raw satellite data or any derived product) can be uniquely identified by the CollectionId.resTitle pair. From the perspective of processing automation, considerable overhead is introduced by the EOLI standard Search operation that provides as minimal output an entire tree of meta-data while a minimal product identification would suffice to proceed to data retrieval. The "Present" operation is already defined to provide additional information about any given result: thematic content is also added to the output of that operation via its key-value pair array extensibility.

Based on the SSE interface [43], two operations are provided to obtain information about and use a service. The "RFQ" (Request For Quotation) is used to obtain information about available product typologies for a given collection. Input to the operation is a list of collection identifiers and the output provides, for each collection identifier, the list of available product typologies, as listed in Table 4.3. The "Order"

operation is used to request the products to be available for direct download.

Product type ID	Description	Notes
CLS	preliminary classification maps with basic geo-reference information	Default format is GeoTIFF
PROJ-AAAA:NNNNN	Re-projected outputs, possibly with improved geo-reference accuracy. Where: • AAAA can be either EPSG or ESRI, indicating the related standard identification; • NNNNN is a valid EPSG / ESRI projection code.	Projection codes are defined by established standards [44] e.g., EPSG:32663 for World Equidistant Cylindrical. Default format is GeoTIFF.
SAT	Original Data (when available)	Direct download of original data from which the map has been derived.

Table 4.3: ASQuLD service interface: defined product typologies.

The Order operation mitigates the meta-data overhead issue by allowing a simple list of *resTitle* identifiers, grouped by collection and requested product typology identifiers, as input. As output, for each requested product, the structured set depicted in Figure 4.9 is returned. This is a minimal set of information required for an automated software component to understand and use the provided information: for each collection a set of dataset links is returned, each in turn composed of links to files composing it. Status and identification attributes are added to the dataset elements to allow automatic interpretation of the results. Status can be one of: Complete, Incomplete or Empty and it is set depending on which data types have links provided in the dataset versus the types requested (e.g. a Complete dataset may contain just one link to the only requested data type). Available typologies for a given collection are retrieved with the RFQ operation.

4.3 Remapping on the Earth Fixed Grid

In the DGGS defined in section 3.2, ATSR data belongs to grid level zero that has a reference GSD of one kilometre. Since the grid is based on the simple cylindrical projection, a classification map in that projection would already be remapped on a given grid level. The ASQuLD catalogue delivers such a map, over which the improved remapping algorithm for AATSR is also applied. This layered approach allows to keep

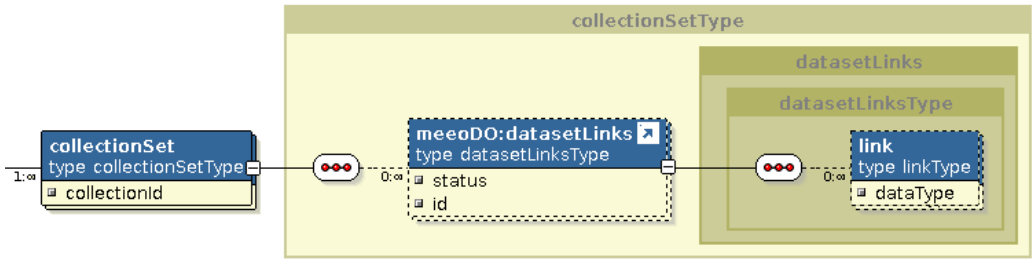


Figure 4.9: ASQuLD service: structured output for data provision consisting of one collection set type for each collection in the input request, in turn containing one dataset links element for each ordered dataset, in turn consisting of one link for each requested product typology available for direct download.

the geo-location correction (a task strictly dependant from the specific instrument) within the system that produces derived products from instrument data (i.e. the automated classification system).

The availability of a projected map matching a grid level reduces the ingestion process to the generation of cuts of the map aligned with grid Tiles of 64 by 64 pixels. At level zero each grid Tile is aligned to quarters of degree and, given the origin of grid coordinates at $(90, -180)$ the geo-location of a grid Tile is addressed via geographic coordinates by the following equations, defining also the area covered by corresponding grid Tiles:

$$x = \text{floor}((90 - Lat) \times 4)$$

$$y = \text{floor}((180 + Lon) \times 4)$$

Each grid Tile covers a geographic area called Tile zone that can be used to address actual Tile content at a given time or simply a Tile, that is uniquely identified by its Tile zone and a date at day resolution. The triple (x, y, t) thus addresses a single Tile containing the $64 * 64 = 4096$ land cover class values for day t over Tile zone x, y.

In the Tile mapping process, the original data is sampled using the Nearest Neighbour methodology and data is filtered taking into account the scope of the system, that is land cover, hence:

- A Land/Sea Mask (LSM), derived from the United States Geological Survey 1Km Land Sea Mask dataset[45], is used to filter the input. Grid Tiles completely over sea are not processed by the system: grid Tiles are considered valid (over land) if at least four of their pixels are mapped to a land pixel (i.e. the they contain at least 4 land pixels that fall within the land mask);

- All types of cloud classes are removed, this enables also the removal of completely cloudy Tiles from the archive, further reducing the archive size;

The LSM implemented for grid level zero consists of an ordered array of pixels stored in a plain text file, a solution that is suitable for that level but that requires a structured alternative for deeper levels.

Adequacy of the remapping grid to the application goals has been assessed and confirmed by an independent study, whose results have been reported in [46]. That document describes four typologies of tests used in the evaluation:

- Tiles produced from a classified map have been mosaicked to ensure the original map can be restored from them and confirm correct tiling operation;
- Quantitative analysis test between ATSR reflectance values and remapped image values has been performed;
- Synthetic datasets has been created and used to verify the remapping process with known expected results;
- Visual of images from different instruments has been performed for images containing targets with clearly identifiable shapes.

4.4 The Automated Data Ingestion

Following the ingestion rules and control flow defined in section 3.6, an interface is provided for the management of the data processing system. As shown in Figure 4.10, a compact tabular display is used to provide an overview of the archive status and direct control over data processing at month resolution[1]. This approach allows the simple management of multiple data sources and related archives by providing the same functionality and display for each of them.

The ingestion control interface drives the processing system that consists of several concurrent processing threads, each operating on a single month. The number of concurrent threads is configured on a per grid level basis by specification of $(GridLevel, numberOfThreads)$ pairs plus the number of concurrent threads to be

[1] To be noted that the ingestion control partitions only the temporal domain of the source archive while control is applied to the whole globe as its geographic area. Ingestion control limited to geographic areas is feasible but includes complexities both in interface and meta-data management that are not addressed by this system.

AATSR ATSR2 AVNIR2

YEAR	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
2002	S: X GP: 0	S: X GP: 0	S: X GP: 0	S: X GP: 0	S: X GP: 0	S: X GP: 0	S: X GP: 0	S: P GP: 4339	S: P GP: 3561	S: P GP: 4121	S: P GP: 3946	S: P GP: 4590
2003	S: P GP: 4650	S: P GP: 3243	S: P GP: 4053	S: P GP: 4400	S: P GP: 4379	S: P GP: 4513	S: P GP: 4743	S: P GP: 4075	S: P GP: 4173	S: P GP: 4685	S: P GP: 4036	S: P GP: 4353
2004	S: P GP: 4677	S: P GP: 4307	S: P GP: 4768	S: P GP: 3877	S: P GP: 4739	S: P GP: 4049	S: P GP: 4186	S: P GP: 4553	S: P GP: 4508	S: P GP: 4601	S: P GP: 3713	S: P GP: 3890
2005	S: P GP: 4241	S: P GP: 3935	S: P GP: 4658	S: P GP: 4019	S: P GP: 4780	S: P GP: 4508	S: P GP: 4731	S: P GP: 4187	S: P GP: 4413	S: P GP: 4762	S: P GP: 4447	S: N GP: 4078
2006	S: P GP: 4641	S: P GP: 4288	S: P GP: 4633	S: P GP: 4065	S: P GP: 4686	S: P GP: 4478	S: P GP: 4774	S: P GP: 4002	S: P GP: 3693	S: P GP: 4704	S: P GP: 4536	S: P GP: 3865
2007	S: P GP: 4549	S: P GP: 4199	S: P GP: 4742	S: P GP: 4401	S: P GP: 4127	S: P GP: 4556	S: P GP: 4378	S: P GP: 4719	S: P GP: 3975	S: P GP: 4737	S: P GP: 4553	S: P GP: 4077
2008	S: P GP: 4577	S: P GP: 4391	S: P GP: 4718	S: P GP: 4404	S: P GP: 4749	S: P GP: 4040	S: P GP: 4749	S: P GP: 4541	S: P GP: 4502	S: P GP: 4759	S: P GP: 3952	S: P GP: 4571
2009	S: P GP: 4507	S: P GP: 4265	S: P GP: 4735	S: P GP: 4348	S: P GP: 4198	S: P GP: 454	S: P GP: 4738	S: P GP: 4705	S: P GP: 4518	S: P GP: 4587	S: P GP: 4448	S: P GP: 4651
2010	S: L GP: 3444	S: L GP: 3707	S: U GP: 405	S: U GP: 54	S: R GP: 7617	S: R GP: 4343	S: R GP: 5729	S: R GP: 6415	S: R GP: 2606	S: R GP: 3284	S: R GP: 1981	S: U GP: 2085

Figure 4.10: Automated data processing input rules management interface, each cell displays status information for a month and can be used to set the rule to be applied over it.

executed for Rolling Archive processing. The control system loops over the months configured for each instrument by an ordered set of $(year - month, rule)$ pairs and executes one thread per instrument, until the allowed number of threads for the corresponding grid level is reached. Then, once a thread completes the next month for the next instrument in that grid level is launched. This configuration allows to adjust the processing system resource demand to the hosting hardware configuration and expected workload.

Each thread performs the same operations, depending on the configured rule for the given month, in particular, when Update is set: a Search operation is performed on the data provider interface, constrained by the given month, to get a list of available data. This Search operation is affected by the aforementioned meta-data overhead detected in the EOLI standard: when the number of results is high, as is commonly the case for a global search over an entire month, the time needed to transfer all the unused meta-data negatively affects the ingestion process, causing a delayed start. Then, the availability of the re-projected map is assumed and that product typology is ordered for the images: one hundred maps are requested for each Order operation. As soon as a map is downloaded it is cut into Tiles that are then stored into the Tile repository while meta-data for each Tile is computed and stored in the Tile database.

4.5 The Tile Archive

The goals of the Tile storage system, or Tiles Archive (TAR), is to provide adequate storage capacity for the Tiles resulting from processing of the entire archive of AATSR and ATSR-2 dataset and to provide very fast identification of Tiles matching spatio-temporal and thematic content query parameters. This goals are attained by dividing the archive in two distinct components: a repository to store the Tiles and a database to store their meta-data. The use of a Tiling system that does not require use of GIS extensions to the database, along with a direct mapping for Tile access in the repository and direct file system storage for fast retrieval provide a solution that aims at scalability and ease of extension.

4.5.1 Tile meta-data database

The Tile meta-data database, referred to as "the database" throughout this section, is the component devoted to fast query provision for spatial-temporal-semantic query for retrieval of thematic based Tiles. A first version of the database has been presented in [47]. From the described version only the compact classification field has been kept while the database structure and access functions have been replaced by the ones herein described. Detailed tests, performance results and considerations for the meta data database are reported in section A.3.

The meta-data consists of a single table for each instrument collected in the archive. The table structure is depicted in Figure 4.11. Geographic location is coded with integer grid Tile coordinates x (along Longitude) and y (along Latitude); temporal information is stored with a timestamp; the thematic content is stored in a bit string field that allows data compression by reducing the number of needed bits to hold it from 912 to 741, saving 171 bit per record.

The particular typology of data stored and likelihood of subsequent queries over sub-regions of any geographic location (as it is being studied or searched) drives the choice of partitioning data along the geographic dimensions, in particular the use of a two-dimensional addressing allows ease definition of square sub-regions (corresponding to square selections over the geographic coordinates system). A square sub-region can be simply defined by four boundaries on the two dimensions as follows:

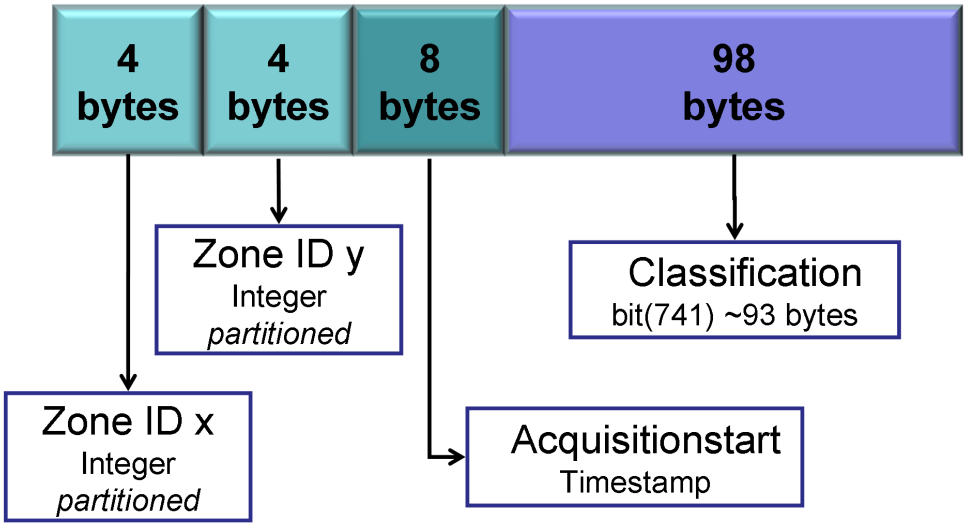


Figure 4.11: Tile metadata database: geographic location is coded with integer grid Tile coordinates x and y; temporal information is stored with a timestamp; the thematic content is stored in a bit string field that allows data compression by reducing the number of needed bits to hold it from 912 to 741.

$$\begin{cases} leftmostTileZone \leq x < rightmostTileZone \\ topmostTileZone \leq y < bottommostTileZone \end{cases}$$

As reported in section A.3, this scheme permits to keep also the complexity of queries over rectangular areas minimal with respect to a mono-dimensional addressing scheme. Moreover, the processing time required to perform a query on the meta-data stored into a single big table (the number of records managed by the system is about 500 millions) is much larger than the time expected for interactive access, hence a partitioning scheme for the database is required to improve performances. Partitions are implemented with native support of the postgresql database management system (supported since the recent version 9.0). Partitions are defined on the bi-dimensional grid Tile addressing, as follows:

$$X_{parts} = 12^{\circ}\ across\ 360^{\circ}\ Longitude$$
$$Y_{parts} = 6^{\circ}\ across\ 180^{\circ}\ latitude$$

For grid level zero that means partitions defined in terms of x and y values as follows:

$$(i = 0..29, j = 0..29)$$

$$partition_{ij} = \begin{cases} 48 * i \leq x < 48 * (i+1) \\ 24 * j \leq y < 48 * (j+1) \end{cases}$$

For a total of 30 across Latitude * 30 across Longitude + 1 master table = 901 tables per instrument (sensor). Given the simplicity of constraints defined on each partition that number of partitions is adequate to provide good query performances to our implementation.

Further performance improvement could be obtained by having more than one core processing the same query concurrently (a function not implemented nor foreseen for implementation in postgresql) or by partitioning also over time on different database servers. Preliminary results of experiments with a mixed approach to have multi core queries via parallel query execution on several database instances on the same machine are reported in section A.2. The rationale is to leverage deployment of multiple instances on the same server as a mean to obtain multi-core query processing: each instance processes the same query on a portion of the entire dataset. The resulting concurrent disk access would present an ideal configuration to leverage the high performance random access feature of SSD technology used to store meta-data. Results are promising but standard tools seem not yet adequately optimized for parallel query performances.

An example of a four dimensional query over the database is presented in Listing 4.1 where Tiles in summer time, across 4 years (1999 to 2002) are requested, over France, having three semantic parameters above 1 percent.

Listing 4.1: Example of spatial-temporal-thematic query for content based retrieval

```
SELECT zone_id_x, zone_id_y, acquisitionstart
FROM tiles_atsr2
WHERE ((
(acquisitionstart BETWEEN '1999-06-21 00:00:00' AND '1999-09-22 23:59:59') OR
(acquisitionstart BETWEEN '2000-06-21 00:00:00' AND '2000-09-22 23:59:59') OR
(acquisitionstart BETWEEN '2001-06-21 00:00:00' AND '2001-09-22 23:59:59') OR
(acquisitionstart BETWEEN '2002-06-21 00:00:00' AND '2002-09-22 23:59:59') OR
) AND (
(zone_id_x BETWEEN 699 AND 753 ) and (zone_id_y BETWEEN 155 AND 190)
) AND (
(((  classification << (10) *13)::bit(13)) >= (40.96 * 1)::integer::bit(13)) AND
(((  classification << (12) *13)::bit(13)) >= (40.96 * 1)::integer::bit(13)) AND
(((  classification << (18) *13)::bit(13)) >= (40.96 * 1)::integer::bit(13))
))

UNION ALL

SELECT zone_id_x, zone_id_y, acquisitionstart
FROM tiles_aatsr
WHERE ((
(acquisitionstart BETWEEN '1999-06-21 00:00:00' AND '1999-09-22 23:59:59') OR
(acquisitionstart BETWEEN '2000-06-21 00:00:00' AND '2000-09-22 23:59:59') OR
(acquisitionstart BETWEEN '2001-06-21 00:00:00' AND '2001-09-22 23:59:59') OR
(acquisitionstart BETWEEN '2002-06-21 00:00:00' AND '2002-09-22 23:59:59') OR
) AND (
(zone_id_x BETWEEN 699 AND 753 ) and (zone_id_y BETWEEN 155 AND 190)
) AND (
```

```
(((  classification << (10) *13)::bit(13)) >= (40.96 * 1)::integer::bit(13)) AND
(((  classification << (12) *13)::bit(13)) >= (40.96 * 1)::integer::bit(13)) AND
(((  classification << (18) *13)::bit(13)) >= (40.96 * 1)::integer::bit(13))
));
```

To be noted the built in functions that allow efficient access to the classification content in compressed form (the 741 bit string classification field) with a shift, a truncation (bit(13) type cast) and conversion to integer (integer cast operation). The complexity of the query, in terms of operations per record to be performed, rises as the number of years and classes to compare rises, while the geographic location of a square selection (common in case of geographic bounding box) is always determined by four comparisons (per instrument), independently of its size. It is also notable the use of the union operator (that could be rendered also concurrent at DBMS level) to support an arbitrary number of instruments per grid level.

4.5.2 The Tile repository

Paired with the Tile meta-data database, the Tile repository provides the storage system for actual Tile maps (thematic maps corresponding to a grid Tile at a given time). This component is made accessible over the network via a basic HTTP interface providing a getTiles operation that accepts its input request as "post data". A properly formatted text string is used to request a list of tiles addressed by their identifier, that is the quadruple $(sensor, zone_x, zone_y, dateTime)$. One interface is provided for each grid level that provides data for all sensors available at that level, that is selected by the "sensor" parameter. This interface allows only retrieval of known Tiles by their identifiers, without any parameterised query capability. That is exactly what is needed to complement the fast query functionality provided by the separate meta-data database and to allow straight data retrieval.

The output is provided as an XML document that allows complete interoperability with any calling component at the expense of some data overhead. The output schema is provided in Figure 4.12. A list of tiles is returned with one tile element for each requested Tile; Tile identification information is provided as per the identification quadruple; Tile data is encoded in the "pixel" element as a hexBinary type that is a string encoding binary data with hexadecimal digits, doubling the data size (two hex digits for each byte).

Within the system this is used as an interchange format, while map data to be displayed is encoded in compressed image files, hence the overhead does not noticeably

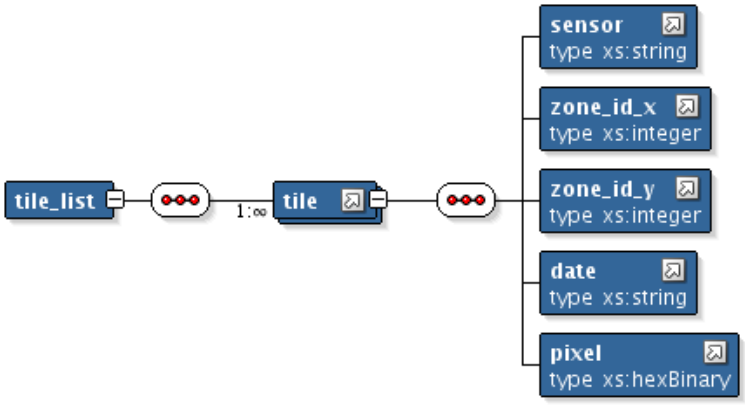


Figure 4.12: Tile repository output schema. A list of tiles is returned with Tile identification information and Tile data as an hexBinary type that is a string encoding binary data with hexadecimal digits.

affect performances. An example of output for a request of a single Tile is provided in Listing 4.2, where Tile data inside pixel element has been omitted for brevity.

Listing 4.2: Example of Tile archive output for a single Tile request; Tile data inside pixel element is omitted.

```
<?xml version="1.0" ?>
<tile_list>
        <tile>
                <sensor>AATSR</sensor>
                <zone_id_x>281</zone_id_x>
                <zone_id_y>45</zone_id_y>
                <date>2002-08-01 00:10:22</date>
                <pixel>03030303[...]000000</pixel>
        </tile>
</tile_list>
```

The repository interface is backed up by an archive based on the XFS file system that, among other features, provides efficient tree structures for fast searches and rapid response, even for directories with tens of thousands of entries[48]; furthermore it allows an unlimited number of files and directories to be stored[2]. Finally, paired with Linux, XFS provides a stable and reliable storage system that is capable of providing throughput levels close to the hardware limits for highly parallel workloads[49].

