Monte Carlo Verification of IMRT treatment plans on Grid

Andrés GÓMEZ[a], Carlos FERNÁNDEZ SÁNCHEZ[a], José Carlos MOURIÑO GALLEGO[a], Javier LÓPEZ CACHEIRO[a], Francisco J. GONZÁLEZ CASTAÑO[b], Daniel RODRÍGUEZ-SILVA[b], Lorena DOMÍNGUEZ CARRERA[b], David GONZÁLEZ MARTÍNEZ[b], Javier PENA GARCÍA[c], Faustino GÓMEZ RODRÍGUEZ[c], Diego GONZÁLEZ CASTAÑO[c], Miguel POMBAR CAMEÁN[d]

[a]*Fundación Centro Tecnológico de Supercomputación de Galicia (CESGA), Santiago de Compostela, Spain*
{agomez,carlosf,jmourino,jlopez}@cesga.es

[b]*Departamento de Ingeniería Telemática, University of Vigo, Spain*
{javier,darguez}@det.uvigo.es

[c]*Departamento de Física de Partículas, University of Santiago de Compostela, Spain*
{javierpg,faustgr}@usc.es

[d]*Hospital Clínico Universitario de Santiago, Santiago de Compostela, Spain*
mrpombar@usc.es

Abstract: The eIMRT project is producing new remote computational tools for helping radiotherapists to plan and deliver treatments. The first available tool will be the IMRT treatment verification using Monte Carlo, which is a computational expensive problem that can be executed remotely on a GRID. In this paper, the current implementation of this process using GRID and SOA technologies is presented, describing the remote execution environment and the client.

Keywords: Radiotherapy, Monte Carlo, treatment plan verification, IMRT, EGEE, GRID, gLite.

1. Introduction

Intensity Modulated Radiation Therapy (IMRT) is a state-of-the-art technique in radiation therapy that allows the delivery of a non-uniform photon fluence for each incident angle of the X-ray beam generated by a medical linear accelerator (linac). It presents clinical advantages over conformal radiation techniques (CRT), which only adjust the shape of the radiation beam to the shape of the tumour. Usually, the calculation of the directions of incidence and the shape of the radiation fields that have to be delivered to build up the desired dose distribution (treatment planning) is done using local software tools called treatment planning systems (TPS) such as Pinnacle,

XiO, Oncentra, Corvus, etc., running on workstations at hospital premises. Specialized personnel (radiotherapists) compute treatment plans either employing their previous knowledge and experience, trial-and-error class-solutions or, for more complex treatment plans, built-in optimization tools. Treatments are tailored to deliver uniform doses to the planned target volumes (PTVs) while keeping the dose to surrounding tissues, especially to the organs at risk (OARs), within the prescribed tolerances. In both cases the goals to be achieved are specified by the radiation oncologists. There are two common methods to deliver an IMRT treatment in computer controlled linacs: step-and-shoot, where the leaves of the multileaf collimator (MLC) are moved in discrete steps between two consecutive irradiations, and dynamic-MLC, where the leaves are moved continuously during irradiation.

The requirements in maximum computation time force TPS tools to perform approximations both in dose calculation engines and optimization algorithms. The most accurate dose calculation techniques included in those codes are based on convolution/superposition methods (C/S) [1], which suffer from certain limitations in high density gradient regions. Due to the complexity of the IMRT plan and the compulsory approximations during dose optimization, each IMRT treatment has to be experimentally verified prior to its actual delivery to the patient, involving the final accelerator and dose measurement units. These in-phantom expensive measurements require large amounts of time and could be avoided or minimized by employing Monte Carlo techniques [2] to simulate the treatment *in silico*.

