Astronomical Data Analysis Software and Systems XVI
ASP Conference Series, Vol. 376, 2007
R. A. Shaw, F. Hill and D. J. Bell, eds.

Astro-WISE: Chaining to the Universe

Edwin A. Valentijn, John P. McFarland, Jan Snigula, Kor G. Begeman, Danny R. Boxhoorn, Roeland Rengelink, Ewout Helmich, Philippe Heraudeau, Gijs Verdoes Kleijn, Ronald Vermeij, Willem-Jan Vriend, Michiel J.Tempelaar

OmegaCEN, Kapteyn Astronomical Institute, Groningen University, Groningen, The Netherlands

Erik Deul, Konrad Kuijken

Sterrewacht Leiden, University Leiden, Leiden, The Netherlands

Massimo Capaccioli, Roberto Silvotti

INAF, Osservatorio Astronomico di Capodimonte, Napoli, Italy

Ralf Bender, Mark Neeser, Roberto Saglia

Universitäts-Sternwarte München, Germany

Emmanuel Bertin, Yannick Mellier

Terapix, Institut d'Astrophysique de Paris, Paris, France

Abstract. The recent explosion of recorded digital data and its processed derivatives threatens to overwhelm researchers when analysing their experimental data or looking up data items in archives and file systems. While current hardware developments allow the acquisition, processing and storage of hundreds of terabytes of data at the cost of a modern sports car, the software systems to handle these data are lagging behind. This problem is very general and is well recognized by various scientific communities; several large projects have been initiated, e.g., DATAGRID/EGEE[1] federates compute and storage power over the high-energy physical community, while the international astronomical community is building an Internet geared Virtual Observatory[2] (Padovani 2006) connecting archival data. These large projects either focus on a specific distribution aspect or aim to connect many sub-communities and have a relatively long trajectory for setting standards and a common layer. Here, we report first light of a very different solution (Valentijn & Kuijken 2004) to the problem initiated by a smaller astronomical IT community. It provides an abstract scientific information layer which integrates distributed scientific analysis with distributed processing and federated archiving and publishing. By designing new abstractions and mixing in old ones, a Science Information System with fully scalable cornerstones has been achieved, transforming data systems into knowledge systems. This break-through is facilitated by the full end-to-end linking of all dependent

[1]http://www.eu-egee.org/

[2]http://www.euro-vo.org/pub/

data items, which allows full backward chaining from the observer/researcher to the experiment. Key is the notion that information is intrinsic in nature and thus is the data acquired by a scientific experiment. The new abstraction is that software systems guide the user to that intrinsic information by forcing full backward and forward chaining in the data modelling.

1. Introduction

The classical paradigm to handle data streams of large physical experiments, such as CERN-HEP, astronomical telescopes, or scientific satellites, is characterized by different layers in the project that apply certain algorithms to the stream of data and subsequently deliver the results to the next layer, following a so-called tier architecture. This architecture can be characterized as data-driven and feed-forward. The construction of the different layers is often grown historically and relates to implicit or explicit project management decisions facing sociology, geography and interfacing of working groups with different expertise. Projects operated under this paradigm work with releases of datasets, which are obtained with a certain version of the code meeting certain quality control standards. Operators have a task to push the input data through the stream, often by means of a pipeline irrespective of whether the derived data items are actually used by the end users. Examples of successful forward workflow projects are numerous (e.g., Valentijn & Kuijken 2004; da Costa Rite, & Slijkhuis 2003; Hiriart et al. 2003; Jung et al. 2003; Mehringer & Plante 2003; Pierfederici et al. 2003). When funding for the operation centers dries up, a final release of processed data is made, and the project comes to an end.

2. The Problem

This classical paradigm has the advantage of publishing complete sets of data with well-described methods, qualifications, and calibrations. However, it has bad scalability when exposed to modern huge data streams: reprocessing all the raw data and storing the results for new releases when new computational methods, calibration strategies, insights or improved code becomes available is very difficult to impossible. And, moreover, why should operators re-process all data as long as it is even not known whether there are customers for the individual products? Though the classical approach has the obvious advantage of facilitating established releases, it cannot facilitate the demands of the end-user/researcher, who actually might want to evaluate specific questions and specific results obtained with specific methods from a subset of the flood of physical observational data perhaps including even holiday digital pictures or scanned documents of national libraries. Key to addressing the merits of a system for a researcher is to evaluate the question of what analysis on uncertainties the researcher facilitates in that system before (s)he submits his/her 4 sigma result to a scientific journal.

