NATIONAL TECHNICAL UNIVERSITY OF ATHENS

School of Electrical and Computer Engineering

Division of Information Transmission Systems

and Material Technology

Control System and Data Acquisition for the Remote Monitoring of the personnel in the ATLAS experimental infrastructure at CERN

A thesis presented by

Eleni S. Adamidi

for the degree of

Doctor of Philosophy, PhD

July 2019

NATIONAL TECHNICAL UNIVERSITY OF ATHENS

School of Electrical and Computer Engineering

Division of Information Transmission Systems

and Material Technology

Control System and Data Acquisition for the Remote Monitoring of the personnel in the ATLAS experimental infrastructure at CERN

A thesis presented by

Eleni S. Adamidi

for the degree of

Doctor of Philosophy, PhD

July 2019

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ

ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

Σύστημα Ελέγχου και Συλλογής Δεδομένων για την απομακρυσμένη παρακολούθηση του προσωπικού στο πειραματικό περιβάλλον του ATLAS στο CERN

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

της

ΕΛΕΝΗΣ Σ. ΑΔΑΜΙΔΗ

ΑΘΗΝΑ

Ιούλιος 2019

NATIONAL TECHNICAL UNIVERSITY OF ATHENS

School of Electrical and Computer Engineering

Division of Information Transmission Systems

and Material Technology

PhD THESIS

Eleni S. Adamidi

Supervisory Board: Prof. Konstantina Nikita (Supervisor)

Prof. Andreas Georgios Stafylopatis

Prof. Evangelos Gazis

Exam committee:

…………..	…………..	…………..
K. Nikita	A. G. Stafylopatis	E. Gazis
Professor NTUA	Professor NTUA	Professor NTUA

…………..	…………..	…………..	…………..
G. Stamou	S. Maltezos	T. Alexopoulos	M. Kokkoris
Professor NTUA	Professor NTUA	Professor NTUA	Professor NTUA

July 2019

……………………

Adamidi S. Eleni

Διδάκτωρ Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Abstract

Personnel safety, supervision and real-time monitoring are important key parameters while performing activities in risky environments and extreme environmental conditions such as the ATLAS cavern at CERN. The hazardous environments are not user friendly in terms of frequent access, performing regular or sudden interventions, monitoring, and supervision activities. The engineers and the personnel need to perform complex activities like installation and maintenance work in the heavy machinery. In most of the cases, it is not possible to memorize the maintenance procedures of a complex machinery. If the machines contain radioactive elements, then the situation is even more crucial for the health and safety of the personnel.

The objective of this research is to develop and optimize the Control System (CS) and Data Acquisition (DAQ) System for the Personnel Safety System used in the ATLAS cavern. We propose a monitoring system to supervise the health status of the personnel and provide guidance during the performance of complex activities inside the ATLAS cavern. Radiation background monitoring can also be achieved through the communication of the system with a gamma camera placed in the cavern. This system is developed to supervise multiple interventions in real-time.

An adaptability and scalability research, from a Service Based Application (SBA) perspective, has also been conducted to examine the use of this system in other extreme environmental conditions.

Keywords: Data Acquisition System (DAQ), Control System (CS), Safety Systems, and Radiation Protection

Περίληψη

Η ασφάλεια του προσωπικού, η επίβλεψη και η παρακολούθηση των εργασιών σε πραγματικό χρόνο, αποτελούν σημαντικές βασικές παραμέτρους κατά την εκτέλεση δραστηριοτήτων σε επικίνδυνα περιβάλλοντα και ακραίες περιβαλλοντικές συνθήκες, όπως το περιβάλλον του πειράματος ATLAS στο CERN. Τα επικίνδυνα περιβάλλοντα δεν είναι φιλικά προς το χρήστη όσον αφορά τη συχνή πρόσβαση, την εκτέλεση τακτικών ή μη προγραμματισμένων παρεμβάσεων, την παρακολούθηση και την εποπτεία. Οι μηχανικοί και το προσωπικό πρέπει να εκτελούν πολύπλοκες δραστηριότητες όπως εργασίες εγκατάστασης και συντήρησης σε βαριά μηχανήματα. Στις περισσότερες περιπτώσεις, δεν είναι δυνατή η απομνημόνευση των διαδικασιών συντήρησης ενός σύνθετου μηχανήματος. Εάν τα μηχανήματα περιέχουν ραδιενεργά στοιχεία, τότε η κατάσταση είναι ακόμα πιο κρίσιμη για την υγεία και την ασφάλεια του προσωπικού.

Στόχος της παρούσας έρευνας είναι η ανάπτυξη και βελτιστοποίηση του Συστήματος Ελέγχου (CS) και του Συστήματος Εξαγωγής Δεδομένων (DAQ) για το Σύστημα Ασφαλείας Προσωπικού που χρησιμοποιείται στο σπήλαιο της ATLAS. Προτείνουμε ένα σύστημα παρακολούθησης για την επίβλεψη της κατάστασης υγείας του προσωπικού και την παροχή καθοδήγησης κατά την εκτέλεση σύνθετων δραστηριοτήτων στο εσωτερικό του περιβάλλοντος του πειράματος ATLAS. Παρακολούθηση της ακτινοβολίας υποβάθρου μπορεί επίσης να επιτευχθεί μέσω της επικοινωνίας του συστήματος με μια γάμμα κάμερα τοποθετημένη στο σπήλαιο. Το σύστημα αυτό αναπτύσσεται για να εποπτεύει πολλαπλές παρεμβάσεις σε πραγματικό χρόνο.

Έχει επίσης διεξαχθεί μια έρευνα προσαρμοστικότητας, από την οπτική εφαρμογών υπηρεσιών (SBA), προκειμένου να εξεταστεί η χρήση αυτού του συστήματος σε άλλες ακραίες περιβαλλοντικές συνθήκες.

Λέξεις κλειδιά: Σύστημα Συλλογής Δεδομένων, Σύστημα Ελέγχου, Συστήματα Ασφαλείας, Ακτινοπροστασία

Εκτεταμένη Περίληψη

Η παρούσα διδακτορική διατριβή εστιάζεται στην ανάπτυξη και βελτιστοποίηση του Συστήματος Ελέγχου και Συλλογής Δεδομένων για την απομακρυσμένη παρακολούθηση του προσωπικού στο περιβάλλον του πειράματος ATLAS στο CERN με κύριο στόχο την ενίσχυση της ασφάλειας και της ακτινοπροστασίας των εργαζομένων. Η επίβλεψη, η επικοινωνία και η καθοδήγηση του προσωπικού σε πραγματικό χρόνο αποτελεί στόχο της έρευνας. Η έρευνα της διδακτορικής διατριβής χρηματοδοτήθηκε από την υποτροφία Marie Curie στα πλαίσια του προγράμματος EDUSAFE (Education in advanced Augmented Reality (AR) Safety Systems for Maintenance in Extreme Environments). Η διατριβή χωρίζεται σε πέντε μέρη. Στο πρώτο μέρος παρουσιάζεται η μελέτη της ακτινοβολίας υποβάθρου στο σύνθετο πειραματικό χώρο του ανιχνευτή ATLAS. Η κατανόηση της πολυπλοκότητας και της επικινδυνότητας του πειραματικού χώρου καθιστά αναγκαία την ανάπτυξη ενός συστήματος ελέγχου που έχει ως κύριο στόχο την ακτινοπροστασία του προσωπικού.

Στο δεύτερο μέρος περιλαμβάνεται η ανάπτυξη και βελτιστοποίηση του Συστήματος Ελέγχου (Control System, CS) για την ασφάλεια του προσωπικού που απασχολείται στον υπόγειο πειραματικό χώρο του ανιχνευτή ATLAS με κύριο στόχο την αποφυγή ραδιενεργών σημείων του πειραματικού χώρου. Οι αλγόριθμοι που αναπτύχθηκαν δέχονται ως είσοδο μετρήσεις από βιολογικούς και περιβαλλοντικούς αισθητήρες. Το σύστημα που αναπτύχθηκε στο πλαίσιο της διατριβής δοκιμάστηκε σε μια περιοχή του υπόγειου πειραματικού χώρου (USA15), στην οποία η ακτινοπροστασία είναι ιδιαίτερα κρίσιμη, καθώς αποτελεί μία από τις τέσσερις συνολικά περιοχές στις οποίες επιτρέπεται η πρόσβαση ακόμα και σε περιόδους λειτουργίας του επιταχυντή.

Το τρίτο μέρος περιλαμβάνει την ανάπτυξη και βελτιστοποίηση του Συστήματος Συλλογής Δεδομένων (Data Acquisition, DAQ) για την Ασφάλεια του Προσωπικού, το οποίο χρησιμοποιείται στον υπόγειο πειραματικό χώρο του ανιχνευτή ATLAS. Η συλλογή δεδομένων πραγματοποιείται ασύρματα σε μορφή μηνυμάτων JSON (Java

Script Object Notation) από διάφορες πηγές εξοπλισμού βιολογικών και περιβαλλοντικών μετρήσεων καθώς και των επιπέδων ραδιενεργού υποβάθρου για την ασφάλεια και παρακολούθηση της υγείας των εργαζομένων. Το σύστημα DAQ αναπτύχθηκε με δυνατότητα προσθήκης νέων αισθητήρων σε διάφορες ακραίες περιβαλλοντικές συνθήκες. Αυτή η δυνατότητα επιτρέπει την προσαρμοστικότητα του συστήματος σε διάφορες περιπτώσεις όπου η ανθρώπινη ασφάλεια είναι κρίσιμη.

Το τέταρτο μέρος της διατριβής παρουσιάζει την σχεδίαση και την ανάπτυξη του συστήματος απομακρυσμένης παρακολούθησης (Remote Monitoring System, RMS) για την επίβλεψη της κατάστασης υγείας του προσωπικού και την παροχή καθοδήγησης κατά την εκτέλεση σύνθετων δραστηριοτήτων στο εσωτερικό του περιβάλλοντος του πειράματος ATLAS με χρήση τεχνολογιών Επαυξημένης Πραγματικότητας (Augmented Reality, AR). Σχεδιάστηκε ένα ολοκληρωμένο περιβάλλον χρήστη (EDUSS Supervision User Interface) για την απομακρυσμένη παρακολούθηση της υγείας του προσωπικού με επικοινωνία με εικόνα και ήχο σε πραγματικό χρόνο. Το σύστημα προσφέρει εξατομικευμένη και συλλογική επίβλεψη δεδομένων και εργασιών με δυναμικά γραφήματα. Μέσω του συστήματος αυτού επιτυγχάνεται η καθοδήγηση των εργαζομένων σε πραγματικό χρόνο με βασικό στόχο την αποφυγή των ραδιενεργών σημείων του χώρου και κατά συνέπεια την μείωση του εργασιακού άγχους.

Τέλος το πέμπτο μέρος της διατριβής παρουσιάζει την έρευνα προσαρμοστικότητας όλων των υποσυστημάτων (Σύστημα Ελέγχου, Σύστημα Συλλογής Δεδομένων και Σύστημα Απομακρυσμένης Παρακολούθησης) σε άλλες ακραίες περιβαλλοντικές συνθήκες. Η διατριβή ολοκληρώνεται στο Κεφάλαιο 8 με παράθεση γενικών συμπερασμάτων και συζήτηση για τις κατευθύνσεις μελλοντικής έρευνας.

Λέξεις κλειδιά: Σύστημα Συλλογής Δεδομένων, Σύστημα Ελέγχου, Συστήματα Ασφαλείας, Ακτινοπροστασία

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά όλους όσους συνέβαλαν με οποιονδήποτε τρόπο, από τον Νοέμβριο του 2014 έως και σήμερα, στην εκπόνηση και ολοκλήρωση της παρούσας Διδακτορικής Διατριβής.