Nowadays, in developed countries, more than 40,000 people per million inhabitants have been diagnosed a cancer [3] yearly and approximately 50% of them receive radiotherapy. In 1999, more than 56,000 patients were irradiated only in Spain [4]. All treatments follow a planning protocol to ensure the quality and effectiveness of the session, and the treatments should be planned in a short period of time (the mean time between the first visit and the begining of radiotherapy treatment is 18,87 days in Spanish public hospitals [4]). Many radiotherapists have to plan over 600 to 1,200 patients per year, with a mean value of 925 [5]. This situation puts a high pressure on them, raising the need of new optimization tools. There are over 10,000 accelerators worldwide that irradiate around 4 millions patients yearly [3]. Only in Spain, there are over 115 particle accelerators in 70 hospitals with radiotherapy facilities [6], four of them in Galicia including the *Complexo Hospitalario Universitario de Santiago*, which is a partner in the eIMRT [7] project.

As a result of the extremely long CPU time and the large amount of plans to calculate, Monte Carlo verification is a clear best-case of GRID technologies exploitation. This is the aim of the eIMRT project which is devoted to produce new tools based on computational intensive algorithms for helping the radiotherapist to plan and verify the radiotherapy treatments. One of these new tools is the verification of planned treatments using Monte Carlo methods, which is presented in this paper. We hope that this tool has a significant impact due to the high number of hospitals that may benefit from it.

The paper is divided in four sections. First of all, a brief description of eIMRT architecture is given. The next section presents the verification process. The third section describes the computational implementation on GRID and discusses the associated difficulties. The paper ends with a final section dedicated to future work and conclusions.