Given the Tile identification scheme and the requirement to provide only direct mapping from a list of identifiers to related Tile data, a straightforward approach is used for the structure of the file system directory tree. The directory structure is based on a fixed naming convention of both directories and files. Furthermore, it is organized

[2]Actually the limit is given by the addressing space that allows a million TeraBytes file system. The ext4 file system supports also an "unlimited" number of files. During stress tests however, the latter presented erratic behaviour with display of Kernel logs for potential data loss, hence it has been discarded in favour of the more stable XFS alternative (see section A.4).

in three levels of depth to keep the number of files in a single directory manageable from the system administrator point of view. The directory hierarchy levels are defined as follows:

Level 0 The file system root (for grid level 0);

Level 1 Tile Latitude directories: Latitude of upper left corner of the tile, composed of six characters according to the template "sLL.ll" , where:

s is the coordinate "sign", can be either "+" (north) or "-" (south);

LL are the two digits of the Latitude's integer part (zero padded on the left);

ll are the two digits of the Latitude's decimal part (at grid level 0 these two digits can be 00, 25, 50 or 75);

Level 2 Tile Longitude directories: Longitude of upper left corner of the tile, composed of seven characters according to the template "sLLL.ll" , where:

s is the coordinate "sign", can be either "+" (east) or "-" (west);

LLL are the three digits of the Longitude's integer part (zero padded on the left);

ll are the two digits of the Longitude's decimal part (at grid level 0 these two digits can be 00, 25, 50 or 75);

Level 3 Tile Year directories: the year of the Tile date, represented by four digits.

The aforementioned structure leads, for grid level 0, to 720 first level directories, each containing 1440 second level directories, each containing one directory for each year where data is available (that are 16 as of February 2011) for a nominal value of 1036800 directories for the geographic partitioning, times 16 for the time partitioning, to cover the entire globe across 16 years. The geographic part of that number is reduced to the number of valid Tiles defined in the Land/Sea Mask as only land data is considered. The total number of directories is then 372154 times covered years. Inside each Year directory there are Tile data files, named according to the template "SSS_*SpCl*CC_YYYYMMDD_HHmmss_XXXXYYY" where:

SSS three characters that identify the underlying acquiring sensor;

SpCl identifies content type, that is spectral classification;

CC two digits identifying the thematic classification level (number of classes);

YYYYMMDD year, month and day of Tile content date;

HHmmss hour, minutes and seconds of the earliest Tile content time;

XXXXYYY seven digits for grid zone coordinates (concatenation of x and y values)[3].

An example of the complete path to a Tile file in the repository is:

`+35.75/+006.50/2002/ATS_SpCl57_20020727_095649_0313226`.

The fixed Tile size of 4096 bytes, without associated meta-data, is also aligned to the file system parameters: the allocation unit is tuned to store exactly 4096 bytes. The described file system based Tile access, paired with its web interface, provides a fast and efficient implementation based on a simple, direct mapping logic, that exploits the efficiency of the XFS file system to perform the Tile look up and retrieval operations and delegates the execution of the semantic query to the separate Tile meta-data database.

[3]In the first implementation this part of the file name was equivalently used to hold the 7 digit zone identifier of the linear grid coordinate system. Although this information could be omitted from file names, thanks to the directory structure, it is kept for completeness; allowing to completely identify file contents by its name.

Chapter 5

Implementation: providing interactive analysis

This chapter describes data access and presentation functions, including the Graphical User Interface (GUI), implemented to provide the features defined in chapter 3. Details on the model matching engine that allows temporal-thematic searches on the archive is also presented together with its model editor interface. The components herein described are built on top of the Tile archive system, described in section 4.5, and used as interfaces to query and access Tile data.

The first part of this chapter describes the user interfaces that are built using Django (as the web application development framework) and Python (as the server side programming language) technologies. Technologies selection were proposed in [50] and used to implement their early prototype versions and some of them kept for the first complete prototype described in [51]; the proposed Google™API (as the provider of geographic map functions) has been replaced by the OpenLayers open source solution. OpenLayers supports recent and upcoming standards for map content delivery proposed by the Open Geospatial Consortium (OGC), such as the Web Map Service (WMS) used to display background maps. The use of OpenLayers also allows users to easily display their own background maps on the interfaces by publishing and accessing them via WMS. The Django / Python server side technologies have been kept as they proved adequate to the system goals and opened also the opportunity to test alternate ways to distribute processing while the evolution model concepts and the matching engine have been considerably extended in both expressive power and data analysis thoroughness.

The second part is about the matching engine that is also implemented using

server side technologies and distributed over the network using the same web server framework that provides the user interfaces. Computation is performed at pixel level and distributed by partitioning the domain over the geographic dimension. A practical and effective solution that is possible thanks to the single pixel based analysis that does not require data exchange among "sibling" processors, controlled by a single parent process.

5.1 On-line data analysis interfaces

The GUI provided by MEA consists of three main interfaces, all accessible over the web and built using web 2.0 technologies for a high degree of interactivity. A complete documentation of the user interfaces and related functions is out of the scope of this thesis and is provided in [52]. This section describes the main elements of the so called Expert user Visual Analysis Tool (EVAT), that is the interface devoted to provide the full set of analysis tools to Expert users (users in the "Expert" role). To recall from section 3.4, the interface is composed of three tabs devoted to the three main phases in data analysis and their related functions; these tabs are:

Time Series Analysis is the first tab, devoted to provide exploratory functions focused on the feature-temporal dimension, contextualised over a geographic map. This tab allows both Tile and pixel level visual analysis;

Evolution Model Editor is the second tab, devoted to definition of evolution models. It provides graphical tools to create, display and configure model elements along with model meta-data management interfaces;

Evolution Model Matching is the third tab, that corresponds to the simplified view provided to Standard users of the system. It is devoted to provide access to the thematic-temporal evolution search feature that is provided by the evolution models matching engine. It provides search results over a geographic map and includes a tool for detailed matching result analysis at pixel level and results export for off-line analysis.

5.1.1 Time Series Analysis

The first tab of the Expert user interface is designed to provide tools for visual analysis that are focused on the temporal domain. It is depicted in Figure 5.1 and is composed

of three main areas, each devoted to specific functions as follows:

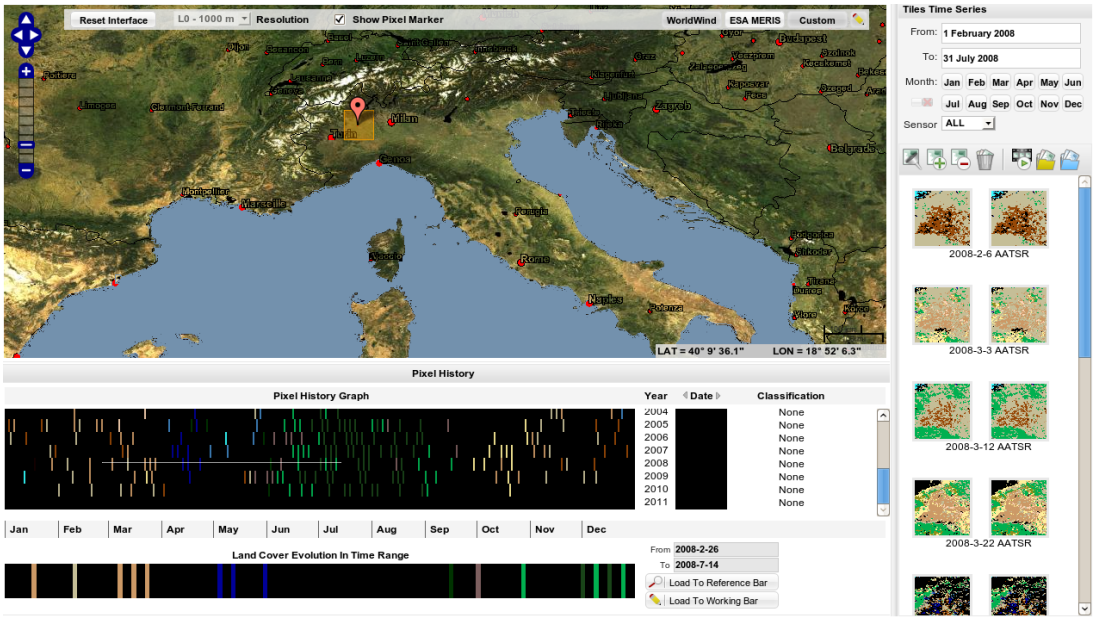


Figure 5.1: The EVAT interface for visual analysis (exploratory "Time Series Analysis" tab) provides an upper left map area for the geographic dimension, a time series frame on the right for the time dimension, and a lower left frame for thematic-temporal pixel level analysis (screen shot taken from version 1.2 of the system).

Map Area that is the upper left section of the tab and is used to select the grid level to display, the study area and a pixel of interest within it for the other display functions. On the map the classification at a selected time is also overlaid on top of a WMS provided background. This area is oriented *toward the geographic dimension* of the archive but can also be used to display thematic maps in time sequences.

Tiles Time Series that is the columnar area on the right of the tab and is used to select the temporal range(s) for area level analysis, that is the display of maps over an area composed by several Tiles in a square, also called the study area. This area is oriented *toward the temporal dimension*, providing sorted display of thematic maps of the study area at different dates. Displayed maps can be directly overlaid on the map area and the right column of maps can provide a view filtered only on a specific thematic class to analyse its evolution in the area over time.

Pixel History that is the lower left section of the tab and is oriented *toward thematic-*

temporal analysis. It displays the Pixel History graph, defined in section 3.4 on page 44 to provide an on screen view of the thematic evolution profile of the single selected pixel across the entire temporal range covered by database content at the appropriate grid level. Along with the graph, tools are provided in order to select an evolution pattern that can be sent to the model editor in the second tab to be used as reference for the definition of an evolution model.

For the Tiles Time Series (TTS), the use of three dimensional display solutions (such as animated carousel effects) have been also explored in visualization over the time dimension but they appeared to add no visual aid in letting the user display the time series when compared to the simpler plain columnar solution adopted. Conversely, superimposition of parts of tiles produced by existing three dimensional display libraries tends to obstruct user vision of the data while adding client side computational load. These effects where thus discarded from interface elements.

The TTS feature potentially poses an interactivity burden to the interface as it downloads study area previews to the browser: if hundreds of previews result from a query, they fill the browser's download queue, rendering other elements unresponsive until all downloads are completed. A prioritized download queue would be a good solution to be made available on future browsers to allow at least "urgent" and "background" image loading functions. Until then we implemented a workaround by adding preview links dynamically, as soon as previous ones are loaded. This allows to keep the interface responsive "between" downloads to serve user requests to download other data, such as the Tile overlay allowed from the pixel history graph.

The use of a tabular view to display the thematic evolution subdivided along years poses some constraint to the minimum width of the interface display in terms of pixels. In particular, the minimum width of the Pixel History Graph must be multiple of 366 and the minimum multiplier for a clear display is 2, that leads to 732 pixels. Considering the other graphical elements to be displayed around it, a minimum resolution of 1280x800 pixels is required. This resolution is standard for modern laptop displays and is also compatible with 1280x1024 pixels that is commonly available on modern 4:3 monitors.

Several functions do not fit in the interface display and are provided as commands that can be selected via the panel placed above the columnar view on the TTS area. The command buttons for that panel, depicted in Figure 5.2, are provided with icons

related to the following commands (the numbers in the Figure refers to this numbered list only and are not displayed on the actual panel):

1. **Land Cover Type Filter** this command gives access to the thematic search functionality that allows defining percentage ranges for thematic classes (aggregated in macro categories) that must be present in the retrieved Tiles composing the study area;

2. **Get Tile Time series** is used to execute a search for tiles, according to the thematic filter above and the specified geographic area and time range selected (matching Tiles are previewed in the columnar view, mosaicked at day level over the study area). To be noted that only one Tile per day is allowed to be shown in the mosaicked view, that is the one corresponding to the most recent acquisition if more than one exists in the archive;

3. **Remove Tile** is used to remove the selected study area map (possibly composed of several Tiles) from the displayed time series;

4. **Discard and Report Tile** is the command that integrates user feedback on thematic data in the system. Via this command users can report problematic, or so appearing, Tiles to the system maintainers, giving direct feedback on data that is a valuable asset to improve its quality. Reported data is immediately removed from the archive as seen by users so that quality of the data is immediately improved by removal of errors. Users can thus continue their analysis without requiring administrative assistance;

5. **Play Time Series** is used to sequentially display on the map area the time series currently displayed in the time series;

6. **View All Tiles** displays an almost full-size window overlay that presents an extended view of the time series currently displayed in the time series left column: this function provides the user with an overview of many maps of the study area, sorted by time from left to right then from top to bottom;

7. **View All Filtered Tiles** is equivalent to the previous one for the time series right column (Tiles filtered by selected class).

Figure 5.2: This command panel is displayed above the columnar time series view of the Tile Time Series area. It provides access to functions of the interface tab that do not fit directly on it (screen shot taken from version 1.2 of the system).

5.1.2 Evolution Model Editor

The second tab of the EVAT is devoted to graphical editing tools to aid the user in the definition of an Evolution Model, simply called model throughout this section. All the content of the interface is placed in its upper area that is depicted in Figure 5.3. It consists of two bars where models are displayed as sequences of model elements, as defined in section 3.3. The upper bar, called reference bar, is for reference in editing the model in the lower bar, called working bar. The reference bar is read only and can display models from the catalogue of models, provided the user has read access to that model, or draft models derived from data received from the pixel history area on the first tab.

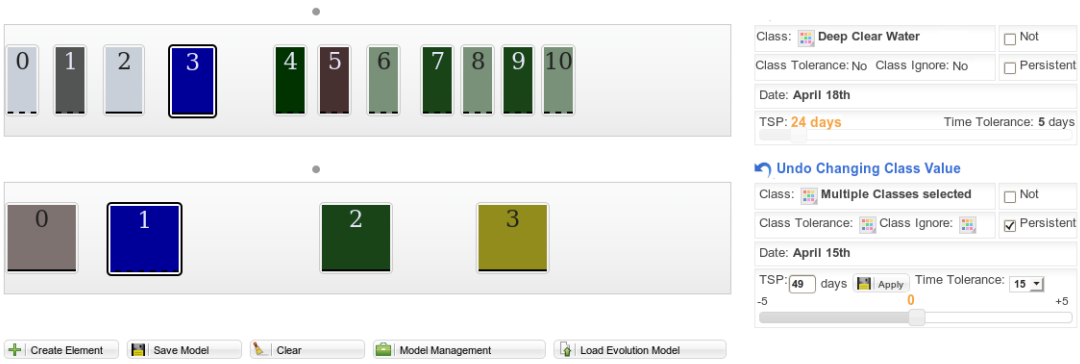


Figure 5.3: The EVAT model editor tab displays Evolution Model elements as boxes on two bars, an upper reference bar and a lower editable (working) bar. Width and spacing of elements are proportional to their temporal parameters while their color depends on selected main classes. To the right of each bar, element parameters are accessible via details form (screen shot taken from version 1.3 of the system).

On the right of each bar a form to display and edit (for the working bar) the configuration of the selected model element is provided. All the parameters of the element can be configured from these forms where aiding tools are provided to ease selection: for example, the class selector that is displayed over the interface when

the user chooses to configure one of the class sets of the element, is shown in Figure 5.4. The selector shows three columns, corresponding to three aggregation levels of thematic classes: the selection can be done at each level to ease configuration of the element.

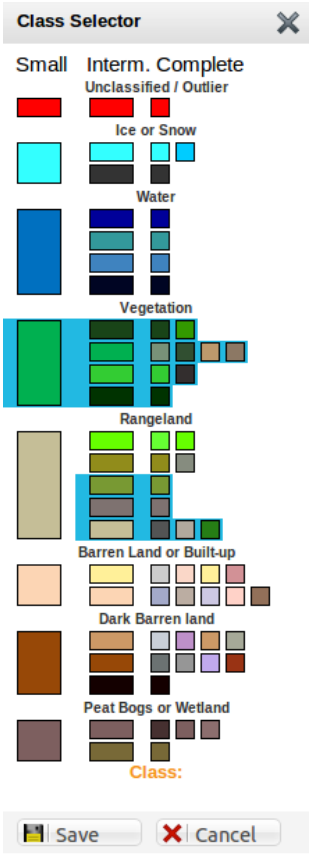


Figure 5.4: The class selector provided by the model editor displays three columns, corresponding to three aggregation levels of the 56 thematic classes provided by the classification system. The selection can be done at each level, but actual element configuration will always use the "Complete" level (screen shot taken from version 1.3 of the system)

Besides the class selection, temporal parameters can also be configured from the configuration form. A single level of "Undo" is provided to cancel the effect of the last operation performed on the model element (or on the model itself in case of addition of a new element for example). The spacing and width of elements displayed in the editor bars are proportional to their TSP and TT temporal parameters respectively. The model editor is thus designed to provide a graphical overview of the selected model, along with tools for its editing, including drag and drop functions and interactive model adaptation to insert new elements.

To be noted that the three classification levels shown on the class selector are also provided for the functions of the first tab of the interface. The rationale for such a choice is to let users choose the level of detail that they wish to use in data exploration. It is a user preference and three levels are provided in this implementation, according

to the classification levels provided by the SOIL MAPPER® software: Small (that provides only 9 broad classes), Intermediate (that includes 26 main classes) and Complete (that provides access to all 56 classes). The high number of classes in the Complete set is more sensitive to issues in the data that can affect the classification accuracy while the broad classes are much more stable across different acquisitions. Nonetheless, some of the classes in the complete set may be used to effectively discriminate among features presenting similar patterns at the broader level.

5.1.3 Evolution Model Matching

Once an Evolution Model has been defined, it is saved in the system catalogue and can then be used to search the data archive for occurrences of the modelled pattern, the third tab of the Expert interface is devoted to this function: it provides a geographic map in the upper area, as shown in Figure 5.5, that is used both for user input in the geographic dimension and output result maps display. The lower part of the tab is itself tabbed to provide a different display, based on the processing state and map configuration. Figure 5.5 shows the sub-tab devoted to collect user input in the temporal dimension and model selection.

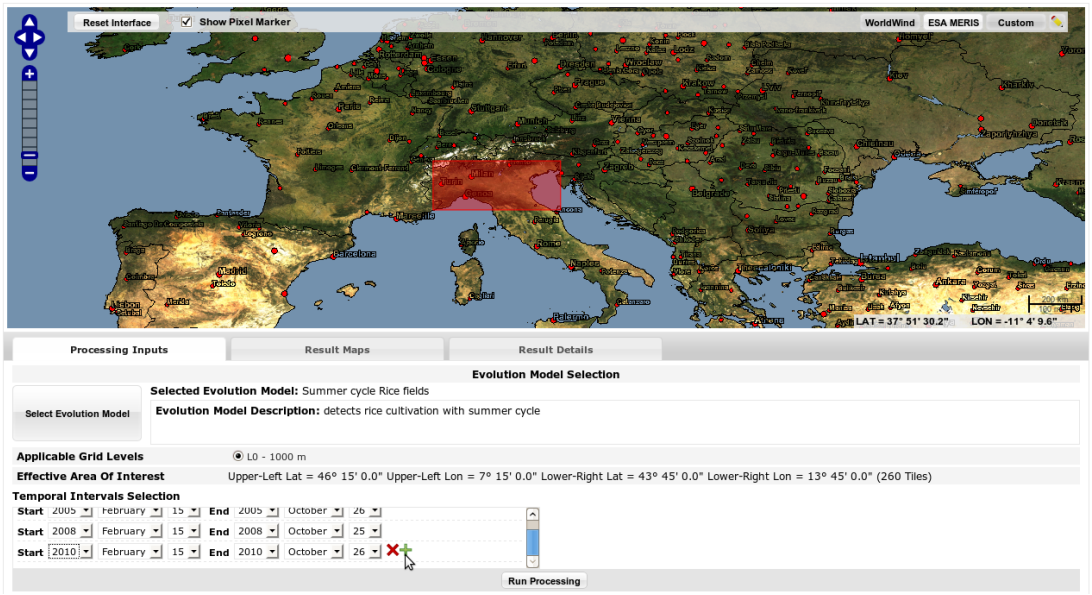


Figure 5.5: Screen shot of the model matching tab of EVAT interface. The upper portion provides a geographic map display while the lower portion is tabbed to display content depending on the operational context (screen shot taken from version 1.3 of the system).

Once an Evolution Model and an AOI over which to search for its occurrences

are selected, one or more temporal ranges can be selected. The temporal range can consist in an interval covering exactly the model duration (as appropriate for well known seasonal phenomena that can be searched at known points along the time line) or it can be an extended interval where the model will be tested for matches using the sliding window approach. The four possible outcomes of a matching result are then displayed for each pixel in a thematic map, coloured according to the outcome: Green is assigned to matching pixels ("Match"), Yellow to "Matches within tolerance", Red to "Not matches" and Black to "No data". Figure 5.6 shows result maps produced by running a draft model for detection of rice cultivations over northern Italy across 1998, as previews in the lower portion of the interface and overlaid on the Earth image on the upper portion.

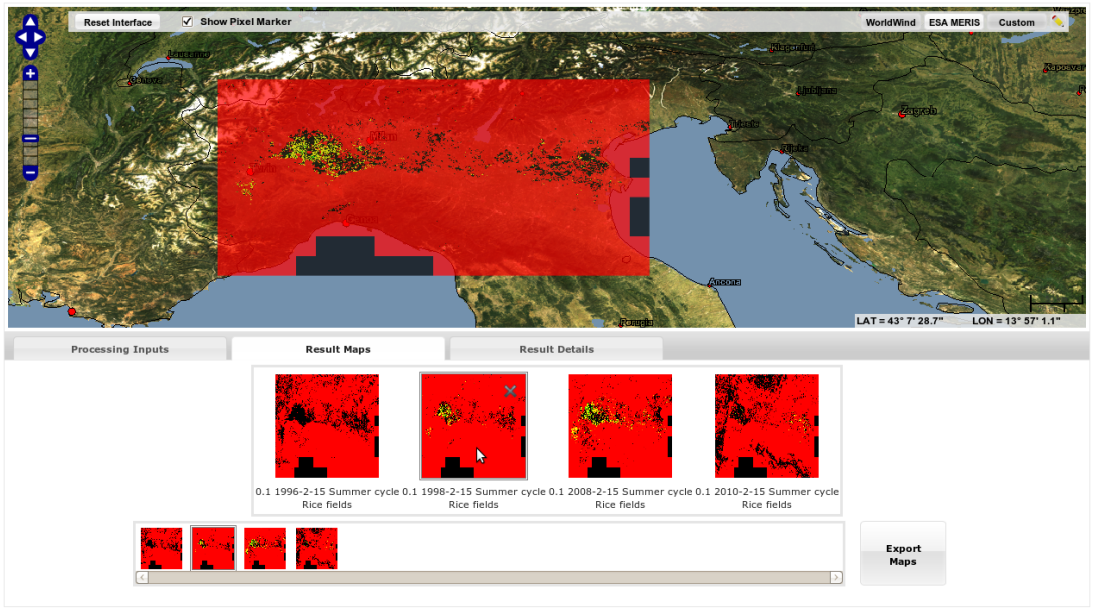
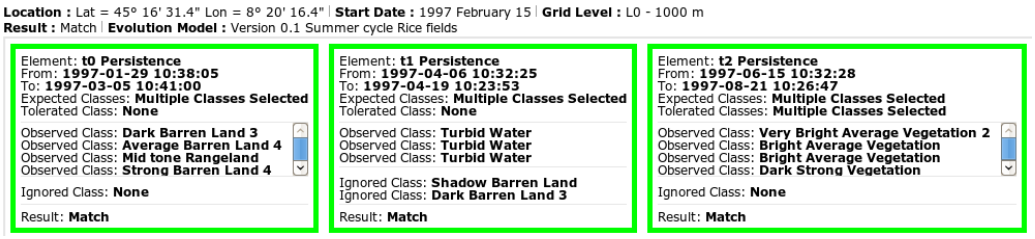


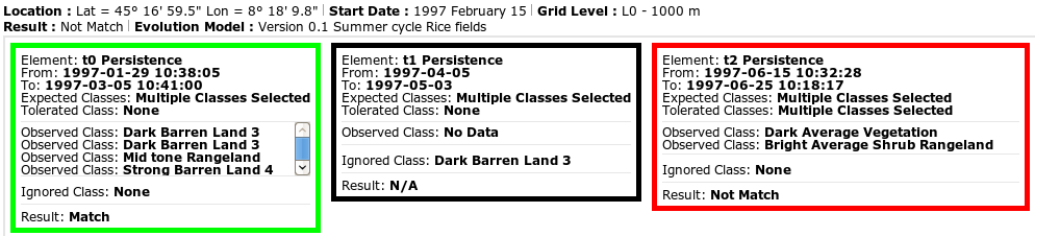
Figure 5.6: Screen shot of the model matching tab of EVAT interface with second sub-tab displaying result maps. The upper map displays results of a model over northern Italy while the lower portion shows its second tab that permits to manage result maps. (screen shot taken from version 1.3 of the system, after running a draft model for detection of rice cultivations over four different years).