Πρωτίστως θα ήθελα να ευχαριστήσω την Καθηγήτρια κ. Κωνσταντίνα Νικήτα, Επιβλέπουσα της Διδακτορικής μου Διατριβής, για την εμπιστοσύνη που μου έδειξε, την καθοδήγηση και την υποστήριξη που μου παρείχε και για την πολύτιμη συνεργασία που είχαμε όλα αυτά τα χρόνια. Αποτελεί πάντα πρότυπο για μένα η εντυπωσιακή ερευνητική της πορεία. Ευχαριστώ επίσης θερμά τον Καθηγητή κ. Ανδρέα Γεώργιο Σταφυλοπάτη, μέλος της τριμελούς επιτροπής της Διατριβής μου.

Θα ήθελα να ευχαριστήσω ολόψυχα τον Καθηγητή μου κ. Ευάγγελο Γαζή, μέλος της τριμελούς επιτροπής, αλλά και Καθηγητή κατά τις προπτυχιακές μου σπουδές στη Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών για τη συνεχή καθοδήγηση του και τις πολύτιμες συμβουλές του σε κάθε στάδιο της παρούσας Διδακτορικής Διατριβής και όχι μόνο. Η εμπιστοσύνη που μου έδειξε όλα αυτά τα χρόνια με τιμά ιδιαίτερα και αποτελεί μια πολύτιμη παρακαταθήκη για το μέλλον.

Ευχαριστώ όλους τους συνεργάτες και φίλους μου στο CERN που ενίσχυσαν την πεποίθηση μου ότι η έρευνα δεν είναι μια μοναχική πορεία και ότι μέσω της διαδραστικότητας των ερευνητών μπορούν να παραχθούν εντυπωσιακά αποτελέσματα.

Θα ήθελα να εκφράσω ένα μεγάλο ευχαριστώ στον συνεργάτη και φίλο μου Alexandre Alves Dos Santos για τις ατελείωτες ώρες δουλειάς και δημιουργικών συζητήσεων πάνω στο ερευνητικό αντικείμενο της Διατριβής καθώς και για την ανεκτίμητη συμπαράστασή του σε όλη αυτή τη πορεία.

Ευχαριστώ επίσης τους συνεργάτες μου Mark, Πάνο, Χρήστο και τον φίλο μου Krzysztof για τη συνεργασία και τη πολύτιμη βοήθειά τους σε όλη τη διάρκεια της παραμονής μου στο CERN.

Ευχαριστώ τους φίλους μου Θεοφάνη και Μαρίνα για τα πολύ σημαντικά διαλείμματα που μου προσέφεραν.

Το μεγαλύτερο ευχαριστώ ανήκει στην οικογένεια μου, τη μητέρα μου, Θεώνη, τον πατέρα μου, Σωτήρη και τα τρία αγαπημένα μου αδέρφια, Σοφία, Κωνσταντίνο και Νίκο, που είναι πάντα δίπλα μου και με στηρίζουν.

Ένα μεγάλο επίσης ευχαριστώ στη Βάσω μου, που χωρίς εκείνη όλα θα ήταν πολύ πιο δύσκολα. Ευχαριστώ τον Γιώργο και τη Λέττα για όλες τις φορές που με βοήθησαν χωρίς καν να το σκεφτούν.

Η Διατριβή μου είναι αφιερωμένη στον άντρα μου, Σπύρο και στην κορούλα μου, Μελίνα που με κάνουν τόσο ευτυχισμένη.

Acknowledgments

The author wishes to thank all other members of the EDUSAFE consortium. The research of EDUSAFE has been supported by a Marie Curie Initial Training Network Fellowship of the European Community's FP7 Program under contract number PITN-GA-2012-316919.

Στον άντρα μου Σπύρο και στην κορούλα μου Μελίνα.

Contents

List of Figures

Chapter 1

1 The ATLAS detector at the Large Hadron Collider

1.1 The Large Hadron Collider

The fundamental structure of the universe is studied by physicists and engineers at the largest particle physics laboratory in the world, the European Organization for Nuclear Research, CERN [1]. Particle collisions close to the speed of light produce results that guide research on particle interaction and the fundamental laws of nature. The instruments used at CERN are purpose-built particle accelerators, which boost beams of particles to high energies before the beams are made to collide with each other or with stationary targets, as well as detectors, which observe and record the results of these collisions. Founded in 1954, the CERN laboratory sits astride the Franco-Swiss border near Geneva. It was one of Europe's first joint ventures and now has 22-member states [1].

The Large Hadron Collider (LHC) is the world's largest and most powerful particle accelerator, colliding proton beams, that started operating successfully in 2009 at CERN, in order to extend the frontiers of particle physics with an unprecedented high energy and luminosity [2] [3]. The LHC is a two-ring-superconducting-hadron accelerator and collider installed in the existing 26.7 km tunnel that was constructed between 1984 and 1989 for CERN's previous accelerator, the Large Electron-Positron collider (LEP) machine, and it is located in the borders between France and Switzerland near Geneva [4]. There are seven LHC experiments using detectors to analyze the myriad of particles produced by collisions in the accelerator.

Proton-proton collisions of 14 TeV at a design luminosity of 10^{34} $cm^{-2}s^{-1}$, will be provided from up to 10^{11} protons (p) colliding 40 million times per second inside the LHC [2]. The proton beams are split in bunches of oblate volume and are successively accelerated by four accelerators, the LINAC2 (50 Mev), the BOOSTER (1.4 GeV), the PS (26 GeV) and the SPS (450 GeV), to be later on injected in the LHC superconducting hadron accelerator ring, with an energy of E_p=7TeV for each proton in a bunch (see *Figure 1*) [5]. Heavy ions (A) collisions will also take place inside the collider, in particular lead nuclei, at 5.5 TeV per nucleon pair, at a design luminosity of 10^{27} cm^{-2} s^{-1} [2]. Those A-A collisions along with the p-p collisions are probed from two general purpose detectors, **ATLAS** (**A** **T**oroidal **L**arge Hadron Collider **A**pparatu**S**) and **CMS** (**C**ompact **M**uon **S**olenoid) [2]. These experiments are the biggest out of the seven currently running at the LHC. The primary aim of the ATLAS and CMS experiments is to discover the Higgs boson and to investigate the existence of new particles predicted in theories beyond the Standard Model (SM) such as Supersymmetry [6].

There are also two low luminosity experiments: **LHCb** for B-physics, aiming at a peak luminosity of L = 10^{32} $cm^{-2}s^{-1}$, and **TOTEM** for the detection of protons from elastic scattering at small angles, aiming at a peak luminosity of L = 2x 10^{29} $cm^{-2}s^{-1}$ with 156 bunches. The LHCb specializes in investigating the slight differences between matter and antimatter by studying a type of particle called the “beauty quark”, or “b quark” [7]. The 5600-tonne LHCb detector is made up of a forward spectrometer and planar detector, it is 21 meters long, 10 meters high and 13 meters wide [7]. About 700 scientists from 66 different institutes and universities make up the LHCb collaboration [7]. The TOTEM, or 'Total, elastic and diffractive cross-section measurement' experiment, studies particles thrust forward by collisions in the LHC emerging at small angles. The TOTEM experiment involves about 100 scientists from 16 institutes in 8 countries [8]. Along with TOTEM, **LHCf** (Large Hadron Collider forward) is the second smaller experiment in LHC. The LHCf experiment involves 30 scientists from 9 institutes in 5 countries. The LHCf experiment uses particles thrown forward by collisions in the Large Hadron Collider as a source to simulate cosmic rays in laboratory conditions [9].

In addition to the proton beams, the LHC also operates with ion beams. The LHC has one dedicated heavy-ion experiment, **ALICE** (A Large Ion Collider Experiment), aiming at a peak luminosity of L = 10^{27} $cm^{-2}s^{-1}$ for nominal lead-lead ion operation [4]. It is designed to study the physics of strongly interacting matter at extreme energy densities, where a phase of matter called quark-gluon plasma forms. The ALICE detector is 26 m long, 16 m high, and 16 m wide sitting in a vast cavern 56 m below ground. The collaboration counts more than 1000 scientists from over 100 physics institutes in 30 countries [10].

Additionally, in 2010 the LHC collaboration approved its seventh experiment: The Monopole and Exotics Detector at the LHC (**MOEDAL**). The prime motivation of MOEDAL is to search directly for the magnetic monopole – a hypothetical particle with a magnetic charge. The MOEDAL is also looking for highly ionizing Stable Massive Particles (SMPs), predicted by theories beyond the Standard Model [11].

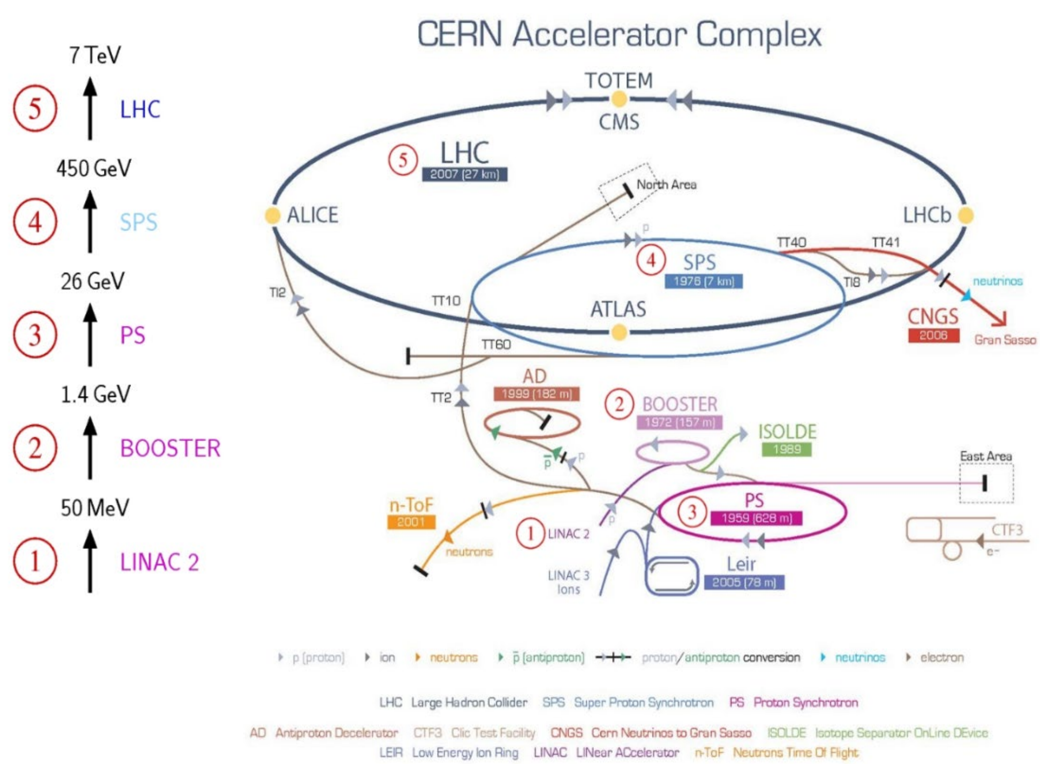

Figure 1. Overview of the CERN Accelerator Complex.

1.2 The ATLAS detector at the LHC

ATLAS is a general-purpose High-Energy Physics (HEP) experiment at the LHC at CERN with more than 3000 scientists of 174 institutions in 38 countries collaborating. It is a particle physics apparatus designed to detect significant physics processes and new particles predicted in theories beyond the SM, by measuring types, orbits and energies of the particles produced during collisions. A major discovery took place in 2012 with the detection of the Higgs Boson, a missing piece of the Standard Model detected at $\sqrt{s} = 8TeV$, with a mass m_H around 125 GeV, or 133 times the proton mass [12]. The ATLAS detector is a complicated infrastructure distributed over a cylindrical volume of 25m diameter and 46m length, located at Point 1,100 meters below the surface [13] [6]. It is the largest detector ever built with an overall weight of approximately 7000 tons [2]. The experimental infrastructure of the detector has approximately 100 million electronic channels and 3000km of cables [14]. A Cut-away view of the ATLAS detector is shown in Figure 2 [3]. It is consisted of four major components, the Inner detector, the Calorimeters, the Muon spectrometer and the magnet systems.