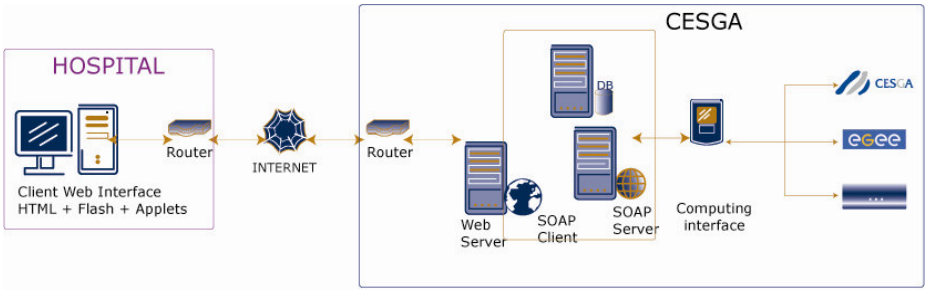

Figure 1. High-level eIMRT architecture

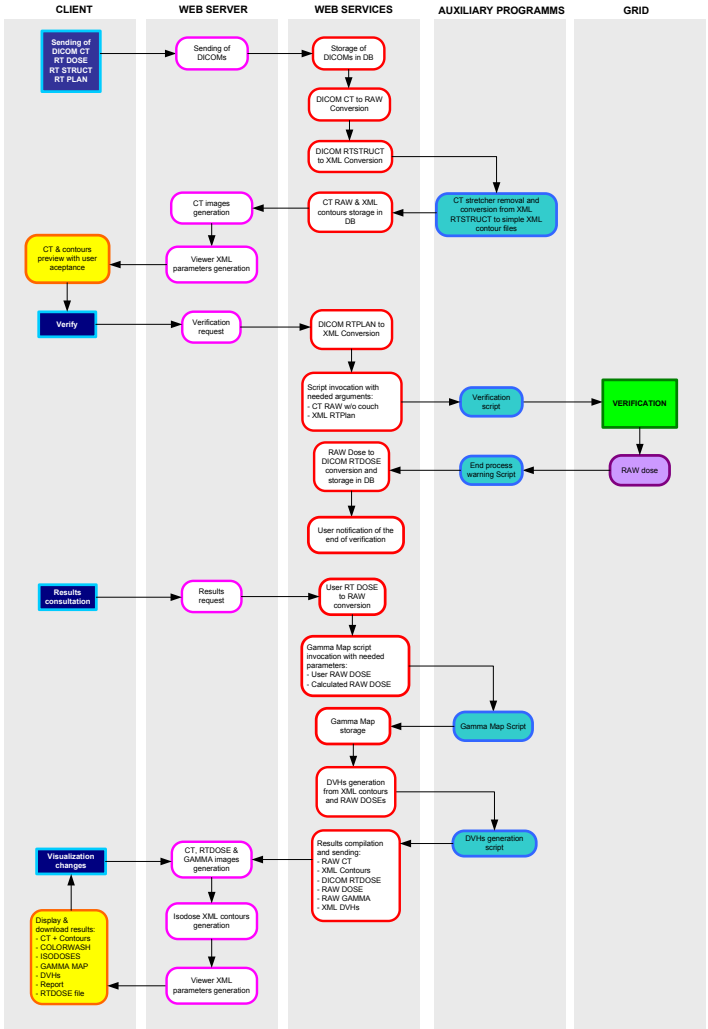

Figure 2: Schematic flow chart of the verification process**.**

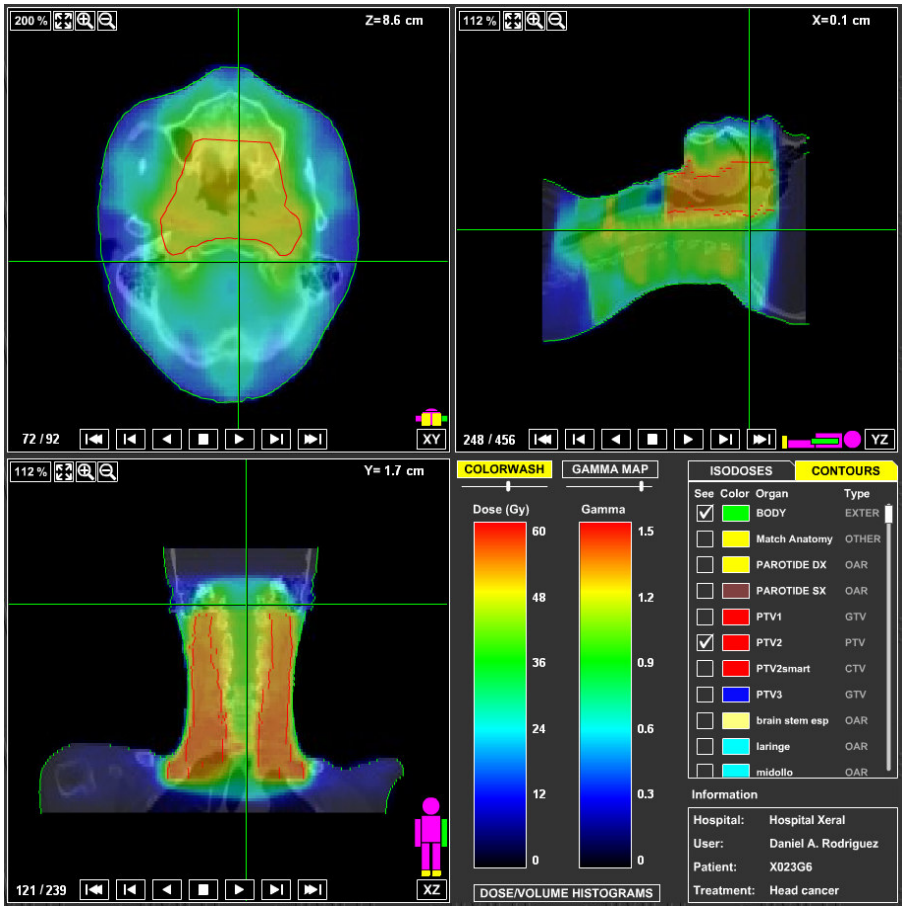

Figure 3. User visualization interface: It shows the slices of the patient in three directions (XY, YZ and XZ), presenting information about the calculated doses, the areas to consider and the comparison between the reference dose (calculated by Monte Carlo) and the treatment dose (calculated by the TPS) using gamma maps.