Such demands combined with the increased data rates from instruments and world-wide communities ask for a different approach. A first interesting step is set by the Space Telescope Science Institute, operating the Hubble Space

Telescope, where users can ask for a certain data product which will then run the Calibration pipeline on-demand using the best available calibrations (Swade et al. 2001), a case of forward chaining. But what to do when a user discovers a very faint object and wants to inspect the robustness of the result by checking the dependencies on uncertainties of calibration parameters and applied methods? The user wants to interact with the data, derive his own result following his insights and methods, and preferably this knowledge is fed back into the system at the disposal of other researchers who optionally want to take advantage of the progress of insight. In the 1980s, solutions were explored in expert systems. Nowadays, with distributed communities working on enormous data floods, new artificial intelligence-like, distributed information systems are required to facilitate the scientific endeavour.

3. The Solution

A new generation of wide-field astronomical imaging cameras includes the MEGACAM at the CFHT and OmegaCAM (Kuijken et al. 2004) at the VLT Survey Telescope (VST). OmegaCAM, with a 1 $\deg^2$ field of view and a pixel size of $0\farcs2$, will deliver 256 mega-pixel images of the sky every few minutes, for 300 nights per year for many years to come. These experiments will produce hundreds of terabytes per year of raw and processed data, e.g., to trace the effects of dark matter and dark energy. These requirements inspired an international consortium, Astro-WISE, to design and implement a completely different approach to connect the end-user to the experiment. Facing the problems sketched above we build an information system in which the role of the user (observer) is central. The system is built to handle queries by the user for his/her desired result, which we call a target. A target could be an astrometrically and photometrically calibrated image, or the results for a set of calibration parameters, or a list of parameter values describing an astronomical object, or whatever target that is facilitated by the system. In the abstractions used for the software the target processing resembles that of the UNIX make metaphor, but in addition the dynamic aspects resembles the goal in artificial intelligence systems.

4. How it Works

Following a query by the user, the system checks whether the target has been processed before. If not, it will be processed on-the-fly, recursively following all its dependencies in an object model, similar to the Unix make metaphor. See the target diagram in Figure 1. If the target already exists, the system will check all its dependencies up to the raw data taken at the telescope (Figure 1). If all its dependencies are up-to-date the target object is returned. If all or some dependencies are not up-to-date they will be recomputed on-the-fly, again according to the make metaphor, but with optional, tunable depth.

All data production in the system follows this schema, whether it concerns an individual user asking for a single special result (target), a calibration scientist determining the instrument behaviour over long periods or a production scientist deriving results for whole nights of observing. They all add knowledge to the