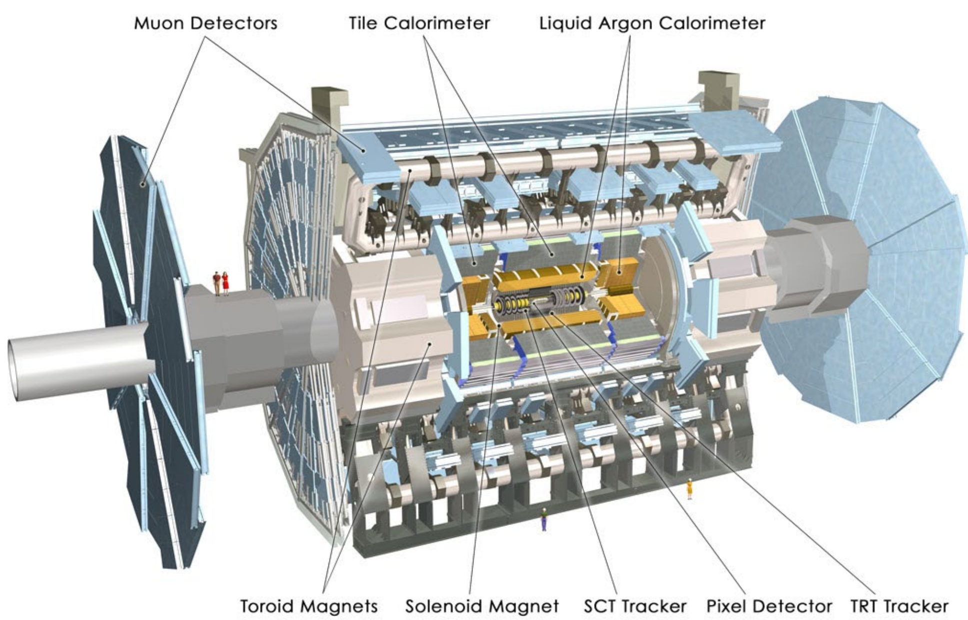

Figure 2. The ATLAS detector layout.

The ATLAS Inner Detector (ID) is designed to provide hermetic and robust pattern recognition, excellent momentum resolution and both primary and secondary vertex measurements for charged tracks above a given p_T threshold (nominally 0.5 GeV, but as low as 0.1 GeV in some ongoing studies of initial measurements with minimum-bias events) and within the pseudorapidity range $|\eta| < 2.5$ [2]. Additionally, it provides electron identification over $|h| < 2.0$ and a wide range of energies between 0.5 GeV and 150 GeV [2]. The ID is constructed on a cylindrical form of length ±3512mm and of radius 1150 mm, within a solenoidal magnetic field of 2 T [2]. The high-radiation environment imposes stringent conditions on the inner-detector sensors, on-detector electronics, mechanical structure and services [2]. Over the ten-year design lifetime of the experiment, the pixel inner vertexing layer must be replaced after approximately three years of operation at design luminosity [2].

The calorimeters measure the energy of charged and neutral particles and are situated around the thin superconducting solenoidal magnet of the Inner Detector [2]. There are two basic calorimeters covering the pseudorapidity range of $|\eta|<4.9$, the High granularity liquid-argon (LAr) electromagnetic sampling calorimeters at the range of $|\eta|<3.2$, and the hadronic calorimetry in the range of $|\eta|<1.7$ [2]. Calorimeter depth is an important factor in providing good containment for electromagnetic and hadronic showers as well as limitation of punch-through into the muon system [2].

The calorimeter is surrounded by the Muon Spectrometer, designed to detect charged particles exiting the barrel and end-cap calorimeters and to measure their momentum in the pseudorapidity range $|\eta|<2.7$ [2]. It is also designed to trigger on these particles in the region $|\eta|<2.4$.

The ATLAS magnet system is consisted of one solenoid and three toroids (one barrel and two end-caps), featuring a unique hybrid system of four large superconducting magnets. This magnetic system is 22 m in diameter and 26 m in length, with a stored energy of 1.6 GJ [2].

One of the most fundamental aspects of this type of detector is the huge amount of data which are being processed during the various phases of data acquisition and analysis [3]. The detector generates unmanageably large amounts of raw data: about 25 megabytes per event, multiplied by 40 million beam crossings per second in the center of the detector, producing a total of 1 petabyte of raw data per second. The need to have a sophisticated data selection (trigger) first and then handle data down to permanent storage, derives from the fact that data are acquired at the clock speed of 40 MHz at the terabytes level where no technology exists today to handle and store such volumes of data [3]. The current LHC luminosity levels produce 0.2 billion p – p events per second. The mean data size for reading out all fired detector channels belonging to the same bunch crossing (BC) is about 1.5 MB. However, the processing and storage capabilities set an upper bound of about 100 selected events per second. To achieve this enormous reduction factor of the order of 107 without losing interest events, the existence of a trigger system is necessary. A block diagram of the ATLAS trigger and data acquisition systems is shown in Figure 3 [2]. This trigger system is briefly described in the following subchapter.

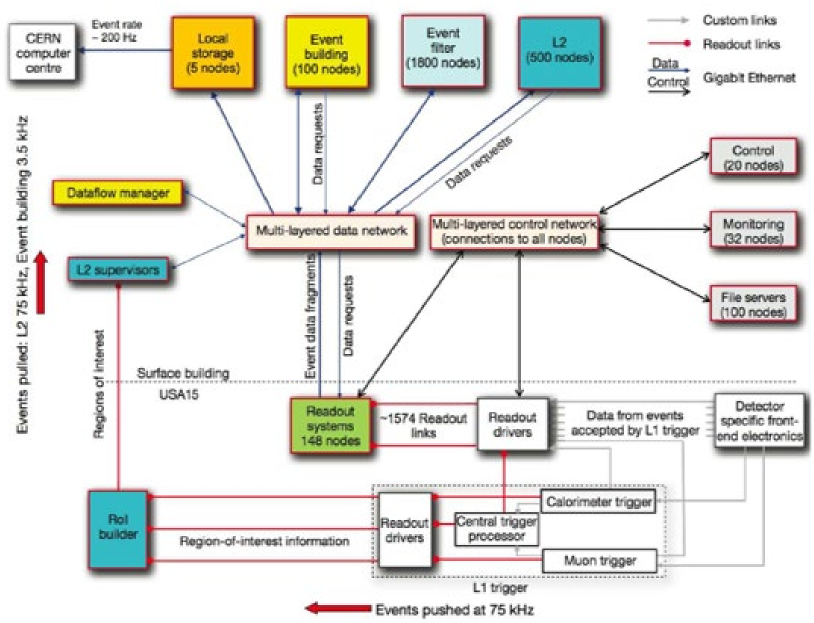

Figure 3. The ATLAS trigger and data acquisition systems diagram (see next section for further details) [2].

1.3 The ATLAS Trigger System

The Trigger system of ATLAS consists of three levels: L1, L2 and the event filter. In each level the decisions that are made define the next level often introducing additional selection criteria. The data acquisition system receives and buffers the event data from the detector-specific readout electronics, at the L1 trigger accept rate, over 1600 point-to-point readout links [2]. The Level-1 Trigger uses muon and calorimeter signals to determine “Regions of Interest” (RoI) and reduces the event rate to 75 kHz by counting clusters of jets, τ, electron/γ and missing ET at various energy thresholds. The Level-2 Trigger uses Level-1 candidates and look at detailed physics properties to achieve a further reduction in rate to 2-3 kHz. Finally, a third level (event filter) uses full event information and decides upon storage of the event for offline analysis with a final rate of 300-400 Hz [15].

The L1 trigger system is in total constituted by the Central Trigger Processor (CTP) which is fed by signals from hardware trigger calorimeter (L1 Calo) and the spectrometer (L1 Muon. The calorimeter selection is based on information from all the calorimeters (electromagnetic and hadronic; barrel, end-cap and forward). The L1 Calorimeter Trigger (L1Calo) aims to identify high-ET objects such as electrons and photons, jets, and τ-leptons decaying into hadrons, as well as events with large missing transverse energy E_T^{miss} and large total transverse energy. Furthermore, a trigger on the scalar sum of jet transverse energies is also available. For the electron/photon and τ triggers, isolation can be required. Isolation implies that the energetic particle must have a minimum angular separation from any significant energy deposit in the same trigger. The information for each bunch-crossing used in the L1 trigger decision is the multiplicity of hits for 4 to 16 programmable E_T thresholds per object type [2].

The L1 muon trigger is based on signals in the muon trigger chambers: RPC’s in the barrel and TGC’s in the end-caps. The trigger searches for patterns of hits consistent with high-p_T muons originating from the interaction region. The logic provides six independently-programmable p_T thresholds. The information for each bunch-crossing used in the L1 trigger decision is the multiplicity of muons for each of the p_T thresholds. Muons are not double-counted across the different thresholds [2]. The architecture of the L1 trigger system is shown in the following Figure 4.

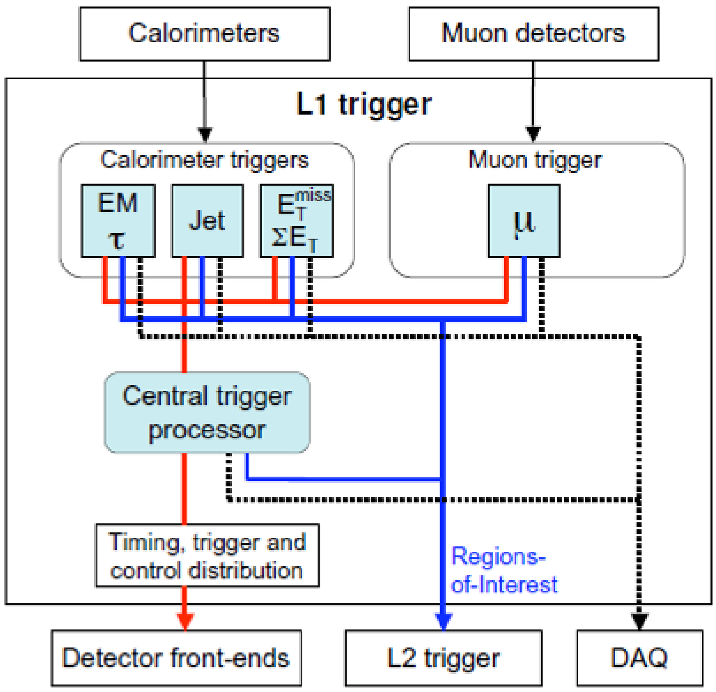

Figure 4. Block diagram of the L1 trigger [2].

The L2 trigger is software based and uses algorithms that run on multiple computers of multiple processors each, which are connected via a fast network. This level uses a L1 RoI as a seed and performs a partial reconstruction of the event starting from the RoI and uses full granularity data from all detectors in contrast to the first level. The full computing power is based on complex decision algorithms that reduce trigger frequency at about 3.5 kHz, with a corresponding average processing time of 40 ms, according to the complexity of the event. The events are sent from the readout servers to events constructors for further analysis [2].

The last level (L3) is the event filter and is also software based. It uses the RoIs of the second level and the full granularity of the detector for entire event processing. The rate now reaches 400 Hz with size event of 1.3 Mbyte. The mean processing time is 4 sec. The events of interest for each different physical analysis are separated into streams of events. The events that have finally passed the filter events are stored for further analysis. Level 2 and event filter consist the high-level trigger [2].

1.4 USA15 experimental area

The ATLAS experimental cavity (UX15) is surrounded by a variety of other caverns and access shafts. The detector cavern is UX15, the adjacent service caverns are USA15 and US15. PX14 and PX16 are the installation shafts for surface access, PM15 and PX15 are the two elevators [16]. The USA 15 cavern, of 20 m width and 62 m length, is designed to accommodate most of the electronics that are necessary for carrying out the experiment (see Figure 5). It is divided into two floors 1.8 m below and 3.6 m above the beam axis. Access for personnel and work for a long time are necessary conditions, even during operation of the LHC at high luminosity, but without exposing to high radiation levels. From USA15 there are several service tunnels leading to the experimental cavity (ULX16, UPX16, ULX14 and UPX14) and service ducts for cables (TE14 and TE16). Moreover, there is a second cavity US15, in which some of the electronics of the ATLAS are hosted, but access is prohibited during LHC operation [17]. With circulating beam, the radiation doses in the Large Hadron Collider (LHC) underground areas reach high levels and therefore access during beam operation must be prohibited for the major part of the underground structure. However, access during beam operation is required for a few underground areas (USA15, upper part of PX24, USC55, part of UX85) and therefore extensive shielding calculations had to be performed [6]. The tests that were performed during the development of the work presented in this thesis, took place in USA15, where human radioprotection is crucial mainly due to the fact that USA15 is one of the four areas where access is allowed even during beam operation.

