

Queueing Aspects of Integrated Information and Computing Systems in Geosciences and Natural Sciences

Claus-Peter Rückemann
*Westfälische Wilhelms-Universität Münster (WWU),
Leibniz Universität Hannover,
North-German Supercomputing Alliance (HLRN)
Germany*

1. Introduction

Modern geosciences widely rely on information science technologies. In most cases of scientific research applications, tools, and state of the art hardware architectures are infamously neglected and therefore their development is not pressed ahead and not documented, a fault for continuous and future developments. Information Systems and Computing Systems up to now live an isolated life, rarely integrated and mostly lacking essential features for future application. Although in general technology advances and new tools arise, there is a number of aspects that prevent interest groups from building complex integrated systems and components on a long term base. These issues, from hardware and system architecture aspects to software, legal, and collaborational aspects, are top in the queue for realisation show-stoppers. This chapter presents the current status of integrated information and computing systems. It discusses the most prominent technical and legal aspects for applications in geosciences and natural sciences. Today's state of the art information systems provide a plethora of features for nearly any field of application. Present computing systems can provide various distributed and high end compute power. Compute resources in most cases have to be supported by highly performing storage resources. The most prominent disciplines on up to date resources are natural sciences like geosciences, geophysics, physics, and many other fields with theoretical and applied usage scenarios. For geosciences both information systems as well as computing resources are essential means of day to day work. The most immanent limitation is that there are only a very few facilities with these systems combining the information systems features with powerful compute resources. The goal we have to work on for the next years is to facilitate this integration of information and computing systems. Modern information systems can provide various information and visualise context for different purposes beyond standard Geoscientific Information Systems (GIS). Fields demanding for handling, processing, and analysing geoscientific data are manifold. Geophysics as well as applied sciences provide various methods as to name magnetic methods, gravity methods, seismic methods, tomography, electromagnetic methods, resistivity methods, induced polarisation, radioactivity methods, well logging and various

assisting and integrated methods and techniques. Integrating these methods with information systems and the support of remote sensing, cartography, depth imaging, and infrastructural and social sources a more and more holistic view on the earth system will be possible. This will help to gain insight in the fields of seismology, meteorology, climatology, prospection and exploration, medical geology, epidemiology, environmental planning and many more disciplines. The resulting information systems and applications are not only used for scientific research but for public information, education, disaster management, expert systems and many more. In various application areas the surplus value arises with intelligent combination of information. A cartographic system only displaying spatial data is of less significance for a seismological disaster management application if there are no additional information and features provided. Provisioning these information will in many case result in interactive computation. For some use cases requests for points of interest, dynamical cartography, event programming, flexible event triggering, long-term monitoring like in seismology, catastrophe management and meteorology are necessary, for others simulation or modeling of scenarios are essential. All these fields of application contain tasks that cannot be handled in extend for large and complex systems on one local compute and storage resource only. Processes like processing jobs, visualisation, traveling salesman problems, and multimedia production have to be transferred to systems with the capacity necessary for multiple requests at the same time. We should not isolate scholarly research from long term information science concepts and architectures used in geosciences disciplines. Therefore the overall goal is to integrate systems, concepts, software, hardware and other components on a higher level of collaboration and strategical decision. As many application scenarios arise from geosciences and natural sciences, a number of examples are given based on implementations from these disciplines and case studies done over the last decade. Present activities and future work to be done on development and strategies level are presented to help overcome the stagnancy in the evolution of integrated systems. This chapter will show the components that in most other cases are discussed independently and presents a basic concept for integrating systems as successfully used with geosciences and natural sciences case studies.

2. Capability versus capacity

The high end computing world vastly used by geosciences researchers can be described with capability computing on the one hand and capacity computing on the other hand which provide complex tools to expand the means and methods of research by continuously expanding the limits of feasibility.

2.1 Capability computing

Capability computing means to target the grand challenge problems in certain fields. This will for example be the case with earthquake simulation, multi-dimensional modeling of the earth's underground, tornado simulation, atmosphere simulation, galaxy cluster simulation, non-linear computation of complex structures, life-sciences and epidemiological simulation, archaeological and architectural simulation. For the foreseeable future complex information and computing systems will be a topic on this list as soon as resources evolve. Systems for capability computing have to provide capacity for a few large jobs. Job processing for single-job scenarios can handle larger problems or faster solution. For these single jobs have

main potential for insight on their own, merely not from context. Capability computing will usually not be the first association with information systems.

2.2 Capacity computing

The more interesting along with the development of the next generation of system architectures is the opening of capacity computing for complex information systems. These systems provide capacity for many small or medium sized jobs. Job processing of several parallel jobs is suitable for these resources. The single job itself may be used for parameter studies, design alternatives, exploring pre-development stages, and in general these jobs on their own have less potential for insight. Making use of capacity computing resources many instances of compute jobs will run on a resource. This is what we need to enliven conventional information system implementations with new features, leading to new insights.

3. Computing resources

3.1 Paradigms

There is a number of different paradigms and resources that can be considered for this purpose and used from various geosciences disciplines. The topmost category for High End Computing (HEC) are High Performance Computing (HPC) and Supercomputing resources. These will in nearly all use cases be used in a non distributed manner. The lower end, Distributed Computing (DC) and services computing resources, are for example built on base of paradigms like Sky Computing, Cloud Computing, Grid Computing, Cluster Computing, Mobile Computing.

3.2 Performance pyramid

What do we have to expect to be the machines behind these paradigms? The answer is a performance pyramid. The performance pyramid for computing resources shows the following structure.

Top sector:

- International supercomputers,
- National supercomputers,
- Regional supercomputers.

Medium and bottom sector:

- Local and dedicated compute servers and clusters,
- Workstations,
- Mobile devices.

4. Computing obstacles

The obstacles for operating complex resources with interactive systems with efficient and effective operation can be overcome while reacting on several levels, architectures, hardware, frameworks and middleware, applications, energy consumption, competence resources, consultancy, and support.

As far as High End Computing (HEC) being a genus for High Performance Computing and various other ambitious computing paradigms is an issue of national interest for most countries, reliability and security are the most important factors for operating these services. Science and Research is depending on the results of their computations. Just with this, everyone is depending on systems and operating systems used. So problems most imminent arise especially with the

- Large number of cores,
- Large number of nodes,
- Distributed memory usage,
- Large number of large hard disks,
- Read and write speed of storage.

With the increasing number of requests and interactivity the communication size, size of data, transfer band width, scalability, and mean times for failure get more important. An intelligent arrangement and configuration of system components and an overall management of system components gets into the focus.

The most prominent problem with the next generation of resources is quantity of components. The handling of quantity leads –besides many other challenges– to increased demands for encryption, IO, PCI, on-chip features, error correction (ECC), research and development, scientific and academic staff and supporting maintenance, operative and administrative staff, as well as for secondary dependencies like energy resources and unbreakable power supplies.

4.1 Consumption

The most prominent problem with quantity, besides the computing obstacles, is consumption. State of the art power and energy measures are for example Low Voltage memory (LV DIMM), Light Load Efficiency Mode (LLEM), multiple Power Supplies, watercooler chassis & air conditioning, higher temperature cooling, hot water cooling, hybrid cooling systems, Energy and Power Manager (Active Energy Manager, AEM and others), application/energy frequency optimisation, energy reduced low frequency Processors, Power Management, and Energy Management.

4.2 Shortcomings

Besides that modular, dynamical applications are rare, even in geosciences, shortcomings regarding application context and how to handle these aspects are obvious:

- Architectures (CPU, GPU, GPGPU, FPGA, ...),
- Languages (high level languages, CUDA, ...),
- Memory,
- Fast and broad band Networks,
- Efficiency,
- Manageability, ...

5. Hardware resources

Most scientific projects consider software and hardware issues to be treated separately. This would most likely be a problem for developing integrated systems on a solid holistic base. As for overall costs, for example with power consumption and staff, only very few institutions will be able to operate and develop those systems. The more complex these systems get, the less can the distinction between infrastructure resources and systems resources be recognised. As for understanding the complexity of these issues to be inseparable in the dimension of future integrated systems, the following paragraphs will illustrate some most important hardware components.