Result maps displayed on the map area can be browsed and zoomed and pixel level information is displayed by hovering the mouse pointer on any pixel covered by the map. The Export function allows also to download the maps as standard GeoTIFF files (that the user can load on any external GIS application) and as a text file in comma separated values format (that can be read by an external software, for example to perform further processing based on the reported coordinates of match). Moreover,

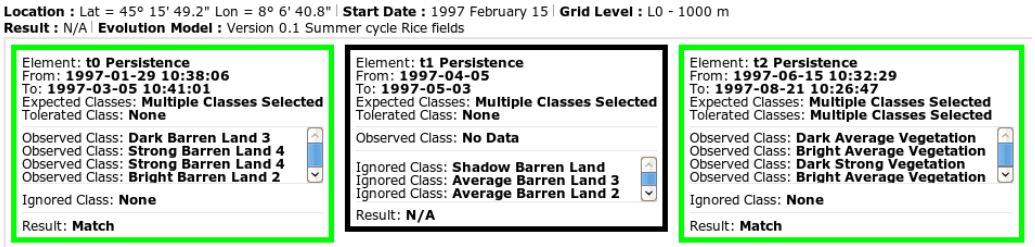
a detailed view of the model matching results at pixel level can be observed clicking on a pixel of interest on the result map. The detailed result view, shown in Figure 5.7 is then displayed in the lower part of the tab to provide a graphical representation of the evolution model as a sequence of boxes, each providing detailed information about the underlying data observed to determine the element outcome.



(a) Pixel Detail panel displaying a matching pixel.



(b) Pixel Detail panel displaying a not matching pixel.



(c) Pixel Detail panel displaying a pixel that cannot be decided (No Data outcome).

Figure 5.7: Pixel level result details provide a detailed view on the model matching outcome, allowing users to examine the underlying data, observed by the matching engine. Details are shown for (a) “Match”, (b) ”Not Match” and (c) “No Data” outcomes. Each box corresponds to a model element and provides details on its outcome and observed data (screen shots taken from version 1.3 of the system; only the details portion of the lower sub-tab of the interface is displayed).

The ”Evolution Model Matching” tab of the Expert interface is also known as the ”Visual Analysis Client”, that is the interface to the system as seen by users in the “Standard” role. It provides these users access to published models (so flagged by the authors of a model) that they can search through a catalogue and run over their selected AOI and temporal intervals to obtain result maps interactively. With this

approach, as soon as a relevant and useful model is available from the Expert users community, it can be immediately made available to be run on demand.

5.2 The model matching engine

The model matching operation that searches through the data archive for data matching a modelled evolution pattern is by far the most demanding in terms of computational requirement, since it has to provide reasonably fast responses to user requests that can cover regional to national scale areas. Following the features provided on the third tab of the EVAT interface, two distinct matching routines are provided: a distributed one that provides no detail to be fast over an area of interest that can cover several Tiles, and a detailed one that is executed on-demand by a single thread over a single pixel to provide the result details at pixel level.

5.2.1 Concurrent distributed matching over area

This section describes the model matching engine that performs distributed computation over an AOI, distributing data according to a model aimed at providing fast response to user requests for interactive on-line analysis. The implemented distribution model is depicted in Figure 5.8 where three cores are assumed to be available to the system, one on the first worker server and two on the second one. The depicted model is an example to explain the model that is then scalable to many cores across several servers with a change in the configuration of the main thread.

As depicted in Figure 5.8, the distributed matching involves the following message flow:

1. A model matching request is sent to the main web server, hosting also the user interface; that request involves one or more time ranges, an AOI, one or more Tiles and the identifier of a model to be matched;

2. The request is passed to the main tread that takes care of partitioning its spatial extent into single Tiles, preparing an empty result map to be sent back to the caller. It then executes a configurable number of local sub-region threads, passing the request for a single different Tile to each of them in smaller requests: if the number of Tiles is greater than the number of available processing cores, a list of the remaining ones is kept on the main thread that waits the completion of

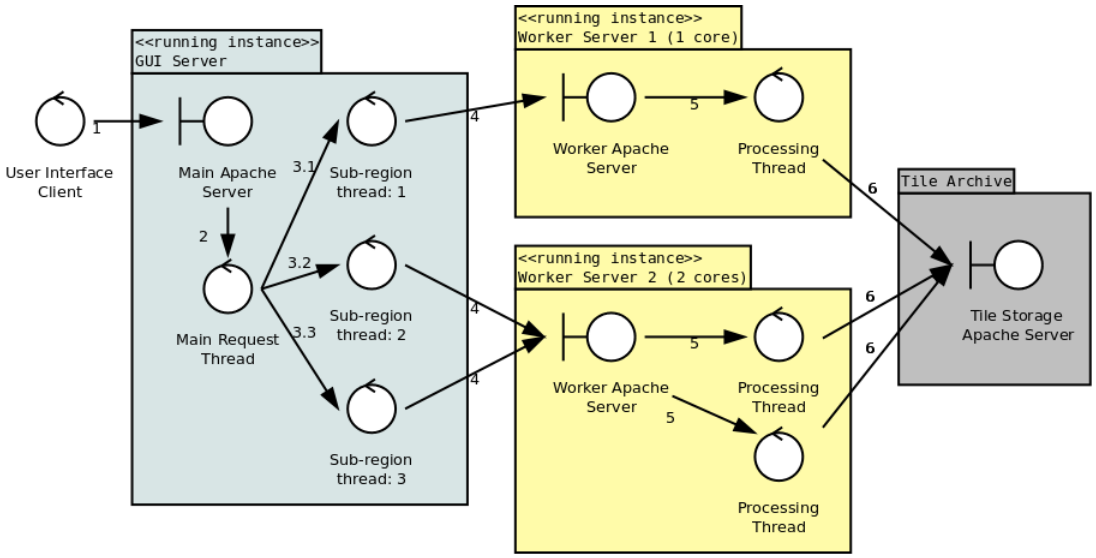


Figure 5.8: The distribution model of the model matching engine is based on server side processing, distributed over Apache Web server technology.

one of the running sub-region threads before starting the next one;

3. Each sub-region thread receives a request for a single Tile in the original AOI that it can process independently, since no communication is required with its "siblings" for a pixel-based analysis. It sends a remote processing request to an available worker server that implements the matching algorithm, operating on a single Tile;

4. Each Worker server receives independent request messages for matching and executes them on local threads to provide the result over the received sub-region;

5. Local threads execute the matching algorithm over Tile data retrieved from the Tiles archive to produce (partial) results to be sent back to the caller;

6. The Tile archive receives concurrent requests for different regions and possibly different time frames and serves the requested data to the processing threads.

The data flow is then the inverse of the message flow as the time series of maps are distilled into a single (partial) result map that is merged into a single result map by the main thread, which sends it back to the calling client. This adheres to the model of the postgresql query plan on partitioned data that is executed by the Tile meta-data database on its tables. This approach allows efficient and simple load distribution resulting in a scalable architecture for distributed processing with the only bottleneck

being the main thread in its data fusion operation, that is however done as soon as data arrives from the sub-region threads as a simple copy operation. As reported in section 6.1.4, the actual implementation is bottlenecked by the Tile Archive in serving tiles to the processing threads, leading to a linear performance vs. the number of Tiles to be actually retrieved from the storage.

The choice to use an Apache Web server as a distributed processing platform, with server side components to perform the computation posed an unforeseen control issue upon cancelling a request: there is no way to pass a connection lost event from the server to a running python engine to cancel a task. Investigation suggested that this kind of communication is probably broken at the scripting engine interface level, since a property exists to test the connection but it is never changed as seen by the running process. This issue leaves running instances of the sub-region threads running upon connection loss or user cancellation request, wasting resources. A second control issue was related to the management of resources to avoid overload (that can lead to sensible performance degradation): we tried using Apache itself as the limiting agent, by specific maximum request limit configuration. The fast processing cycle of some worker server with respect to the others however, caused resources to appear falsely available, as further threads are in fact waiting to start, leading to accumulation of queued requests. Even if that accumulation does not overload the worker nodes, it causes an overload on the server hosting main threads and unpredictable wait times for queued tasks as the concurrency level to get resource raises. To solve both issues, a database centric communication model has been implemented to:

- Keep track of running tasks and flag them as cancelled upon connection loss (detected by periodic connection polling) or user cancellation by the main thread. Interleaved status polling by the sub-region threads avoids stale tasks to complete "long" tasks;
- A table listing resources is provided, these get acquired by running tasks upon start and released upon termination. Direct resource availability checks allow accurate overload avoidance.

5.2.2 Detailed on-demand over pixel

The detailed view of the results at pixel level is backed-up by a dedicated model matching function that is run locally on the main web server to compute the matching

outcome for a single pixel, provided with additional details on each element. This separation has been done to avoid computing unwanted details for the entire AOI, thus reducing the amount of time needed for computation of matching results over large areas. A dedicate function to get pixel level details is also fast to execute on demand only for selected pixel since it is very likely to use cached data from the tile archive and performs computation only for a single pixel.

For that single-pixel analysis the result of the matching process is a series of parameters to be displayed in the result detail form on the interface, instead of a result map and this data cannot be directly exported as is the case with result maps.

Chapter 6

Results and discussion

This chapter provides an analytical report on several aspects of the presented system. Performance aspects, critical for usability of the system as an interactive tool are included, along with a description of known and intrinsic system limitations. An overview of relevant features for visual data analysis and examples of searches that can be done with evolution models are also provided. Finally, the usability and effectiveness of MEA as a tool for data exploration, analysis and with capabilities to characterize features based on temporal evolution has been assessed by a selection of end users, performing an independent evaluation. Use cases covered were monitoring of agricultural practices and detection of burned areas, along with the critical evaluation of system features. A summary of the received feedback concludes this chapter.

6.1 Performance level and response time

Since driving goals of the presented system implementation were fast pixel based classification of the entire ATSR archive and quick interactive response to users, the following sub-sections provide performance results for the four main processing functions. Measurements were taken on the system loaded by processing the 15 years of (A)ATSR data on the hardware detailed in section A.1. After the ingestion was completed, without merging multiple Tiles on the same day (and without including the improved cloud filter, introduced to reduce cloud contamination in the classification), the number of records in the database was 498.226.758, with a monthly average of 2.896.667 Tiles.

6.1.1 Classification and remapping performances

The classification system operated a first complete classification with processing elements deployed on processing nodes (pn1 and pn2) within seven months. Now process-

ing elements are also deployed on database nodes (db1 and db2). Related measured performances are reported for an average strip and extrapolated over the entire archive to provide an estimate and considerations for a second run that is foreseen with an improved classification system version and the cloud borders filter.

Processing time were taken by processing, on both machines, the strips `"ATS_TOA_1PNPDE20080114_201514_000044542065_00100_30715_7111.N1"`, and `"AT2_TOA_1PTRAL19990615_073242_000000001043_00306_21703_0000.E2"` as representatives of an average AATSR and ATSR-2 datasets respectively. The results reported in Table 6.1 highlight the difference in the processing times, due to the different processing powers of the two nodes. To be noted that about 20% of the time is spent to read, calibrate and classify the dataset, while the remaining 80% is spent to re-sample it to the simple cylindrical projection: a step that includes application of the accurate geo-location correction system implemented for AATSR.

	AATSR Processing		ATSR-2 Processing	
Machine	real time	CPU time	real time	CPU time
pn1	10m36.604s	41m56.350s	11m5.749s	39m57.290s
db1	4m17.676s	18m22.860s	4m6.646s	17m40.720s

Table 6.1: Classification and remapping processing time. There is a clear difference between times on the two machines, reflecting their difference in processing power and there is also a factor of four difference among real time (wall clock time) and CPU time reflecting the efficiency of the parallel implementation of the remapping system.

From both of the two strips, 11 valid granules were produced out of a nominal total of 20, as expected from the presence of night time acquisition in half of the covered orbit that leads to the generation of granules with no classification data (the used classifier needs daylight acquisitions). There are two database nodes available for processing that, together, are roughly equivalent to five processing nodes in terms of processing time, hence we can deduce an overall computational speed equivalent to 7 processing nodes. Considering the aforementioned strips representative of the average strip and rounding up the wall time to 12 minutes, a processing speed of about 7 strips in 12 minutes can be obtained, equivalent to $60 * \frac{7}{12} = 35 strips/hour$.

Considering a count of 76397 strips available up to all 2010, an estimate of the processing time required to process both archives is 2183 hours, equivalent to *91 days*, not considering network transfer times. To be noted that a continuous download to the local buffer allowed us to store enough data locally during periods of high bandwidth

availability to compensate for the reduced bandwidth periods, thus avoiding network bottleneck issues on the first run, that used only two processing nodes. Considering an average strip size of 350 MB, the minimum bandwidth to sustain the data rate required to continuously process AATSR data is $minBW$ = strip size × processing speed = 11.96GB/h that is more than half the nominal bandwidth capacity of a T3 link (19.20 GB/h). Considering we computed a conservative estimate of the processing speed, that bandwidth requirement makes even more evident the need to move processing close to the data in the RS field, especially when the output data size is much lower than the input one, saving network resources and avoiding network bottleneck issues.

6.1.2 Tile ingestion performances

Tile ingestion is the operation that processes classification maps (granules derived from ATSR strips) producing Tiles over the Earth Fixed Grid system at level 0. The process is executed concurrently by the Tile ingestion system that allows to modulate the resource demand on the specific hardware. In particular, optimal throughput for data ingestion is obtained when the storage device is just below its overloading threshold (i.e. under constant 100% utilization). This subsection reports results of performance tests on the ingestion system executed to determine its configuration for best throughput.

A query to the database is used to determine the current time and corresponding number of records at hourly intervals:

```
select now(), sum(granules)
       from ((select count(id) as granules from granules_aatsr)
            UNION (select count(id) as granules from granules_atsr2))
as tempTable;
```

preliminary tests, performed both on the ext4 and XFS file systems gave an estimate of around 800 granules per hour to generate and store Tiles on an empty file system. Half of the storage array was used for each file system. Over three hours we got 13 granules/minute on ext4 and 13.3 granules/minute on XFS. Ext4 was still presenting stability issues, reported in sectionA.4, at the time of testing. A second round of performance observations was taken in June 2010, with 8 concurrent processes, on an almost complete archive, leading to an average speed of 6.72 granules/minute (about 403 gr/h).

Lower performances were expected for a filled file system with respect to the early stages as the tree structures holding file meta-data grow in depth. Finally, the need to

re-process several months of data provided the opportunity to perform a thorough on field performance test. The performances attained by the Tiling system with a different number of concurrent threads are shown in 6.1. It appears that the best throughput can be obtained by using a number of processors between 3 and 5. Detailed disk sub-

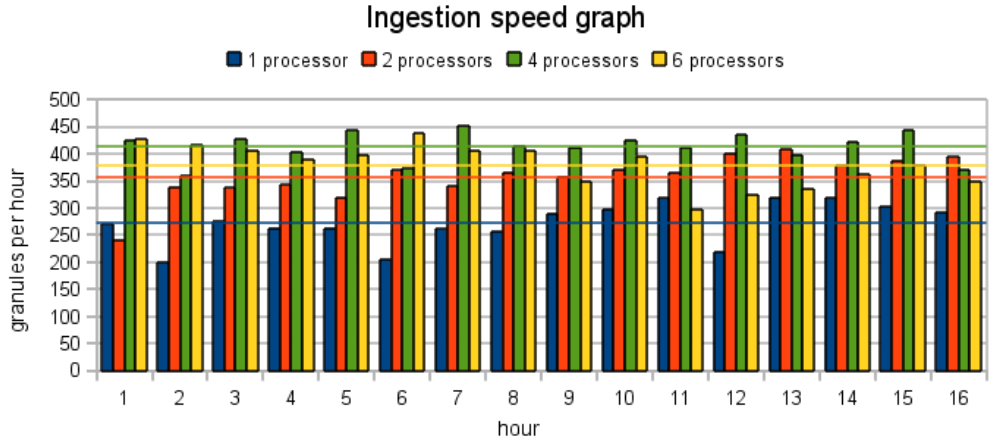


Figure 6.1: Multi-threaded ingestion: performances with different concurrency levels

system workload analysis allowed to highlight a strict dependence between disk device saturation and throughput performance level. As can be seen in Figure 6.2 using a high number of concurrent ingestion processes saturates the disk subsystem capabilities, leading to lower throughput, when compared to using a number of processes that keep the device just below saturation level.

We have seen that device saturation lowers the performance as well as its under-utilization, hence an adaptive (throttled) solution would be optimal to keep the disk subsystem close to 90% busyness, for constant optimal performance. Linux provides several systems for disk subsystem performance control, such as the "ionice" utility, paired with the CFQ I/O scheduler[53]; preliminary testing seemed to indicate that the "niceness" only applies to read requests while the ingestion process consists almost only of writes. XFS provides also a unique feature for disk bandwidth reservation: Guaranteed rate I/O system (GRIO)[54]. Investigation of these tools was outside the scope of this work so an ad-hoc configuration was chosen from the measured performances during low system load: 3 concurrent threads are set, delivering the best average of 450 granules/hour.

Considering the granule count in the ASQuLD storage of 813.664 granules as of mid February 2011, with an ingestion speed of 450 granules/hour, gives an estimate

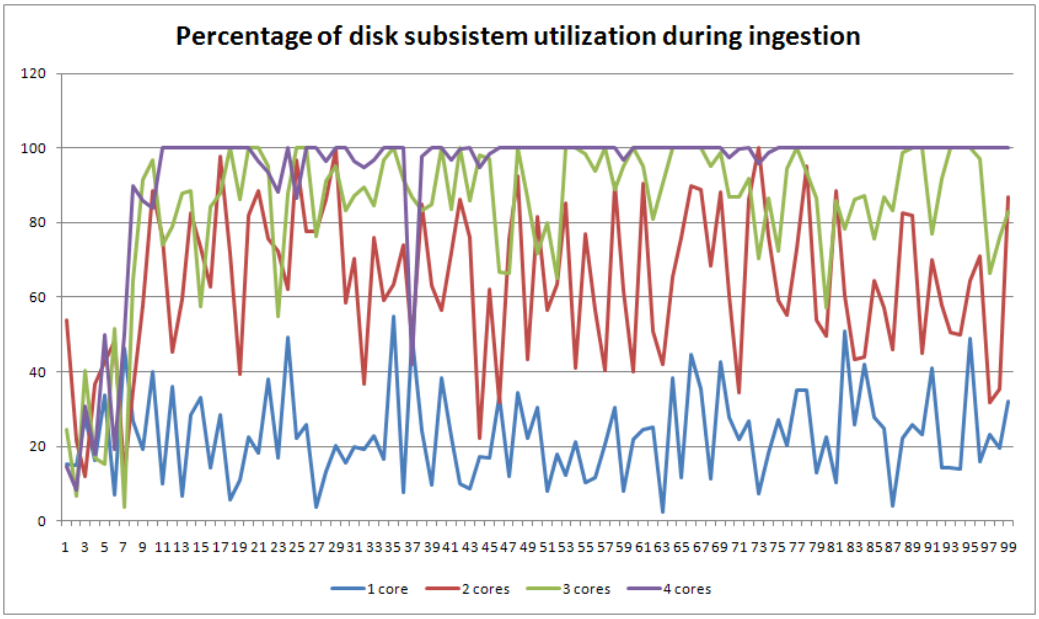


Figure 6.2: Multi-threaded ingestion: device utilization with different concurrency levels. With one and two processes (blue and red lines) the device is underutilized while with four processes (violet line) the device is at almost constant saturation level. Three processes (green line) results in optimal sub-saturation level.

of about *75 days* to re-process the entire classified maps archive. An operation that can be done with a minimal time delay with respect to classified maps production. From the aforementioned estimate on strip processing we can see that the Tile storage system would influence the lower bound of processing time for processing a full pass from the original data if just one processing node is added to the system. Bottleneck for this operation is confirmed to lie in the disk subsystem for Tile storage, since the majority of CPU time is spent in IOwait state to copy tiles in the archive. Replacement of the storage unit with a more performing one will be the best option for immediate performance gain.

Tile merge operation

Besides ordinary ingestion to load data into the archive, the Tile merge operation is also to be performed to remove (merge together), multiple observations during the same day, over the same Tile. This function has not been optimized for performance since the system can be used without it (the most recent Tile is returned in case of multiple observations) and its benefit is most relevant close to the poles. Since it provides considerable reduction of Tiles' quantity, it is worth to introduce such optimization in

future developments to allow faster merging. A test on the whole month of January 1997 has been performed: the elaboration started on Mon May 10 11:05:17 CEST 2010 and completed on Tue May 11 17:30:30 CEST 2010, taking 30,5 hours to merge 3.787.371 Tiles into 2.517.648, with a reduction of 1.269.723 Tiles from the archive, corresponding to 4.84 GB of storage space and 33% of the original Tiles' quantity. That gives a rough estimate of six months to perform merge operation on the entire archive.