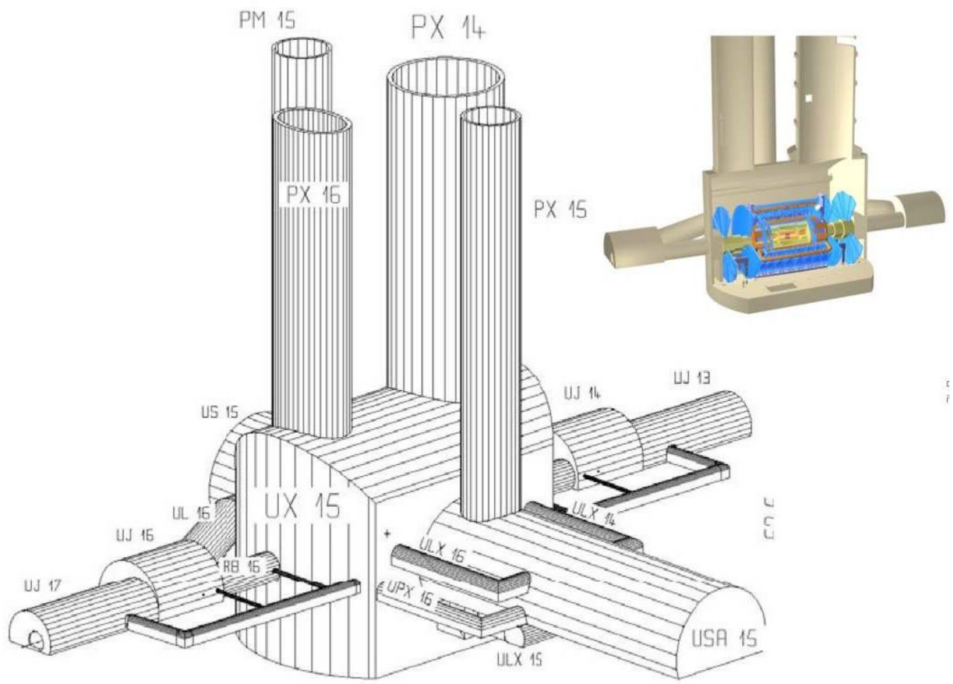

Figure 5. Tfhe experimental area with its individual cavities, tunnels and access shafts, including USA15 [17].

Chapter 2

2 Background radiation and shielding in ATLAS

The ATLAS cavern located 100m underground is not accessible during operation of the experiment because of ionizing radiation, except of some regions that access is needed even when the beam is on. In this chapter we will make an introduction to the background radiation of the experimental area to understand the need of safety systems and radioprotection. We will also refer to the shielding strategies that are used in the area.

2.1 Background radiation in ATLAS

The dominant primary source of background radiation at the LHC, when operating at design luminosity, derives from collisions at the interaction point [2]. Very high energies and collision rates in the ATLAS experiment result in extremely high levels of low-energy neutron, high-energy hadron and photon radiation. Consequently, radiation damage to silicon detectors and electronics, increased detector occupancy, background signals resulting spurious triggers in some of the detector systems and creation of residual radionuclides, are some of the radiological hazards that can impact access and maintenance interventions. Radiation in the LHC underground areas and in the accelerator tunnel is produced when protons interact with the nuclei of the residual gas atoms (beam gas interactions) or with the nuclei of the atoms of every other material surrounding the beams such as beam screens collimators, magnets, cables, cryostats or the beam dump (point losses). When there are circulating beams in the LHC there is a small but continuous loss of protons along the ring. These lost protons will interact with the material that is closed to the beams. These primary interactions produce secondary particles (neutrons, pions, kaons, other protons), with some of them having sufficient energy to interact again and cause the production of tertiary particles and so on, resulting in a hadronic cascade. The fragments of the struck nuclei produced in the hadronic cascade are radioactive and decay in a timescale between a fraction of a second and many days. The accelerator thus continues to produce radioactivity even though there are no more circulating beams. Most of the energy from the primary particles is dumped into two regions in ATLAS, the Target Absorber Secondaries (TAS) collimators, and the Forward Calorimeters (FCal), consisting sources of secondary radiation [2]. A large amount of shielding around these TAS collimators has been designed to achieve very large reductions in the expected background rates in the muon spectrometer [2]. The shielding strategies used in ATLAS will be presented in this chapter.

The impact of background radiation falls into several deleterious effects [18] such as:

a) inefficiencies, degraded resolutions, and increased rates of fake tracks in tracking detector.
b) Spurious trigger rates will increase if the background radiation consists of penetrating tracks.
c) Radiation damage of silicon detectors and electronics.
d) Interactions leading to anomalous deposits of local radiation causing disruption of electronic signals (single event upsets), or destruction of components (single event damage.)
e) Wire detectors can experience “ageing” (reduced gain and therefore efficiency) due to polymerized deposits on the wires caused by radiation interacting with certain components of the detector gas.
f) Nuclear interactions in dense materials, leading to the creation of residual radio-nuclides. The resulting dose rates from radio-activation of certain materials will lead to radiological hazards, which impact access and maintenance scenarios.

Optimization of the shielding in ATLAS is achieved by using different particle physics simulation programs such as FLUKA and GCALOR for simulating hundreds of different geometrical options [2] and by well-defined cross-comparisons for possible transport-code differences [18]. An example of such a GCALOR calculation of the ionizing dose in the region closest to the interaction point is shown in Figure 6, where the locations of the inner detector sub-systems, of the different calorimeters and of the inner end-cap muon stations are indicated [2]. The forward calorimeters will be exposed to up to 160 kGy/y, whereas the corresponding number for the end-cap electromagnetic calorimeters is 30 kGy/y, leading to very large integrated doses over the full lifetime of the experiment [2]. For that purpose, only the LAr (Liquid Argon) technology with its intrinsically high resistance to radiation is used in the end-cap and forward regions [2]. According to this kind of results different shielding materials are investigated and thoroughly tested for radiation hardness.

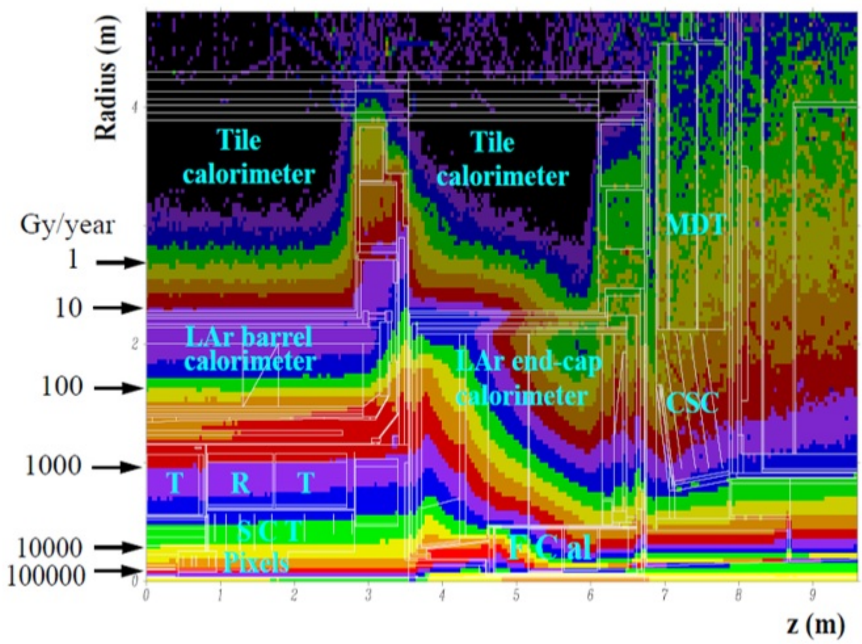

Figure 6. The total ionizing dose per year calculated by GCALOR. The scale on the left gives the integrated dose per year corresponding to the various iso-lines [2].

Studies of induced radioactivity have resulted in more than 600 radiation maps for different regions, running times and cooling-off times available on the ATLAS activation web-pages [19]. The basic conclusion of these studies is that the beampipe will be the major source of radiation in ATLAS. Figure

7 shows the radiation levels in the two access areas during standard access. The predicted dose rates in the two access areas are also shown. The calculation was done for one year of running at 10^{34} $cm^{-2}s^{-1}$ and five days of cooling off. It is recommended that the area around the beampipe, out to a radius of about one meter, is fenced off by a cage-like structure. This will ensure that people working in ATLAS during standard access will not receive more than 0.1 mSv/h and that people thus will be able to work for at least 2-3 weeks before reaching their yearly dose-limit. The only detector that will be inside the cage is the inner detector and so only maintenance of the inner detector will be severely limited during standard access. With the beampipe in place it will in any case be difficult to do extensive maintenance to the inner detector and so the limited availability of the inner detector might not be a problem [18].

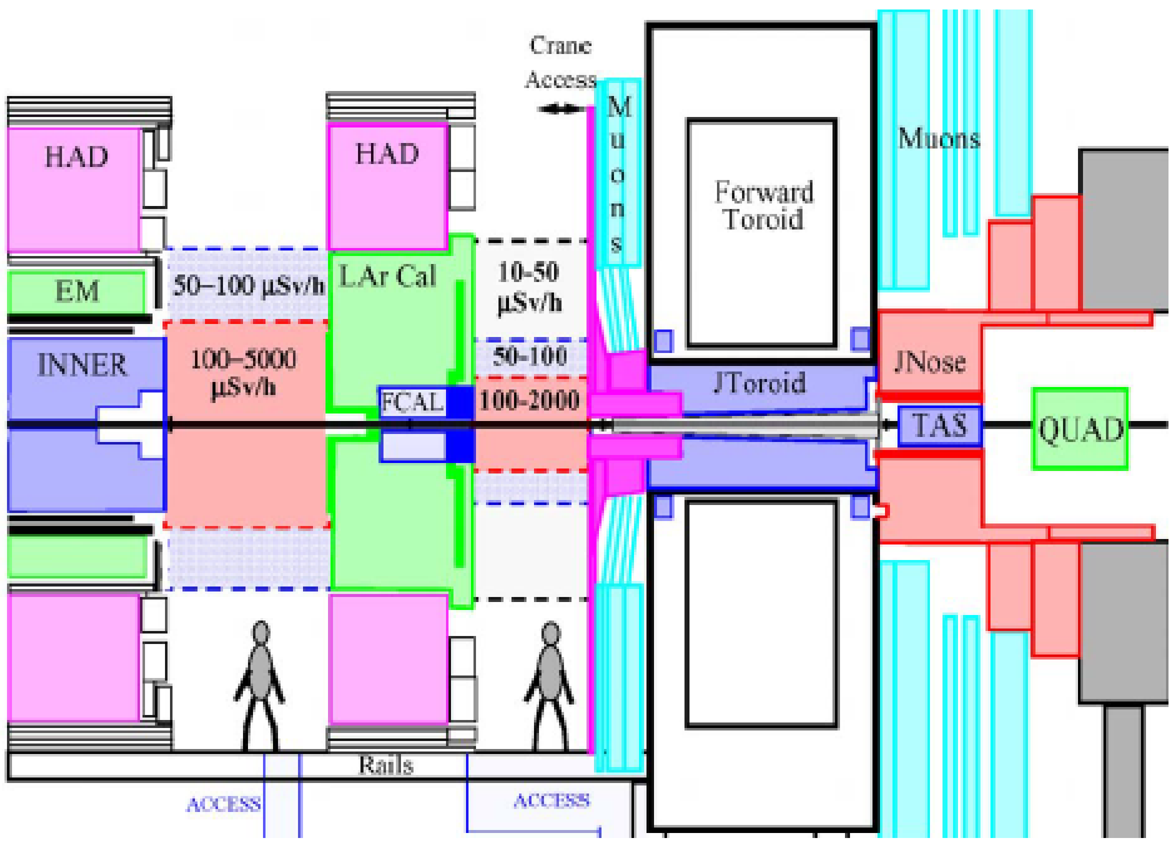

Figure 7. One half of the inner region of the ATLAS experiment during standard access [18].