5.1 Resources infrastructure

An unadmissible premise for safe and reliable operation complex and large systems are concepts and implementation of power resources, unbreakable power supplies, air conditioning, electronics, physical security and many more infrastructure components. Figure 1 shows infrastructure components necessary to operate a larger computer installation for the purpose of scientific computing: generator, air conditioning, power supplies, and electrical and physical security measures.



Fig. 1. Infrastructure components necessary to operate larger computer installations.

5.2 Resources cooling

Besides the infrastructure, various measures associated directly with the computer systems are necessary and this will depend on the type of installation. Figure 2 shows one type of rack water-cooling (SGI, 2011).

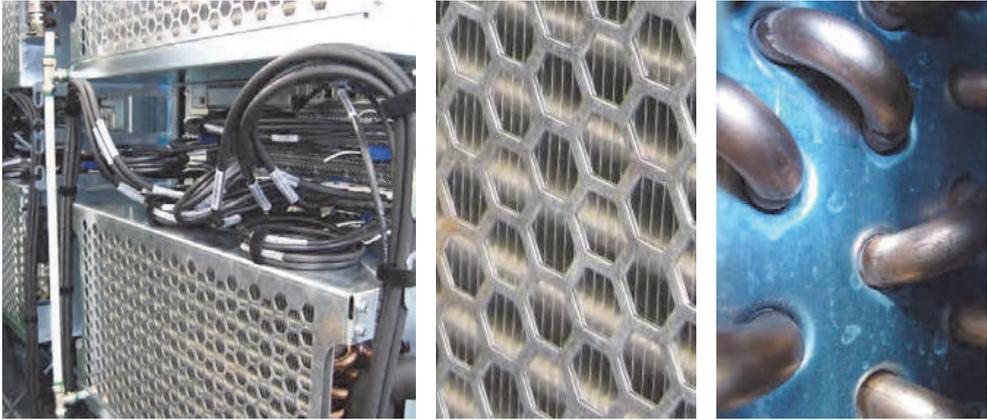


Fig. 2. Water-cooling in rack.

5.3 System core resources

For the main purpose of computing, large numbers of compute nodes are needed. The main system resources are cores and memory. The first two images in Figure 3 show a rack with compute nodes (SGI, 2011) and some thousand memory sticks needed for one supercomputer installation.

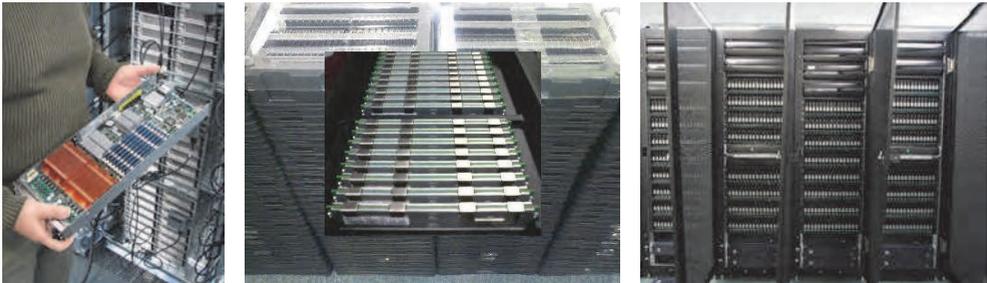


Fig. 3. Core resources and storage: compute nodes, memory, and disk storage unit.

5.4 System networks

With the increase of core resources, the more networks infrastructure is needed to operate these resources and make them accessible as a system. Figure 4 shows cabling and switches. Currently the significance of networks is rapidly increasing. Fibre optics are used to efficiently and effectively implementing networks. No wonder that in the year 2009 the physics Nobel Prize was dedicated to fibre optics, for the ground breaking achievements concerning the transmission of light in fibers for optical communication.

5.5 System storage

Besides cores, memory, and networks large storage capabilities are necessary for permanent storage. Figure 3 shows a disk storage unit consisting of several racks of hard disk drives, controllers, and servers.



Fig. 4. Networks, cabling, switches.

5.6 Systems connection

Linking high end resources is an important factor for ensuring economical use and enabling for access. In many cases these connections are built for redundancy, in order to switch connection for maintenance or emergencies.

5.7 System redundancy

Not only systems connections can be created using fallbacks. Figure 5 shows a network switch connecting some compute resources on redundant paths. With large numbers of components the rate of failure increases. Redundancy and appropriate concepts will minimise the risk due to component failures. Large resources, as we have seen, not only need redundant cores and memory but various additional redundancies. Figure 5 further shows redundant rack power supplies, redundant disk drive enclosures with redundant disks, for example with appropriate RAID level, and redundant meta data storage servers.

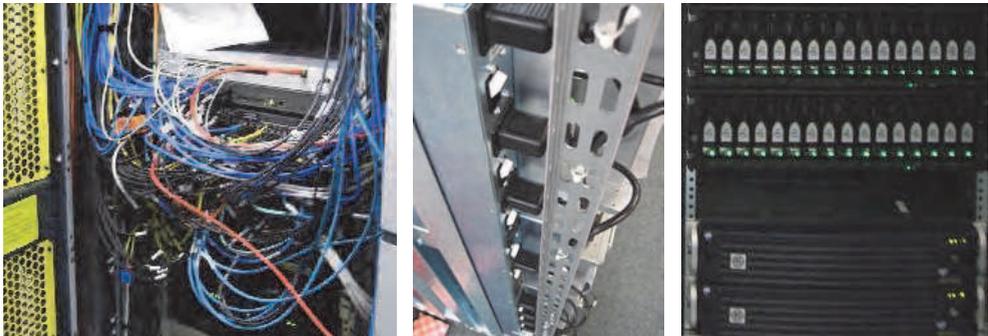


Fig. 5. Redundancy with linking resources, rack power supplies, disk drive enclosures and disks, meta data storage servers.

5.8 System operating

System operating will have to ensure local system access, component and services monitoring as well as hardware, physical and logical maintenance. Figure 6 illustrates operating access using local console, remote access, monitoring and physical maintenance.



Fig. 6. Operating using local console access, remote access, monitoring and physical maintenance.

6. Resources prerequisites

The complexity of these aspects shows why the integration of computing resources has takes so much time and why this is on the turn now. When we want do the planning for future resources and consumptive prerequisites there are essential requirements for technical and competence resources. For the technical resources we have to implement efficient and effective general purpose system installations, for loosely as well as for massively parallel processing:

- Architecture (e.g. MPP Massively Parallel Processing / SMP Symmetric Multi-Processing),
- Accessing Computing Power (MPI / OpenMP / loosely coupled interactive),
- Efficiency (Computing Power / Power Consumption),
- Storage and Archive.

For the competence resources a sustainable infrastructure has to be built, regarding research, scientific consulting, staff, operation, systems management, technical consulting, and administrative measures. Goals with using these resources are dynamically provisioning of secondary information, calculation and computation results, modeling and simulation. For exploiting the existing and future compute and storage resources, the basic “trust in computing” and “trust in information” requirements have therefore to be implemented in complex environments. For many scenarios this leads to international collaborations for data collection and use as well as to modular development and operation of components. The geosciences as other natural sciences cannot fulfill these requirements without interdisciplinary research and collaboration.

7. Software needs hardware

The next sections will present some examples on how these resources are used in geosciences and geoinformatics with integrated systems. It will show the complexity of the next generation of system architectures and usage that arises from integrating the necessary components and resources.