6.1.3 Thematic content query performances

Recalling that the thematic content is stored in a bit string that allows to save 171 bits per record and considering 500 million Tiles, a total reduction of about 10 GB can be obtained. This can improve performance, in line with the results described in [47] for tables that can fit in the server's RAM (data is more likely to stay cached).

The developed catalogue of Tile meta-data provides largely improved performance in the identification of Tiles, as from the preliminary results reported in [47], from 1.5 seconds, over a single season with linear addressing on a synthetic database of 300 millions records, *4.5 seconds across 12 seasons* with two-dimensional addressing on the real data database with 500 millions records.

Moreover the reported time of our tests is observed after an explicit system cache clear operation with the command `echo 1 > /proc/sys/vm/drop_caches` hence in an almost cold state. Without cache clearing we obtain a query time of about *0.6 seconds* for a semantic query across 12 season over a 2000 Tiles area (approximatively an area covering France) with three semantic conditions. Given the considerable amount of RAM on the database servers the latter condition is likely to persist, after the first query over any given region, for the entire user session with the system. Other solutions to accelerate the semantic search function have been investigated, including parallel query execution; a feature not yet available in postgreSQL, up to version 9.0.3. The pgpool-II middleware[55] has been used for testing and it led to promising results in the execution times on single database instances; then an unacceptable performance loss was registered in the final step of providing the unified results, details are provided in sectionA.2.

Another thematic content query function that is demanding in terms of data access and that must be served in a reasonable amount of time for interactive analysis is the

Pixel History graph. It requires analysis of all available maps for a given Tile across the entire archive to build its land cover profile over time. This function is Tile based, in the sense that it provides faster responses over the same Tile for subsequent requests as the time series, once loaded, remains cached for the entire Tile, not only the selected pixel. For intra-Tile request, besides the first one, the time to generate the graph was always under 2 seconds for all performed tests. The time elapsed for the first request displays more variability and higher values, as shown in Table 6.2, with an average time of 5,65 seconds, however, it is still acceptable for interactive analysis. Figure 6.3 depicts the distribution along Latitude of the test pixels and elapsed times.

Elapsed Time	Pixel Lat.	Pixel Lon.
4,897588	-14,830078	24,306641
5,161291	-16,763672	15,560547
4,873162	-23,619141	17,626953
5,133647	-27,033203	28,724609
4,834967	-18,595703	47,271484
3,944556	12,166016	75,833984
4,678427	13,791016	44,017578
4,870872	14,802734	37,208984
5,816015	56,638672	13,697266
5,838758	55,583984	26,353516
4,703252	53,560547	-7,044922
6,207648	58,044922	38,392578
5,946446	58,044922	54,654297
6,300115	54,966797	69,154297
5,533276	58,130859	79,087891
5,364648	52,419922	12,291016
8,284396	65,779297	-70,501953
9,486191	63,142578	-98,451172
8,052967	60,591797	-121,568359
5,907955	40,642578	-116,556641
5,656765	30,181641	-106,802734
3,904719	16,998047	-91,947266
3,955887	-0,580078	-72,435547
4,311126	-4,974609	-48,001953
5,004018	-19,826172	-66,458984
5,481127	-42,853516	-68,658203
4,016162	-0,841797	-78,587891
8,016287	62,439453	-156,283203
7,815969	63,228516	15,806641

Table 6.2: Time elapsed for Pixel History Graph generation over 30 test site pixels. Pixels where chosen arbitrarily over land areas to represent a sparse geographic distribution across major continental areas.

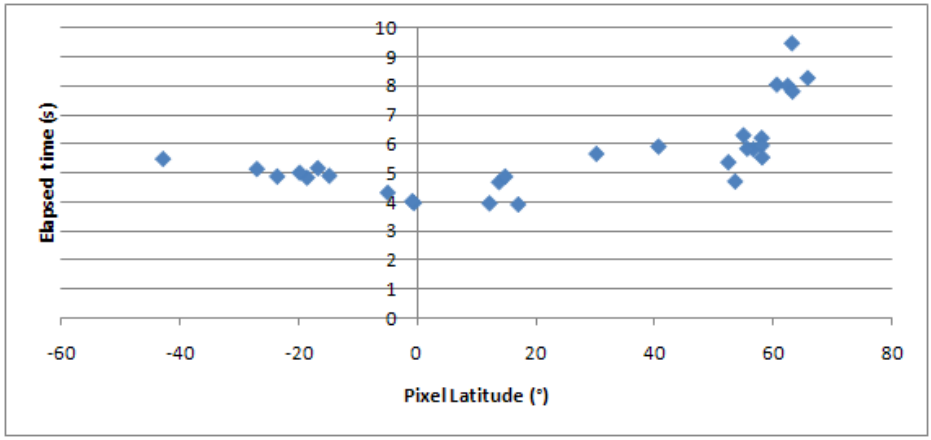


Figure 6.3: Elapsed times for Pixel History Graph generation with respect to pixel Latitude.

6.1.4 Multi temporal matching performances

The matching engine is distributed across four processing servers with a maximum availability of 38 cores to service a single request: 12 cores are enabled on database machines (db) and 7 on processing node (pn) ones. The tests reported in this subsection are performed varying several parameters for different models in an attempt to profile the engine performances. The AOI is varied from 4 to 400 Tiles, resulting in a variable area covered on Earth's surface for the same AOI size, depending on its Latitude. The area covered by a 400 Tiles AOI over northern Italy is shown in Figure 6.4. Test execution was automated for pseudo-random AOI (constrained to be selected over land) and temporal parameters (constrained by data availability) selection. For test reproducibility a fixed seed is used for the pseudo-random numbers generator.

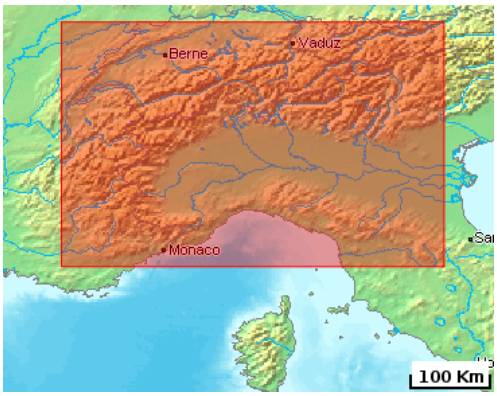


Figure 6.4: Area covered by a rectangular selection of 400 Tiles over northern Italy, Latitude about 45 degrees North.

Seasonal phenomena Matching

This performance was tested using seasonal models (tested over a fixed temporal reference for their first element) made of Elements with TT of 15 days, separated by a TSP of 60 days (resulting in the analysis of Tiles for every other month), with persistence enabled and all classes selected (full analysis of all available data). The variables are the AOI and the number of Model Elements. The results are shown in Figure 6.5. Cached results are obtained by repeating the single test four times, then averaging the execution time after discarding the first one. The cached result test is aimed at profiling the effect of the storage system hardware (disk subsystem) on model matching performances. The 8 element model, taking almost the same time to generate results

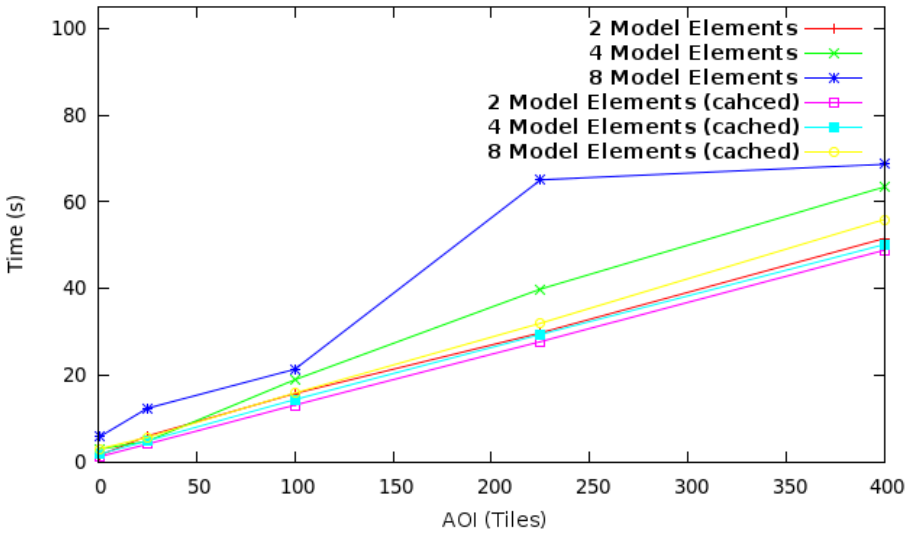


Figure 6.5: Seasonal Evolution Model matching performance. The graph shows the time required to produce a result map with respect to the number of Tiles in the AOI and the number of elements in the model. Cached results are obtained by averaging the execution time of four consecutive runs, discarding the first one.

for a 225 and a 400 Tiles AOI, reveals a strong dependency of the matching operation on the actual number of Tiles found (which is almost 20.000 for both observations). The corresponding reduction of the time with cached results highlights a linear dependence with the AOI when disk subsystem latency is removed. *An average response time of 27 seconds and an average speed of 160 Tile/sec are obtained.* A maximum response time of 70 seconds is obtained for 8 elements and more than 20.000 Tiles to process over a 400 Tiles AOI, which represent a reasonable upper limit for the expected use.

Transitional phenomena Matching

The behaviour of the matching engine is designed to reduce execution time for the matching of generic evolution models over a search window with "shortcuts" in the processing logic: as soon as a "Not match" outcome is determined for an element, the sliding window is moved forward one day (since details of all elements are provided only for on demand, pixel level requests), moreover, as soon as a "Match" outcome is found, the sliding window is moved forward of its duration to avoid multiple count of the match result. These two features allow an average transitional match to take an execution time to complete similar to the seasonal ones. Since data acquisition is the most demanding part of the matching operation in case of frequent "Not Match" and "Match" outcomes, taking 60-70% of execution time, the effect of the tiles storage bottleneck is evident. It is the single hardware element to be improved for immediate performance gain in this operation. The worst case for this matching function would be a model always returning a match within tolerance, that has not been assessed since it is not expected to be the common case for the expected use.

This performance was tested using a transitional model made of 4 Elements with TT of 15 days, separated by TSP of 60 days, with persistence enabled and an infrequent class was used to test the "Not match" timing (one test on a single tile per day). The variables are the AOI and the sliding window size (2, 4, 8 months). A test model with a 4 month window and about 200 Tiles for the AOI should represent a reasonable upper boundary for the expected system use. As expected, performances are similar to the seasonal seasonal test, with linear dependency from AOI and number of Tiles to be processed, as shown in Figure 6.6. Execution time is less than 80 seconds in all cases with an average processing speed of 150 Tiles/sec and average response time of 30 seconds. The maximum response time is 77.6 sec for 400 Tiles for the AOI and a test window sliding over 120 days; the number of Tiles found in the archive for this observation was over 16569, that is the highest of the entire test series. Again cached data tests show relevant improvement of the response time.

6.2 Extensibility and scalability

The MEA system is highly modular, designed for reuse and extensibility: any of the main components implementing the main functions can be replaced in favour of a dif-

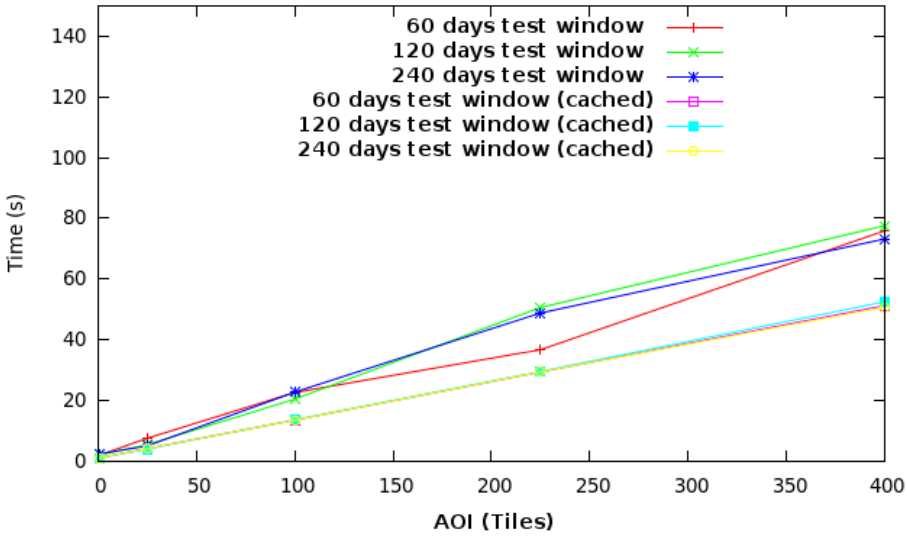


Figure 6.6: Generic Evolution Model matching performance. The graph shows the time required to produce a result map with respect to the number of Tiles in the AOI and the number of days covered by the test window (that is the difference between the time coverage of the model and the search window). Cached results are obtained by averaging the execution time of four consecutive runs, discarding the first one.

ferent solution thanks to the clear boundaries and interoperable networked interfaces between them. However, use of a different component may result in reduced functionality or inapplicability of others if the semantics or data structure are also changed in the process.

The Evolution Models Engine provides a highly reusable pattern matching engine that can operate on any bounded categorical feature set of geographic time series Tiled over the Earth Fixed Grid (EFG) and it has been developed to perform concurrent distributed computation to accelerate processing over large geographic areas using a configurable set of processing resources (in terms of nodes and cores) for task distribution. To avoid resource overload, that would increase the amount of time needed for all tasks, each node is tested for processing slot availability when distributing tasks to assign them according to actual system load; being an interactive system, if an overload condition is detected new tasks are rejected to allow faster processing of ongoing requests. The user interface to edit the pattern sequence of evolution models is also completely reusable for different thematic classifications as its categories definition is provided in a single configuration file and its selector display accounts for different levels of mutually exclusive aggregation sets. The replacement of a custom stored function, in favour of native SQL operators (bitString type functions) to ac-

cess the highly customised tile meta-data database eases also its extension to different thematic contents (with different number of classes), provided the Tile size remains of 4096 elements.

Presently, the current Tile repository is the bottleneck for processing scalability as its performance is barely adequate to the current processing power of the system. However, since it implements a file system based solution with a direct mapping layer between Tile identifiers and their storage location (file path) and since the identified bottleneck is I/O bounded (due to hard disks performances), this issue can be solved by adding an overall mapping level over different storage arrays. The further mapping layer would provide an efficient solution, allowing to utilize higher-performance (and cost) storage solutions for immediate performance gain.

With respect to data sources, classification maps coming from additional sensors can be integrated by extension of the remapping component (or by providing them directly in simple cylindrical projection) and the addition of a priority system for data sources can be added to the Tile merge function in case of "duplicate" data. Change of domain, such as reuse of the system for Particulate Matter (PM) concentration analysis, requires also a replacement (or extension) of the modelling and matching component, since the presented system is designed for nominal measurement scale with a finite set of classes, since lacks operators and functions that would prove essential for ordinal or interval scales. Current system would however be directly applicable to Air Quality Index Maps for example, to visually explore the index variation over time and search for occurrences on transitions to a particular quality class.

6.3 limitations and known issues

6.3.1 Inherent limitation in post-classification

Being the search based on Evolution Models over a classification method, it can be at most as accurate as the underlying classification system: quality of classification maps is paramount and an effort in improving them by user collaboration has been made by including the issue reporting function, inspired by the crowd sourcing philosophy, as a way to leverage the great benefits of user contribution to a system. Direct feedback on and exclusion of problematic data (at Tile level) is available to users to continuously improve the overall data archive quality, since problematic data are reported during data exploration (reduce errors) and feedback is used to identify problematic areas for

the data source or classification system (improve classification and detect problematic data).

6.3.2 Model expressive power

The evolution pattern that can be defined with the presented system, although already suitable for versatile modelling of known land cover change phenomena, has still relevant features that should be added to improve its applicability to detect a wider range of evolution patterns:

- With a fixed TSP parameter, the models are limited in temporal flexibility (no dynamic or range distance between elements can be specified). This is mitigated but not resolved by Time Tolerance of elements: a phenomenon characterised by a high variability in the relative distance among its land cover transitions over time cannot be modelled. Adding a TSP tolerance value could be a potential solution to model, for example, a burned area followed by vegetation recovery at a later (undefined) time;

- Compared to the most known patter matching system over text (regular expressions), this system still lacks capabilities that could be assessed for applicability to the thematic-temporal domain.

6.3.3 Surface calculations

The Simple Cylindrical projection permits easy representation and processing in a computer. However it has the drawback of making complex the calculations of Earth surfaces, since it is not equal-area. That issue can be addressed by using geodesic calculations behind the scene[46]: the area covered by a grid element can be roughly determined dividing by 64 the surface of the Tile containing it, computed using the geodesic length (meters) of the Tile's arc. This solution has not been implemented into the system and should be added to automatically obtain surface calculations (such as the extent of the area matched by an Evolution model in hectares). However, surface calculations can also be performed on the exported result maps, using an external tool, if needed by a user.

6.4 A versatile multi-temporal data exploration system

The following subsections provide examples of applications of the provided visual analysis features over available data, both at pixel and study area levels. Examples of Evolution Models, demonstrative of some typologies of searches that they allow to perform over the archive are also provided.

6.4.1 Visual analysis

Area level

At study area level the "Tile Time Series" frame provides spatio-temporal query with semantic filter capability to browse the evolution of the area across the temporal dimension. Among the other features of that frame, there is the possibility to filter and manipulate results to select maps most relevant to an evolution of interest. Selected maps can be viewed together, sorted by time to highlight relevant patterns, such as the time series map display, shown in Figure 6.7. The depicted area is the surroundings of the city of Vercelli in Piedmont, northern Italy, where there is a vast and diffuse rice growth agricultural practice[56]. The extent and subsequence of the various phases of the growth cycle are clearly notable using 1-Km resolution data, including flooding across April.

Pixel level

At pixel level, the provision of the Pixel History graph representation permits visual analysis of land cover patterns across several years, with the aim of showing behaviours potentially leading to insight on the underlying phenomenon, also related to preceding or following behaviour. As basic cases, the graph for a pixel in a urban area is shown in Figure 6.8, where the stable profile is evident considering the colour variability is mostly limited to sub-classes of the bare soil and rangeland macro classes (see in Figure 6.9 an excerpt of the class selector with highlighted classes, observed at different detail level, that may be used for urban typology characterization). A consistent change in data availability and typology in the winter season is also made evident, together with few outliers (with respect to the time series) represented by azure (snow), as well as green (vegetation) observations in the warm seasons.

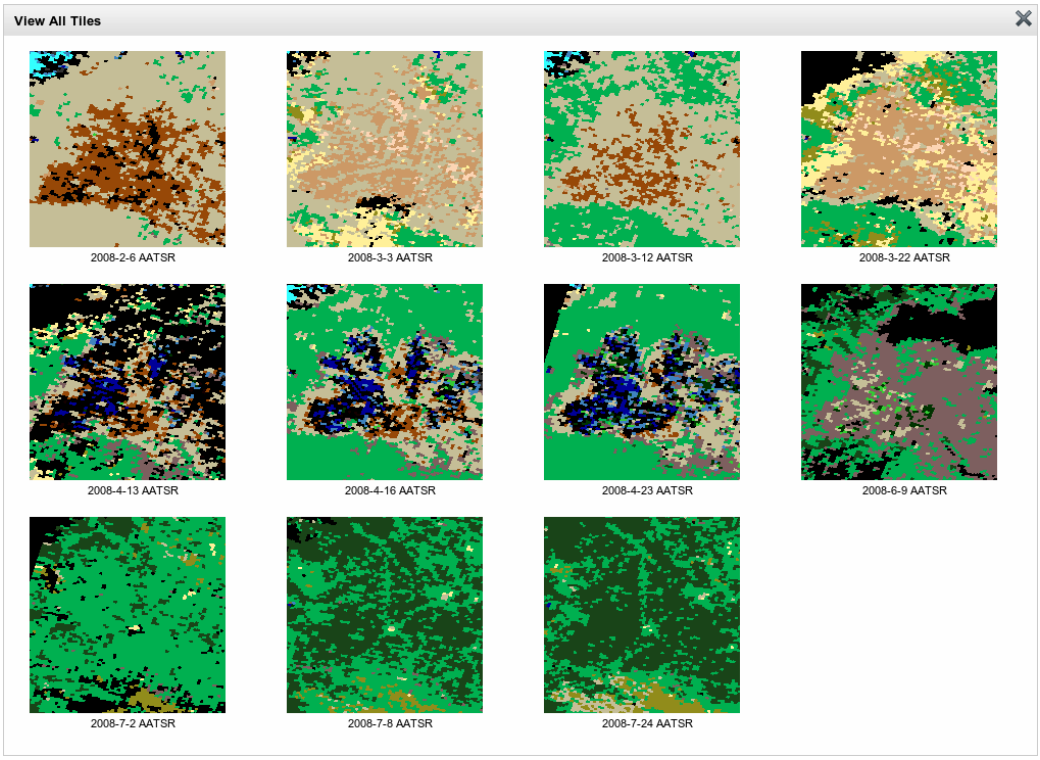


Figure 6.7: Time series view of a study area showing the area around the city of Vercelli in Piedmont, northern Italy. A major trend for the area is evident from this view and the large flooding phenomena (blue and light blue pixels in the first three maps of the second row) is in line with the known presence of vast rice fields in the area.

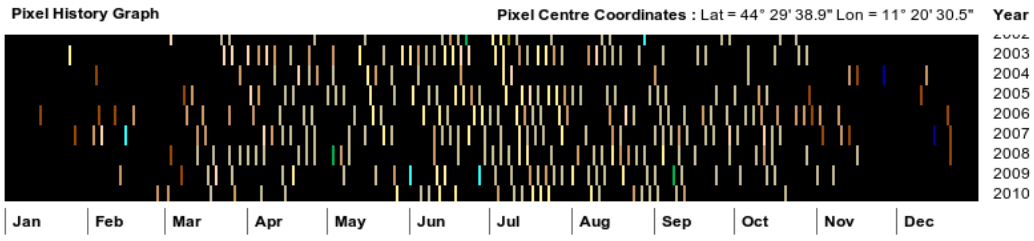


Figure 6.8: Pixel History Graph of a pixel covering a urban area in the city of Bologna, in the Emilia Romagna region in northern Italy. Colour variability is limited to specific sub-classes of the bare soil and rangeland categories across the warm and hot seasons.