2.2 Radiation in the USA15 cavern in ATLAS

The wall thickness between the experimental cavern and USA15, described in sector 1.4, the plug thickness at the top of the two vertical shafts, the fluence and dose rate in the surface buildings as well as the maximum dimension allowed for the lateral passage ducts were computed by Ferrari et al. in 1995 [20] [21] . They concluded that the USA15 wall should be 2 m thick, giving a maximum dose equivalent rate in USA15 of ~3 μSv/h (see Figure 15) to foreseen future lower radiation dose limits. The predicted dose equivalent rate of 3 μSv/h was sufficiently low at the time; considering that there were no experimental data from the today's LHC highest beam energy operation of 14TeV proton-proton interactions. In the future, radiation doses will increase mainly due to material activation and higher energy in particle collisions. It is assumed that the wall starts at 1300 cm from the beamline. The results are given for three vertical slices and have been averaged over the full horizontal length of the wall.

2.3 Radiation measurements through a network of detectors

Radiation measurements in the experimental infrastructure can be achieved through a system of small silicon pixel detectors, fully capable of delivering real-time images of fluxes and spectral composition of different particle species. These silicon detectors are operated via active USB cables and USB-Ethernet extenders by a PC placed in the USA15 counting room. The hybrid silicon pixel device consists of a silicon detector chip, 300 mm thick with 256x256 pixels, bonded to a readout chip. Each of the 55 mm x 55 mm pixels is connected to its respective readout chain integrated on the chip. Settings of the pulse height discriminators determine the input energy window and at the same time provide noise suppression. The pixel counter determines the number of interacting quanta of radiation falling within this window. These devices can be used for position and energy sensitive (5 Kev

– tens of MeV) spectroscopic detection of radiation. They are also capable of counting particle fluxes at rates in excess of GHz/cm^2 [12].

This system can be used in both tracking and counting modes, to record tracks or counts caused by x-rays, gamma-radiation, neutrons, electrons, MIPs and ions. The silicon detectors are partially covered by neutron converters for neutron detection. The tracking mode is based on electronic visualization of tracks and traces of 100 individual radiation quanta in the sensitive silicon volume. In the case of count rates above 5x103 events/cm2s, the devices are operated in counting mode, in which charge deposition in the pixels is counted at different threshold settings. Calibration of the devices enables the conversion of the individual tracks observed and/or counts measured into fluxes of respective types of radiation and dose rates. These pixel devices will be placed inside ATLAS: four devices on the LAr calorimeter facing the inner detector, four devices on the tile calorimeter, four devices near the muon chambers in the inner end-cap muon station, and two devices near the forward shielding and close to the outer end-cap muon station [12].

2.4 Shielding strategies

The purpose of this shielding in the ATLAS experimental infrastructure is to reduce the number of background particles in regions that the radiation background is more intense, such as near the beampipe and specifically at the Inner Detector and the Muon spectrometer, to a manageable level and to protect people working in the electronics cavern (USA15) [18]. Total shielding weight of 2825 tons (1887 tons of metal, 920 tons of concrete, and 18 tons of plastic) has been installed around the detector [2]. A multi-layered shielding approach has been used, since different types of radiation can be stopped with different types of shielding materials [2].

Ideally, the shielding materials should be thick enough to absorb all charged particles. Since different types of radiation require different types of shielding materials, a multi-layered shielding approach is used. The inner layer needs to be as dense as possible in order to stop high energy hadrons and secondaries, so it is built from materials that give a large number of interaction lengths into a limited volume, such as copper and iron. In the case of iron, a minimum carbon content of a few percent is advantageous, since it efficiently moderates the neutron energies down to lower values.

The remaining neutrals are mostly neutrons, which can travel long distances, losing their energy gradually. The second layer is consisting of doped polyethylene, which is used to moderate the neutron radiation escaping from the first layer. The low energy neutrons are then captured by a dopant (either boron or lithium).

Photon radiation is created during the neutron capture process. The third shielding layer, which consists of steel or lead, is made to stop these photons. Lead is more effective in stopping photons but it has the disadvantage of giving off more neutron radiation than steel.

In ATLAS, most of the energy from the primary particles is dumped into the TAS collimators, and the Forward Calorimeters (FCal) which are among the strongest sources of secondary radiation and are somewhat self-shielding. Since they are compact, they have been further shielded with layers of dense material and cladding. Although self-shielding, the FCal sits inside the endcap calorimeters, and is not far from the first forward muon station, so there is relatively little space for shielding in this "corner" of the detector, and many design iterations have been done to optimize the shielding shapes and materials in this region.

Shower lengths will be determined by the material-dependent radiation lengths for electrons and photons, and hadronic interaction lengths for hadrons. Shower lengths increase logarithmically with particle energy so, as the energy increases, shower maximum moves deeper into the material and leakage out the back of the shielding increases [18]. Radiation lengths and interaction lengths for the materials more commonly used in the shielding process are presents in Table 1.

Material	***ρ (g/cm³)***	***λ(cm)***	***X_0 (cm)***	***Comment***
Pure Cu	8.9	17.5	1.45	expensive
Cu Alloy	8.6 – 8.8	18.0 – 18.4	1.4 – 1.35	machinable
Pure Fe	7.9	19.1	1.8	n resonances
Steel	7.8	19.2	1.8	low carbon
Caste Fe	7.2	20.4	2	3% carbon
Pb	11.4	18.9	0.56	γ filter
Pure W	19.3	10.3	0.35	elemental
W Alloy	18.2	11.2	0.38	expensive
Concrete	2.4	46.9	10.9	walls
C	2.3	50.0	18.8	moderator
Al	2.7	37.2	8.9	structural
Polyethylene	0.94	92.4	47.0	moderator

Table 1. Radiation lengths and interaction lengths for the materials more commonly used in the shielding.

2.5 Shielding Regions of ATLAS

The shielding in ATLAS (see Fig.8) is divided in six subprojects presented in this subchapter.

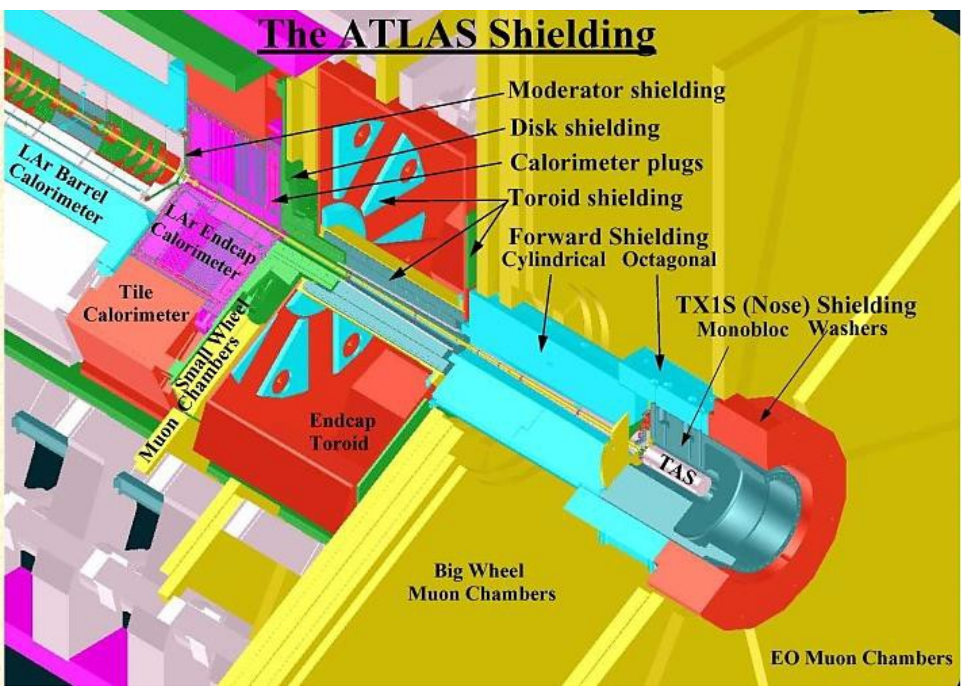

Figure 8. Shielding projects in ATLAS [2].

2.5.1 The Forward shielding (JF)

The two Forward Shielding assemblies are designed to protect the middle and outer end – cap muon stations from background particles created in secondary interactions in the beampipe, the calorimeters and the TAS collimators. This shielding is removable and will be stored in the surface building during maintenance of ATLAS. The JF shielding consists of 2x387=775 tons of cast iron, 2x24=48 tons of steel plates and 2x5.5=11 tons of polyethylene for a total weight of 2x418=836 tons.

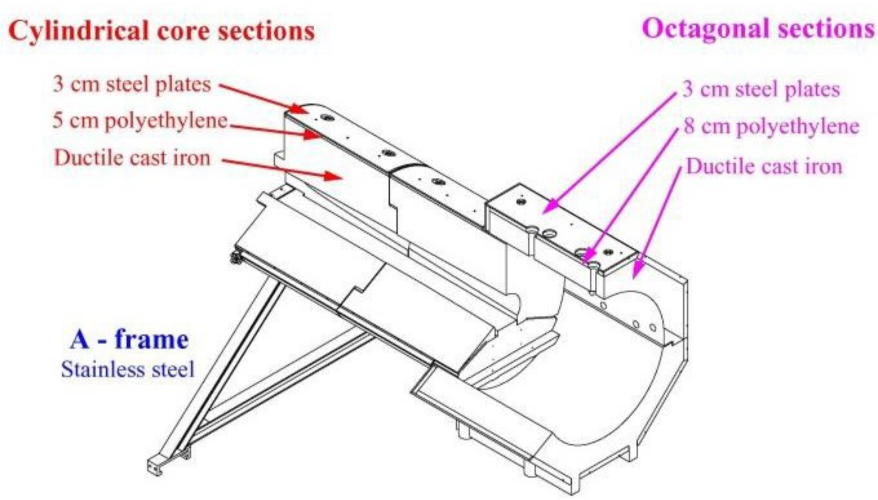

Figure 9. JF Shielding [2].

The shielding consists of two parts: The cylindrical core, which is enclosed in a 5 cm polyethylene layer and the octagonal back that is surrounded by a thick layer of 8 cm. Three pieces called JFC1 ("the bridge"), JFC2 and JFC3 are used for the core and two pieces called "JFS3 upper" and "JFS3 lower" for the octagonal back. All pieces are made of cast ductile iron, which has a large carbon content useful as a moderator of neutron radiation and they are surrounded by a layer of polyethylene doped with boron in the form of H3BO3 and followed by a 3 cm thick steel layer. These polyethylene layers are made of 10000 bricks of three different shapes [2] [22].

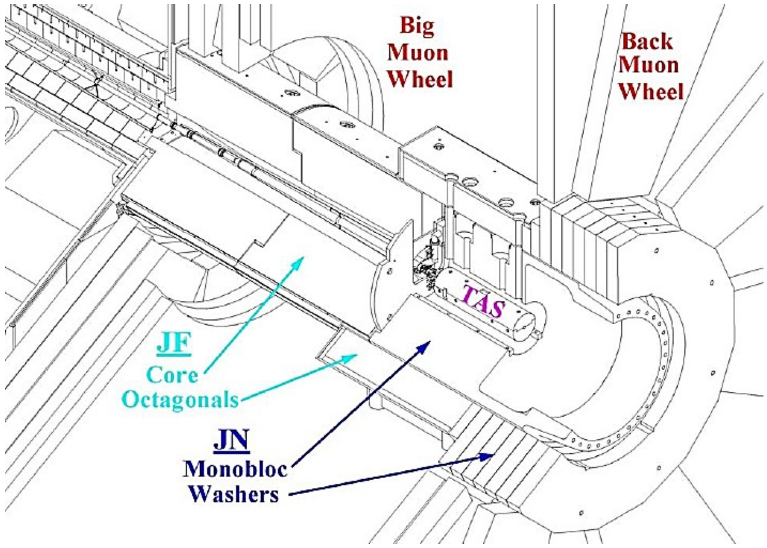

Figure 10. The massive shielding in the forward region (JF, JN) and TAS collimator [18].