7.1 Geoexploration and integrated systems

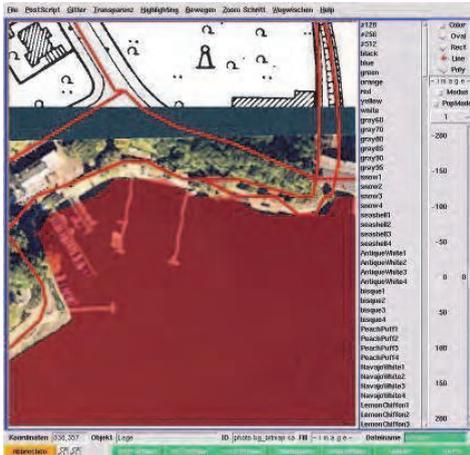
These sections present information system use cases and geoscientific application components that will profit from integrated information and computing systems using high end resources. It discusses features, concepts and frameworks, legal issues, technical requirements, and techniques needed for implementing these integrated systems. High Performance Computing has been recognised as one of the key technologies for the 21st century. It shows up with the problems existing today with implementing and using computing resources not only in batch mode but in combination with quasi interactive applications and it gives an outlook for future systems, components, and frameworks, and the work packages that have to be done by geoscientific disciplines, services and support, and resources providers from academia and economy.

There are two main objectives for interfacing modular complex integrated information and computing systems: “trust in computing” and “trust in information”. This goal for a long-term strategy means to concentrate on implementing methods for flexible use of envelope interfaces for use with integrated information and computing systems for managing objects and strengthen trust with systems from natural and geosciences, spatial sciences, and remote sensing as to be used for application, e.g., in environment management, healthcare or archaeology. Spatial means are tools, for the sciences involved. Therefore processing and computing is referred to the content which is embedded and used from the visual domain. Over the last years a long-term project, Geo Exploration and Information (GEXI) (GEXI, 1996, 1999, 2011) for analysing national and international case studies and creating as well as testing various implementation scenarios, has shown the two trust groups of systems, reflected by the collaboration matrices (Rückemann, 2010a). It has examined chances to overcome the deficits, built a collaboration framework and illuminated legal aspects and benefits (EULISP, 2011; Rückemann, 2010b). The information and instructions handled within these systems is one of the crucial points while systems are evolving by information-driven transformation (Mackert et al., 2009).

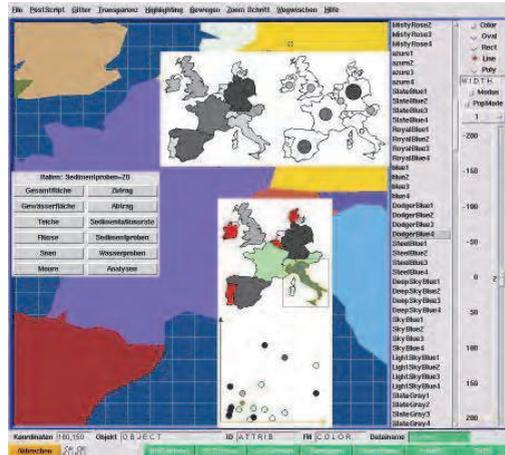
For computing and information intensive systems the limiting constraints are manifold. Recycling of architecture native and application centric algorithms is very welcome. In order to reuse information about these tasks and jobs, it is necessary to enable users to separate the respective information for system and application components. This can be done by structured envelope-like descriptions containing essential workflow information, algorithms, instructions, data, and meta data. The container concept developed has been called Compute Envelope (CEN). The idea of envelope-like descriptive containers has been inspired by the good experiences with the concept of Self Contained applications (SFC) (Rückemann, 2001). Envelopes can be used to integrate descriptive and generic processing information. Main questions regarding the topics of computing envelope interfaces are: Which content can be embedded or referenced in envelopes? How will these envelope objects be integrated into an information and computing system and how can the content be used? How can the context and environment be handled?

7.2 Geoscientific Information Systems and Active Source

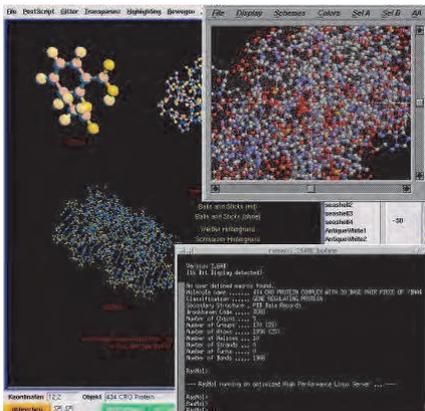
One of the most essential components for integrating geoexploration components and the access to computing resources are dynamical Geoscientific Information Systems (GIS) (Rückemann, 2001). Some screenshots of a dynamical GIS (Rückemann, 2009) illustrate the facets: Using dynamical data, raster, vector, secondary data, and events (Figure 7a), embedding dynamical components into components (Figure 7b), interacting with external components (Figure 7c), extending the user interface (Figure 7d).



(a) Raster, vector, secondary data, and events.



(b) Embed dynamical components into components.



(c) Interact with external components.



(d) Extend the user interface.

Fig. 7. Facettes of an Integrated Information System.

7.3 Information databases

Information databases are the base for many components in Information Systems. There are various concepts built on classical database architectures, special information structures,

and data collections. These architectures can be centralised or distributed. For the different purposes the implementations will use database software, file systems structures, meta data collection or combinations of more than one mechanism. In any case there should be a flexible and standardised way to interface and access the information. The information created within the LX Project has been subject of a long-term research initiative. It covers and combines for example educational treatises, individual and definitions and descriptions, scientific results, as well as Point of Interest (POI) information. It is capable of handling categorisation, ergonomic multi-lingual representations as well as information alternatives for various different purposes. It integrates typesetting issues and publishing, including formulas, customised sorting, indexing, and timeline functions for the information objects.

Listing 1 shows a simple example for an LX Encyclopedia entry.

1	Caldera	%-GP%-XX%---: Caldera	[Vulkanologie, Geologie]:
2		%-GP%-DE%---:	\lidx{Schüsselkrater},
3		%-GP%-DE%---:	\lidx{Chaldera},
4		%-GP%-DE%---:	\lidx{Caldera},
5		%-GP%-DE%---:	\lidx{cauldron},
6		%-GP%-DE%---:	\lidx{Kessel}.
7		%-GP%-DE%---:	...
8		%-GP%-EN%---:	\lidx{Chaldera},
9		%-GP%-EN%---:	\lidx{Schüsselkrater},
10		%-GP%-EN%---:	\lidx{Caldera},
11		%-GP%-EN%---:	\lidx{cauldron},
12		%-GP%-EN%---:	\lidx{Kessel}.
13		%-GP%-EN%---:	...
14		%-GP%-DE%---:	s. auch Capping stage
15		%-GP%-EN%---:	s. also Capping stage
16		%-GP%-XX%---:	%SRC: ...
17		%-GP%-XX%---:	%NET: http://...

Listing 1. Example LX Encyclopedia entry.

7.4 High End Computing

The integration of interactive and batch resources usage with dynamical applications does need flexibility in terms of interfaces and configuration. Various use cases studied distributed resources like capacity resources with Condor for example on ZIVcluster resources and on the other hand with High Performance Computing resources for example on ZIVSMP and HLRN (HLRN, 2011; ZIVGrid, 2008; ZIVHPC, 2011; ZIVSMP, 2011).

8. Integration framework – resources, services, disciplines

As well as analysing and separating the essential layers for building complex integrated systems, it is essential that these allow a holistic view on the overall system, for operation, development, and strategies level. The framework developed (Rückemann, 2010b) and studied for integrated information and computing is the Grid-GIS house (Figure 8) (Rückemann, 2009). An implementation of this kind is very complex and can only be handled with a modular architecture being able to separate tasks and responsibilities for resources (HEC, HPC), services, and disciplines. Figure 9 illustrates the logical components for integrated monitoring, accounting, and billing architecture for distributed resources and data usage. The legal aspects of combined usage of geo applications with services and distributed