(a) Classes resulting by aggregation of observations into a model element at intermediate classification level (screenshot of a portion of the class selector).

(b) Classes resulting by aggregation of observations into a model element at complete (56 classes) classification level.

Figure 6.9: Land cover classes observed for a pixel over a urban area. At intermediate classification level (a) four macro classes are crossed; at complete (and finer) classification level (b) only three very specific classes are observed outside the "bright barren land category", allowing for finer characterization.

A second example is provided in Figure 6.10, depicting the profile of a pixel subject to the effects of agricultural practice. Seasonal change patterns are made evident across all the displayed years: vegetation growth from March to May, persisting high vegetation levels from June to August then harvest in October.

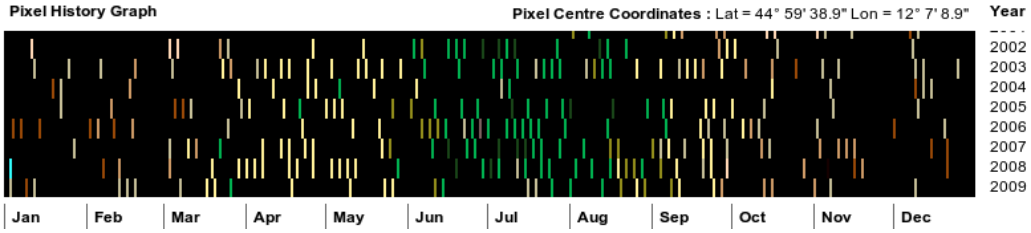


Figure 6.10: Pixel History Graph of a pixel showing agricultural practice influence. Three main phases of the change patterns are vegetation growth from March to May, persisting high vegetation levels from June to August, harvesting in October.

The Pixel History graph is also capable of highlighting disturbance events and recovery times as is the case for the graph shown in Figure 6.11, profiling a pixel over an area struck by wild fires in Greece in 2007. It is evident that the area was extensively burnt by the a wild fire event, identified by the red coloured outlier observation. A recovery period of sixteen months to return to a vegetated area is also evident for the underlying area. The definition of variable TSP models, suggested in 6.3.2 would allow automatic detection of such patterns with determination of recovery times.

6.4.2 Evolution Model examples

Several typologies of EM can be defined with the provided model editor, depending on the number of elements and the extent of the selected search window. This section presents examples of EM explaining possible uses to model expected land cover evolution behaviours.

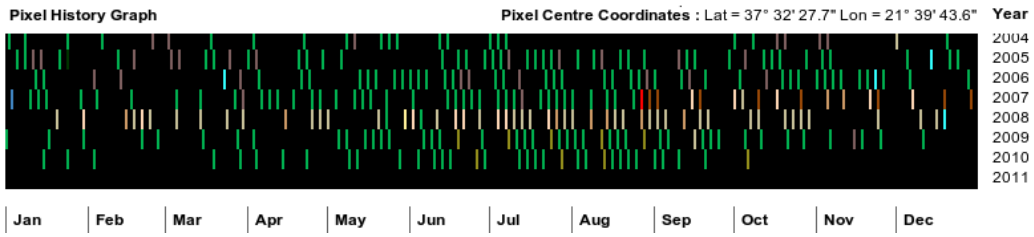


Figure 6.11: Pixel History Graph of a pixel showing disturbance and recovery phenomena relating to the wild fires in Greece in 2007. Both the extensive burn occurred in late August and the sixteen months recovery time for the area are evident.

Single element models for frequency detection

An EM defined with a single element can be used to obtain the extent of a specific land cover at a given time, it can also be used to search for periods of persisting occurrence of that land cover typology over time, such as persistent bare soil across seasons. Using a low TT the element can count occurrences of a cover type across a wide search window; an example can be the frequency of snow cover persisting for three weeks across a year

Two element models for change detection

An EM consisting of two ME provides specific change detection, these kinds of models can be called transitional if they are not bounded to seasonal references. They can be used to search for a specific land cover transition of interest in a given area, across the search window. An example of such model to detect deforestation (and any phenomenon that causes a similar transition) is depicted in Figure 6.12 where a year is set as TSP to test the land cover of three months (TT of 45 days), with respect to the same period of the previous year. With a shorter TSP, the transition may represent seasonal harvesting or candidate burned areas (for example by detailing the bare soil brightness with a finer classification).

Figure 6.12: An evolution model with two elements to detect deforestation: each element covers three months and they are one year apart to detect the land cover transition with respect to the previous year.

Three elements for contextual change detection

An EM consisting of three ME can be used to provide context to an observed class, in search for isolated anomalies in the classification, as it is the case with the EM depicted in Figure 6.13. Such model detects sporadic bare soil classification among persistent vegetation that may represent points worth investigation (e.g. accessing the original

data to better investigate its spectral properties to improve the classification system or to discover its underlying cause).

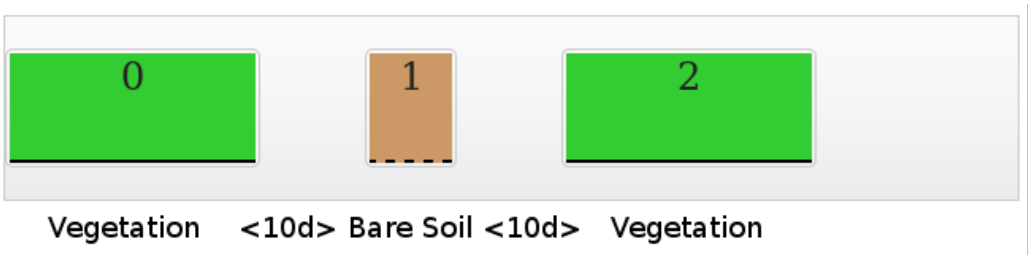


Figure 6.13: Using three elements in an evolution model to search for anomalies in classification: an isolated bare soil occurrence among persistent vegetation, if occurring frequently in the data archive can be further investigated to determine if it is a misclassification and its cause.

The SM classification system reports also outliers as a land cover class: contextualization of such observations can be used to determine their meaning or to improve model discrimination. A strong thermal anomaly is an example of an outlier observation that, if preceded by vegetation, may indicate a fire event, if then followed by bare soil may confirm a burned area. Three element models can also be used to search for a deforestation event followed by vegetation recovery and for any other two-transition events that may be of interest to the user.

Multiple elements for evolution patterns

An EM consisting of multiple ME can model a complex evolution pattern, presenting sequential change behaviour with high stability in the temporal intervals between such changes, like for the phases of crops cultivation. An evolution model designed to potentially detect irrigated winter wheat fields (as identified in the example in Figure 3.7 on page 36) is shown in Figure 6.14. Furthermore, the availability of classes identifying categories other than those strictly related to the NDVI, such as water and wetlands, allow for the definition of models for agricultural practices that foresee flooding, such as rice fields. An example of EM designed to make also use of flood detection for the identification of summer cycle rice fields is shown in Figure 6.15.

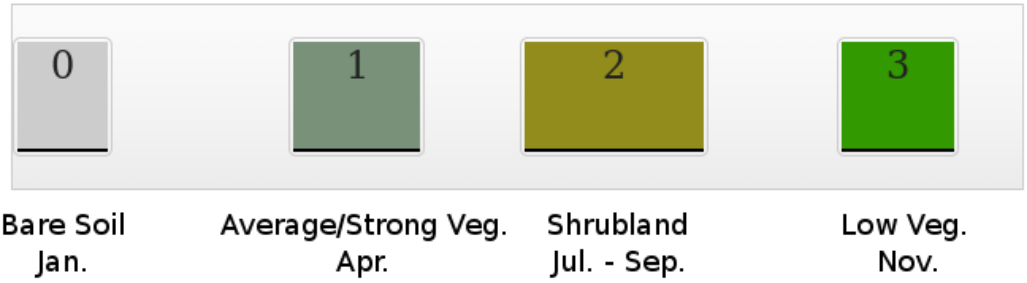


Figure 6.14: An evolution model with four elements, defined along the NDVI profile of winter wheat fields. Each element tests observations in the data archive for expected land cover types at relevant intervals in the seasonal cycle.

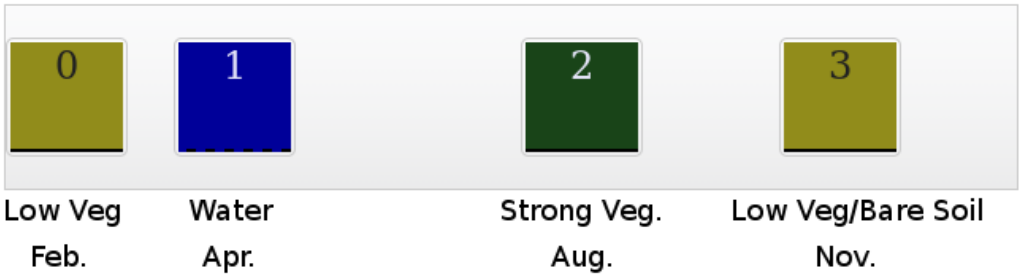


Figure 6.15: An evolution model with four elements, based on observations over the dense rice fields area around Vercelli, in northern Italy. This model makes use of the water and wetland discrimination capability of the classification to discriminate between rice and other similar cycle crop fields.

6.5 Data availability

The MEA system, loaded with 1-Km resolution data from the ATSR archives, provides a global thematic view over land cover with temporal frequency ranging from daily to several days, with a nominal rate of once in three days. A heatmap showing the geographical distribution and the count of Tiles with valid observations across the entire archive is depicted in Figure 6.16. The map is plotted on Grid Tiles coordinates on the x and y axes and, for the entire addressing space, provides points coloured according to the Tile count at that (x, y) address. The depicted count highlights the great imbalance in observations that present very high count in the $0-100$ and $620-720$ y intervals; as expected from a near polar typology of orbit of the satellite hosting the instrument, driving its swaths to overlap with subsequent orbits toward the poles. The

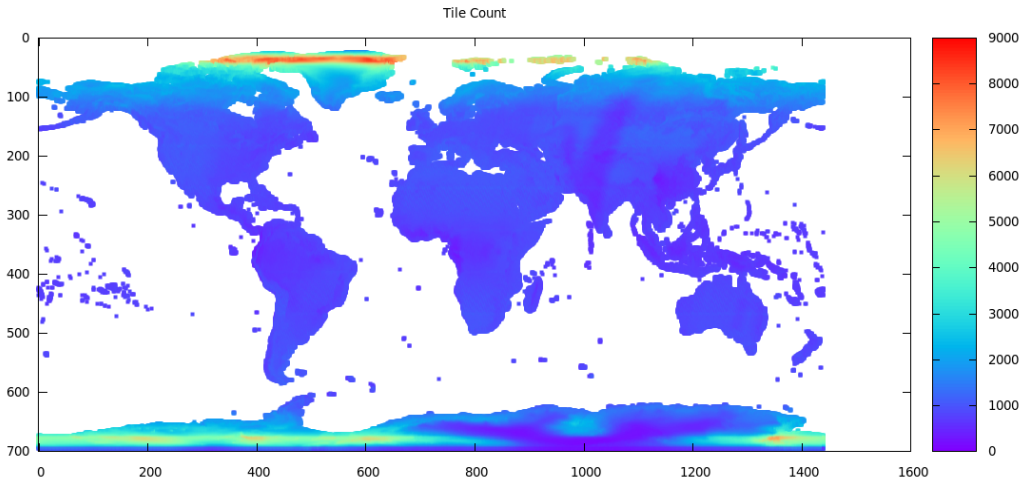


Figure 6.16: This heatmap shows tile count for each Grid Tile coordinate pair, resulting in a simple cylindrical projection with 0.25 x 0.25 degree dots, coloured according to data availability across the entire 15 years archive. This map includes all data, including multiple acquisition for a single day. The imbalance of data toward the poles is evident.

Tile merge operation has been defined with the purpose of merging together Tile maps that are acquired several times a day. The projected data availability, over the entire archive, after a complete merge operation aligning the count to the system temporal resolution of one day, is depicted in Figure 6.17. The change in data distribution is evident as we obtain a more uniform count across Latitude. Using a single day temporal resolution reduces the imbalance and reduces the data volume of about 35% by ensuring a maximum temporal frequency to once a day. The projected Tile count,

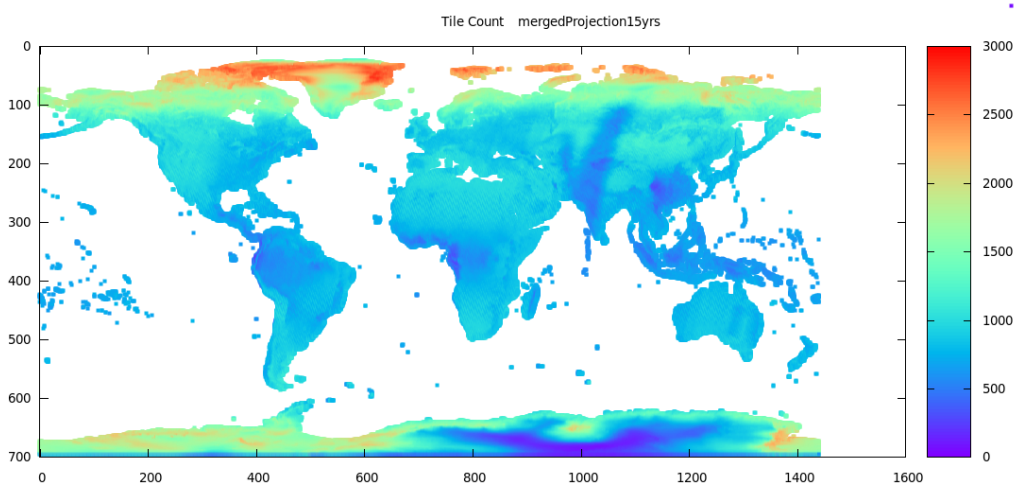


Figure 6.17: This heatmap shows Tile count for each Grid Tile coordinate pair, resulting in a simple cylindrical projection with 0.25 x 0.25 degree dots, coloured according to data availability across the entire 15 years archive. This map depicts a projection of available data at single day resolution, that is after the merge of multiple Tiles for the same day into a single one is completed. A more uniform distribution of data across Latitude is evident.

at single day resolution, for the most recent complete year (2010) is depicted in Figure 6.18, where the higher data availability provided by the AATSR instrument is made evident with respect to the ATSR-2 instrument, which provided half of the data for the cross-sensor year 2002, shown in Figure 6.19, where observations from both ATSR-2 and AATSR are available and the effect of low bandwidth strips over land is evident.

6.5.1 Quality assessment on classified data

Since the system can be at most as accurate as the underlying classification system, the detection of (known or unknown) issues in it, such as misclassification from cloud contamination (that may lead to classification of clouds as snow/ice land cover) that is here used as an example of visual quality assessment and data issue reporting.

An immediate visual identification of outliers in the time series is possible by examining the PXH and displaying the related Tile on the map, as shown in Figure 6.20. In the considered pixel (around the northern border of Greece), the presence of a single snow cover pixel among an hot season's constant vegetation cover is used to load the corresponding map of the study area for that day for area level analysis.

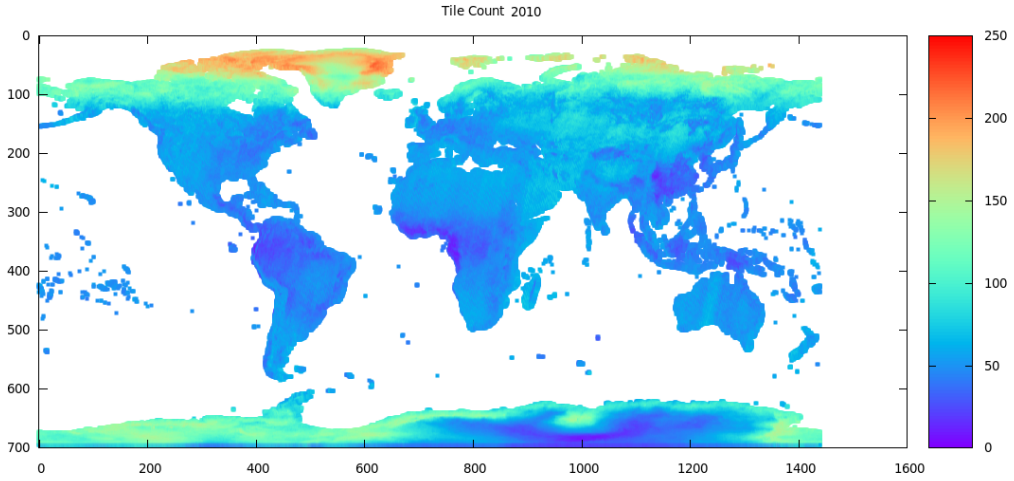


Figure 6.18: This heatmap plots tile count for each Grid Tile coordinate pair, resulting in a simple cylindrical projection with 0.25 x 0.25 degree dots, coloured according to data availability. This map depicts a projection of available data at single day resolution (after Tile merge) for year 2010, data is provided by processing AATSR datasets only.

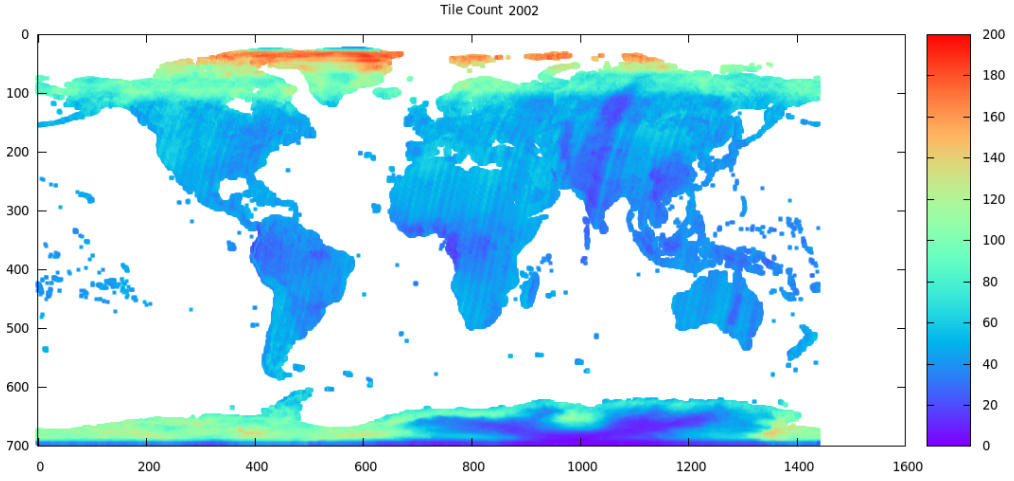


Figure 6.19: This heatmap plots tile count for each Grid Tile coordinate pair, resulting in a simple cylindrical projection with 0.25 x 0.25 degree dots, coloured according to data availability. This map depicts a projection of available data at single day resolution (after Tile merge) for year 2002, data is provided by processing both ATSR-2 (first half of the year) and AATSR datasets.

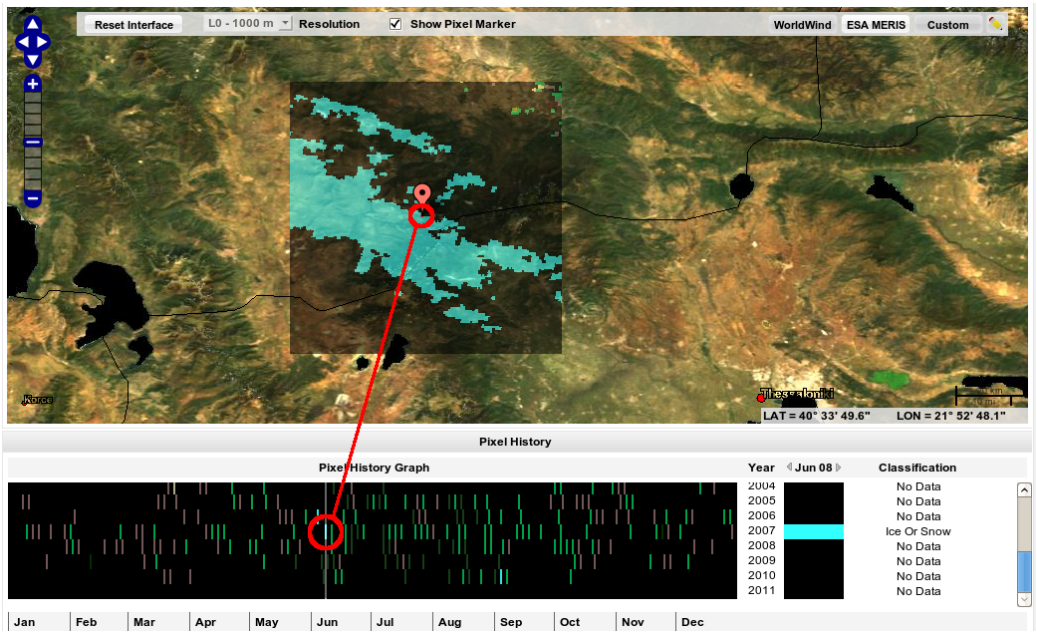


Figure 6.20: Rapid detection of outliers in time series is possible via the Pixel History graph; the circled observation represents snow cover in summer, that is quite unexpected, even across the northern boundary of Greece. Directly loading the corresponding study area map allows detection of a cloud misclassification issue, as suggested by the "No Data" (cloud pixels are discarded) surrounding the snow pixels.

