

SOFTWARE FRAMEWORK QAClient FOR MEASUREMENT/AUTOMATION IN PROTON THERAPY CENTERS

A. Mayor, O. Actis, D. Meer, B. Rohrer, Paul Scherrer Institut, 5232 Villigen PSI, Switzerland

Abstract

Paul Scherrer Institute (PSI) operates a proton center for cancer treatments. For calibration measurements and quality assurance procedures which have to be executed on a frequent basis and involve different systems and software products, an in-house software framework (QAClient) was developed.

QAClient provides a configurable and extensible framework communicating with PSI control systems, measurement devices, databases and commercial products as LabVIEW and Matlab. It supports automation of test protocols with user interaction, data analysis and data storage as well as generating of reports.

INTRODUCTION

The Paul Scherrer Institute (PSI) was a pioneer in the field of proton therapy for cancer treatment by being the first center to implement spot scanning for dose delivery back in 1996 [1].

Different treatment areas as Gantry 2 (PSI in-house developed advanced scanning gantry [2]), Gantry 3 (a commercial gantry from Varian) and OPTIS2 (for ocular tumor treatments) are constantly being improved and extended for new irradiation techniques and have to be maintained permanently.

Especially for extensive and periodic quality assurance (QA [3]) test procedures but as well for long lasting calibration and measurement tasks there was an increasing need for an automation tool which could integrated all involved control systems, measurement devices and perform data analysis and storage.

It was therefore decided in 2011 to start the development of such a software tool. The tool had to be modular and extensible, suitable for a wide range of tasks and easy to use and adapt for different protocols and procedures without much programming efforts.

To be as platform independent as possible and to be able to integrate it easily into other PSI systems and environments, Java [4] was chosen as the programming language and XML [5] as configuration format. QAClient was developed as an extensible software framework and is still being extended and used in a variety of applications on daily basis.

Motivation

The complexity of all involved systems in proton therapy used for medical patient treatment (Fig. 1) requires extensive and periodic testing and QA. This testing has to be

efficient, reproducible, easy to execute (also by less specialized personnel) and has to produce QA documentation and publish data to QA databases.

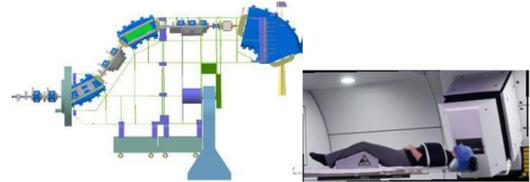


Figure 1: Beamline and treatment at Gantry 2.

Integrated Systems

In order to be able to automatize QA procedures QAClient has to integrate many different technical systems, actuators and detectors:

- Mechanical systems such as gantry, nozzle, patient couch and beam blockers
- Electromagnetic devices such as magnets, motors and valves
- Detectors such as ionization and strip chambers
- Control systems such as delivery system, verification system, couch robot and beam tune system
- Commercial systems such as Matlab [6], LabVIEW [7] and laser trackers

It has to be able to control these systems and to process data sent by these systems. Some measurement devices are controlled and read out via commercial tools such as LabVIEW which can be integrated in QAClient.

Data analysis and modelling tasks area often delegated to Matlab which also can be integrated in QAClient.

Requirements

To cover regulatory aspects of QA and to ensure that it can be executed reliably and efficiently also by non-specialized personnel the main requirements for QAClient were defined as follows:

- Full QA test procedure automation
- Integration of different systems and devices
- Interaction with human operators (data visualization)
- Integrated data analysis
- Comprehensible Reporting
- Long-term QA data storage and retrieval
- Easy to use
- Fully configurable (no programming required)

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2019). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