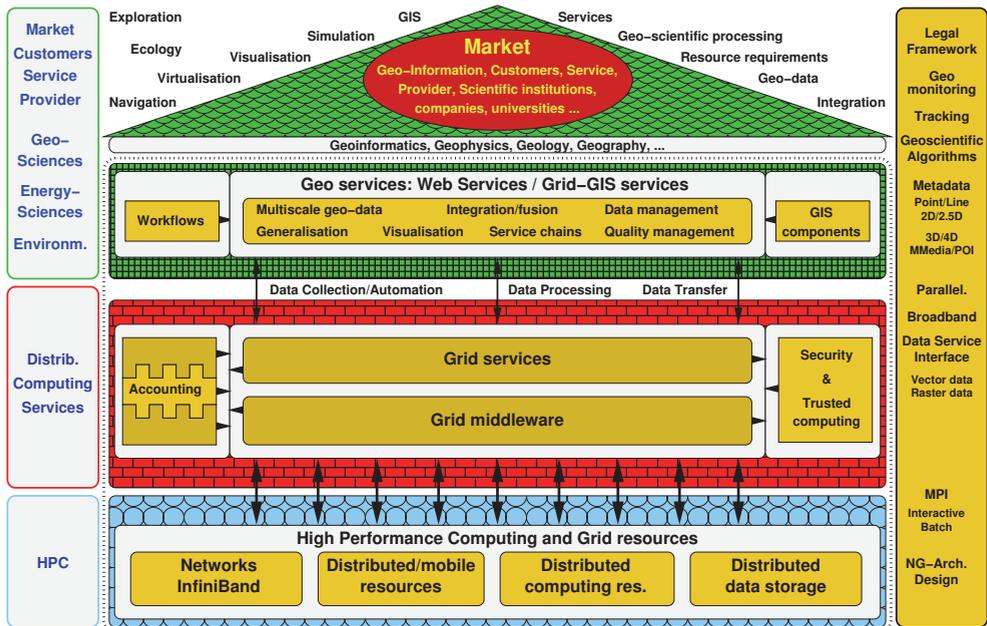


Fig. 8. Framework for integrated information and computing.

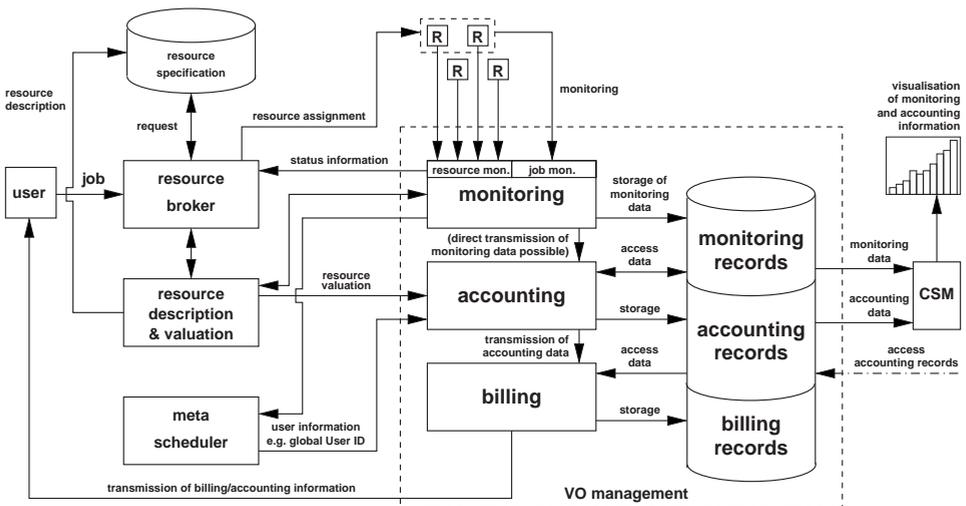


Fig. 9. Integrated monitoring, accounting, and billing architecture.

resources (Rückemann, 2010a;b) is shown in Figure 10. As the various components need dynamical and user-space interfaces, scripting mechanisms are unabdicable. The following sections discuss an example of a dynamical application using different distributed resources via flexible methods. Dynamical components and interfaces are implemented with Tcl and

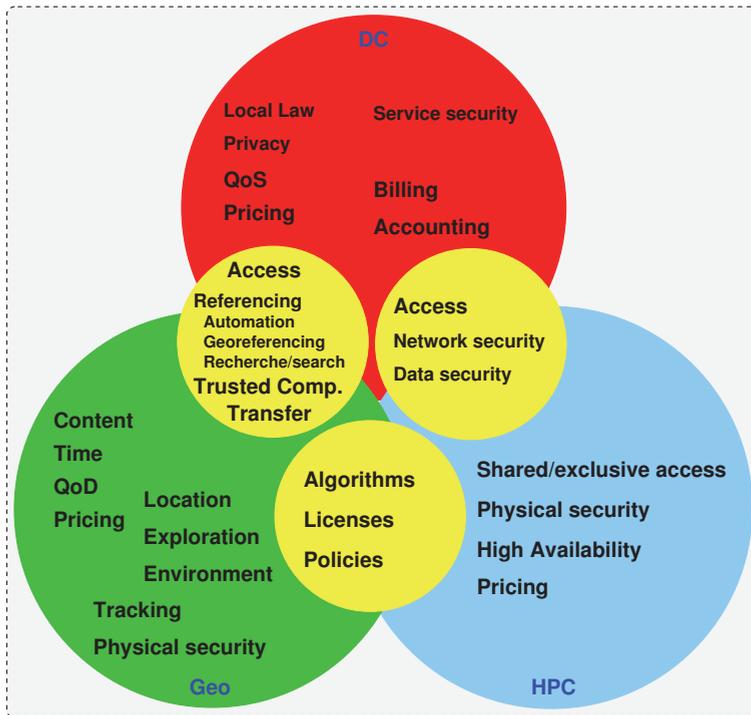


Fig. 10. Legal aspects of combined geo applications, services, and distributed resources usage.

derivatives and C (Tcl Developer Site, 2010), like the actmap application and Active Map procedures and data.

9. Integrated – InfoPoints using distributed resources

Using auto-events, dynamical cartography, and geocognostic aspects, views and applications using distributed compute and storage resources can be created very flexibly. As with the concept presented resources available from Distributed Systems, High Performance Computing, Grid and Cloud services, and available networks can be used. The main components are:

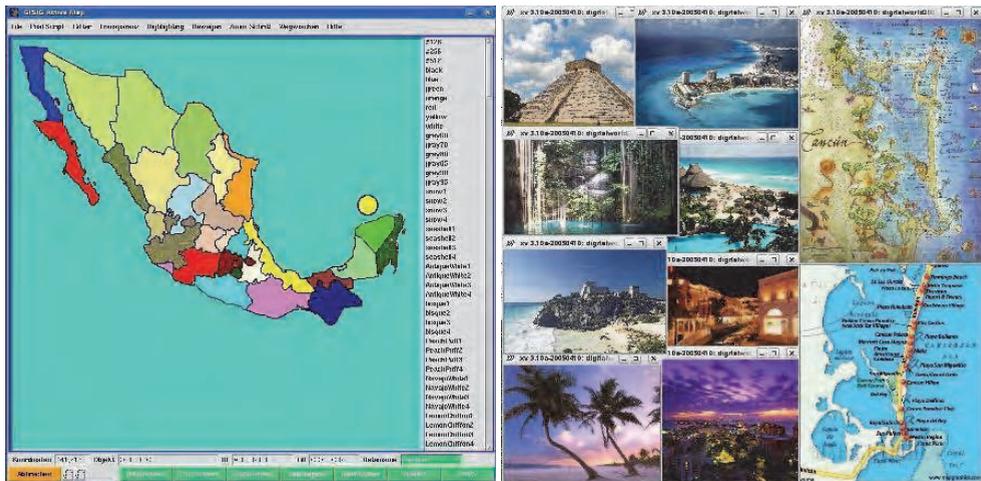
- interactive dynamical applications (frontend),
- distributed resources, compute and storage, configured for interactive and batch use,
- parallel applications and components (backend), as available on the resources,
- a framework with interfaces for using parallel applications interactively.

Besides the traditional visualisation a lot of disciplines like geo-resources and energy exploration, archaeology, medicine, epidemiology and for example various applications within the tourism industry can profit from the e-Science components. These e-Science components can be used for Geoscientific Information Systems for dynamical InfoPoints and multimedia,

Points of Interest based on Active Source (Active POI), dynamical mapping, and dynamical applications.

9.1 InfoPoints and dynamical cartography

With the integration of interactive dynamical components and dynamical cartography various surplus values can be used. Figure 11(a) shows an interactive Map of México (Rückemann, 2009). The yellow circle is an event sensitive Active Source object containing a collection of references for particular objects in the application. This type of object has been named InfoPoint. InfoPoints can use any type of start and stop routines triggered by events. Figure 11(b) shows a defined assortment of information, a view set, fetched and presented by triggering an event on the InfoPoint. The information has been referenced from within



(a) Interactive México with InfoPoint Yucatán.