Outliers and potential misclassification issues may reduce the accuracy of the pattern matching performed during multi-temporal search operation. Those issues can be immediately removed from the data archive by the "Expert" user examining the study area by reporting them to the system administrator. The thematic data provider can then provide further investigation on the issue to correct it or discover an unexpected phenomenon causing its occurrence. Use of the "Discard and Report" function to perform that operation is shown in Figure 6.21, which is also a collaborative function for data quality improvement. Further assessment of the detected phenomenon is possible

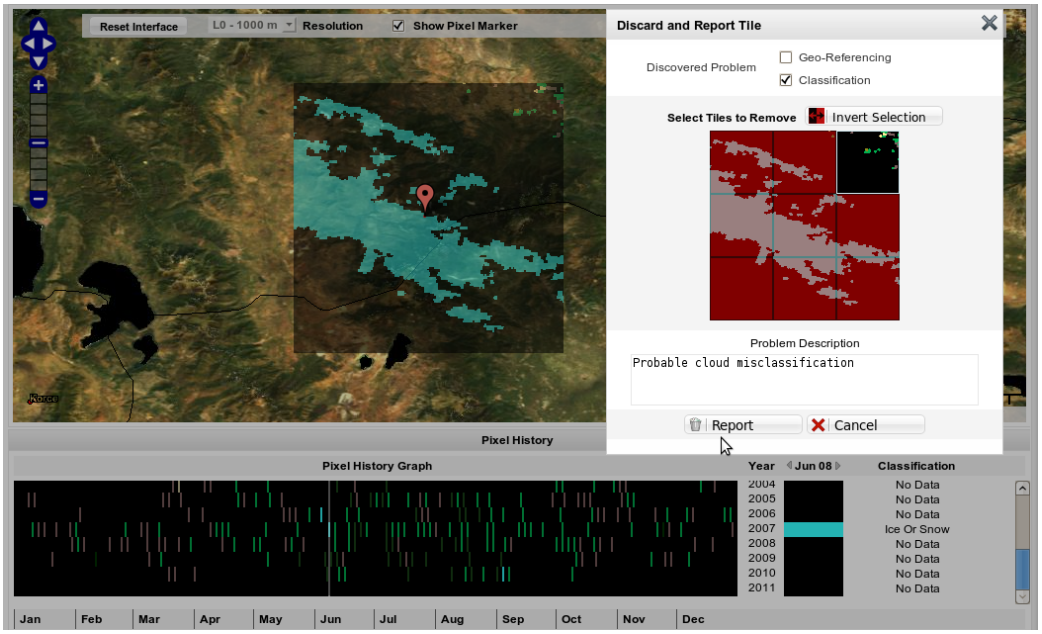
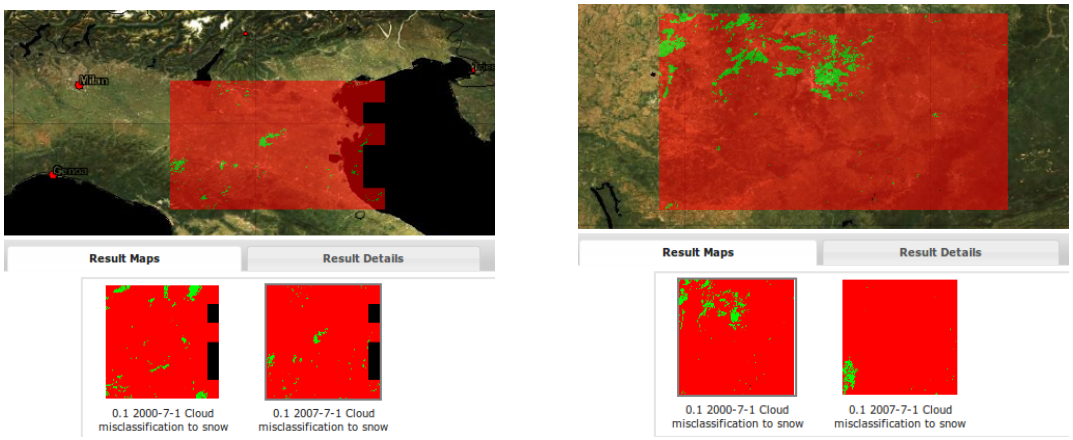


Figure 6.21: Discard and report of outliers. This function can be used to report problematic data and remove it from the data archive to continue with a study of an area without further disturbance and without requiring administrative support.

by defining an evolution model that analyses data to map occurrences of the issue over time on a given region. An evolution model to detect isolated snow elements during hot seasons can provide an example of such assessment in terms of extent and frequency over selected areas. Two selected areas are shown in 6.22 where the issue has been examined over July-August in 2000 and 2007 taking few minutes to define and match the evolution model for a preliminary assessment.

Fast interactive detection of areas affected by classification issues can be used to assess the extent of its occurrence and identify its placement in space and time. With that information further data analysis can be done in search for hypotheses on the cause of the problem (e.g. possible correlation with the underlying land cover typology



(a) Cloud misclassification over northern Italy (b) Cloud misclassification over eastern U.S.

Figure 6.22: Mapped assessment of misclassification issues extent and frequency can be rapidly obtained over an area of interest by searching for isolated or unexpected outliers with an Evolution Model, as is the case with snow cover over summer searched in northern Italy (a) and eastern U.S. (b).

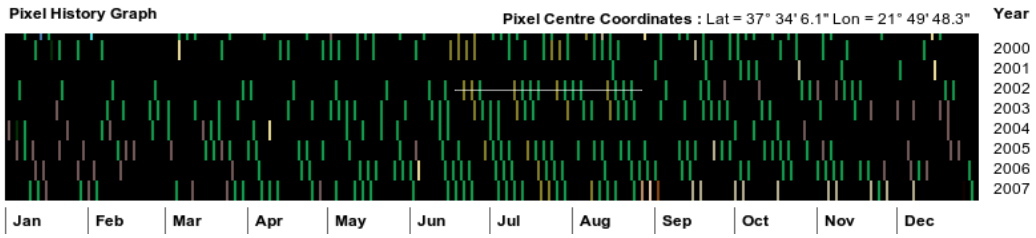


Figure 6.23: Pixel History Graph of a pixel showing systematic alternate classification. The pixel profiled in the graph shows systematic changes in the classification within few days at regular intervals (the pattern highlighted with the horizontal line) that could suggest a study area to search for correlation with its possible cause.

for cloud misclassification). The availability of the multi temporal data archive can also be leveraged to assess these hypotheses with refined models to obtain an effective subset identification for further study. Finally, the Pixel History graph profile can highlight systematic classification issues, such as the alternation shown in Figure 6.23 pinpointing a combination of location, time (or season as it appear to recur also in other years) and land cover behaviour candidate for further investigation.

6.6 Preliminary case study

The first case study used to verify the applicability of MEA as a system for change detection was performed attempting to identify flooding and vegetation growth on flooded areas over the IRAQ marshlands from 2003 to 2007. The study was based

on visual interpretation of a reference set of images from the Moderate Resolution Imaging Spectroradiometer (MODIS)

instrument (a higher resolution sensor with respect to ATSR) compared with the results from the system loaded with a selection of ATSR images taken from the thematic catalogue (the loading was performed manually, since automatic loading was not yet integrated). Two seasonal models to detect transition to water and to vegetation were used and a visual assessment of the result map done by overlaying them to the reference data. Results confirmed that the system "has proved to be a very valuable service for developing and applying land evolution models, moving from a pixel level to a wide area level of analysis and interpretation. The experience confirmed the suitability of the system toward a wide availability of data, both temporally and spatially, its speed of use and a user-friendly graphical interface."[57].

The preliminary version of the model engine allowed only to specify a single class for a "Match" results, with a fixed classification level of 56 classes. Later analysing of the results maps produced in the study confirmed that such limitation led part of the flooded areas to fell in the "Match within tolerance" outcome. After the addition of class aggregation and the possibility to select multiple expected classes, the model for flooding detection used in the study was ported to the new engine, with several classes moved to the main set. This allowed to better characterise (and confirm visually) a "Match" of the flooding, allowing to use tolerance for its intended meaning. In both cases no numerical assessment was made: however the visual comparison made evident the correct detection of the majority of the phenomenon.

6.6.1 Independent validation by user group

The system underwent an independent validation by a group of users selected among the research user community. Users already working with RS data in their fields of study were asked to perform an evaluation of the system with respect to usability metrics and to its ability to assist them in their research topics. The usability was measured according to the model of attributes defined in[25]: memorability (it is easy to remember how to perform the main tasks, even after some period without using the system), learnabilty ("the user can rapidly start getting some work done with the system"), efficiency ("once the user has learned the system, a high level of productivity is possible"), error rate (an error is intended as an action not accomplishing the intended

results, the system should present a low error rate and no unrecoverable errors) and satisfaction ("the system should be pleasant to use" in a subjective sense).

The users' focus was on the "Expert" user interface and on data analysis features, low to no attention was given to content management functions (such as permissions, administrative functions and interfaces separation for different user roles) at this stage of assessment. Overall, the system was rated as pleasant to use with an attractive and clear interface and presented no catastrophic errors. A more detailed output of the evaluation can be summarized according to the three main functions provided by the interface:

Time Series Analysis was considered very easy to learn, well structured and highly efficient to use; no error was reported for it and in resulted to have a very high memorability. The Pixel History graph was reported to be a remarkable visualization tool to display data for pattern identification. The availability of a 15 years time series at the disposal of the user (without the need to search for, collect and prepare it for temporal oriented analysis) was also notified as a highly improving feature in research activities. Improvements were also proposed, most notably the need to display the full Pixel History graph at user's request to give a complete and intuitive visual impression of the thematic behaviour of the location over time;

Evolution Model Editor together with its definition and EM concepts were evaluated hard to learn features that may require too much effort from the user to be learned without a practical (task oriented) user guide. Furthermore, the graphical display has been reported to not provide adequate temporal scale information to be intuitively compared with the patterns identifiable on the Pixel History graph. Nonetheless, once the concepts and parameters have been learnt they have been successfully used to define simple patterns to characterise searched phenomena. The accuracy levels reached however suggested that: higher resolution data should be added to the system, classification accuracy further assessed and model editing features improved toward a more intuitive interface;

Evolution Model Matching was evaluated as easy to learn and very intuitive in its results display functionality.

Besides direct feedback by the users a questionnaire was prepared to get an objective

metric to evaluate the system along the aforementioned usability attributes. The questionnaire consisted of both semantic differential and likert scales on a 1-7 rating scale. Some of the metrics listed in [25] were used to evaluate the various parts of the system and its overall performance, such as: Pleasing/Irritating, Complete/Incomplete, Simple/Complicated. All the most relevant components of the interface on all three tabs were also rated for usefulness on a four level rating scale: Very useful, Improving, Not so useful and Unnecessary. Most of the elements and functions were rated Very useful to Improving with few exceptions in the Not so useful rank (intended as not needed by the user but that may be useful to perform a different task).

In conclusion, although the group consisted of six users, that are not enough to provide an aggregation that can be considered an objective evaluation, the feedback received from the independent assessment of the system confirmed its high potential in being an aiding visualization tool to improve research activities, especially for its provision of a time oriented view on large data archives. The suitability of the system for efficient modelling and detection of phenomena characterised by simple patterns (change detection) has also been confirmed (considering the intrinsic limitation of moderate ground resolution data), while its direct applicability to provide accurate masks of studied phenomena would seem to require more validation activities (also in terms of classification accuracy). The desirability of consolidation of the thematic data archive, with the integration of more datasets coming from different sources, to mitigate data scarcity for some periods, apparently due to frequent cloud coverage, had also emerged. The possibility to provide, with the same access methodology, additional thematic views over satellite image archives, even at lower semantic level (such as vegetation indexes and surface temperature values) and their integration into the modelling system has been also confirmed to be a very promising development direction to fully exploit the data archives. Finally the need to include data at higher resolution levels emerged as the 1-Km resolution, although adequate to roughly characterize phenomena under study, was perceived by most users as a limiting factor to obtain more effective results at sub-regional scales.

Chapter 7

Future work and improvements

This chapter proposes possible solutions to overcome current limitations of the presented system and improve its usability and usefulness, considering also the feed-back received form users invited to perform an independent evaluation of its features and performances and to participate in usability assessment. Aspects of the system that are good candidates for further study and improvement are also summarized with directions for future work. It has to be noted however that almost every main component of this integrated system can benefit from further improvements and present an opportunity for further applied research, since each of them covers a different field of study by itself. The presented system, without any pretence of being the optimal solution for multi-temporal analysis, implements a working example on the possibilities offered by the synergistic integration of the diverse technologies available to the EO user community to provide an enabling framework to exploit large data archives. Both data visualization and interactivity are made available over the web via a highly usable and responsive user interface that fosters thematic-temporal pattern identification and analysis. The MEA system makes readily available to its user community an unprecedented view over a global dataset that can provide insightful information in a very short amount of time when compared to classical multi-temporal study methodologies requiring time consuming data collection and preparation activities. At its present state of development the system is readily usable as a companion tool for researchers and it is designed to already support a broader set of RS data users typologies. This system represents a building block toward a methodological consolidation of the large amount of information and knowledge already present in the EO filed.

7.1 On models and the model editor

Improve usability of the model editor

Following end user feed-back, which has to be the driving element in designing user interfaces that have chances to be really useful and adequate to meet their functional goals, the primary effort in future developments should be directed to provide an immediate, clear and intuitive display of the temporal dimension in the model editor. There are issues to overcome in providing such a functionality without losing generality of display. The presented implementation provides a display of the model elements with some proportionality to their temporal parameters: the width of an element is directly proportional to its TT value, while de distance from its preceding element is directly proportional to its TSP. Two key issues have to be managed with this linear display approach: the minimum width and the need to avoid scattering elements too far apart. A minimum width is of greater relevance considering that a more direct manipulation of the model sequence (along with the current form based input) is also highly desirable to improve usability. If we assume a minimum of three pixels for an element's area to be easily used to perform a "drag" action, and considering three possible options (extend an element to left or right and move it along the time line) a minimum of 9 pixel have to be used for its representation. With a time resolution of one day, to allow specification of elements covering a single day and to avoid overlap of elements close together, 9 pixels should then represent one day on the model editor bars; resulting in a quite impractical size of 3285 pixels required to represent a single year. Moreover, models can be defined to search for the thematic evolution across several years (such as to provide maps of urbanization changes over 2, 5 or even 10 years), in this case a "gap" would cause a discontinuity in the temporal representation (presently the issue is addressed by a limitation in the maximum distance of the elements and of their width, dynamically adjusted according to the screen width). Two possible modification to solve these issues could be the constraint to a minimum temporal coverage of three or five days for model elements, which would be reasonable considering the common practice to "aggregate" multiple temporal acquisitions (e.g. weekly or monthly averages) when performing multi-temporal analysis, and the provision of a scrollable miniature overview of the model, allowing an integral display of a "long" model that could be used to quickly move its visualization along the time line.

While several technologies already exist to implement direct manipulation of elements in a web application, future implementations should be oriented toward portability and minimal requirement of additional software components with respect to the standard browsers implementations. The development of the HTML5 specification is a promising effort toward availability of a web platform ensuring development of more direct manipulation elements with standard implementation. In the meantime, an improving addition toward immediate display of temporal information would be direct element's labelling, as shown in Figure 7.1.

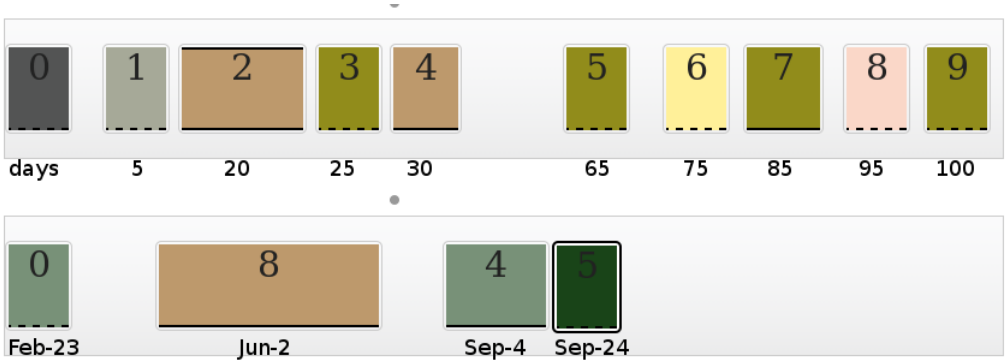


Figure 7.1: Improved time dimension visibility for model editor elements.

Data-driven alternatives

The presented user interface fosters visual pattern identification and provides a simple tool for its selection at pixel level over a single temporal interval, it provides also user controlled aggregation of sets of observed classes into model elements but still offers limited automation in direct model definition from the observed data. One if its design goals was to provide a modelling tool allowing users to precisely define patterns to search, it may however also be a limiting factor for interface efficiency not to increase the automation in model definition from selected patterns. The interactive pattern selection features give also the opportunity to test alternative modelling techniques together with the applicability of the selection features to provide an intuitive system for interactive specification of input elements to data driven techniques. The exploratory features offered by the system can thus be leveraged to ease also driven analysis (especially in the temporal domain where the selection of time windows relevant to the characterization of patterns to be modelled is one of the crucial aspects in temporal pattern recognition). Thanks to the modular design of the presented system, integra-

tion of data driven modelling into the system would be eased, enabling reuse of its thematic archive. A different model editing interface can then be displayed, depending on the model type.

Hierarchical approach automation

For mixed pixels (observations covering a varied set of features on the surface, within the same pixel), the overall observed behaviour results from the mixed contribution of all the various underlying features. The potential availability of an integrated dataset with multiple resolution data can also be an opportunity to explore options to automate multi-level searches. A multi-level model may be an effective tool to execute a fast search through the archive over large areas. Such multi-level model can be implemented by defining a raw characterization of features as seen at coarse resolution (considering possible mixture with surrounding features), then testing on search results for finer discrimination with a model suited for more precise detection (with more specificity in thematic evolution). The addition data at higher resolution layers is then required to verify this approach. This multi-scale approach could greatly reduce the time required to perform a thematic search over vast regions by quickly sub-setting the spatial domain.

The geographic location is also another candidate attribute for automatic selection of different versions of a model to detect the same searched feature that may be strongly influenced by location in its thematic-temporal behaviour. The inclusion of geographic information to the model to indicate its area of applicability, in a machine readable form (such as Geography Markup Language, which is an XML based standard for geo-referenced vector data), can enable a collaborative effort toward globally applicable models that include specific versions with increased local accuracy. For example, a model for an agricultural crop cycle can be defined at large scale and, when used to search for data, a more specific version can be automatically selected for use if available for the search area, to improve search results. A model ranking scale would complete the frame by allowing precedence to be applied for overlapping models.

7.2 On data availability

Increase interoperability

An integrated system should take advantage of standardization efforts to provide the higher level of standards-based interoperability. With respect to data archives produced by the system, interoperable elements are already provided and their standardization would be the the direction to pursue to increase system interoperability. The temporal dimension is a recent addition to supported features of existing open source spatial data management products supporting open standards, such as MapServer[58] and partly supported by others, such as GeoServer[59]. The use of standard protocols to interface the classification maps catalogue and Tiles archive, such as WMS[60] or Web Coverage Service (WCS)[61] should be investigated as further interoperability improvement. Particular care should be taken in ensuring that a standards based implementation can provide adequate performance levels for the kind of interactive browsing and visualization features provided by the presented interfaces. Nonetheless, the use of standard protocols to serve map data would permit further direct exploitation of thematic maps produced by MEA ingestion system. A second component of the system that provides an opportunity for implementation of the recently standardised Web Processing Service (WPS)[62] protocol for geo-spatial processing services is the model matching engine. Providing geospatial content as output of a processing request, parametrized over the model and the search window, this function is not suitable for the other map or content oriented protocols. It has to be noted that both elements of the system (the tiled maps archive and the model matching engine) are already accessible over the HTTP protocol with a well defined interface specification, that is the basis for interoperability. Their adherence to open standards would enable accessibility also with standard client implementations.

Leverage extensibility

The modularity of the system is the key to its extensibility, as suggested also by its users, this extensibility has to be leveraged to fully exploit the potential of the proposed system. The integration of other thematic layers of information would provide a more diverse view on a study area, increasing the potential of insight coming from exploratory data analysis. The applicability of the proposed system to provide air quality map information within the presented framework is ongoing, as well as the

implementation of different spatio-temporal display and management solutions for ordinal and numerical measurements. Other candidate thematic information layers for land cover information have been also proposed to make well known properties available to users, even if these are at a lower semantic level with respect to a land cover classification. Plain Normalized Difference Vegetation Index and Land surface temperature are examples of properties that, profiled interactively over time can foster pattern identification.

Integration of multiple sources

Time series of satellite data are an adequate tool for mapping land use through the analysis of the evolution profile of observed data over time, [12] recently confirmed the applicability of time-series MODIS 250 m NDVI data as a cost-efficient and time-efficient means for large-area crop mapping in the U.S. Central Great Plains and the MODIS sensor has been widely applied for many studies, also thanks to its free of charge distribution policy: as the next step towards implementation of a multi-resolution, global land cover analysis system, the extension of the MEA system to the MODIS archive should be pursued and the availability of a baseline dataset for specific applications can be leveraged to explore the possibility to use the high discrimination capability of the SM software to better characterize land use, while greatly increasing the data availability of the MEA system. The two datasets at similar resolution levels could also be exploited to integrate pan-sharpening applications across them as their resolution ratio would allow[63]. Moving toward higher resolution levels, the need to avoid data duplication in the storage units becomes more relevant and the integration of more efficient solutions to provide efficient storage of data, while maintaining fast interactive display capabilities should be pursued. A continued availability of high resolution satellite data will be ensured by the open access data policy for the upcoming Sentinel missions, which "ensures free-of-charge access to all Sentinel data as well as the products generated via the Internet to anyone interested in using them"[64]. The possibility to sample the data into an equal-area DGGS, that is congruent with respect to the Earth surface would optimize data storage by removing data duplication caused by projection distortions. On the other hand it will introduce the need to re-project data at the time of display request. The advantages of processing data without duplication and the use of pre-calculated "mapping tables" between two different DGGS, one used

for storage and the other for visualization is likely to provide increased performances. A comparative study to assess possible advantages of using two different discrete system for storage and display would be desirable. Existing solutions for geo-spatial data management are capable of on-the-fly projections, that is an operation usually requiring complex computations over raster data. Performance levels attainable with these solutions for temporal aggregation has not been investigated in this study, but they may be feasible at high performance levels among discrete grid system as they can be defined with fixed, direct mappings to one another.

7.3 Support to validation activities

When a data analysis system supports decision making, the accuracy that can be obtained with it becomes an even more critical aspect to be considered. Decision makers require accurate products (to a given precision, dependant on their informative needs), product validation and accuracy estimate becomes necessary to provide these users with this information. Two factors have to be considered: accuracy of the base layer used to visualize time series and to search for patterns in its evolution and the accuracy that can be obtained with a given model in search of a modelled behaviour. For evolution models, its associated metadata can be used to provide validation information in the form of a textual description, describing the output of validation activities. More integrated approached could also be followed and should be investigated, as outlined hereafter.

User feedback

The collaborative environment that can be provided by an integrated and centralized system can be further exploited, extending users' feedback collection features to the models themselves. At its present state, the system integrates features to collect user feedback on interface and system problems (problem reporting), system improvement (suggest features) and thematic data issues (report Tiles), all directed to the system maintainer. For evolution models user feedback should be directed to the model owner and include a clear rating scale to permit assessment user's impressions on it. A model may then require a minimum rating score before it can be published for use by the whole users community.