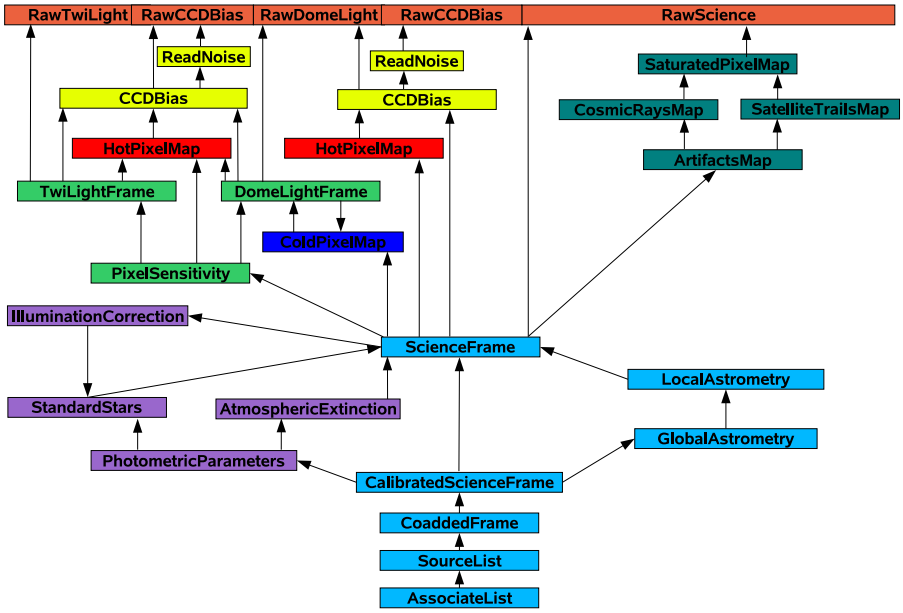

Figure 1. A *target diagram*: slightly simplified view of the dependencies of targets to the ocean of raw observational data of astronomical wide field imaging. Arrows indicate the backward chaining to the raw data.

system. Note that, in the end, it does not matter very much whether or not a target requested by the user has been processed before, it only affects the performance of the system. This way, the work flow in the ocean of data space is fully driven by the users request (see `http://process.astro-wise.org/`).

This behavior is achieved by linking (referencing) all dependent data items in the information system. This is achieved by carefully specifying the observation procedure both for science and calibration observations and subsequently designing a data model, chaining the processing targets to the raw data. Next, the data model is converted into an object model, which in turn is ported in a commercial database (using object oriented user-defined types in a relational database). The database thus has full awareness of all dependencies. When compute scripts are run at a computer, all Class instantiations are automatically made persistent in the database forming a dynamic archive of all targets and added knowledge, while new and nearly sacred raw data is ingested as long as the acquisition continues. Table 1 summarizes some key differences between the Astro-WISE system and the classical approach.

Obviously, this can only be achieved by integrating all computing and storage hardware into a single information system. Figure 2 shows the peer-to-peer network employed for Astro-WISE. The network and its hardware can be viewed as an extension of the telescope and measurement apparatus hardware, together with the abstract software, forming a unity. The user, by requesting a target, triggers the whole chain from target to raw observational data of the measurement apparatus, a real virtual observatory.

Table 1. Characteristics of the Astro-WISE Information System

Classical Paradigm	Astro-WISE Paradigm
forward chaining	backward chaining
tier architecture	target architecture
driven by raw data	driven by user query
process in pipeline	process on-the-fly
operators push data	users pull data
results in releases	information system
static archive	dynamic archive
raw data is obsolete	raw data is sacred

All top level software in the system is written in the Python scripting language (see `http://www.python.org/`) which in turn calls C libraries. Following the LISP language, Python achieves a very high level of integration. It facilitates another crucial unification in the system: apart from the GUIs, all usage of the system is done from a Python prompt via a Python binding to the database. Novice users, advanced users, and scientific programmers/developers alike use the same environment, allowing a continuous transition between all levels. All Python open source facilities, e.g., visualization libraries (MatPlotLib), numerical libraries (NumPy), etc., are at direct disposal of the user and are connected to the output of user queries to the information system. Users apply scripts provided by a federated CVS distribution of the standard Astro-WISE code base, but can also modify scripts, fine-tuning them or adding methods as long as they do not violate the object model. This way, both a simple standard version and user tuned versions of processing targets can be obtained simultaneously (different versions of the code are tagged and can be traced).

The database is partitioned in instruments and projects, facilitating individual read and write privileges for persons, groups and projects. Users set a context relating to their project and thus see only their partition in the ocean of data. The smallest project, the individual or anonymous Web based user doing his/her afternoon experiment is facilitated in a MyDB context, which can be upgraded to another, appropriate context/project at the end of the session. Project leaders have the responsibility for the project quality control. The code base supports this by built-in verify, compare and inspect methods for each Class, where verify involves an internal code check, compare involves a database query to different instantiations of the same Class and inspect is a visual inspection by the user. Eventually, all quality control is maintained by a time-stamping mechanism on each Class instantiation, and a GUI is build for calibration scientists to supervise and alter the timestamps, the ultimate point where human insights add knowledge to the system.