2. eIMRT Architecture

The eIMRT architecture has been described in a previous paper [7]. Figure 1 shows a general overview. It comprises four layers: client, application server, computing interface/data server and computing elements. The client has been developed for demonstration purposes because all the system is designed following the SOA (Service Oriented Architecture) paradigm. It is divided in two layers: a web server based on Coocon [8], which calls the web services and transforms SOAP messages to HTML, and an Internet navigator. Also, there are two special plugins, one based on Java to upload the DICOM files and anonymize them, and another one based on Flash for visualization of results (see Figure 3 for a view of the visualization interface). Note that the client interface is rather simple. Complexity is completely enclosed at the server side, which accesses high-throughput computing via web services [9]. Due to the open architecture, we can use several computing services. In fact, we have implemented the

computing interface for using the local cluster facilities through a queue system (Sun Grid Engine) or for submitting the jobs to the GRID, in this case using the gLite middleware [10]. Nowadays, the authentication with the GRID infrastructure is done using a single certificate for all the tasks, but in the near future personalized authentication based on certification will be implemented.

Currently, the most important web services are:

- UserManagement, which manages all the information related to user and control sessions.
- FileManagement, which makes all the operations for uploading and controlling the files related to the treatments as DICOM CT, DICOM RTPlan, etc.
- TreatmentManagement, for managing the information and operations related to a treatment.
- Verification, which submits and controls the operations related to the verification of a treatment.
- MapManagement, for generating different maps to compare two dose distributions. This web server provides an open interface for different types of maps. Currently the gamma maps [11] are implemented, but other maps can be supported.
- Monitorization, which allows the monitorization of the status of a computational operation, such as verification, and alerts the final user when it ends.

In the near future, other services will be implemented for the optimization of treatments or the characterization of accelerators (currently an expensive off-line process).

3. Verification of IMRT Treatment Plans

To **validate a treatment**, the end user employs the system to check the dose distribution that has been calculated at the hospital (for instance, with a TPS) against the dose distribution associated to the same treatment resulting from a more accurate dose calculation method (Monte Carlo at the current stage). A schematic representation of the full process is shown in figure 2. The output of both methods is compared, for example using the gamma maps generated by the MapManagement service. The eIMRT project has implemented a Monte Carlo verification process based on the well known and validated BEAMnrc package [12], and comprises five phases (see figure 4 for a schematic view):

- **Phase 1: Accelerator simulation**. It takes the data describing the geometry of the linear accelerator, its radiation source and the treatment to be verified. Basically it takes the information from the input DICOM RTplan of the treatment and produces the input files in the right format for the BEAMnrc Monte Carlo code. There is a single input file for the accelerator head simulation (from the bremsstrahlung target to the bottom of the collimators) for each of the following:
  a) Field in CRT treatments. In Conformal Radiation Therapy the shape of the radiation field is adjusted to fit the profile of the tumour according to the beam's eye view.
  b) Segment in IMRT step-and-shoot treatments.
  c) Control point in IMRT dynamic-MLC treatments.

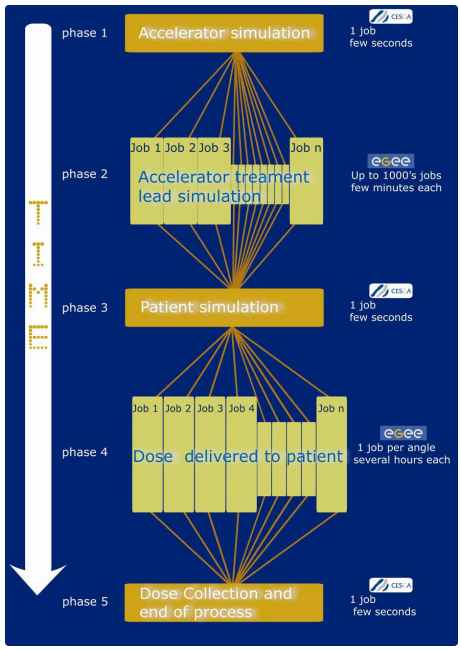

Figure 4. Schematic view of the Monte Carlo verification module. The output of phase 5 can be compared with the input dose distributions using,for example, the gamma maps generated by the MapManagement service.

Due to the reduced CPU time, this step can be executed locally at the server side. It produces from few files for CRT treatments (about one per angle) to several thousands in case of IMRT.