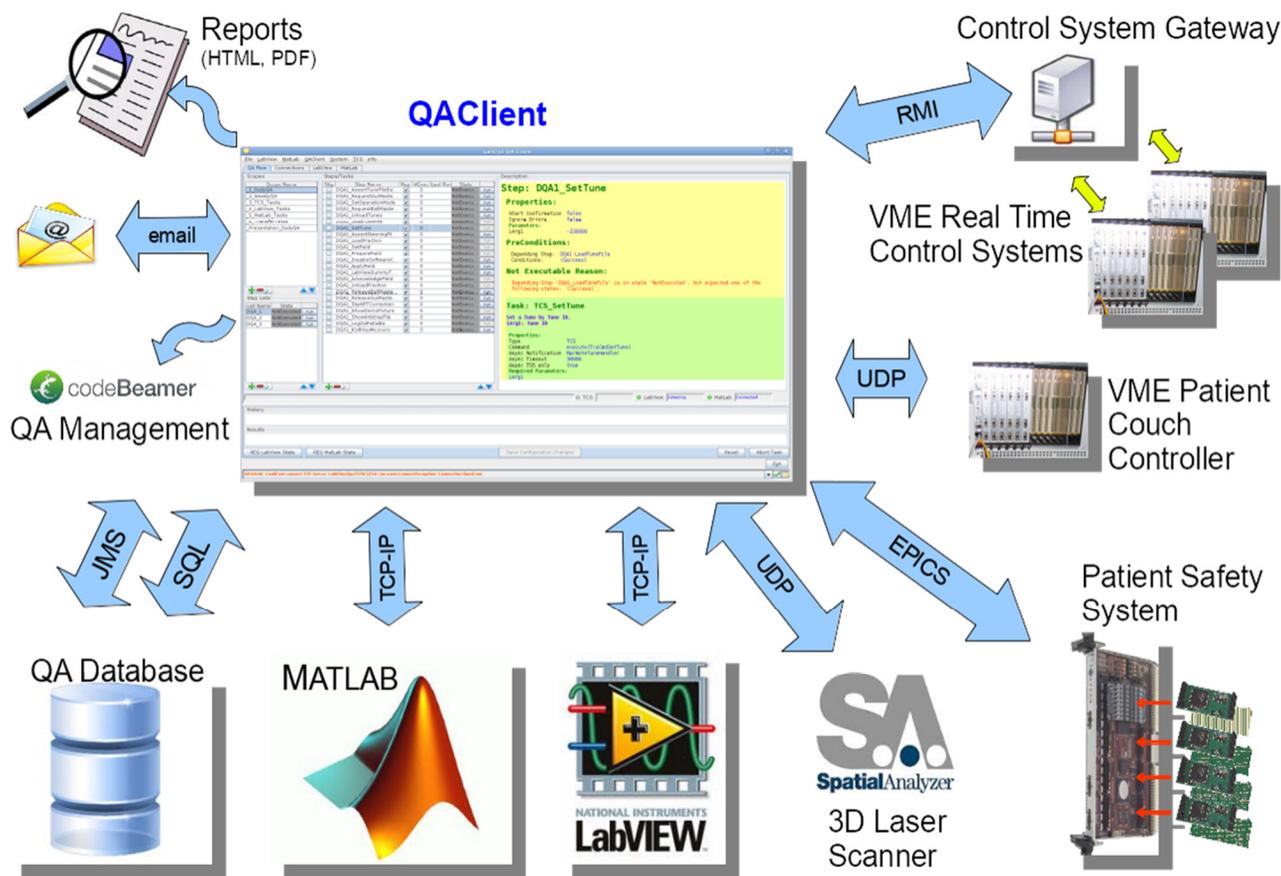


Figure 2: Overview of system integration and communication protocols supported by QAClient.

ARCHITECTURE AND REALISATION

Overview System Integration

Most systems which have to be integrated offer a well-defined API and communication protocols (Table 1). QAClient supports these standard protocols where necessary and possible (see Fig. 2). Configurable standard tasks hide the internals of the different protocols from the user and the engineer configuring a QAClient application.

Table 1: Protocols and Formats Used for Communication

System	Protocols / Format
Java Servers	RMI
Control Systems	RPC, UDP, EPICS
Matlab	TCP-IP (bi-directional)
LabVIEW	TCP-IP (bi-directional)
Laser Scanner	UDP
QA Database	JMS, SQL
codeBeamer	codeBeamer API
Reports	PDF, HTML, Email, text

Custom TCP-IP Protocols

For tools like MatLab and LabVIEW simple and reliable (TCP-IP) protocols have been implemented.

A small Java service has been integrated into MatLab which receives commands on one port and which sends status notifications and data on a second port. Commands are generic and consist of a MatLab script name and a variable list of parameters. Further API commands allow the program or the user to abort executing commands, to request status information or processed data and to shut down the service. Matlab Scripts can send messages, notifications, status information, data (as XML) and files via custom commands to QAClient.