(b) Sample view set of InfoPoint Yucatán.

Fig. 11. Integrated interactive dynamical components and InfoPoints.

the World Wide Web in this case. InfoPoints can depend on the cognitive context within the application as this is a basic feature of Active Source: Creating an application data set it is for example possible to define the Level of Detail (LoD) for zoom levels and how the application handles different kinds of objects like Points of Interest (PoI) or resolution of photos in the focus area of the pointing device.

9.2 Inside InfoPoints

The following passages show all the minimal components necessary for a fully functional InfoPoint. The example for this case study is mainly based on the Active Source framework. Triggered program execution ("Geoevents") of applications is shown with event bindings, start and stop routines for the data.

9.3 InfoPoints bindings and creation

Listing 2 shows the creation of the canvas for the InfoPoint and loading of the Active Source via bindings.

```

1 #
2 # actmap example -- (c) Claus-Peter R\"uckemann, 2008, 2009
3 #
4 #
5 #
6 # Active map of Mexico
7 #
8 #
9 erasePict
10 $w configure -background turquoise
11
12 pack forget .scale .drawmode .tagborderwidth \
13     .poly .line .rect .oval .setcolor
14 pack forget .popupmode .optmen_zoom
15
16 openSource     mexico.gas
17 removeGrid
18
19 ##EOF:

```

Listing 2. Example InfoPoint Binding Data.

This dynamical application can be created by loading the Active Source data with the actmap framework (Listing 3).

```

1 /home/cpr/gisig/actmap_sb.sfc mexico.bnd

```

Listing 3. Example creating the dynamical application.

9.4 InfoPoints Active Source

The following Active Source code (Listing 4) shows a tiny excerpt of the Active Source for the interactive Map of México containing some main functional parts for the InfoPoint Yucatán (as shown in Figure 11(a)).

```

1 #BCMT-----
2 ###EN \gisigsnip{Object Data: Country Mexico}
3 ###EN Minimal Active Source example with InfoPoint:
4 ###EN Yucatan (Cancun, Chichen Itza, Tulum).
5 #ECMT-----
6 proc create_country_mexico {} {
7     global w
8     # Yucatan
9     $w create polygon 9.691339i 4.547244i 9.667717i \
10         4.541732i 9.644094i 4.535433i 9.620472i 4.523622i \
11         9.596850i 4.511811i 9.573228i 4.506299i 9.531496i \
12     ...
13     -outline #000000 -width 2 -fill green -tags {itemshape province_yucatan}
14 }
15
16 proc create_country_mexico_bind {} {
17     global w
18     $w bind province_yucatan <Button-1> {showName "Province_Yucatan"}
19     $w bind province_quintana_roo <Button-1> \
20         {showName "Province_Quintana_Roo"}

```

```

21 }
22
23 proc create_country_mexico_sites {} {
24 global w
25 global text_site_name_cancun
26 global text_site_name_chichen_itza
27 global text_site_name_tulum
28 set text_site_name_cancun "Cancún"
29 set text_site_name_chichen_itza "Chichén_Itzá"
30 set text_site_name_tulum "Tulum"
31
32 $w create oval 8.80i 4.00i 9.30i 4.50i \
33 -fill yellow -width 3 \
34 -tags {itemshape site legend_infopoint}
35 $w bind legend_infopoint <Button-1> \
36 {showName "Legend_InfoPoint"}
37 $w bind legend_infopoint <Shift-Button-3> \
38 {exec browedit$t_suff}
39
40 $w create oval 9.93i 4.60i 9.98i 4.65i \
41 -fill white -width 1 \
42 -tags {itemshape site cancun}
43 $w bind cancun <Button-1> \
44 {showName "$text_site_name_cancun"}
45 $w bind cancun <Shift-Button-3> \
46 {exec browedit$t_suff}
47
48 $w create oval 9.30i 4.85i 9.36i 4.90i \
49 -fill white -width 1 \
50 -tags {itemshape site chichen_itza}
51 $w bind chichen_itza <Button-1> \
52 {showName "$text_site_name_chichen_itza"}
53 $w bind chichen_itza <Shift-Button-3> \
54 {exec browedit$t_suff}
55 ...
56 }
57
58 proc create_country_mexico_autoevents {} {
59 global w
60 $w bind legend_infopoint <Any-Enter> {set killatleave \
61 [exec ./mexico_legend_infopoint_viewall.sh $op_parallel ] }
62 $w bind legend_infopoint <Any-Leave> \
63 {exec ./mexico_legend_infopoint_kaxv.sh }
64
65 $w bind cancun <Any-Enter> {set killatleave \
66 [exec $appl_image_viewer -geometry +800+400 \
67 ./mexico_site_name_cancun.jpg $op_parallel ] }
68 $w bind cancun <Any-Leave> {exec kill -9 $killatleave }
69
70 $w bind chichen_itza <Any-Enter> {set killatleave \
71 [exec $appl_image_viewer -geometry +800+100 \
72 ./mexico_site_name_chichen_itza.jpg $op_parallel ] }
73 $w bind chichen_itza <Any-Leave> {exec kill -9 $killatleave }
74 ...
75 }
76

```

```

77 proc create_country_mexico_application_ballons {} {
78 global w
79 global isl
80 gisig:set_balloon $isl.country "Notation_of_State_and_Site"
81 gisig:set_balloon $isl.color "Symbolic_Color_od_State_and_Site"
82 }
83
84 create_country_mexico
85 create_country_mexico_bind
86 create_country_mexico_sites
87 create_country_mexico_autoevents
88 create_country_mexico_application_ballons
89 scaleAllCanvas 0.8
90 ##EOF

```

Listing 4. Example InfoPoint Active Source data.

The source contains a minimal example with the active objects for the province Yucatán in México. The full data set contains all provinces as shown in Figure 11(a). The functional parts depicted in the source are the procedures for:

- `create_country_mexico:`
The cartographic mapping data (polygon data in this example only) including attribute and tag data.
- `create_country_mexico_bind:`
The event bindings for the provinces. Active Source functions are called, displaying province names.
- `create_country_mexico_sites:`
Selected site names on the map and the active objects for site objects including the InfoPoint object. The classification of the InfoPoint is done using the tag `legend_infopoint`. Any internal or external actions like context dependent scripting can be triggered by single objects or groups of objects.
- `create_country_mexico_autoevents:`
Some autoevents with the event definitions for the objects (Enter and Leave events in this example).
- `create_country_mexico_application_ballons:`
Information for this data used within the Active Source application.
- Call section: The call section contains function calls for creating the components for the Active Source application at the start of the application, in this case the above procedures and scaling at startup.

Any number of groups of objects can be build. This excerpt only contains Cancun, Chichen Itza and Tulum. A more complex for this example data set will group data within topics, any category can be distinguished into subcategories in order to calculate specific views and multimedia information, for example for the category `site` used here:

- `city` (México City, Valladolid, Mérida, Playa del Carmen),
- `island` (Isla Mujeres, Isla Cozumel),

- archaeological (Cobá, Mayapan, Ek Balam, Aktumal, Templo Maya de Ixchel, Tumba de Caracol),
- geological (Chicxulub, Actun Chen, Sac Actun, Ik Kil),
- marine (Xel Há, Holbox, Palancar).

Objects can belong to more than one category or subcategory as for example some categories or all of these as well as single objects can be classified *touristic*. The data, as contained in the procedures here (mapping data, events, autoevents, objects, bindings and so on) can be put into a database for handling huge data collections.