Validation data layers

The use of the OpenLayers client, supporting standard protocols, such as WMS allows for comparative display of existing thematic layers for cross-reference and validation. These layer are usually more "static" in nature, in the sense that they provide aggregate temporal information at broader temporal intervals (e.g. bi-monthly to yearly in the case of the GlobCover products[13]) but they can provide also a higher semantic level such as classification of land use in the Corine Land Cover products[65]. These thematic layers, used as background map can provide a reference for accuracy assessment of both the base thematic classification and the evolution model search accuracy in the land cover domain.

Ground control points

An improving addition would be the connection with vector data, such as ground control points or reference ground truth datasets that can be used for immediate display but should also be considered for the implementation of automated validation procedures on the base thematic data layers. Integration of vector data could also enable investigation of multi-source data fusion[66] to deliver improved content maps for decision making. Once again the ongoing standardization effort in geo-spatial data access proves to be an enabling effort: serving these vector features via Web Feature Service (WFS) would enable their prompt inclusion in the visualization system, while further analysis is required on their use for automatic validation, including their mapping to raster data in the system archive and to its thematic content.

Chapter 8

Conclusions

Archives of satellite data contain large amounts of information that can be exploited both for research activities and decision support. Accessibility to satellite data is a key requirement for researchers that use them to extract meaningful information to gain knowledge about observed phenomena. To exploit such information an interactive system for thematic-temporal analysis has been proposed and a working implementation realized to confirm its feasibility. The MEA system provides a foundation for further development toward an integrated system to deliver information to users in a way that fosters exploratory data analysis. This implementation, loaded with a global scale dataset has been used to assess on the field its provided benefits and its usability.

In this thesis I have proposed a series of graphical interfaces specifically designed to provide a thematic view over the temporal domain to users interested in studying the dynamic behaviour of the Earth's surface. To give a dynamic view of the data, an interactive environment is provided, that requires implementation of an efficient spatio-temporal query and storage system, designed for fast identification and retrieval of data. On modern commodity hardware, the system delivers adequate performances for interactive analysis allowing the execution of complex spatio-temporal-thematic queries within few seconds to identify data units and deliver access performances close to hardware limits over an archive of about five hundred millions data elements. The set of provided data access and analysis features is thus suitable for on-line, interactive use with a response time of few seconds for time series analysis elements, and an average performance of less than 6 seconds to profile a pixel's evolution over 15 years.

The user interfaces are designed not only to foster identification of thematic-temporal patterns over time series of data but also to define models to automate detection of these patterns in the archive. I have proposed a basic pattern matching

system that can be used to define expected behaviour of the classes in time at pixel level. Being based on a sequential pattern definition, such system has several known limitations in its expressive power but has also the advantages to allow the implementation of a fast, distributed matching engine, suitable for interactive use and to let the user directly define models in terms of expected thematic content over time, enabling direct investigation of search outcomes at pixel level directly on the graphical user interface. The implemented system can be used to perform thematic evolution searches over regional extent areas and possibly larger areas, depending on the temporal coverage of the model. For large scale searches (national areas and above) the system can still be used but the response time would not be adequate for interactive mode as it can take minutes to deliver results. Over small study areas, this interactive pattern matching system and its detailed results analysis function can be used to quickly define and verify models based on the behaviour of selected study areas, either based on knowledge of the area or assess the occurrence of an interesting observed behaviour to investigate its cause. Or even simply to test and verify patterns over the data archive to explore its content or search for problematic areas for the classification system that may uncover issues in the source dataset. The data reporting function enables also direct feedback from the users on data quality, that is a valuable asset for its continuous validation and improvement. Furthermore, the potential of this interactive pattern matching system is to deliver masks of any modelled phenomenon at the user's request that can be a practical surveying tool to be used for decision making, for the identification of interesting areas for further study or for further processing with external systems. The presented system is thus a tool provided to its user community to directly support research activities and to enable prompt usability of its products to the wider public.

I have designed a highly modular system for the implementation of a prototype adequate to demonstrate and assess the potential advantages of the principles of usability and exploratory data analysis, applied to an interactive data access system over satellite data at global scale. All the data handling components provide an interoperable (although yet non standard) interface accessible over the HTTP protocol, allowing for almost transparent technology changes in any of the components. Finally the system is ready to support multi-resolution data and is easily extensible to other thematic domains. A discrete global grid system has been used to define a multi-scale Earth

Fixed Grid enabling system extensibility to multiple sensors at different resolutions. The verification of this solution for cross-resolution data analysis has not been verified in this work but its support has been coded into the various component of the system to ease its implementation in future work. Furthermore an authorisation policy model and a data processing management systems have been defined that can be applied to other systems accepting editable user provided content, allowing multi-user access and dealing with data archives with long temporal coverage.

EO data users, participating in validation activities and asked to provide an independent assessment of the system for measurement of its usefulness, agreed on the suitability of such a data visualization system to provide an insightful view of the dynamics of phenomena that can be characterised over time. The model editor interface has been reported as the only component to require improvement for a higher degree of usability. It is required to flatten its learning curve by securing the user orientation on the temporal dimension and providing direct manipulation of the graphical representation of the model to configure its parameters. Nonetheless, the modelling functionality has been reported adequate to quickly model (once the user familiarises with the interface) and detect relevant patterns in the thematic classification[1].

The availability of a large scale archive of thematic data, already mapped to a reference system permitting temporal analysis has been also valued for its potential to greatly reduce the data gathering and preparation phases required in almost every study involving satellite data. Even if direct access to raw data is not enabled, the availability of one (and possibly more) thematic layers (including well known parameters, such as the NDVI), directly browsable toward the time dimension has been recognized as one of the most improving features of the presented system. The Pixel History graph, in particular, has been reported as a very insightful way of displaying classification data for pattern detection.

Its user interfaces are designed to permit users to access the underlying data with two different models of interaction: an exploratory analysis to visually identify relevant pattern in the classification over time and a model based analysis to define an expected land cover evolution over time, that can then be automatically searched for occurrences

[1]The data used for this implementation has still unresolved issues in geo-location that may require the definition of a systematic co-registration process and provide a radiometric detail that lowers the accuracy attainable by the selected classification system, yet promising results are already attainable with the analysed data.

through the time series with pattern matching. For both models, the same interface layout is used as it fosters adherence to the three main steps in research: examine data (visual analysis), make an hypothesis (model definition), hypothesis testing (matching of the defined model), then further verification of its results (export and off-line verification). Results provided can be used both to gain knowledge of the modelled phenomena and to obtain specific thematic views of an area under study.

The ubiquity of World Wide Web and the recent development of web 2.0 technologies are basic conditions to ease keeping the computation close to the data, while interactively deliver results to remote users. All the interface features of the system are made accessible over the web, fostering the abstraction of underlying technology to the user, in line with the Application as a Service (AaaS) paradigm of cloud computing. Web 2.0 is the key element in obtaining the desired high level of interaction and ease of use on the interface, without requiring installation of specific software packages on the client system: a standard modern web browser is all that is needed to access the application. The modular design of the system defines all components with clear function separation and each component presents networked interfaces based on standard protocols (HTTP and SQL connectors), making the system ready to scale up by adding computing and storage resources. The advantages of web based and distributed platforms applied to the RS field are thus made evident. Finally, when compared to existing web-based GIS solutions, MEA provides an unique, multi-temporal, thematic view over a world-wide archive of RS data, that is made available to the scientific community to foster data exploitation.

During the last three years I have been working on the analysis and development of data access and processing elements in the framework of the ESA ground segment, working with emerging technologies and systems in support of research and satellite data exploitation. I have been in contact with both satellite data users and service maintainers while studying systems for, data access, data processing, systems design and integration practices and theoretical background in Remote Sensing for Earth Observation. I have realised the potential of systems integration to deliver a user oriented framework to foster data exploitation. During my PhD studies I have been deeply involved in all phases of the realization of the presented system applying a methodological approach to the design, verification, assessment and validation of all system components. Performing investigation on critical aspects and research on the various fields

covered by an integrated system that crosses all the main activities in satellite data processing: data storage and archiving systems, Earth Observation and geo-spatial data processing elements, parallel and distributed processing models, concurrency and resource management, database optimization, data access and presentation, user interfaces design, validation and assessment methodologies. I have also coordinated the installation and configuration of the system at ESRIN premises, dealing with constraints and requirements related to the deployment of hardware and software systems in an operational environment of an international agency and its contextual interaction with other systems.

The presented system provides a building block toward a methodological consolidation of the large amount of information and knowledge present in the EO filed, made available to the EO scientific community. It provides a highly modular design supporting extension to different data sources and thematic domains. Provided the features to be analysed are in nominal finite set domain, the pattern definition and matching system can operate with a simple change in its configuration, while the clear interface model can be used to dynamically integrate other data visualization tools for different data types.

Appendix A

Additional Details

This chapter contains relevant details about some of the elements of the system that have been assessed during system development and the hardware configuration used for performance measurements.

A.1 hardware system details

This section provides details on the hardware system where the implementation of the system has been deployed. Three kind of nodes plus a direct attach storage units are used. The Operating system on all nodes was "Ubuntu 10.04.1 LTS" Linux distribution.

Application server and central storage: STOR

This node is the core of the system, hosting the web application server, and all processing coordination components; its hardware characteristics are provided in Table A.1. Connected to an internal SAS controller is a Sun StorageTek™ 2530 SAS Array, providing storage for the Tile repository. It holds 8 x 1TB 3GB SATA 7,2 Krpm drives in Raid 5 configuration with 1 hot spare, for a total storage capacity of 6 TB.

Subsystem	Configuration	Total capacity
CPU	2 x Intel Quad-Core Xeon E5310 @ 1.60GHz	8 procs
HDD	6 x 3.5" SATA 500 GB 7.2krpm GB RAID5 + 1 spare	2 TB
RAM	8 x 1024 MB DDR2 667 MHz	8 GB
NET	2 x Dual 1 Gbit NIC (4 x 1Gbit ports)	1 x 2 Gbit/s (2 ports)

Table A.1: Application server and storage controller hardware.

Processing Node: PN

There are two of those systems that are devoted to provide processing power for distributed processing elements. Hardware configuration is provided in Table A.2.

Subsystem	Configuration	Total capacity
CPU	2 x Intel Quad-Core Xeon E5310 @ 1.60GHz	8 procs
HDD	4 x 2.5" 73 GB 10krpm GB RAID5	227 GB
RAM	4 x 512 MB DDR2 667 MHz (2 GB)	
	4 x 1024 MB DDR2 667 MHz (4 GB)	6 GB
NET	1 x Dual 1 Gbit NIC	1 x 2 Gbit/s (2 ports)

Table A.2: Processing Node hardware.

Database server

There are two of those systems, whose main function is to provide high performance databases for fast Tile metadata query. They are also used to provide processing power for distributed processing elements with lower priority. Hardware configuration is provided in Table A.3.

Subsystem	Configuration	Total capacity
CPU	2 x Intel Quad-Core Xeon X5560 (2.8GHz)	16 procs (HTT)
HDD (system)	2 x 146 GB SAS (RAID1)	146 GB
HDD (database)	5 x 32 GB 2.5-Inch SATA Solid State Drive (RAID 0)	160 GB
RAM	12 x 4096 MB RDIMM DDR3-1333 MHz	48 GB
NET	2 x Dual 1 Gbit NIC (4 x 1Gbit ports)	1 x 2 Gbit/s (2 ports)

Table A.3: Database node hardware.

A.1.1 Solid State Disk performances

This section reports on SSD device performance tests. Several configuration have been tested in search for optimal device configuration but none of them delivered the expected nominal random read performance levels. Since the cause has not been discovered, results are reported here for reference and further study. The fio[67] benchmark tool has been used, which comes with a specific solid state device test suite: the default configuration options for SSD testing delivered with fio packaged distribution were used.

As expected from the drives specification[68] a single drive should deliver 35000 IOPS at 4k data size, for a 136.71 MB/sec bandwidth with random reads. A 5 units RAID 0 configuration should be able to deliver five times the performance of

a single drive, hence the expected nominal bandwidth performance is: $5 * 136.71 = 683 MB/sec$.

Considering only one physical connector (i.e. four distinct SATA 3 Gb ports at 300MB/sec each) can be connected to the drive bay due to cabling constraints on the SAS/SATA controller for the Fire X4270 Sun servers, the maximum theoretical bandwidth for the drive subsystem is: 4 * 300 MB/sec = 1200 MB/sec. For random reads the full SSD potential of 683 MB/sec should then be exploitable, while for sequential reads the higher limit of 1250 MB/sec would not be attainable.

Test results

Table A.4 summarizes test results with several configurations. None of the tested configurations provided performance levels close to the theoretical performances. Partition misalignment is known to be a common factor of performance degradation for solid state drives and the hardware controller may be a cause of misaligned reads if it writes unaligned meta-data at the beginning of the devices. Other sources of performance degradation should be searched along the data transfer chain (e.g. in the controller firmware routines themselves, in the kernel driver). Software RAID was configured with the standard Linux kernel software RAID driver (md) after configuring the hardware controller to show the SSD drives as simple volumes.

Number of Devices	Raid type	Partition alignment (sec)	Stripe unit (KB)	IOPS	Bandwidth (KB/s)	Queue Depth
5	Hardware	default	256	9765	39060	4
5	Hardware	128	64	8352	33412	4
5	Hardware	640	64	-	-	4
5	Hardware	0	64	9150	36604	4
5	Hardware	0	64	14829	59319	32
1	Simple vol	0	n/a	8241	32965	4
1	Simple vol	0	n/a	12702	50810	32
5	Software	0	128	7363	29456	4
5	Software	0	4	8672	34692	4
5	Software	0	1024	7326	29307	4
5	Hardware	default	256	11303	45215	4
5	Hardware	0	16	11081	44327	4
5 elevator=noop	Software	0	1024	6659	26638	4

Table A.4: Bandwidth and IOPS for different SSD configurations (4k reads with libaio) The second half of the table shows results with hardware controller read cache disabled.

Hardware details

Ubuntu distribution with Kernel:

`Linux 2.6.32-22-server #36-Ubuntu SMP x86_64 GNU/Linux.`

The disk controller, as seen by the system:

```
$ sudo lspci -vv | grep -A 29 13:

13:00.0 RAID bus controller: Adaptec AAC-RAID (rev 09)
Subsystem: Sun Microsystems Computer Corp. Device 0286
Control: I/O+ Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop-
ParErr+ Stepping- SERR+ FastB2B- DisINTx-
Status: Cap+ 66MHz- UDF- FastB2B- ParErr- DEVSEL=fast >TAbort-
 <TAbort- <MAbort- >SERR- <PERR- INTx-
Latency: 0, Cache Line Size: 256 bytes
Interrupt: pin A routed to IRQ 26
Region 0: Memory at fae00000 (64-bit, non-prefetchable) [size=2M]
Expansion ROM at fad80000 [disabled] [size=512K]
Capabilities: [98] Power Management version 2
Flags: PMEClk- DSI- D1+ D2- AuxCurrent=0mA PME(D0-,D1-,D2-,D3hot-,D3cold-)
Status: D0 PME-Enable- DSel=0 DScale=0 PME-
Capabilities: [a0] Message Signalled Interrupts: Mask- 64bit+ Queue=0/1 Enable-
Address: 0000000000000000  Data: 0000
Capabilities: [d0] Express (v1) Endpoint, MSI 00
DevCap: MaxPayload 512 bytes, PhantFunc 0, Latency L0s unlimited, L1 <1us
ExtTag- AttnBtn- AttnInd- PwrInd- RBE+ FLReset-
DevCtl: Report errors: Correctable+ Non-Fatal+ Fatal+ Unsupported-
RlxdOrd+ ExtTag- PhantFunc- AuxPwr- NoSnoop+
MaxPayload 128 bytes, MaxReadReq 512 bytes
DevSta: CorrErr+ UncorrErr+ FatalErr- UnsuppReq+ AuxPwr- TransPend-
LnkCap: Port #0, Speed 2.5GT/s, Width x8, ASPM L0s, Latency L0 <128ns, L1 unlimited
ClockPM- Suprise- LLActRep- BwNot-
LnkCtl: ASPM Disabled; RCB 64 bytes Disabled- Retrain- CommClk-
ExtSynch- ClockPM- AutWidDis- BWInt- AutBWInt-
LnkSta: Speed 2.5GT/s, Width x8, TrErr- Train- SlotClk+ DLActive- BWMgmt- ABWMgmt-
Capabilities: [90] Vital Product Data <?>
Capabilities: [100] Advanced Error Reporting <?>
Kernel driver in use: aacraid
Kernel modules: aacraid
```

The Kernel driver configuration:

```
$ sudo modinfo aacraid
filename:       /lib/modules/2.6.32-22-server/kernel/drivers/scsi/aacraid/aacraid.ko
version:        1.1-5[2461]-ms
license:        GPL
description:    Dell PERC2, 2/Si, 3/Si, 3/Di, Adaptec Advanced Raid Products,
HP NetRAID-4M, IBM ServeRAID & ICP SCSI driver
author:         Red Hat Inc and Adaptec
srcversion:     1BF4B6F7B8DDD726B5421FC
alias:          pci:v00009005d00000288sv*sd*bc*sc*i*
[...]
alias:          pci:v00001028d00000001sv00001028sd00000001bc*sc*i*
depends:
vermagic:       2.6.32-22-server SMP mod_unload modversions
```

The disk controller as seen by the kernel driver:

```
------------------------------------------------------------------------
Controller information
------------------------------------------------------------------------
   Controller Status                        : Optimal
   Channel description                      : SAS/SATA
   Controller Model                         : Sun STK RAID INT
   Physical Slot                            : 1
   Installed memory                         : 256 MB
   Copyback                                 : Disabled
   Background consistency check             : Disabled
   Automatic Failover                       : Enabled
   Global task priority                     : High
   Stayawake period                         : Disabled
   Spinup limit internal drives             : 0
   Spinup limit external drives             : 0
   Defunct disk drive count                 : 0
   Logical devices/Failed/Degraded          : 2/0/0
   ------------------------------------------------------
   Controller Version Information
   ------------------------------------------------------
   BIOS                                     : 5.2-0 (16732)
   Firmware                                 : 5.2-0 (16732)
   Driver                                   : 1.1-5 (2461)
   Boot Flash                               : 5.2-0 (16732)
```

The SSD Drives:

```
      Device #2
         Device is a Hard drive
         State                              : Online
         Supported                          : Yes
         Transfer Speed                     : SATA 3.0 Gb/s
         Reported Channel,Device(T:L)       : 0,10(10:0)
         Reported Location                  : Enclosure 0, Slot 2
         Reported ESD(T:L)                  : 2,0(0:0)
         Vendor                             :
         Model                              : SSDSA2SH032G1SB
         Firmware                           : 845C8855
         Size                               : 30517 MB
         Write Cache                        : Disabled (write-through)
         FRU                                : None
         S.M.A.R.T.                         : No
         S.M.A.R.T. warnings                : 0
         NCQ status                         : Disabled
```

Logical device information:

```
Logical device number 1
   Logical device name                      : ssdArray
   RAID level                               : 0
   Status of logical device                 : Optimal
   Size                                     : 151990 MB
   Stripe-unit size                         : 16 KB
   Read-cache mode                          : Disabled
   Write-cache mode                         : Enabled (write-back)
   Write-cache setting                      : Enabled (write-back)
   Partitioned                              : No
   Protected by Hot-Spare                   : No
   Bootable                                 : No
   Failed stripes                           : No
   Power settings                           : Disabled
```

```
-----------------------------------------------------
Logical device segment information
-----------------------------------------------------
Segment 0                                : Present (0,10)
Segment 1                                : Present (0,11)
Segment 2                                : Present (0,12)
Segment 3                                : Present (0,13)
Segment 4                                : Present (0,14)
```

A.2 Parallel concurrent queries tests

This section provides preliminary results obtained by experimentation with a mixed approach to leverage concurrency and parallelism for improved query performances. The basic idea was to take advantage of multiple cores (a feature not available in current versions of postgreSQL, up to 9.0) using parallel query configuration. Having an SSD enabled database a configuration is set to host several instances of the DBMS running concurrently on the same physical server, each hosting a partition of the database. The standard pgpool-II middleware is used[55], along with a specialized routine written in C++. The resulting multi-core concurrent disk access would present an ideal configuration to leverage the high performance random access feature of SSD technology, used to store meta-data.

	Single	Module 4	Module 8
Thread 1	7,445	4,185	0,722
Thread 2	—	4,266	1,045
Thread 3	—	5,141	1,152
Thread 4	—	5,295	2,969
Thread 5	—	—	3,072
Thread 6	—	—	3,181
Thread 7	—	—	3,392
Thread 8	—	—	3,766
Pg-pool	6,492	19,234	15,300
C++ module	7,445	5,348	3,889

Table A.5: The parallel query execution time reports measured times at each interface level: Thread N indicates time to get results from each database server instance while Pg-pool and C++ module refers to the front end (middle-ware) interface. Instances have been loaded with data according to the year module the server instance (source [69]).

From results reported in Table A.5, it is evident that single tasks (reported in the table as Thread N) take advantage of data partitioning and deliver good performance levels, the pgpool-II tool (version 3.0.1 was used) however, providing support for that configuration, appears to negate the concurrency benefits adding considerable overhead

in the results fusion phase. Conversely, a prototype function written in C++ language that puts unsorted results together seem to confirm the performance gain possibility. The standard solution however has the advantage to support a vast range of query types, while the test implementation supported only sorted or unsorted select queries. Nonetheless, the approach of using parallel queries is confirmed to be a good direction to follow for increased performances as it allows a second level partitioning of the data archive that in our case has been done over the time domain.

A.3 A note on mono-dimensional geographic addressing

A first version of the Tile archive meta-data was designed to take advantage of the Earth Fixed Grid and Tiling concepts to reduce the classical two dimensional geo-location problem (based on the geographic latitude and longitude coordinate system) to a mono-dimensional problem (based on linear Grid Tile zone identification).

Since AOI selections are mainly not linear on a 2D map, (in the sense that they do not consists of an horizontal line), domain subset caused query planner to issue many (one per crossed Tile line) filter conditions on index scans that provide sub-optimal queries and results in partitioning complexity. Only bands of latitude could be defined using two zone identifiers boundary values.

Another factor that reduces performance was inherent to the grid structure and addressing scheme, that is to define rectangular area wider that a quarter degree in latitude, an additional zone id range has to be defined and tested for each quarter of degree. That leads to complex calculations for bitmap OR and AND over indexes.