For LabVIEW a very similar bi-directional protocol has been implemented in a specific VIServer. It is possible to send calls and parameters to VITasks. LabVIEW applications can as well send messages, notifications, status information, data (as XML) and files to QAClient.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2019). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

Technologies

To meet all functional and technical requirements as well as to be able to integrate it into the existing infrastructure QAClient as a framework uses a number of different technologies:

- Swing / AWT Graphical User Interface
- JMS (Java Message Service, ActiveMQ)
- RMI (Remote Method Invocation) to Gateways
- EPICS (Experimental Physics and Industrial Control System) to Patient Safety and Accelerator Control Systems
- HTTP Service
- HTML and PDF Reports
- SMTP (Simple Mail Transfer Protocol)
- different proprietary TPC-IP / UDP protocols
- OJDBC database access
- XML configuration

As PSI uses different operating systems (Windows 7, 10, Linux, etc.) for different purposes and to reduce portability efforts QAClient is programmed in Java. The only requirement on the platform it runs on is that it provides at least an OpenJDK 8 Java runtime environment.

Features

QAClient offers the following features:

- Graphical user interface (with dynamic “expert”/“simple” mode and optional color textures to help color-blind operators)

- XML based application configuration using predefined standard tasks
- Generic mechanism of data transfer between different tasks
- Dynamic flow control (conditions based on data or user decision)
- Programming loops
- Input file driven measurement loops
- Data checks and validations
- Sending Commands to different Control Systems, LabVIEW, Matlab
- Processing Notifications from Control Systems, LabVIEW, Matlab
- Logging framework with adjustable severity level
- LDAP user identification and authorization
- Persistent states to continue aborted tasks
- Generic dialogs to show/visualize data and ask for user decisions
- Archiving of QA data and measurement results in external database
- Generating and printing of summary reports (HTML, PDF)
- Visualization of archived QA data

Application Configuration

All QAClient applications run on the same program version and are defined by pure XML configuration, there is no need for application specific QAClient versions or programming. The XML uses a strict declaration schema (XSD) which defines the standard blocks and their relationship and which validates the configuration XML.