9.5 Start an InfoPoint

Listing 5 shows the start routine data (as shown in Figure 11(b)). For simplicity various images are loaded in several application instances (*xv*) on the X Window System. Various other API calls like `Web-Get fetchWget` for fetching distributed objects via HTTP requests can be used and defined.

```

1 xv -geometry +1280+0 -expand 0.8 mexico_site_name_cancun_map.jpg &
2 xv -geometry +1280+263 -expand 0.97 mexico_site_name_cancun_map_hot.jpg &
3
4 xv -geometry +980+0 -expand 0.5 mexico_site_name_cancun.jpg &
5 xv -geometry +980+228 -expand 0.61 mexico_site_name_cancun_hotel.jpg &
6 xv -geometry +980+450 -expand 0.60 mexico_site_name_cancun_mall.jpg &
7 xv -geometry +980+620 -expand 0.55 mexico_site_name_cancun_night.jpg &
8
9 xv -geometry +740+0 -expand 0.4 mexico_site_name_chichen_itza.jpg &
10 xv -geometry +740+220 -expand 0.8 mexico_site_name_cenote.jpg &
11 xv -geometry +740+420 -expand 0.6 mexico_site_name_tulum_temple.jpg &
12 #xv -geometry +740+500 -expand 0.3 mexico_site_name_tulum.jpg &
13 xv -geometry +740+629 -expand 0.6 mexico_site_name_palm.jpg &

```

Listing 5. Example InfoPoint event start routine data.

9.6 Stop an InfoPoint

Listing 6 shows the stop routine data. For simplicity all instances of the applications started with the start routine are removed via system calls.

```

1 killall -9 --user cpr --exact xv

```

Listing 6. Example InfoPoint event stop routine data.

Using Active Source applications any forget or delete modes as well as using Inter Process Communication (IPC) are possible.

9.7 Integration and trust

Integrating components for mission critical systems does expect methods for handling “Trust in computation” and “Trust in information”. This is what Object Envelopes (OEN) and Compute Envelopes (CEN) have been developed for (Rückemann, 2011). Listing 7 shows a small example for a generic OEN file.

```

1 <ObjectEnvelope><!-- ObjectEnvelope (OEN)-->
2 <Object>
3 <Filename>GIS_Case_Study_20090804.jpg</Filename>
4 <Md5sum>...</Md5sum>
5 <Shalsum>...</Shalsum>
6 <DateCreated>2010-08-01:221114</DateCreated>
7 <DateModified>2010-08-01:222029</DateModified>
8 <ID>...</ID><CertificateID>...</CertificateID>
9 <Signature>...</Signature>
10 <Content><ContentData>...</ContentData></Content>
11 </Object>
12 </ObjectEnvelope>

```

Listing 7. Example for an Object Envelope (OEN).

An end-user public client application may be implemented via a browser plugin, based on appropriate services. With OEN instructions embedded in envelopes, for example as XML-based element structure representation, content can be handled as content-stream or as content-reference. The way this will have to be implemented for different use cases depends on the situation, and in many cases on the size and number of data objects. Listing 8 shows a small example for an OEN file using a content DataReference.

```

1 <ObjectEnvelope><!-- ObjectEnvelope (OEN)-->
2 <Object>
3 <Filename>GIS_Case_Study_20090804.jpg</Filename>
4 <Md5sum>...</Md5sum>
5 <Shalsum>...</Shalsum>
6 <DateCreated>2010-08-01:221114</DateCreated>
7 <DateModified>2010-08-01:222029</DateModified>
8 <ID>...</ID><CertificateID>...</CertificateID>
9 <Signature>...</Signature>
10 <Content><DataReference>https://doi...</DataReference></Content>
11 </Object>
12 </ObjectEnvelope>

```

Listing 8. OEN referencing signed data.

9.8 Implemented solution for integrated systems with massive resources requirements

For most interactive information system components a configuration of the distributed resources environment was needed. In opposite to OEN use, making it necessary to have referenced instead of embedded data for huge data sets, for CEN it should be possible to embed the essential instruction data. So there is less need for minimising data overhead and communication. Envelope technology is meant to be a generic extensible concept for information and computing system components (Rückemann, 2011). Figure 12 shows the workflow with application scenarios from GEXI case studies (Rückemann, 2010b). Future objectives for client components are:

- Channels for limiting communication traffic,
- Qualified signature services and accounting,
- Using signed objects without verification,
- Verify signed objects on demand.

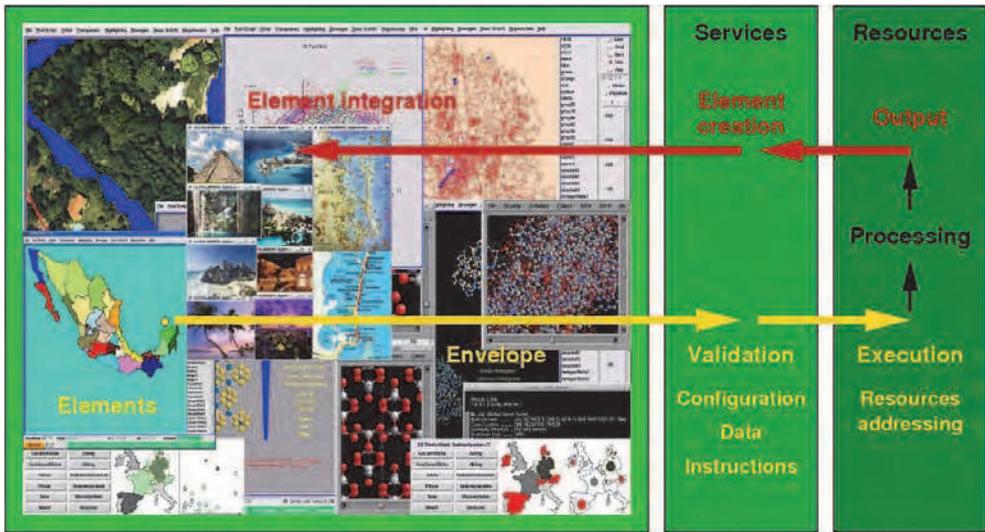


Fig. 12. Workflow with application scenarios from the GEXI case studies.

The tests done for proof of concept have been in development stage. A more suitable solution has now been created on a generic envelope base. An end-user public client application may be implemented via a browser plugin, based on appropriate services. The current solution is based on CEN files containing XML structures for handling and embedding data and information. This is so important because even in standard cases we easily have to handle hundreds of thousands of compute request from these components. In the easiest case it will be static information from information databases, for advanced cases it is for example conditional processing and simulation for thousands of objects like multimedia data or borehole depth profiling.

9.9 Integrated components in practice

When taking a look onto different batch and scheduling environments one can see large differences in capabilities, handling different environments and architectures. In the last years experiences have been gained in handling simple features for different environments for High Throughput Computing like Condor (Condor, 2010), workload schedulers like LoadLeveler (IBM, 2005) and Grid Engine (SGE, 2010), and batch system environments like Moab / Torque (Moab, 2010; Torque, 2010). Batch and interactive features are integrated with Active Source event management (Rückemann, 2001). Listing 9 shows a small example of a CEN embedded into an Active Source component.

```

1 #BCMT-----
2 ###EN \gisignip{Object Data: Country Mexico}
3 #ECMT-----
4 proc create_country_mexico {} {
5 global w
6 # Sonora
7 $w create polygon 0.938583i 0.354331i 2.055118i ...
8 #BCMT-----
    
```

```

 9  ##EN \gisignip{Compute Data: Compute Envelope (CEN)}
10  #ECMT-----
11  #BCEN <ComputeEnvelope>
12  ##CEN <Instruction>
13  ##CEN <Filename>Processing_Bat_GIS515.torque</Filename>
14  ##CEN <Md5sum>...</Md5sum>
15  ##CEN <Shalsum>...</Shalsum>
16  ##CEN <Sha512sum>...</Sha512sum>
17  ##CEN <DateCreated>2010-09-12:230012</DateCreated>
18  ##CEN <DateModified>2010-09-12:235052</DateModified>
19  ##CEN <ID>...</ID><CertificateID>...</CertificateID>
20  ##CEN <Signature>...</Signature>
21  ##CEN <Content>...</Content>
22  ##CEN </Instruction>
23  #ECEN </ComputeEnvelope>
24  ...
25  proc create_country_mexico_autoevents {} {
26  global w
27  $w bind legend_infopoint <Any-Enter> {set killatleave \
28    [exec ./mexico_legend_infopoint_viewall.sh $op_parallel ] }
29  $w bind legend_infopoint <Any-Leave> \
30    {exec ./mexico_legend_infopoint_kaxv.sh }
31  $w bind tulum <Any-Enter> {set killatleave \
32    [exec $appl_image_viewer -geometry +800+400 \
33    ./mexico_site_name_tulum_temple.jpg $op_parallel ] }
34  $w bind tulum <Any-Leave> \
35    {exec kill -9 $killatleave }
36  } ...