2.5.2 The Disk shielding (JD)

The Disk shielding (JD) has a diameter of 872 cm and a total weight of 2x87=174 tons and serves a threefold purpose. Firstly, it supports the muon chambers in the first end-cap muon station (Small Wheel). Secondly, it shields these chambers from background radiation emerging from the calorimeters. Also, it provides a well- defined way for the magnetic field flux return from the solenoid magnet. The bulk of this shielding disk, which supports end – cap muon trigger chambers, consists of a vertical steel disk with a diameter of 872 cm. Three such disks are used for the 8 cm thick "large disc". Two disks of diameter of 540 cm are used for the 5 cm thick "small disk". Two cast iron ribs of 1.6 tons attached to the large disk are used to increase the mechanical stability of the JD.

The magnetic field is lead back through the electronic boxes of the Tile calorimeter by using 31 ring segments, with a weight of 161 kg each, attached to the large disk.

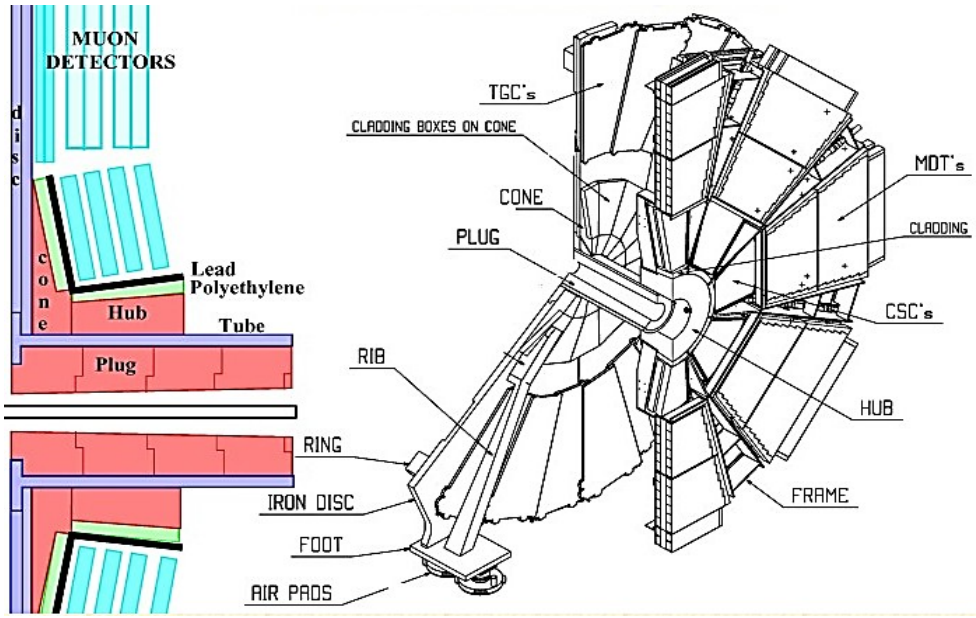

Figure 11. JD Shielding [2].

At the center of the large disk and surrounding the beam-pipe is a stainless-steel tube of length 208 cm, weight 5.4 tons and diameter 106 cm stainless steel tube containing a set of cylindrical shielding pieces made of leaded red brass (85% Cu, 5% Pb, 5% Sn, 5% Zn), which also supports Cathode Strip Chambers (CSCs) and Monitored Drift Tubes (MDTs).

Figure 12. Left: The stainless-steel tube. Right: The front of the JD Shielding [22].

The total weight of the muon chambers is about 13 tons (SW 12 tons and TGCs 1 tone) which brings the total weight of one JD/SW assembly to 98 tons [2] [22].

2.5.3 The Toroid Shielding (JT)

The toroid shielding consists of two parts: JTT which is outside the toroid and surrounds the beampipe, and JTV which is neutron shielding situated inside the endcap toroid cryostat.

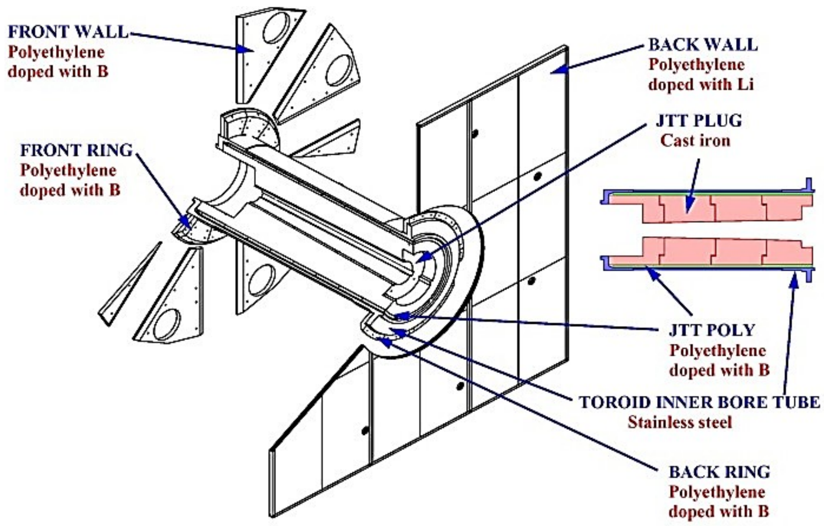

Figure 13. JT Shielding [2].

The JTT is a cylindrical structure made of ductile cast iron of overall length about 473.6 cm, which surrounds the beampipe on the inside of the two end-cap toroid cryostats. The shielding consists of four plug pieces numbered (1 – 4) from the back. The front piece has a large hole in the center, into which the stainless-steel tube of the JD fits. On the outside of the cast iron is a polyethylene layer doped with B2O3 (5%) that is used for neutron shielding. The photons created in the polyethylene layer are stopped by the stainless-steel ECT bore tube, which supports the shielding in the end-cap toroid. The JTT shielding consist of 2x55=110 tons of cast iron and 2x1.3=2.6 tons of polyethylene for a total weight of 113 tons.

Chapter 3

In this chapter we will refer to the Safety Systems that are currently used in ATLAS experimental area and the radioprotection rules and methods used for the personnel.

3 Safety and personnel radioprotection in ATLAS

3.1 Anticipated hazards and existing safety systems

The main risks are in the underground experimental area, especially in the main cavern and the adjacent technical-service caverns (see Fig.5) [2]. These areas are accessible to the personnel for maintenance activities and significant risks can occur. Gases found in Ultra High Vacuum Systems: H2, CH4, H20, CO, and CO2 can leak, which can cause fire incidents, local oxygen deficiency levels and detector damages. There are several safety systems for gas leaks and fire incidents, such as Sniffer Systems, Hydrants, Fire extinguishers, Water Mist Systems, High expansion foam, Air conditioning, Sputter ion-pumps and Titanium sublimation pumps. Oxygen Deficiency detectors are crucial to detect possible Oxygen Deficiency Hazard (ODH) caused due to the cryogenic technologies used in the cavern and the presence of large quantities of CF4 gas that can lead to ODH. LHC is the largest cryogenic system in the world and one of the coldest places on Earth. Cryogenic-fluid leaks can take place around the Cryogenic Infrastructures (Super Conducting Magnets, LAr Calorimeters). Personal Protective Equipment (PPE), Equipment/system leak tightness, Ventilation/extraction systems, ODH detectors, Emergency procedures & evacuation plans as well as low temperature compatible materials are safety systems that can protect the personnel from hazards linked to Cryogenic-fluid leaks. A synopsis of possible anticipated hazards in the working environment and the current safety systems that are used are presented in Table II.

Environmental Hazards	**Hazard Sources**	**Safety Systems**
Radiation Hazards (Personnel Doses, Material activation/aging/destruction)	Beam on (Prompt Ionizing Radiation), Beam off (Ionizing Radiation due to radioactivity)	Shielding, ALARA regulations, CERN's Safety Code, Dosimetry, Radiation Monitoring Systems
Gas leaks/Smoke Detection	Fire, Gases found in Ultra High Vacuum Systems: H2, CH4, H20, CO, and CO2	Sniffer Systems, Hydrants, Fire extinguishers, Water Mist Systems, High expansion foam, Air conditioning, Sputter ion-pumps, Titanium pumps
Oxygen Deficiency Hazard (ODH)	Cryogenic technologies, Presence of large quantity of CF4 gas	Oxygen Deficiency detectors
Cryogenic-fluid leaks (Helium, Nitrogen and Argon)	Cryogenic Infrastructures (Super Conducting Magnets, LAr Calorimeters)	Personal Protective Equipment (PPE), Equipment/system leak tightness, Ventilation/extraction systems, ODH detectors, Emergency procedures/ evacuation plans

Table 2. Anticipated environmental hazards.

Several safety systems have been designed and implemented to detect at a very early stage any possible sources of danger in the underground work environment and to activate alarms and trigger the required safety actions [2]. These systems have been implemented under the direct supervision of the ATLAS GLIMOS (Group Leader In Matters Of Safety) leading the ATLAS safety organization and of the CERN Safety Commission [2]. These safety systems have been designed and implemented so as to detect at a very early stage any event which might endanger the safety of personnel, environment or ATLAS equipment and the readout of most of these systems uses the software that is provided by the DCS described in this chapter [2].

The enormous size of complexity of the ATLAS environment, creates the obvious need of developing a supervision system that will enable monitoring of the various critical processes that take place in this experimental infrastructure. The inner tracker, the calorimeter and the muon system are the three detectors that consist the ATLAS experiment, each of which is composed of several sub-detectors. The individual detector components as well as the common experimental infrastructure are supervised by the Detector Control System (DCS), by using a highly distributed system of 140 servers, to ensure safe operation [23]. The DCS enables equipment supervision using operator commands, reads, processes and archives the operational parameters of the detector, allows for error recognition and handling, manages the communication with external control systems, and provides a synchronization mechanism with the physics data acquisition system [23].

The ATLAS Control System architecture, shown in Figure 14, consists of various front-end systems and the back-end Supervisory Control and Data Acquisition system (SCADA). The back-end is organized into three functional horizontal planes, a Local Control Station (LCS), a Subdetector Control Station (SCS) and a Global Control Station (GCS). This is a flexible hierarchy that reliably models the natural ATLAS sub-detecting parts and subsystems [2].

The LCS layer transmits information regarding the state of subdetectors and subsystems. The LCS may execute orders received from higher hierarchy levels or can perform its own special features. The SCS layer is the middle level of the hierarchy. Each subdetector has its own control station which allows full local control. At this level of the hierarchy the subdetectors are connected to external systems such as the magnet system, the LHC, the detector safety system, and trigger and data acquisition system. The GCS layer forms the upper part of the hierarchy and is responsible for the overall operation of the detector. This level summarizes all the functions of each state of the detectors and subsystems as well as malfunctions [2].

The front-end system consists of various parts arranged near the detector or in nearby rooms. It is directly linked with the hardware of the detector, providing reading, digitization and in some cases, signal processing and data transfer to the back-end. On the other hand, it receives and executes instructions from the back-end. The equipment of the front-end consists of sensors, controllers, digitizers, processors and computing systems pf autonomous basis. All the ATLAS front-end equipment has to adapt to different requirements regarding radiation tolerance (radiation levels can reach 100 kGy/year in areas close to the IP), operating under strong magnetic fields (magnetic field amplitude can reach 4 T in the cavern), the long lifetime (because the experiment is planned to operate for more than a decade), and low cost [2].