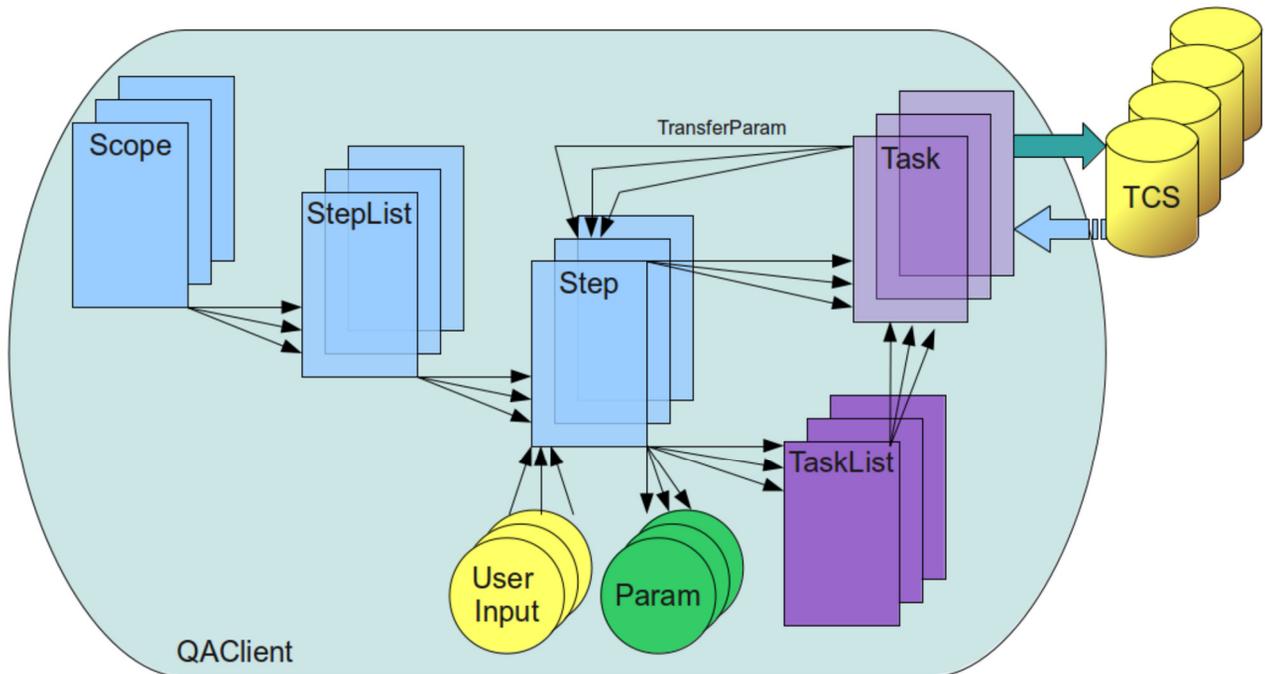


Figure 3: Application configuration entities and their relationships.

As depicted in Fig. 3, the main entities listed in Table 2 can be used in a QAClient application configuration:

Table 2: XML Configuration Entities (not complete)

Entity	Usage
Scope	Full QAClient Application (measurement, test etc.)
StepList	Full measurement or test procedure.
Step	Smallest execution entity with all its parameters, input data and conditions.
Task	Predefined standard task with a set of required input parameters
TaskList	Optional grouping entity to combine a set of tasks into a more global and re-usable task list
UserInput	Generic user dialog to present data, files and decisions to the operator
Param	Configured input values for standard tasks
Transfer-Param	Data transferred produced by one step/task and used by another one as input data.

QAClient also provides a graphical configuration tool to assemble such step lists, however it does not support all features and possible relationships and flow control mechanisms.

Standard Tasks

QAClient currently provides 228 predefined standard tasks which can be used with different parameters (see Table 3).

Table 3: Standard Task Examples

Task Type	Examples
Operating System	OpenFile, ExecuteSystemCommand, ExecuteSSHCommand...
Control System	StartupBeamline, MoveToPosition, StartAutomaticSequence, SetTune...
QAClient	ForEachFile, CreateData, ReceiveUDP...
LabVIEW	DataRequest, StartVITask...
Matlab	StatusRequest, ML_Command...
Database	SendData, ShowDBData...

Graphical User Interface

The graphical user interface supports two display modes which can be switched dynamically:

- **“Simple Mode”**: reduced amount of information just showing the StepLists / Steps and their execution state (Fig. 4).
- **“Expert Mode”**: offers full control and detailed information on every step and data received and sent (Fig. 5).

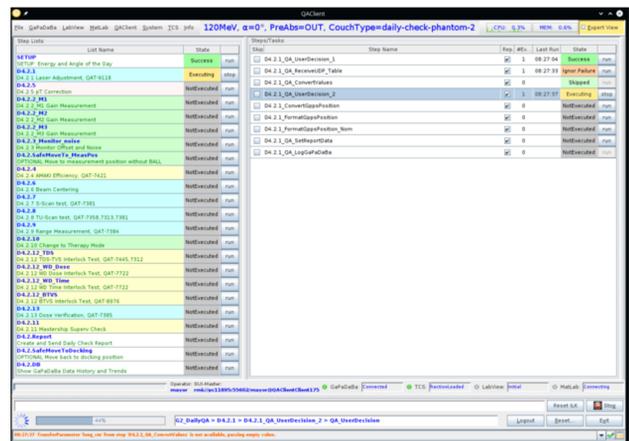


Figure 4: QAClient user interface in “simple mode”.

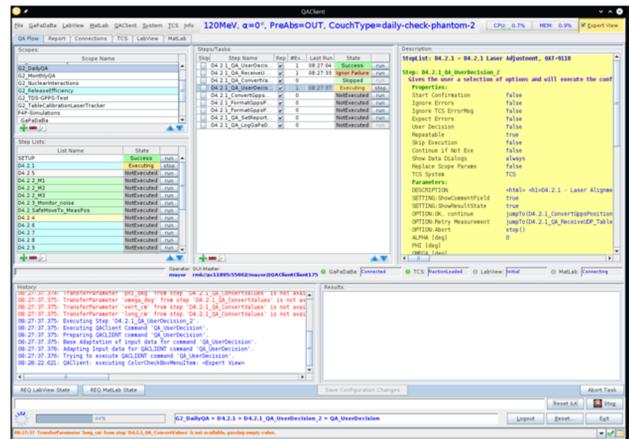


Figure 5: QAClient user interface in “expert mode”.