Listing A.1: Excerpt of query check condition with mono dimensional Tile addressing. One condition has to be check for each Tile line crossed along Latitude by the AOI

```
(
(zone_id_tiles BETWEEN 232530 AND 232675 ) OR
(zone_id_tiles BETWEEN 233970 AND 234115 ) OR
(zone_id_tiles BETWEEN 235410 AND 235555 ) OR
(zone_id_tiles BETWEEN 236850 AND 236995 ) OR
(zone_id_tiles BETWEEN 238290 AND 238435 ) OR
(zone_id_tiles BETWEEN 239730 AND 239875 ) OR
(zone_id_tiles BETWEEN 241170 AND 241315 ) OR
(zone_id_tiles BETWEEN 242610 AND 242755 ) OR
(zone_id_tiles BETWEEN 244050 AND 244195 ) OR
(zone_id_tiles BETWEEN 245490 AND 245635 ) OR
(zone_id_tiles BETWEEN 246930 AND 247075 ) OR
(zone_id_tiles BETWEEN 248370 AND 248515 ) OR
(zone_id_tiles BETWEEN 249810 AND 249955 )
)
```

Listing A.2: Excerpt of query check condition with two-dimensional Tile addressing. Only four conditions has to be check independent of AOI size

```
((zone_id_x BETWEEN 699 AND 753 ) and (zone_id_y BETWEEN 155 AND 190))
```

Testing the check condition has to be done for each record found, leading to noticeable overhead in case of hundreds of thousands checks. Moreover a two-dimensional addressing allows a more efficient partitioning of data as it allows partitions to be defined across both dimensions. Conversely, mono-dimensional addressing leads to "latitude bands" partitions (defined with a pair of tile_zone values).

A.4 The ext4 issue

During stress tests on the Tiles archive storage unit, the ext4 file system presented erratic behaviour that we were unable to resolve or trace to a possible cause, it is reported for reference in Listing A.3. The issue occurred during a Tile merge operation that intensively read, modifies, deletes and rewrites millions of 4 kB files. Note that only the last digit of seconds in the temporal log reference has been kept and the kernel log reformatted for clarity.

Listing A.3: The ext4 file system error observed during stress tests

```
[8.834417]
        mptscsih: ioc0: attempting task abort! (sc=ffff880213450200)
[8.834426]
        sd 3:0:0:0: [sdb] CDB: Synchronize
        Cache(10): 35 00 00 00 00 00 00 00 00 00
[9.328547]
        mptbase: ioc0: LogInfo(0x31140000):
        Originator={PL}, Code={IO Executed}, SubCode(0x0000)
[9.328624]
        mptscsih: ioc0: task abort: SUCCESS (sc=ffff880213450200)
[9.348775]
        end_request: I/O error, dev sdb, sector 5856032200
[9.349059]
        Aborting journal on device sdb1-8.
[9.361121]
        EXT4-fs (sdb1): delayed block allocation failed
        for inode 808452097 at logical offset 0 with
        max blocks 1 with error -30
[9.361669]

[9.361671]
        This should not happen!!  Data will be lost
[9.361914]
        EXT4-fs error (device sdb1) in
        ext4_da_writepages: Journal has aborted
[9.362921]
        EXT4-fs error (device sdb1):
        ext4_journal_start_sb: Detected aborted journal
[9.363391]
        EXT4-fs (sdb1): Remounting filesystem read-only
[9.363747]
        EXT4-fs (sdb1): ext4_da_writepages:
        jbd2_start: 347 pages, ino 808452097; err -30
```

The reported error drove the choice to stop the assessment of the two solutions in favour of the XFS file system that presented no errors under the same workload on the same device.

A.5 Data duplication assessment

Moving away from equator, the Earth-fixed grid chosen as the fixed spatial reference system introduces distortion on the tiles equivalent to that produced by the simple cylindrical projection. To assess the amount of data duplication introduced ad different latitudes an empirical approach is used. Selected classified granules are processed replacing the classification value with an unique value for each pixel, while keeping the geo-location information unaltered. Over the obtained synthetic dataset, selected Tiles belonging to 1x1 degree boxes are re-sampled into Tiles. In the resulting Tiles, approximately four pixel are expected to present the same value inside a tile (re-sampling is done at least at twice the original resolution): exceeding pixel are counted as duplicates; zero valued pixels are also counted. Results of an assessment over an early prototype of the re-sampling system are presented in Table A.6.

LAT	LON	Duplication %	Zeroes
74	-93	78,77	60
70	-90	73,76	83
65	-85	67,69	95
60	-81	61,92	81
55	-78	56,53	13
50	-75	51,31	32
45	-73	46,61	0
24	-80	31,52	0
20	-81	29,62	0
15	-81	27,77	0
9	-82	26,17	0
0	-75	25,15	0
-1	-75	25,19	0

Table A.6: Empirical measure of data duplication across Latitude: LAT and LON are upper left coordinates of the 1x1 degree box used to count synthetic values. Duplication is the amount of values above the expected count.

The presence of zeroes at high Latitude indicated a problem in the remapping software that has been corrected. As expected, the duplication increased toward the North pole. This test methodology proved useful in the assessment of software implementation correctness. To ensure correctness of the assessment software it was also run on the synthetic granule, leading to the expected zero duplication and a single zero

valued pixel (present by design at a corner of the granule, that is never used as test area).

The number of Tiles produced by processing a granule varies also in consequence of the distortion of the sampling grid: At the equator about 1200 are produced, raising to about 2214 at 40-58N and reaching its maximum around the South Pole (where the Land/Sea Mask discards fewer Tiles with respect to the northern polar region) of about 9000 Tiles (e.g. a granule from 2005-01-16 11:56:47 to 2005-01-16 12:02:59 delivers 8929 valid Tiles over the South Pole, from Lat -76 to -84).

Bibliography

[1] European Space Agency. Classification application-services and reference datasets, project page. Web site. available online at `http://earth.eo.esa.int/rtd/Projects/CARD/index.html`.

[2] European Space Agency. Support by pre-classification to specific applications, project page. Web site. available online at `http://earth.eo.esa.int/rtd/Projects/SPA/index.html`.

[3] Alan Maceachren and John H. Ganter. A pattern identification approach to cartographic visualization. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 27(2):64–81, Summer 1990.

[4] J.W. Tukey. *Exploratory data analysis*. Addison-Wesley series in behavioral science. Addison-Wesley Pub. Co., 1977.

[5] Alexander Szalay and Jim Gray. The World-Wide Telescope. *Science*, 293(5537):2037–2040, 2001.

[6] Mihai Datcu, Klaus Seidel, Oscar Guerra, and Manfred Schroeder. *KIM Knowledge Driven Information Mining in Remote Sensing Image Archives*. Project executive summary, ESA, Nov 2002. downloaded from `http://earth.esa.int/rtd/Documents/KIM_Executive_Summary.doc` on Dec-18-2006.

[7] Google™. Google™Earth engine web site. Website. available online at `http://earthengine.googlelabs.com/#intro`.

[8] J. R. Mahoney, G. Asrar, M.S. Keinen, J. Andrews, M. Glackin, C. Groat, W. Hohestein, and L. Lawson. The u.s. strategic plan for the climate change science program. a report by the climate change science program and the subcommittee on global change research. Technical report, Climate Change Science Program Office, 2003.

[9] D. Lu, P. Mausel, E. Brondízio, and E. Moran. Change detection techniques. *International Journal of Remote Sensing*, 25(12):2365–2407, 2004.

[10] C. Homer, C. Huang, L. Yang, B. Wylie, and M. Coan. Development of a 2001 national land-cover database for the United States. *Photogramm. Eng. Remote Sensing*, 70:829–840, 2004.

[11] R. Haines-Young and J.L. Weber. Land accounts for Europe 1990–2000. Technical report, European Environment Agency, Copenhagen, Denmark, 2006.

[12] Brian D. Wardlow and Stephen L. Egbert. Large-area crop mapping using time-series MODIS 250 m NDVI data: An assessment for the u.s. central great plains. *Remote Sensing of Environment*, 112(3):1096 – 1116, 2008.

[13] Sophie Bontemps, Pierre Defourny, and Eric Van Bogaert. GLOBCOVER 2009 product description and validation report. Technical report, Universitè Catolique de Louvain, 2010.

[14] S. S. Stevens. On the theory of scales of measurement. *Science*, 103(2684):677–680, 1946.

[15] Requirements & Standards Division. *European Cooperation for Space Standardization*. ECSS Secretariat, ESA-ESTEC, Noordwijk, The Netherlands, 2009.

[16] Services support environment. Web site. http://services.eoportal.org/.

[17] M. C. Terzi P. Nencioni, S. Gianfranceschi. *Earthnet Online XML Font End*. Intecs, Apr 2006. Issue 2.4, available online at `http://services.eoportal.org/portal/documents/eoli_24.pdf` as of February 2011.

[18] Oscar Gómez and Ferran Páramo. Methodological guidebook - data processing of land cover flows. Technical report, Universitat Antònoma de Barcelona, Barcelona, Spain, July 2005.

[19] K. Sahr, D. White, and A. J. Kimerling. Geodesic discrete global grid systems. *Cartography and Geographic Information Science*, 30(2):121–134, 2003.

[20] A. J. Kimerling, K. Sahr, and D. White. Global scale data model comparison. In *Proceedings of Auto Carto 13*, 1997.

[21] Declan Butler. Virtual globes: The web-wide world. *Nature*, pages 776–778, 2006.

[22] V. Pascucci and R.J. Frank. Global static indexing for real-time exploration of very large regular grids. In *Supercomputing, ACM/IEEE 2001 Conference*, pages 45.1–45.8, Nov. 2001.

[23] Jeffrey Scott Vitter. External memory algorithms and data structures: dealing with massive data. *ACM Comput. Surv.*, 33:209–271, June 2001.

[24] Jesse James Garrett. Ajax: A new approach to web applications. Web site, February 2005.

[25] Jakob Nielsen. *Usability Engineering*. Academic Press, Boston, 1993.

[26] Daniel P. Bovet and Marco Cesati. *Understanding the Linux Kernel*. O'Reilly Media, Oct. 2000.

[27] Antonio Plaza, David Valencia, Javier Plaza, and Pablo Martinez. Commodity cluster-based parallel processing of hyperspectral imagery. *Journal of parallel and distributed computing*, 66:345–358, 2006.

[28] S. Sommer, J. Hill, and J. Mégier. The potential of remote sensing for monitoring rural land use changes and their effects on soil conditions. *Agriculture, Ecosystems & Environment*, 67(2-3):197 – 209, 1998.

[29] Marco Folegani, Simone Mantovani, and Stefano Natali. SOIL MAPPER® system and products description. Web site, September 2010. downloaded from `http://www.meeo.it/index.php?action=download_resource&id=13&module=resourcesmodule&src=%40random4846d46ecae6e` on Oct-01-2010.

[30] A. Baraldi, V. Puzzolo, P. Blonda, L. Bruzzone, and C. Tarantino. Automatic spectral rule-based preliminary mapping of calibrated Landsat TM and ETM+ images. *IEEE Trans. Geosci. And Remote Sensing*, 44(9):2563–2586, Sep 2006.

[31] S. Natali and G. Piva. ESM-TN-MEEO-GS-0101 - internal report. Technical report, MEEO, 2009.

[32] Color shaded-relief image of the Earth. available online at `http://www.ngdc.noaa.gov/mgg/topo/img/glshade2.gif`.

[33] A. Beccati, M. Folegani, S. D'Elia, R. Fabrizi, S. Natali, and L. Vittuari. The multi-sensor land classification system (LCS): automatic multi-temporal land use classification system for multi-resolution data. In W. Wagner and B. Székely, editors, *IAPRS*, volume XXXVIII, Part 7B, pages 74–79, Vienna, Austria, July 2010. ISPRS.

[34] Alan Beccati, Roberto Fabrizi, and Alessandro Candini. *PreQu ASQuLD Interface Control Document*. MEEO, August 2010. PreQu project document number PRQ-ICD-MEEO-GS-0426.

[35] The ATSR project web site. Web site, February 2011.

[36] J. Murray, P. Bailey, A. Birks, and D. Smith. *ATSR-1/2 User Guide*, June 1999. A short guide to the ATSR-1 and -2 instruments and their data products. Edited by Chris Mutlow.

[37] Roberto Fabrizi and Mario Cavicchi. Radiometric calibration assessment tool and classification processing chain. Internal report PRQ-TN-MEEO-GS-0422, MEEO, December 2009.

[38] Roberto Fabrizi and Mario Cavicchi. Evaluation ATSR-2 data collection and cosmetic filling on ALCS. Internal report PRQ-TN-MEEO-GS-0410, MEEO, June 2009.

[39] Andrew R. Birks. Instrument pixel co-ordinates in AATSR products. Draft Technical note, May 2009.

[40] Roberto Fabrizi and Mario Cavicchi. (A)ATSR georeferencing performance evaluation and misplacement correction algorithms. Internal report PRQ-TN-MEEO-GS-0400, MEEO, June 2009.

[41] Mario Cavicchi, Alan Beccati, Sergio D'Elia, Andrea Della Vecchia, Andrea Baraldi, and Stefano Natali. A twelve years ATSR-2/AATSR preliminary classification maps database. In Huguette Lacoste-Francis, editor, *Proceedings of the 2nd MERIS/(A)ATSR User Workshop*, volume SP-666, page 229, Frascati, Rome, Italy, September 2008. ESA.

[42] Alan Beccati, Mario Cavicchi, Roberto Fabrizi, and Stefano Natali. ASQuLD: an advanced semantic query system for large satellite database. In Huguette Lacoste-Francis, editor, *Proceedings of the PV (Ensuring Long-Term Preservation and Adding Value to Scientific and Technical Data) Conference*, volume poster, page n/a, Fillafranca del Castillo, Madrid, Spain, December 2009. ESA.

[43] SSE Team. *Service Support Environment Interface Control Document*. SpaceBel, February 2008. Issue 1.6, available online at `http://services.eoportal.org/massRef/documentation/icd.pdf`, as of February 2011.

[44] Web site. List of projection codes available on line at `http://spatialreference.org/`.

[45] U.S.G.S. global 1-Km land-sea mask product, June 2008. available on line at `http://edc2.usgs.gov/1KM/land_sea_mask.php`.

[46] Roberto Fabrizi, Mario Cavicchi, Luca Vittuari, and Andrea Lugli. Remapping system performance evaluation and validation. Internal report PRQ-TN-MEEO-GS-0434, MEEO, December 2009. SPA Project Technical Note.

[47] Giacomo Piva. Customised catalogue and search system for large satellite imagery database. Laurea specialistica in informatica, Università degli Studi di Ferrara - Facoltà di Scienze Matematiche Fisiche e Naturali, 2008.

[48] Silicon Graphics (SGI). Open source xfs™for linux®. Web site, 2006. Datasheet available on-line at `http://oss.sgi.com/projects/xfs/datasheet.pdf`.

[49] Dave Chinner and Jeremy Higdon. Exploring high bandwidth filesystems on large systems. In *Proceedings of the Ottawa Linux Symposium*, Ottawa, Canada, July 2006.

[50] Damiano Barboni. Customized graphic user interface for large database data mining purpose. Laurea specialistica in ingegneria informatica e dell'automazione, Università degli Studi di Ferrara - Facoltà di ingegneria, 2008.

[51] A. Beccati, M. Folegani, S. D'Elia, D. Barboni, and S. Selmi. Multi sensor evolution analysis (MEA): Land use and land cover analysis applied to (A)ATSR time series. In Huguette Lacoste-Francis, editor, *Proceedings of ESA Living Planet Symposium*, volume SP-686, pages 076–D4, 2010.

[52] Alan Beccati, Damiano Barboni, and Stefano Selmi. *Software User Manual and ICD for MEA*. MEEO, February 2011. SPA project document number SPA-ICD-MEEO-GS-0143, Issue 1.6.

[53] Jens Axboe. ionice manpage. Linux distribution, August 2005. Also available on-line at `http://linux.die.net/man/1/ionice`.

[54] Mike Holton and Raj Das. XFS: A next generation journalled 64-bit filesystem with guaranteed rate I/O. Web site. available on-line at `http://www.uoks.uj.edu.pl/resources/flugor/IRIX/xfs-whitepaper.html`, accessed on January 2011.

[55] Pgpool-II project page. Web site. `http://pgpool.projects.postgresql.org/`.

[56] Official ente nazionale risi web site. Web site. `http://www.enterisi.it/ser_statistiche.jsp`, accessed on December 2010.

[57] Alessandro Lolli. Multitemporal analysis of satellite imagery: Implementation of a land evolution model in ALCS (AATSR land classification system). Laurea specialistica in ingegneria per l'ambiente e il territorio, Alma Mater Studiorum - Università Di Bologna - Facoltà Di Ingegneria, 2009.

[58] Jeff McKenna. Mapserver web site - wms time. Web site. `http://mapserver.org/ogc/wms_time.html`.

[59] Geoserver welcome page. Web site. `http://geoserver.org/display/GEOS/Welcome`.

[60] Open Geospatial Consortium. Opengis web map service (WMS) implementation specification. Web site. `http://www.opengeospatial.org/standards/wms`.

[61] Open Geospatial Consortium. Web coverage service (wcs) implementation standard. Website. `http://www.opengeospatial.org/standards/wcs`.

[62] Open Geospatial Consortium. Web processing service. Web site. `http://www.opengeospatial.org/standards/wps`.

[63] Manfred Ehlers, Sascha Klonus, Pär Johan Åstrand, and Pablo Rosso. Multi-sensor image fusion for pansharpening in remote sensing. *International Journal of Image and Data Fusion*, 1(1):25–45, 2010.

[64] ESA member states approve full and open Sentinel data policy principles. Web site, November 2009.

[65] George Büttner, Jan Feranec, Gabriel Jaffrain, László Mari, Gergely Maucha, and Tomas Soukup. The CORINE land cover 2000 project. In *EARSeL eProceedings*, volume 3, pages 331–346. European Association of Remote Sensing Laboratories, 2004.

[66] Jixian Zhang. Multi-source remote sensing data fusion: status and trends. *International Journal of Image and Data Fusion*, 1(1):5–24, 2010.

[67] J. Axboe. FIO - flexible I/O tester. `http://freshmeat.net/projects/fio/`.

[68] Intel. *Intel® X25-E SATA Solid State Drive – Product Manual.* Downloaded on 25/10/2010 from `ftp://download.intel.com/design/flash/NAND/extreme/extreme-sata-ssd-datasheet.pdf`.

[69] Alan Beccati, Damiano Barboni, Mario Cavicchi, Alessandro Candini, and Loreto Pellegrini. MEA Tiles database structure - system analysis. Internal report SPA-TN-MEEO-GS-154, MEEO, February 2011. SPA Project Technical Note, Issue 1.2.

Acronyms

(A)ATSR AATSR and ATSR-2

AaaS Application as a Service

AATSR Advanced Along Track Scanning Radiometer

ACL Access Control List

AJAX Asynchronous JavaScript and XML

AOI Area Of Interest

API Application Programming Interface

ASQuLD Advanced Semantic Query system for Large satellite Database

ATSR Along Track Scanning Radiometer

ATSR-2 Along Track Scanning Radiometer 2

AVNIR-2 Advanced Visible and Near-Infrared Radiometer 2

CARD Classification Application-services and Reference Datasets

CFQ Completely Fair Queuing

CMP Classified Maps Provider

CT Class Tolerance

DAS Direct-Attached Storage

DB Data Base

DBMS Data Base Management System

DGG Discrete Global Grid

DGGS Discrete Global Grid System

EDA Exploratory Data Analysis

EFG Earth Fixed Grid

EM Evolution Model

EO Earth Observation

EOLI Earthnet OnLine Interactive

ESRIN European Space Research INstitute

ES Earth Science

ESA European Space Agency

ESM Enhanced SOIL MAPPER

EVAT Expert user Visual Analysis Tool

FTP File Transfer Protocol

GDGG Geodesic Discrete Global Grid

GIS Geographic Information System

GRIO Guaranteed rate I/O system

GSD Ground Sampling Distance

GUI Graphical User Interface

HTML HyperText Markup Language

HTTP HyperText Transfer Protocol

KIM Knowledge-based Information Mining

LEAC Land and Ecosystem Accounts

LSM Land/Sea Mask

LULCC Land Use and Land Cover Change

ME Model Element

MEA Multi-sensor Evolution Analysis

MEEO Meteorological and Environmental Earth Observation

MODIS Moderate Resolution Imaging Spectroradiometer

NDVI Normalized Difference Vegetation Index

NOAA National Oceanic and Atmospheric Administration

OGC Open Geospatial Consortium

PM Particulate Matter

PXH Pixel History graph

RAL Rutherford Appleton Laboratory

RAM Random Access Memory

RS Remote Sensing

RSS Research and Services Support

RTD Research and Technology Development

SM SOIL MAPPER®

SPA Support by Pre-classification to specific Applications

SSD Solid State Drive

SSE Service Support Environment

TAR Tiles Archive

TOA Top Of Atmosphere

TSP Time Since Previous

TT Time Tolerance

TTS Tiles Time Series

VO Virtual Observatory

WCS Web Coverage Service

WFS Web Feature Service

WMS Web Map Service

WPS Web Processing Service

XML EXtensible Markup Language

Acknowledgements

I wish to thank all the people who made the realization of this thesis possible, starting with my advisor Eleonora Luppi and my thesis reviewers Sergio D'Elia and Lucio Colaiacomo, for their precious advice.

I would also like to acknowledge the support to my research activities provided by MEEO and its staff. In particular I would like to thank: Stefano Natali, Simone Mantovani and Marco Folegani, the company owners, for letting me conduct research and development activities within the framework of company projects, which led to the fully fledged instance of the presented system, now running at ESA premises; Mario Cavicchi (who has also proven to be a valuable brainstorming companion on many of the subjects herein presented), Damiano Barboni, Marco Bascetta, Moris Pozzati, Alessandro Candini and Stefano Selmi.

I would also like to acknowledge the cooperation of PhD Fellow Piero Campalani who contributed to the integration of the OpenLayers software and is currently working on the extension of the presented system to the analysis of Particulate Matter Maps for air quality monitoring; Maria Grazia Veratelli, Luca Vittuari, Matteo Mattiuzzi and Teresa Bras for their support in validation activities.

Last but not least, a thank goes to Andrea Della Vecchia, Michele Iapaolo, Carlos Ferrao, Giancarlo Rivolta and the other people at ESA that I have been working with throughout the last three years.