```

Listing 9. CEN embedded with Active Source.

Interactive applications based on Active Source have been used on Grid, Cluster, and HPC (MPP, SMP) systems.

9.10 Resources interface

Using CEN features, it is possible to implement resources access on base of validation, verification, and execution. The sources (Listing 10, 11) can be generated semi-automatically and called from a set of files or can be embedded into an actmap component, depending on the field of application.

```

 1 <ComputeEnvelope><!-- ComputeEnvelope (CEN)-->
 2 <Instruction>
 3 <Filename>Processing_Batch_GIS612.pbs</Filename>
 4 <Md5sum>...</Md5sum>
 5 <Shalsum>...</Shalsum>
 6 <Sha512sum>...</Sha512sum>
 7 <DateCreated>2010-08-01:201057</DateCreated>
 8 <DateModified>2010-08-01:211804</DateModified>
 9 <ID>...</ID>
10 <CertificateID>...</CertificateID>
11 <Signature>...</Signature>
12 <Content><DataReference>https://doi...</DataReference></Content>
13 <Script><Pbs>
14 <Shell>#!/bin/bash</Shell>

```

```

15 <JobName>#PBS -N myjob</JobName>
16 <Oe>#PBS -j oe</Oe>
17 <Walltime>#PBS -l walltime=00:10:00</Walltime>
18 <NodesPpn>#PBS -l nodes=8:ppn=4</NodesPpn>
19 <Feature>#PBS -l feature=ice</Feature>
20 <Partition>#PBS -l partition=hannover</Partition>
21 <Accesspolicy>#PBS -l naccesspolicy=singlejob</Accesspolicy>
22 <Module>module load mpt</Module>
23 <Cd>cd $PBS_O_WORKDIR</Cd>
24 <Np>np=$(cat $PBS_NODEFILE | wc -l)</Np>
25 <Exec>mpieexec_mpt -np $np ./dyna.out 2>&1</Exec>
26 </Pbs></Script>
27 </Instruction>
28 </ComputeEnvelope>

```

Listing 10. Embedded Active Source MPI script.

Examples for using High Performance Computing and Grid Computing resources include batch system interfaces and job handling. Job scripts from this type will on demand (event binding) be sent to the batch system for processing. The Actmap Computing Resources Interface (CRI) is an example for an actmap library (`actlcric`) containing functions and procedures and even platform specific parts in a portable way. CRI can be used for handling computing resources, loading Tcl or TBC dynamically into the stack (Tcl Developer Site, 2010) when given set behaviour_loadlib_actlib "yes".

```

1 <ComputeEnvelope><!-- ComputeEnvelope (CEN)-->
2 <Instruction>
3 <Filename>Processing_Batch_GIS612.pbs</Filename>
4 <Md5sum>...</Md5sum>
5 <Sha1sum>...</Sha1sum>
6 <Sha512sum>...</Sha512sum>
7 <DateCreated>2010-08-01:201057</DateCreated>
8 <DateModified>2010-08-01:211804</DateModified>
9 <ID>...</ID>
10 <CertificateID>...</CertificateID>
11 <Signature>...</Signature>
12 <Content><DataReference>https://doi...</DataReference></Content>
13 <Script><Condor>
14 <Environment>universe = standard</Environment>
15 <Exec>executable = /home/cpr/grid/job.exe</Exec>
16 <TransferFiles>should_transfer_files = YES</TransferFiles>
17 <TransferInputFiles>transfer_input_files = job.exe, job.input
18 </TransferInputFiles>
19 <Input>input = job.input</Input>
20 <Output>output = job.output</Output>
21 <Error>error = job.error</Error>
22 <Log>log = job.log</Log>
23 <NotifyMail>notify_user = ruckema@uni-muenster.de</NotifyMail>
24 <Requirements>
25 requirements = (Memory >= 50)
26 requirements = ( ( OpSys=="Linux" ) || ( OpSys=="AIX" ) ) && ( Memory >= 500 )
27 </Requirements>
28 <Action>queue</Action>
29 </Condor></Script>

```

```
30 </Instruction>  
31 </ComputeEnvelope>
```

Listing 11. Embedded Active Source Condor script.

With Actmap CRI being part of Active Source, calls to parallel processing interfaces, e.g., using InfiniBand, can be used, for example MPI (Message Passing Interface) and OpenMP, already described for standalone job scripts for this purpose, working analogical (Rückemann, 2009).

9.11 Service and operation

With the complexity of the high level integration of disciplines, services, and resources there are various aspects that cannot be handled in general as they will depend on scenario, collaboration partners, state of current technology, and many other. Based on the collaboration framework operation can integrate Service Oriented Architectures (SOA) and Resources Oriented Architectures (ROA). Based on the pre-implementation case studies and application scenarios it will be necessary to define agreements on the low level of services (S-Level) and operation (O-Level). For the S-Level integrated systems will need to collect Service Level Requests (SLR), define Service Level Specifications (SLS), and specify appropriate Service Level Agreements (SLA). According to these, on the O-Level Operational Level Agreements (OLA) have to be arranged. In all practical cases these agreements together with the underlying Service Level Management (SLM) should clearly take up less than two to five percent of the overall capacity for the collaboration for an efficient and effective system. The most important mostly non-technical factor for planning complex integrated systems therefore is to limit the dominance and growth of management, administrative, and operational tasks. Economic target centred contracts can be a solution to set limits to possible usuriousness, for example to restrict against “hydrocephalic” reporting and auditing.

10. Evaluation

The case studies demonstrated that integrated systems can be successfully implemented with enormous potential for flexible solutions. Disciplines, services, and resources level can be handled under one integrated concept. Interactive dynamical information systems components have been enabled to use an efficient abstraction and to handle thousands of subjobs for parallel processing, in demanding cases without the disadvantages of distributed systems. With the results of the case studies we can answer one additional question: What are the essential key factors for long-term use of integrated components? The academic and industry partners involved in the case studies emphasised that the key factors are:

- Information, instructions, and meta data have to be self explanatory.
- Multi-lingual information need appropriate interfaces in order to be editable and processable along with each other.
- Information has to be stored in a common non proprietary way.
- Tools for processing and interfacing the information have to be available without restrictions.
- Component atoms need to be recyclable.
- Information and components have to be widely portable.

11. Outlook

There are a number of aspects that have to be addressed in future work. These are mostly not only on the technology but on collaborative, organisational, and funding level with integrated information and computing systems. For geosciences and natural sciences algorithms and concepts for processing, visualisation, and extended use of data and information are available. The grand challenge with the wisdom will be to succeed in overcoming the fate of decision makers on scientific funding, that gathering a critical mass of acceptance in the society is vital.

12. Conclusion

This chapter has shown some prominent aspects of the complexity of the next generation of system architectures that arises from integrating the necessary components and computing resources, used in geosciences and geoinformatics. With technology advances new tools arise for geosciences research and Information Systems and Computing Systems will become more widely available. Due to the complexity and vast efforts necessary to implement and operate these systems and resources there is a strong need to enable economic and efficient use and operation. Hardware and software system components cannot be neglected anymore and viewed isolated. System architecture issues to software, legal, and collaborative aspects are in the focus and must be handled for operation, development, and strategies level. Various application scenarios from geosciences and natural sciences profit from the new means and concepts and will help to push the development not only of Geoscientific Information Systems and Computing Systems but of Information Systems and Computing Systems for the geosciences.