Some tasks offer displaying generic dialogs with configurable content or content received from other tasks, as shown in Fig. 6. These dialogs can be used to display measured data, plot analysis results, show external files and to prompt the operator for a decision like “OK, continue” or “Retry” or “Abort”. Depending on the user action taken, the workflow can be different.

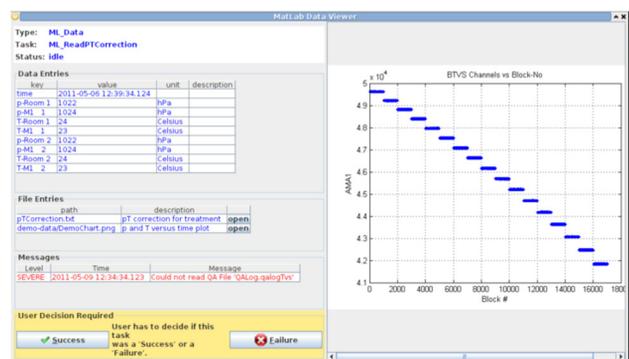


Figure 6: Generic user dialog.

Reporting and Data Archiving

Data can be saved in external files or sent to a database. QAClient can also produce comprehensive PDF and HTML reports (Fig. 7) which can be archived as files or in contentBeamer, sent by email and printed.

Content from this work may be used under the terms of the CC BY 3.0 licence (© 2019). Any distribution of this work must maintain attribution to the author(s), title of the work, publisher, and DOI.

G2 Daily QA Report

Area	G2	Responsible	Date	Signature
Scope	G2_DailyQA	Physicist		
Operator	heller (Martin Heller, MA17, martin.heller@psi.ch, +41 56 310 53 23, M080/0131)	Medical Physicist		
Start	2019-09-13 06:40:53			
End	2019-09-13 07:15:06			
Duration	34 min, 12 sec			
ExecTime	21 min, 21 sec			

SETUP Pre-Absorber	OUT
AcceptIncompletescopes	true
SETUP Energy (MeV)	120
SETUP Alpha (deg)	90

Step List ID	Status	Data	Operator Comments
D4.2.1 Laser Adjustment, QAT-9110	Success	Tolerance: Qat1276.06, +/- 0.01, y121.06, +/- 0.01 3 Tolerance: Qat1280.00, +/- 0.02, y121.06, +/- 0.01 3	
D4.2.5 pt Correction	Success	Temp: 14.97mpm22.7, Phantom22.1 (C) (99) Air Pressure: 800.0 (hPa) pt correction: 1.8026	
D4.2.2_M1 Gain Measurement	Success	Slope: (Measured): 0.0, Nominal: 0.0, residual(Nom): 0.000 Tolerance: (Tolerance): 0, Deviation: 0.183 (%)	
D4.2.2_M2 Gain Measurement	Success	Slope: (Measured): 2.0, Nominal: 2.0, residual(Nom): 0.023 Tolerance: (Tolerance): 0, Deviation: 0.183 (%)	
D4.2.2_M3 Gain Measurement	Success	Slope: (Measured): 0.07, Nominal: 0.07, residual(Nom): 0.000 Tolerance: (Tolerance): 0, Deviation: 0.060 (%)	
D4.2.3 Monitor Offset and Noise	Success	M1_Msc offset: 0.0, Limit(Nom): 0, Limit(Nom): 0.0 (mV) M2_Msc offset: 0.0, Limit(Nom): 0, Limit(Nom): 0.0 (mV) M3_Msc offset: 0.0, Limit(Nom): 0, Limit(Nom): 0.0 (mV) M4_Msc offset: 0.0, Limit(Nom): 0, Limit(Nom): 0.0 (mV)	
D4.2.4 AMMI Efficiency, QAT-7421	Success	AMMI (Measured): 0.00007, Limit(Nom): 0.0003	
D4.2.6 Beam Centering	Success	Offset: (Measured): 0.11, tolerance: 0.11, tolerance: 0.203 (cm)	
D4.2.7 S-Scan test, QAT-7381	Success	Position: (residual): 0.12, tolerance: 0.203 (cm) Beam Size: (residual): 3.74, tolerance: 0.80 (%)	
D4.2.8 TU-Scan test, QAT-7358, 7313, 7381	Success	Position: (residual): 0.14, tolerance: 0.203 (cm) Beam Size: (residual): 3.26, tolerance: 0.80 (%)	
D4.2.9 Range Measurement, QAT-7384	Success	Range: (residual): 1.20, tolerance: 0.80 (cm) Energy: (residual): 0.99, tolerance: 0.80 (MeV)	
D4.2.10 Change to Therapy Mode	Success		
D4.2.12 TDS-TVS Interlock Test, QAT-7445, 7312	Success		
D4.2.12 MD Dose Interlock Test, QAT-7722	Success		
D4.2.12 MD Time Interlock Test, QAT-7722	Success		
D4.2.12 BTVS Interlock Test, QAT-8976	Success		
D4.2.13 Dose Verification, QAT-7385	Success	Chamber1 (Calculated): 0.000, nominal: 0.000, deviation: 0.0143 (cp) Chamber2 (Measured): 0.002, nominal: 0.000	
D4.2.11 Mastership Superv Check	Success		