- **Phase 2: Accelerator treatment head simulation**. The whole accelerator treatment head is simulated for each input file, optimizing the variance reduction techniques to maximize the particle production and scoring for that field shape. This step executes the BEAMnrc code for each input file. It needs about 20 hours of CPU (in a Pentium IV at 3.0GHz) for each treatment. It generates one output file for each input.
- **Phase 3: Patient simulation.** The particles associated to each field shape calculated in the previous step are collected and grouped together attending to the energy of the beam and gantry, table and collimator angles. Consistency checks are also performed in order to ensure that no particles were lost during the simulations of the previous step. CT data describing a patient is converted to densities for further calculation of the dose delivered by each of the fields. It takes into account the data of the characterization of the computerized tomography (CT). The output of this phase produces the input files for simulating the dose deposition inside the patient using also Monte Carlo.
- **Phase 4: Dose delivered to the patient.** The dose-inside-the-patient is calculated using the DOSXYZnrc Monte Carlo code [12]. Since this task is highly parallelizable, it can be divided in many different independent jobs. Currently, the

calculation is divided in one job for each incident angle, although divisions with finer granularity are possible. It needs about 35 hours in a single CPU.

- **Phase 5: Dose collection and end of process.** The results are merged into a final, single dose distribution. The dose distribution normalized per unit of primary fluence (i.e. the amount of radiation that reaches the target of the accelerator where the electrons collide to produce photons) is converted into an absolute or relative dose distribution by comparison with TPS results (i.e. the actual dose that the accelerator delivers).

Once the Monte Carlo process has finished, the radiotherapist can manually compare his independently calculated dose with the dose calculated by the Monte Carlo verification. For this task, we have developed a special service that produces gamma maps, taking the dose maps as input. This task only needs a few seconds of CPUs and the result can be graphically displayed on the client (see figure 3). No single value can be produced to assess the quality of the treatment plan, which is a decision to be taken by the radiotherapist.

4. GRID Execution

The previously described Monte Carlo verification process requires a high computational capacity. For each treatment verification, the execution of hundreds of short jobs is necessary in phase 2 and in phase 4, only few jobs (usually less than 10) run, but each one consumes several hours. So, adding all phases, tenths of hours of CPU are consumed in each treatment verification. Also, it may be the case that a treatment has to be verified several times before achieving a good solution. Therefore, a large amount of computational resources is necessary to reduce the time-to-solution. Since not all the treatments have to be verified, the infrastructures of different hospitals or a GRID infrastructure can be shared to fulfil the expectations of the radiotherapists and get a solution within a proper time frame for clinical purposes.

Each one of the described phases of the treatment verification process has different computational necessities. Consequently, they will be executed in different contexts. They will be described in detail from the computational point of view.

Phase 1, which simulates the accelerator, is sequential and requires a short execution time, so can be executed locally. It needs a single input file that specifies the files that are necessary for the execution of the whole process, like DICOM files of the patient, the files that describe the accelerator, as well as other files related to the process itself that are independent from the treatments. All the necessary files for phase 2 are produced as the output, one for each process, as well as a text file with as many rows as Monte Carlo processes to be executed and three columns: one indicating the input file, another one with the EGSnrc executable to be used for this job and a third one that indicates the file with the data of the accelerator. This intermediate command file is produced to make the verification process independent of the final infrastructure. So, it can be used as the input for different types of infrastructures such as GRID, clusters or any new type of middleware. Even better, in the future we can mix several infrastructures for a single treatment, executing the final jobs in the GRID and in a local cluster simultaneously.