13. Acknowledgements

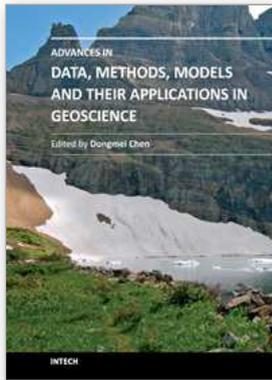
I am grateful to all national and international academic and industry partners in the GEXI cooperations for the innovative constructive work and case study support as well as to the colleagues at the Leibniz Universität Hannover, at the Institut für Rechtsinformatik (IRI), the North-German Supercomputing Alliance (HLRN), the Westfälische Wilhelms-Universität (WWU) Münster, at the Zentrum für Informationsverarbeitung (ZIV) Münster, in the German Grid Initiative D-Grid and the participants of the postgraduate European Legal Informatics Study Programme (EULISP) and of the WGGEOSP work group as well as the colleagues at the last years GEOWS, GEOProcessing, CYBERLAWS, ICDS, and INFOCOMP international conferences for prolific discussion of scientific, legal, and technical aspects as well as to the staff at ZIV and the partner institutes and associated HPC companies for supporting this work by managing and providing HEC resources over the years. I am especially grateful to Hans-Günther Müller, SGI, for fruitful discussion and providing support with hardware photos.

14. References

- Condor (2010). Condor, High Throughput Computing. URL: <http://www.cs.wisc.edu/condor/> [accessed: 2010-10-10].
- EULISP (2011). *Fundamental Aspects of Information Science, Security, and Computing (Lecture)*, EULISP Lecture Notes, European Legal Informatics Study Programme, Institute for

- Legal Informatics, Leibniz Universität Hannover (IRI / LUH). URL: <http://www.eulisp.de> [accessed: 2011-01-01].
- GEXI (1996, 1999, 2011). Geo Exploration and Information (GEXI). URL: <http://www.user.uni-hannover.de/cpr/x/rprojs/en/index.html#GEXI> (Information) [accessed: 2010-05-02].
- HLRN (2011). HLRN, North-German Supercomputing Alliance (Norddeutscher Verbund für Hoch- und Höchstleistungsrechnen). URL: <http://www.hlrn.de> [accessed: 2011-07-10].
- IBM (2005). IBM Tivoli Workload Scheduler LoadLeveler. URL: <http://www-03.ibm.com/systems/software/loadleveler/> [accessed: 2010-10-10].
- Mackert, M., Whitten, P. & Holtz, B. (2009). *Health Infonomics: Intelligent Applications of Information Technology*, IGI Global, pp. 217–232. Chapter XII, in: Pankowska, M. (ed.), *Infonomics for Distributed Business and Decision-Making Environments: Creating Information System Ecology*, ISBN: 1-60566-890-7, DOI: 10.4018/978-1-60566-890-1.ch010.
- Moab (2010). Moab: Admin Manual, Users Guide. URL: <http://www.clusterresources.com> [accessed: 2010-10-10].
- Rückemann, C.-P. (2001). *Beitrag zur Realisierung portabler Komponenten für Geoinformationssysteme. Ein Konzept zur ereignisgesteuerten und dynamischen Visualisierung und Aufbereitung geowissenschaftlicher Daten*, Diss., Westfälische Wilhelms-Universität, Münster, Deutschland. 161 (xxii+139) S., URL: <http://wwwmath.uni-muenster.de/cs/u/ruckema/x/dis/download/dis3acro.pdf> [accessed: 2009-11-16].
- Rückemann, C.-P. (2009). Dynamical Parallel Applications on Distributed and HPC Systems, *International Journal on Advances in Software* 2(2). ISSN: 1942-2628, URL: <http://www.iariajournals.org/software/> [accessed: 2009-11-16].
- Rückemann, C.-P. (2010a). Integrating Future High End Computing and Information Systems Using a Collaboration Framework Respecting Implementation, Legal Issues, and Security, *International Journal on Advances in Security* 3(3&4): 91–103. Savola, R., (ed.), VTT Technical Research Centre of Finland, Finland, URL: http://www.iariajournals.org/security/sec_v3_n34_2010_paged.pdf [accessed: 2011-07-10], URL: <http://www.iariajournals.org/security/> [accessed: 2011-07-10].
- Rückemann, C.-P. (2010b). Legal Issues Regarding Distributed and High Performance Computing in Geosciences and Exploration, *Proceedings of the Int. Conf. on Digital Society (ICDS 2010), The Int. Conf. on Technical and Legal Aspects of the e-Society (CYBERLAWS 2010), February 10–16, 2010, St. Maarten, Netherlands Antilles*, IEEE Computer Society Press, IEEE Xplore Digital Library, pp. 339–344. ISBN: 978-0-7695-3953-9, URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5432414> [accessed: 2010-03-28].
- Rückemann, C.-P. (2011). Envelope Interfaces for Geoscientific Processing with High Performance Computing and Information Systems, *Proceedings International Conference on Advanced Geographic Information Systems, Applications, and Services (GEOProcessing 2011), February 23–28, 2011, Gosier, Guadeloupe, France / GEOProcessing 2011, ICDS 2011, ACHI 2011, ICQNM 2011, CYBERLAWS 2011, eTELEMED 2011, eL&mL 2011, eKNOW 2011 / DigitalWorld 2011*, XPS, Xpert Publishing

- Solutions, pp. 23–28. Rückemann, C.-P., Wolfson, O. (eds.), 6 pages, ISBN: 978-1-61208-003-1, URL: http://www.thinkmind.org/download.php?articleid=geoprocessing_2011_2_10_30030 [accessed: 2011-07-10].
- SGE (2010). Sun Grid Engine. URL: <http://gridengine.sunsource.net/> [accessed: 2010-10-10].
- SGI (2011). SGI, Silicon Graphics GmbH. URL: <http://www.sgi.com> [accessed: 2011-07-10].
- Tcl Developer Site (2010). Tcl Developer Site. URL: <http://dev.scriptics.com/> [accessed: 2010-10-10].
- Torque (2010). Torque Admin Manual. URL: <http://www.clusterresources.com/torquedocs21/> [accessed: 2010-10-10].
- ZIVGrid (2008). ZIV der WWU Münster – ZIVGrid. URL: <http://www.uni-muenster.de/ZIV/Server/ZIVGrid/> [accessed: 2008-12-23].
- ZIVHPC (2011). ZIVHPC, HPC Computing Resources. URL: <https://www.uni-muenster.de/ZIV/Technik/ZIVHPC/index.html> [accessed: 2011-02-20].
- ZIVSMP (2011). ZIVSMP, SMP Computing Resources. URL: <https://www.uni-muenster.de/ZIV/Technik/ZIVHPC/ZIVSMP.html> [accessed: 2011-02-20].



Advances in Data, Methods, Models and Their Applications in Geoscience

Edited by Dr. DongMei Chen

ISBN 978-953-307-737-6

Hard cover, 336 pages

Publisher InTech

Published online 22, December, 2011

Published in print edition December, 2011

With growing attention on global environmental and climate change, geoscience has experienced rapid change and development in the last three decades. Many new data, methods and modeling techniques have been developed and applied in various aspects of geoscience. The chapters collected in this book present an excellent profile of the current state of various data, analysis methods and modeling techniques, and demonstrate their applications from hydrology, geology and paleogeomorphology, to geophysics, environmental and climate change. The wide range methods and techniques covered in the book include information systems and technology, global position system (GPS), digital sediment core image analysis, fuzzy set theory for hydrology, spatial interpolation, spectral analysis of geophysical data, GIS-based hydrological models, high resolution geological models, 3D sedimentology, change detection from remote sensing, etc. Besides two comprehensive review articles, most chapters focus on in-depth studies of a particular method or technique.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Claus-Peter Rückemann (2011). Queueing Aspects of Integrated Information and Computing Systems in Geosciences and Natural Sciences, *Advances in Data, Methods, Models and Their Applications in Geoscience*, Dr. DongMei Chen (Ed.), ISBN: 978-953-307-737-6, InTech, Available from:
<http://www.intechopen.com/books/advances-in-data-methods-models-and-their-applications-in-geoscience/queueing-aspects-of-integrated-information-and-computing-systems-in-geosciences-and-natural-sciences>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the [Creative Commons Attribution 3.0 License](#), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.