The back-end system consists of different software systems which communicate with the front-end providing supervisory control at the detector users. The back-end system, used by the four LHC experiments, carried out using a commercial SCADA package. The SCADA package provides software tools and guidelines ensuring the homogeneity of the back-end in all the different subsystems, their subdetectors and the LHC experiments [2].

The ATLAS automatic control system archives the important parameters in the ATLAS online database, which is only accessible via the ATLAS control network (ATCN), for security and performance reasons. However, an offline base, which is an exact copy of the online database, serves users within the CERN General Public Network (GPN). Although the operating parameters are of the order of 106, the delay of the replication mechanism does not exceed a few seconds [2].

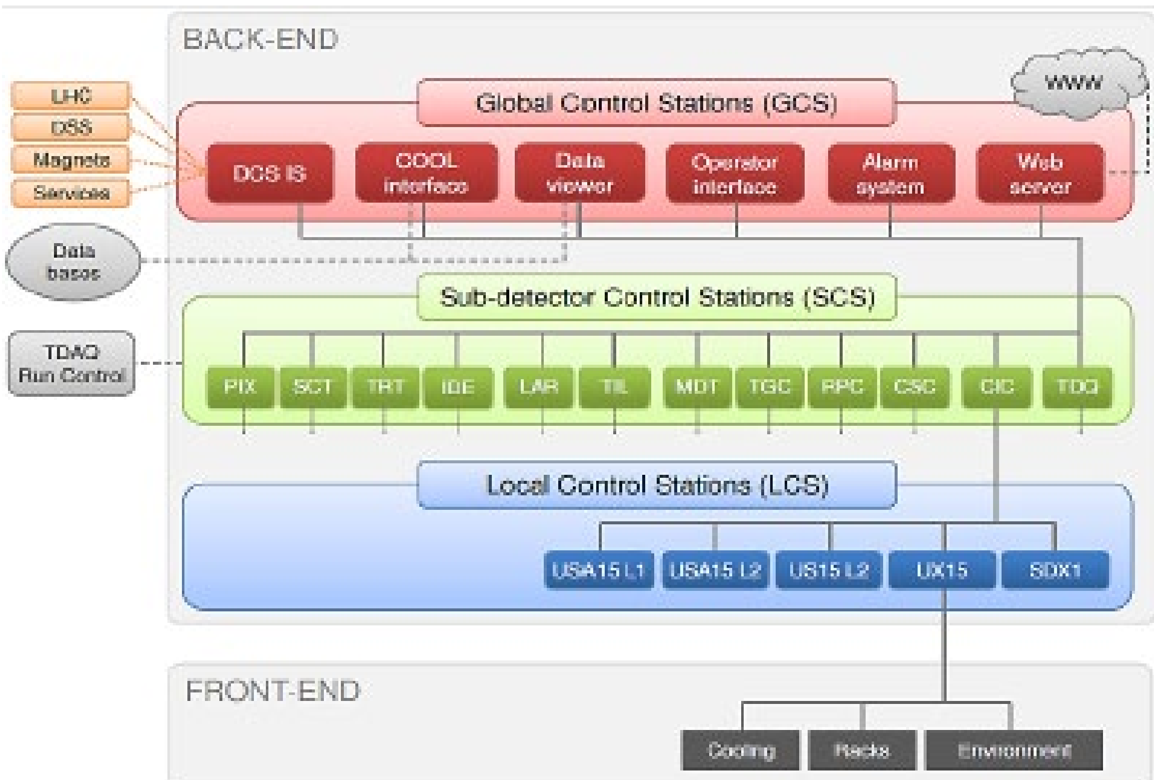

Figure 14. The DCS Architecture of ATLAS [2]

Safety in the ATLAS experiment needs to be ensured for the personnel, the protection of the environment equipment and infrastructure during the installation and the various phases of operation of the detector (data-taking, access and maintenance) [2].

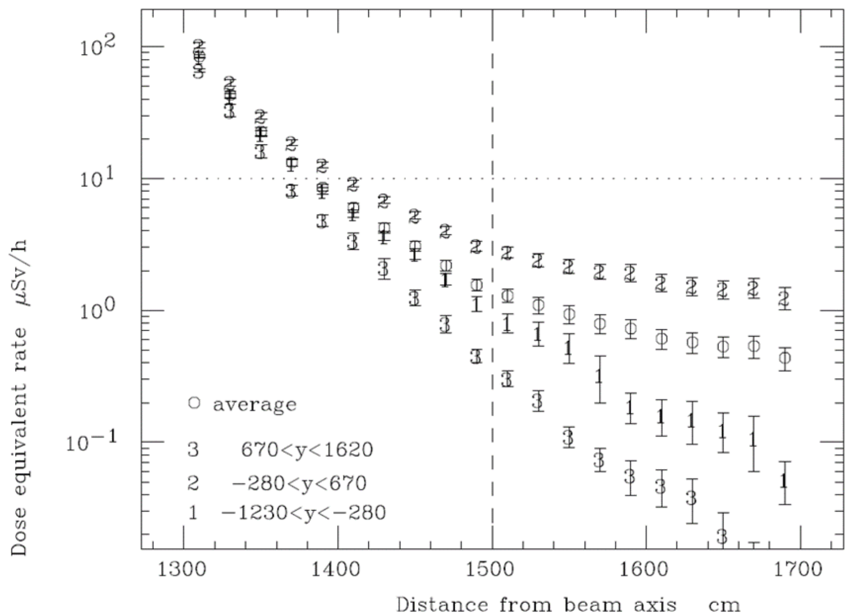

Figure 15. Dose equivalent rate, averaged over the whole wall length versus the distance from the beam line [20].

3.2 Personnel radiation protection

CERN's radiation protection policy stipulates that the exposure of persons to radiation and the radiological impact on the environment should be as low as reasonably achievable (the ALARA principle), and should comply with the regulations in force in the Host States and with the recommendations of competent international bodies [24].

The International Commission on Radiological Protection (ICRP) has specified in its Recommendation 60 [25], that any exposure of persons to ionizing radiation should be controlled and should be based on three main principles, namely:

– justification: any exposure of persons to ionizing radiation must be justified;

– limitation: personal doses must be kept below legal limits;

– optimization: personal and collective doses have to be kept as low as reasonably achievable (ALARA).

These recommendations have been fully incorporated into CERN's radiation safety code [26].

The legal protection limits for radiation are not expressed directly in measurable physical quantities, resulting in the inability of quantifying the biological effects of ionizing radiation exposure of the human body. Therefore, protection limits are expressed in terms of so-called protection quantities, which quantify the extent of exposure of the human body to ionizing radiation from both whole-body and partial-body external irradiation and from the intake of radionuclides. In order to demonstrate compliance with dose limits, so-called operational quantities are typically used, which are aimed at providing conservative estimates of protection quantities. The radiation protection detectors used for individual and area monitoring are often calibrated in terms of operational quantities [24]. The following quantities will be described as they were presented by Forkel et al. [24].

3.2.1 Physical quantities

The **fluence Φ**, is the quotient of dN by da, where dN is the number of particles incident upon a small sphere of cross-sectional area da, and it is measured in units of $1/m^2$:

$$\Phi = \frac{dN}{da} \quad (1)$$

The **absorbed dose D** (measured in units of grays; 1 Gy = 1 J/kg = 100 rad) is the energy imparted by ionizing radiation to a volume element of a specified material divided by the mass of that volume element.

The **kerma K** (measured in units of grays) is the sum of the initial kinetic energies of all charged particles liberated by indirectly ionizing radiation in a volume element of a specified material divided by the mass of that volume element.

The **linear energy transfer L** or LET (measured in units of J/m, but often given in keV/µm) is the mean energy dE lost by a charged particle owing to collisions with electrons in traversing a distance dl in matter. Low-LET radiation (L < 10 keV/µm) comprises X-rays and gamma rays (accompanied by charged particles due to interactions with the surrounding medium), and light charged particles such as electrons that produce sparse ionizing events far apart on a molecular scale. High-LET radiation (L > 10 keV/µm) comprises neutrons and heavy charged particles that produce ionizing events densely spaced on a molecular scale.

The **activity A** (measured in units of Becquerel; 1 Bq = 1/s = 27 pCi) is the expectation value of the number of nuclear decays in a given quantity of material per unit time.

3.2.2 Protection quantities

The **organ absorbed dose D_T** (measured in units of grays) in an organ or tissue T of mass m_T is defined by

$$D_T = \frac{1}{m_T}\int_{m_T} D dm. \quad (2)$$

The **equivalent dose H_T** (measured in units of Sieverts; 1 Sv = 100 rem) in an organ or tissue T is equal to the sum of the absorbed doses $D_{T,R}$ in an organ or tissue caused by different radiation types R weighted by so-called radiation weighting factors w_R:

$$H_T = \sum_R w_R \times D_{T,R} \quad .(3)$$

This equivalent dose expresses the long-term risks (primarily cancer and leukemia) from low-level chronic exposure.

The **effective dose E** (measured in units of Sieverts) is the sum of the equivalent doses, weighted by the tissue weighting factors w_T (where $\Sigma_T w_T$ =

1), for several organs and tissues T of the body that are considered to be the most sensitive:

$$E = \sum_T W_T \times H_T \ . \ (5)$$

3.2.3 Operational quantities

The **ambient dose equivalent H*(10)** (measured in units of Sieverts) is the dose equivalent at a point in a radiation field that would be produced by a corresponding expanded and aligned field in a 30 cm diameter sphere of tissue of unit density at a depth of 10 mm, on the radius vector opposite to the direction of the aligned field. The ambient dose equivalent is the operational quantity for area monitoring.

The **personal dose equivalent H_p(d)** (measured in units of Sieverts) is the dose equivalent in standard tissue at an appropriate depth d below a specified point on the human body. The specified point is normally taken to be where an individual dosimeter is worn. The personal dose equivalent $H_p(10)$, with a depth d = 10 mm, is used for the assessment of the effective dose, and $H_p(0.07)$, with d = 0.07 mm, is used for the assessment of doses to the skin and to the hands and feet. The personal dose equivalent is the operational quantity for monitoring of individuals.

3.2.4 Dose conversion coefficients

The direct calculation of protection or operational quantities from the particle fluence are functions of the particle type, energy, and irradiation configuration. These functions are the dose conversion coefficients and the most commonly used are those for the effective dose and ambient dose equivalent. The former coefficients are based on simulations in which the dose to organs of anthropomorphic phantoms is calculated for approximate actual conditions of exposure, such as irradiation of the front of the body (antero-posterior irradiation) or isotropic irradiation [24].

3.3 Radiation levels

Natural background radiation.

Worldwide, on average, the annual whole-body dose equivalent due to all sources of natural background radiation ranges from 1.0 to 13 mSv, with an average of 2.4 mSv. In certain areas, values up to 50 mSv have been measured. A large fraction (typically more than 50%) originates from inhaled natural radioactivity, mostly radon and radon decay products. The dose equivalent due to radon can vary by more than one order of magnitude: it is 0.1–0.2 mSv per year in open areas, 2 mSv per year on average in houses, and more than 20 mSv per year in poorly ventilated mines [24].

Cosmic ray background radiation.

At sea level, the whole-body dose equivalent due to cosmic ray background radiation is dominated by muons; at higher altitudes, nucleons also contribute. The dose equivalent rates range from less than 0.1 μSv/h at sea level to a few μSv/h at aircraft altitudes [24].

Cancer induction.

The cancer induction probability is about 5% per Sievert on average for the entire population [24].

Lethal dose.

The whole-body dose from penetrating ionizing radiation resulting in 50% mortality in 30 days, assuming no medical treatment, is 2.5–4.5 Gy (RBE-weighted when necessary), as measured internally on the longitudinal center line of the body. The surface dose varies because of variable body attenuation and may be a strong function of energy [24].

Recommended dose limits.

The International Commission on Radiological Protection (ICRP) recommends a limit for radiation workers of 20 mSv effective dose per year averaged over five years, with the provision that the dose should not exceed 50 mSv in any single year. The limit in the EU countries and Switzerland is 20 mSv per year and 50 mSv in United States [24]. Many physics laboratories in the US and elsewhere set lower limits while the dose limit for the general public is typically 1 mSv per year [24].