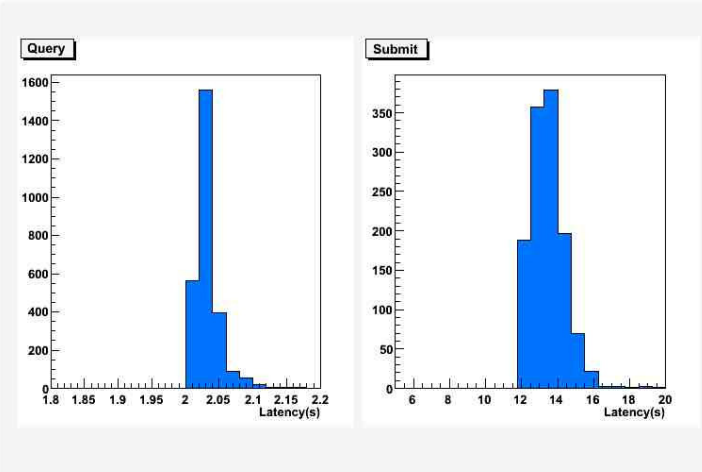

Figure 5: Measured latencies (in seconds) for query (left) and submit (right) operations in EGEE. The mean time for query the status of a job is 2.04 ±0.02 with a minimum latency of 2 seconds and for submitting a single job is 13.46 ± 0.94 seconds.

After this first pre-processing stage, all jobs of phase 2 have to be executed. The number of jobs of this phase depends on the treatment and the accelerator, but it is of the order of several hundreds (more than 1,200 in some cases). Nevertheless, its individual run time is low, in the range of a couple of minutes. Since all of them are independent, these jobs can be sent to the GRID, taking advantage of the many resources available. A JDL file is created for each process, indicating the executable and the necessary input files. The files that depend on the treatment are sent in the InputSandBox, whereas those that are common to all processes (as the executables) are copied from the Storage Element at the beginning of the job. All the outputs are recovered using the OutputSandBox. Due to the current undesirable high latency for sending jobs to the EGEE infrastructure (see figure 5), if compared to the real execution time, the tasks are grouped in a few jobs taking into account this latency and the CPU time, in order to optimize the final elapsed time. Once they are sent to the GRID, the status of all the jobs is continuously monitored to produce useful information for the user (who can ask for the status of the verification process at any time) and to resubmit failed jobs due to infrastructure failures. Again, the high latency of the EGEE infrastructure affects the monitoring process, yielding it infeasible when the jobs are individually submitted. By grouping them, this task becomes feasible because the number of independent jobs is limited.

Once phase 2 has finished satisfactorily, phase 3 gets all the output data of the accelerator simulation and produces the input files for the patient simulation, using the same input file of phase 1 and the results of phase 2. It produces the input files for the next phase and a command file describing the tasks in the next phase. It is a sequential job with reduced elapsed time that can be executed locally in the server.

Phase 4 is similar to the second one. It requires less jobs (initially, one per incident angle, i.e., between 3 and 7 in usual treatment plans, although each job can be parallelized), but each one requires several CPU hours. They are sent to the GRID following the same procedure as in phase 2, but, given the characteristics of the jobs to execute, in this case there is no grouping. Monitoring and fault-tolerant processes are

also executed. This phase is the most computationally expensive, with a long elapsed time. Since each job can be trivially parallelized, we expect to reduce the final elapsed time of this task in the future. This is an on-going work because we want to use the same cluster for each single job in order to reduce the time for uploading and downloading files, and it is a feature only recently available in EGEE.

Finally, in phase 5 all the output files are downloaded from the GRID, post-processed and merged in a single dose file of a few MBs, sending an alerting message to the final user. This is a sequential short task that resides in the server.

5. Conclusions and Future Work

The decoupled eIMRT architecture is a cost-effective solution to speed-up the CPU-greedy processes in advanced radiotherapy planning: accelerator characterization, treatment validation and treatment optimization. As far as we know, there are no similar distributed environments for verification and optimization of radiotherapy treatments, although other desktop tools as DoseLab [13] for dose comparison or CEER [14] are available. In this paper, an implementation of the IMRT verification on the GRID has been presented. The two most computationally demanding steps have been implemented to be executed on a GRID or a distributed environment, because they are well suited for this kind of architectures due to their highly parallel nature. They have been tested on the EGEE infrastructure. We have identified some problems with submission and monitoring due to the high latencies of both tasks in the actual infrastructure. Although the total CPU time is not very high (about 100 hours for each treatment), a short response time is needed to fulfil the expectations of the radiotherapists, who need to check the treatment as soon as possible. Therefore, the problems we have studied represent a good best-case for an interactive GRID environment, as proposed by the Interactive European Grid Project [15].