In fact, an important design criterion was to allow a complete federation of the system, facilitating different research groups spread over Europe to share scientific projects. Currently, the system connects National data centers in the Netherlands (Groningen), Italy (Naples), France (Paris), and Germany (Mu-

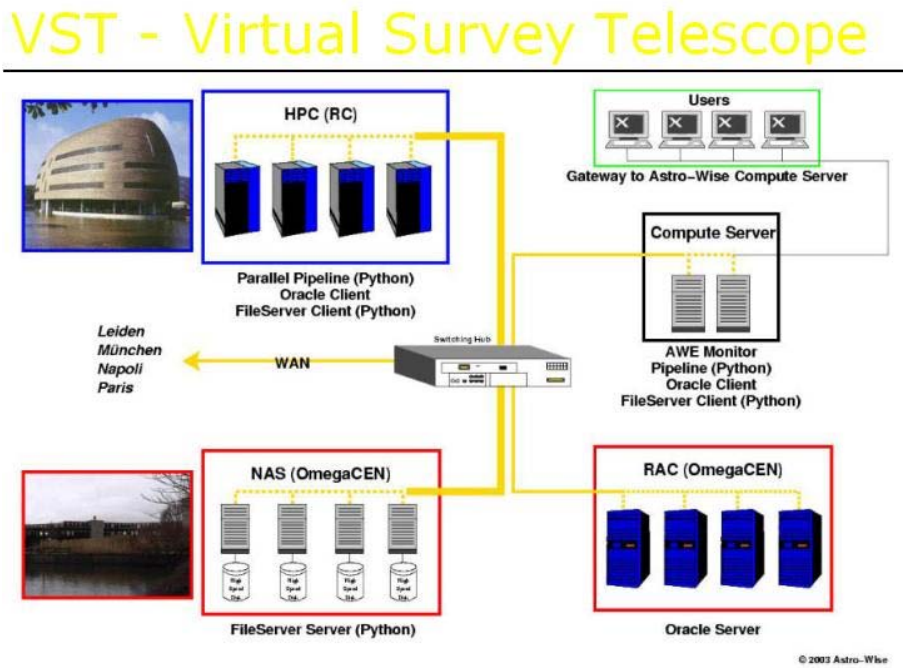

Figure 2. From virtual computer observing to real computer observing. The figure sketches the four fully scalable cornerstones for computer observing in a peer to peer network (with international federations).

nich), which in turn re-distribute the system to satellites (e.g., Bonn and Heidelberg). Similar hardware is installed at the National centers: the code-base is federated, the database is replicated, and the file-servers prevent duplication of massive storage of image data. Network traffic is minimised by allowing for terabyte-sized local cache for frequently used objects. All GUIs are web based. These include the database viewer (`http://dbview.astro-wise.org/`) and the GUI to evaluate and operate target processing at any selected Linux farm of the federation. User (client) altered Python scripts are sent to the processors in the federation as so-called Python pickles (persistent Python objects in a string representation).

The system has been fully implemented and currently processes the raw data of various astronomical wide field imaging cameras at ESO's 2.2-m telescope in Chile (300.000 CCD read-outs), the ING telescope(Canary Islands), the Subaru telescope (Japan) and OmegaCAM for ESO's VLT survey telescope (currently test data only). During the implementation, while taking care of object model purity, we experienced an avalanche of benefits for which relatively little extra effort had to be made:

- The full linking of object dependencies allows full backwards chaining as employed by artificial intelligence (Shachter & Heckerman 1987; Thompson & Thompson 1985) and history tracking. Literally every bit of information that went into the target can be retrieved: the system provides a tell me

everything tool. In fact, this is an *a priori* ontological implementation, in contrast to the fashionable *a posteriori* semantic web search engines.
- Publishing on Internet/EURO-VO (the European Virtual Observatory) is done by raising a flag. It is up to project managers to do this for classical paradigm static results or for Astro-WISE paradigm dynamic results.
- The control by the database over the parallel processing (e.g., SETI@home) on any number of nodes. The enabling of a international compute GRID.
- Enabling global astrometric and photometric solutions with increased accuracy, redirecting global database knowledge back into the system.
- The built-in workflow directly guides the user and no workflow systems are required.