✔ without issue
 ⚠ minor issue(s)
 ⚠ major issue(s)
 ✖ didn't pass
 ⏸ not executed or aborted

Figure 7: QAClient standard report.

OUTCOMES AND USAGE

General Usage

QAClient today is used for a wide range of applications and use cases, these are among others:

- Daily check (automated QA) (Fig. 8)
- Monthly checks (automated QA)
- Patient couch calibration measurements
- Beam tune calibration measurements and modelling
- System integration tests (as test-engine)

Example: Daily Check Gantry 2

One of the main applications of QAClient is the daily check procedure of Gantry 2 [8], which is performed every morning prior to patient treatment. The most important parameters which are checked by the daily check:

- Dose accuracy using ionization chambers
- Proton beam position accuracy
- Proton beam parallelism and position
- Proton beam energy / range in water
- Beam kicker magnet closing efficiency
- Base check on patient safety system (beam on/off, beam interruptions, beam tune)

To achieve this the following equipment (among other) is used:

- Mini-strip detector
- Multi-layer ionization chamber
- Ionization chambers at different depths

The produced output consists of:

- Measurement and analysis reports
- Printed QA document for paperwork
- Long-term storage of QA data
- Sending of data and files to QA archive
- Sending of QA document via email



Figure 8: Daily check phantom with different detectors.

SUMMARY

QAClient provides a flexible software framework which is fully configurable and easily extensible. It offers a wide range of reliable and standardized procedures and offers interfaces to different systems. It ensures that QA checks are executed in a reproducible way and helps to reduce errors. Execution times of QA procedures like the daily check of Gantry 2 could be reduced by 70% and can now also be executed by less specialized personnel without deep knowledge of the involved systems and devices.

REFERENCES

- [1] E. Pedroni *et al.*, “The 200-MeV proton therapy project at Paul Scherrer Institute: Conceptual design and practical realization”, *Med. Phys.*, vol. 22, no. 1, pp. 37-53, 1995. doi:10.1118/1.597522
- [2] E. Pedroni *et al.*, “The PSI Gantry 2: a second generation proton scanning gantry”, *Z. Med. Phys.*, vol. 14, no. 1, pp. 25-34, 2004. doi:10.1078/0939-3889-00194
- [3] B. Arjomandy *et al.*, “An overview of the comprehensive proton therapy machine quality assurance procedures implemented at The University of Texas M. D. Anderson Cancer Center Proton Therapy Center-Houston”, *Med. Phys.*, vol. 36 no. 6 Part1, pp. 2269-2282, 2009. doi:10.1118/1.3120288
- [4] Java Programming Language, <https://www.java.com>
- [5] XML: Extensible Markup Language, <https://www.w3.org/XML>
- [6] Matlab: Data Analysis, <https://mathworks.com/matlab>
- [7] LabVIEW: Laboratory Virtual Instrumentation Engineering Workbench, <https://www.ni.com/labview>
- [8] O. Actis, D. Meer, S. König, D. C. Weber, and A. Mayor, “A comprehensive and efficient daily quality assurance for PBS proton therapy”, *Phys. Med. Biol.*, vol. 62, no. 5, pp. 1661-1675, 2017. doi:10.1088/1361-6560/aa5131