The ongoing project will produce new tools and improvements in the near future. The next step is the inclusion of the treatment optimization process and the characterization of any accelerator (So far only previously characterized accelerators are allowed). From the point of view of GRID technologies, the full system should improve security, including end-to-end authorization based on certificates, the full support of GRID distributed storage or the access to DICOM files directly from the hospitals without having to upload them. Also, we plan to use the Workload Management Server [10] for submitting jobs, included in the last production version of gLite. It allows bulk submission and shared sandboxes, which will be very helpful for phase 2 and 4.

In the near future, a full Monte Carlo TPS system will be available. It will benefit both from the improvements in Monte Carlo simulation and the increasing multi-core CPU power in the workstations. However, due to the large number of treatments to be planned in each hospital, we believe that a distributed GRID environment based on the SOA paradigm will still be useful, and it will easily provide access to computer power, new functionalities and algorithms from desktop computers.

Acknowledgments

This research has been funded by the PGDIT05SIN00101CT grant (*Xunta de Galicia*, Spain), and partially by FSE. We also appreciate the collaboration of the University of Wisconsin and University of Maryland, mainly from Professor Robert Meyer.

This work makes use of results produced by the Enabling Grids for E-sciencE project, a project co-funded by the European Commission (under contract number INFSO-RI-031688) through the Sixth Framework Programme. EGEE brings together 91 partners in 32 countries to provide a seamless GRID infrastructure available to the European research community 24 hours a day. Full information is available at http://www.eu-egee.org.

References

[1] T.R. Mackie, J.W. Scrimger, and J.J. Batista, "A convolution method of calculating do for 15-MV x rays", Med. Phys. 12, 188-196 (1985)

[2] P. Andreo, "Monte Carlo techniques in medical radiation physics", Phys. Med. Biol. 36, No 7, 861-920 (1991)

[3] U. Amaldi, G. Kraft, "Particle accelerators take up the fight against cancer", CERN Courier, Volume 46, No 10, pp. 17-20 (2006)

[4] G. López-Abente Ortega et.al, "La situación del cáncer en España". Ministerio de Sanidad y Consumo. 2005. ISBN: 84-7670-673-1

[5] A. Iglesias Lago, *Planificadores 3D y simulación virtual del tratamiento. Situación en España. Supervivencia asociada a su aplicación.* Santiago de Compostela: Servicio Galego de Saúde, Axencia de Avaliación de Tecnologías Sanitarias de Galicia, avalia-t; 2003. Serie Avaliación de Tecnoloxías. Investigación Avaliativa: IA2003/01

[6] Catálogo Nacional de Hospitales 2005, Ministerio de Sanidad y Consumo, http://www.msc.es/ciudadanos/prestaciones/centrosServiciosSNS/hospitales/home.htm

[7] A. Gómez, et. al., "Remote Radiotherapy Planning: The eIMRT Project", in Challenges and Opportunities of Health Grids, V. Hernández and others, eds., IOS Press, 2006

[8] Web Services, http://www.w3.org/2002/ws/

[9] The Apache Cocoon Project, http://cocoon.apache.org

[10] gLite, http://glite.web.cern.ch/glite/

[11] D. A. Low, W. B. Harms, S. Mutic, J. A. Purdy, "A technique for the quantitative evaluation of dose distributions", Med. Phys. 25, 656-661 (1998)

[12] D. W. O. Rogers, B. A. Faddegon, G. X. Ding, C.-M. Ma, J. We and T. R. Mackie, "BEAM: a Monte Carlo code to simulate radiotherapy treatment units", Med. Phys. 22, 503-524 (1995).

[13] DoseLab, http://doselab.sourceforge.net/index.html

[14] J. Deasy, A. Blanco and V. Clark, "CEER: a computational environment for radiotherapy research", Med Phys. 30, 979-985 (2004).

[15] http://www.interactive-Grid.eu/