Next to facilitating statistical studies, the system is optimized for needle-in-the-haystack kind of searches, finding extremely rare astronomical objects, such as moving, solar system objects, or variable objects like ultra-compact binary systems of white dwarf stars or distant supernovae. Next to all image metadata, the astronomical source parameters are stored in an Oracle database and Oracle partitioning is used to quickly address up to 100-TB data volumes. Fast astronomical object associations (*a posteriori* associations) are made in many-to-many mode using both native database indexing and positional HTM[3] indexing. The linking (joins and references) are maintained at an extreme level. For example, in the KIDs 1500-5000 degree survey nearly 1 TB of linked data items are anticipated.

The avalanche of side products and its rapid implementation are thanks to several unifications which are achieved by merging various pointer mechanisms, such as those provided by object oriented programming (inheritance), class/attribute persistency, database internal links (joins, references), and namespace handling by the Python scripting language. In the context of physical information theory, all these links can be viewed as kinds of memory addresses and facilitate forms of information transfer, gearing the user to intrinsic information, and facilitating the handling of the dispersion caused by the measurement apparatus, camera, and the off-line computing hardware. In particular, the option to inspect partial derivatives of dependent parameters by re-deriving results allows the user to inspect the information content in the data (Frieden 1998).

5. Conclusion

Our key concept, involving novel approaches to maintain data associations in federations of integrated pipelines and archives, can be applied to arbitrary forms of digitized observational data, ranging from DNA sequences to numerical simulations and national libraries (e.g., centuries of Dutch governmental handwritten records being scanned and entered into the system will be processed with pattern recognition techniques). After all, this is all part of our Universe that we observe, and with the coming decade, will be copied into our hardware at multi-petabyte scales to be interpreted by a next generation of scientists.

[3]http://www.sdss.jhu.edu/htm/

Acknowledgments. The project received funding from the European Union through the EC Action *Enhancing Access to Research Infrastructures*, a FP5 RTD programme, the Netherlands Research School for Astronomy NOVA and NWO. Correspondence and requests for materials should be addressed to E.A. Valentijn, Coordinator, Astro-WISE `valentyn@astro.rug.nl`. Supplementary Information can be found on `http://www.astro-wise.org`.

References

da Costa, L., Rite, C. & Slijkhuis, R. G. 2003, in ASP Conf. Ser. 314, ADASS XIII, ed. F. Ochsenbein, M. Allen, & D. Egret (San Francisco: ASP), 3

Frieden, B. R. 1998, *Physics from Fisher Information*, (Cambridge: Cambridge University Press)

Hiriart, R., Valdes, F., Pierfederici, F., Smith, C. & Miller, M. 2003, in ASP Conf. Ser. 314, ADASS XIII, ed. F. Ochsenbein, M. Allen, & D. Egret (San Francisco: ASP), 74

Jung, Y., et al. 2003, in ASP Conf. Ser. 314, ADASS XIII, ed. F. Ochsenbein, M. Allen, & D. Egret (San Francisco: ASP), 764

Kuijken, K., et al. 2004, SPIE, 5492, 484

Mehringer, D. M., & Plante, R. L. 2003, in ASP Conf. Ser. 314, ADASS XIII, ed. F. Ochsenbein, M. Allen, & D. Egret (San Francisco: ASP), 42

Padovani, P. 2006, in ASP Conf. Ser. 351, ADASS XV, ed. C. Gabriel, C. Arviset, D. Ponz, & E. Solano (San Francisco: ASP), 771

Pierfederici, F., Valdes, F., Smith, C., Hiriart, R., & Miller, M. 2003, in ASP Conf. Ser. 314, ADASS XIII, ed. F. Ochsenbein, M. Allen, & D. Egret (San Francisco: ASP), 476

Shachter, R. D. & Heckerman 1987, *Thinking Backward for Knowledge Acquisition*, AI Magazine, 8:3, 55

Swade, D. A., Hopkins, E., & Swam, M. S. 2001, in ASP Conf. Ser. 238, ADASS X, ed. F. R. Harnden, Jr., F. A. Primini, & H. E. Payne (San Francisco: ASP), 295

Thompson, B. A., & Thompson, W. A. 1985, *Inside an Expert System*, Byte, 10, 315

Valentijn, E. A., & Kuijken, K. 2004, Toward an International Virtual Observatory, Proceedings, ed. P. J. Quinn & K. M. Gorski (Berlin: Springer) 19