3.4 Health effects of ionizing radiation

The need for human radiation protection derives from the numerous health effects of ionizing radiation which we will briefly describe in this sub-chapter. Radiation can cause two types of health effects, deterministic and stochastic.

Deterministic effects, usually measured in gray units (1 Gy = 1J/Kg), are tissue reactions which cause injury to a population of cells if a given threshold of absorbed dose is exceeded. The severity of the reaction increases with dose. The quantity used for tissue reactions is the absorbed dose D. When particles other than photons and electrons (low-LET radiation) are involved, a dose weighted by the Relative Biological Effectiveness (RBE) may be used. The RBE of a given radiation is the reciprocal of the ratio of the absorbed dose of that radiation to the absorbed dose of a reference radiation (usually X-rays) required to produce the same degree of biological effect. It is a complex quantity that depends on many factors such as cell type, dose rate, and fractionation [24].

Stochastic effects are malignant diseases and inheritable effects for which the probability of an effect occurring, but not its severity, is a function of dose without a threshold [24].

For each type of deterministic effect (erythraemia, depletion of bone marrow and blood cells, necrosis, vomiting, etc.), there is a dose threshold for the damage to become assessable or visible. The various types of damage observable after acute irradiation, and their dose equivalents are listed in Table 1 [24].

Dose (whole-body irradiation)	Effects
<0.25 Gy	No clinically recognizable damage
0.25 Gy	Decrease in white blood cells
0.5 Gy	Increasing destruction of leukocyte-forming organs (causing decreased resistance to infections)
1 Gy	Marked changes in the blood (decrease in the numbers of leukocytes and neutrophils)
2 Gy	Nausea and other symptoms
5 Gy	Damage to the gastrointestinal tract causing bleeding and ~50% death
10 Gy	Destruction of the neurological system and ~100% death within 24 h

Table 3. Radiation damage to the human body [24].

3.5 Radiological classification at CERN

According to the effective dose a person is liable to receive during his stay in the area under normal working conditions during routine operation, the areas inside CERN's perimeter are classified in the following three types that are in line with the Safety Code F (2006) [26]:

i. Non-designated Areas.
ii. Supervised Radiation Areas.
iii. Controlled Radiation Areas.

The latter two are jointly termed **Radiation Areas**.

The potential external and internal exposures must be considered when assessing the effective dose that persons may receive when working in an area under consideration. Limitation of exposure in terms of effective dose is ensured by limiting an operational quantity, the ambient dose equivalent rate H*(10) for exposure from external radiation, and by setting action levels for airborne radioactivity and specific surface contamination at the workplace for exposure from incorporated radionuclides. The radiological classification used at CERN is shown in Table 4 [24].

	Area	Dose limit [year]	Ambient dose equivalent rate		Sign	
			Work place	Low occupancy		
	Non-designated	1 mSv	0.5 µSv/h	2.5 µSv/h		
Radiation Area	Supervised	6 mSv	3 µSv/h	15 µSv/h	Dosimeter obligatory / Dosimètre obligatoire	
Radiation Area	Simple	20 mSv	10 µSv/h	50 µSv/h	Dosimeter obligatory / Dosimètre obligatoire	Controlled Area
Radiation Area	Limited Stay	20 mSv		2 mSv/h	LIMITED STAY / SÉJOUR LIMITÉ / Dosimeters obligatory / Dosimètres obligatoires	Controlled Area
Radiation Area	High Radiation	20 mSv		100 mSv/h	HIGH RADIATION / HAUTE RADIATION / Dosimeters obligatory / Dosimètres obligatoires	Controlled Area
Radiation Area	Prohibited	20 mSv		> 100 mSv/h	PROHIBITED AREA / ZONE INTERDITE / No Entry / Défense d'entrer	Controlled Area

Table 4. Classification of Non-designated Areas and Radiation Areas at CERN [24].

Chapter 4

4 Remote Monitoring and Data Acquisition Systems

The vision of future smart monitoring and data acquisition systems used for supervision purposes and healthcare, depends on the technological advances on computer, networking and medical fields as well as the collaboration between researchers on those fields. In order to improve the existing remote monitoring systems (RM), it is important to combine large-scale wireless telecommunication technologies such as 4G and Wi-Fi Mesh along with small-scale personal area technologies like radio frequency identification (RFID), Bluetooth, ZigBee, and wireless sensor networks and integrate them in tele-medical systems [27]. Accurate and reliable data acquisition (DAQ) and transmission is of great importance, since according to the processing of the acquired data, emergency conditions that need interventions can be identified. Most of the existing health monitoring systems include one or more types of sensors carried by the patient, forming a Body Area Network (BAN) as presented in chapter 2, and one or more types of sensors deployed in the environment forming a Personal Area Network (PAN). These two are connected to a backbone network via a gateway node. At the application level, the healthcare professionals or other caregivers can monitor the vital health information of the patient in real-time via a graphical user interface (GUI). The emergency situations produce alerts by the application and these alerts and other health status information can be reached via mobile devices like laptop computers, Personal Digital Assistants (PDAs) and smart phones [27].

Apart from wireless telecommunication technologies, other technologies such as Augmented Reality (AR) [28], can be used to form novel remote monitoring systems. AR is a technology, which aids in seeing a real scene with embedded data useful for human activities. An AR system supplements the real world with virtual (computer-generated) objects that appear to coexist in the same space as the real world [28]. This system combines real and virtual objects in a real environment, runs interactively, and in real time, and registers (aligns) real and virtual objects with each other. This technology is used in wide areas namely education, health care, maintenance, construction, gaming, navigation, military, tourism etc. The personnel safety and Augmented Reality based safety architectures can be found in many domains. Poole et al [29] presented the Daresbury personnel safety system. This personnel safety system designed for the Synchrotron Radiation Source (SRS) is a unified system covering the three accelerators of the source itself, the beam lines, and the experimental stations. Kumar et al. [30], presented indoor environmental gas monitoring system based on digital signal processing. The paper focused on the problem of real time processing of carbon monoxide and carbon dioxide gases measurement using a DSP board (TMS320C6455) and then implementing to the proposed gas monitoring system. Pantelopoulos et al [31] described a survey on wearable sensor-based systems for health monitoring and prognosis. An emphasis was given to multi parameter physiological sensing system designs, providing reliable vital signs measurements and incorporating real-time decision support for early detection of symptoms or context awareness. Multi sensor data fusion for fire detection has been described by Zervas et al in [32]. The goal of this work was the deployment of Wireless Sensor Network at Urban-Rural Interface aiming to the detection, monitoring and crisis management of such natural hazard. Mizune et al [33] presented outdoor Augmented Reality for direct display of hazard information. The paper concentrated on a tracking system with high accuracy and real-time processing by template matching for Augmented Reality. Mining risk information in hospital information systems as risk mining, a research was carried out by Tsumoto et al [34]. This paper proposed risk mining where data including risk information are analyzed by using data mining methods and mining results are used for risk prevention.

An AR personnel safety system used for maintenance jobs in radioactive environments such the ATLAS cavern at CERN is presented in subchapter 3.2 as a case study of remote monitoring and data acquisition system used for supervision purposes. This research was conducted under the framework of the EDUSAFE European project described in the following chapter. A novel remote monitoring system using smart data acquisition will be presented and described in the following subchapter.

4.1 Smart mobile end-to-end monitoring architecture

To cope with the exponential increase in population suffering from life-long diseases and their associated costs, many health-care governments decided to shift in to mobile health monitoring systems, where smartphones, pocket personal computers or personal digital assistants are employed as the main coordination and processing module [35]. Some of the benefits of mobile RM systems are long-term monitoring of cognitive disorders like Alzheimer's, Parkinson's or similar cognitive diseases brain disorders as well as reduction of costs from unnecessary hospitalizations. Serhani [35] et al. developed a novel smart mobile end-to-end monitoring architecture (shortly, SME2EM) in order to monitor and visualize life-long diseases [35]. The endeavor of this architecture was to bridge the gap between sensor manufacturers, smart-phone industries, public internet service providers, data storage providers, patients, physicians, and health-care stakeholders in general [35]. This architecture is designed in layers and it is shown in Figure 16. More specifically, the upper layer includes the following five monitoring processes:

1) **Smart Data and signal Acquisition**. This process smartly handles data and signal retrieval and transmission in an effective way. The event driven mechanism is developed within the mobile application to

pro-actively and intelligently collect data when needed or when a specific pre-defined/pre-programmed situation occurs or even when unexpected events happen. During this process, data that are useful for identifying and diagnosing disease patterns are acquired. The mobile application has required preliminary logic to decide on the accuracy, reliability and effectiveness of collected data. This logic is mainly based on continuous control of the sensor's contact with the body as well as the use of vital signs thresholds (e.g., normal min and max values) as provided by physicians. Finally, SME2EM data acquisition is considered smart, as it implements the event-driven mechanism and the data of interests to the diagnosis process is solely tagged for synchronization with the back-end server, while other acquired data remain on the mobile, in case they might be required later [35].

2) **Smart Data and signal Preprocessing**. This process handles data pre-processing and implements a couple of smart features at the mobile devices to effectively retrieve and transmit sensory data. These features include data compression, grouped transmission and delayed transmission. The main motivation behind implementing such features within mobile application is to optimize the transmission cost by reducing required resources (e.g., battery and network). The pre-processing activity is characterized as smart, since each one of these actions, such as compression, is executed only if resources are available and the action will be of benefits [35].

3) **Smart Signal Feature Extraction and Selection**. This process applies only when collected data are complex signals recorded by ECG and Electroencephalography (EEG) sensors. It combines different feature extraction techniques to improve the accuracy of differentiating individual disease's states, which cannot be obtained by individual techniques. Moreover, it improves the processing time that is needed to select relevant features by combing filter and

wrapper selection algorithms into an effective algorithm. The feature extraction and selection are characterized as smart, since the developed algorithms are of proactive nature, meaning that their outputs will be changed according to the embedded learning algorithms and training-sensing EEG (or ECG) signals. Thus, each patient may have different selected features [35].

4) **Smart Data and signal Classification and Diagnosis**. Different unsupervised learning techniques (e.g., hierarchical clustering techniques) are combined to increase precision and robustness of the classification and diagnosis process. From a transactional flow perspective, this process is invoked through Web services whose functionalities are exposed to other mobile applications. The classification algorithm is branded as smart as it is readily a proactive one as well as its input selected features as well [35].

5) **Smart Visualization**. In this process of visualizing data, the following purposes are served: (1) validation of collected data, (2) support physician to make appropriate decisions, (3) report on continuous updates on seizures, and (4) preparation of any interventions in case of critical situations. The data can be presented using different views that include firstly a summary of monitoring results, graphs, and patterns of readings, and secondly, a report on discrepancies of measures and generation of automatic preventive actions that might be suggested. Since the volume, speed, and dynamist of collected data characterize signal-based monitoring, then a visualization activity consists of: (1) pre-processing and formatting of sensory data, (2) personalized views, (3) optimized display, (4) meaningful representation (e.g., 3D representation), and (5) on demand visualization. The visualization features of SME2EM are considered as smart, as they achieve proactive, dynamic and adaptive behaviors, which are implemented in our graphical user interface (GUI) [35].

The two lower layers include tools, platforms, and technologies used to support and provide services (e.g., storage, processing, and analysis) to the upper layer's processes [35]. The communication between the heterogeneous platforms and applications is supported through the use of standard Simple Object Access Protocol (SOAP) and light weight data-interchange format called JavaScript Object Notation (JSON) [35]. The proposed architecture additionally supports high-processing performance, massive data storage and analysis, thanks to the Cloud elasticity, which supports big data tools, processes, and technologies [35].

A similar architecture of the system that was developed for health monitoring and supervision purposes in the extreme environmental working conditions inside the ATLAS cavern at CERN, will be presented in the following chapter, as a case study of a remote monitoring and data acquisition system.

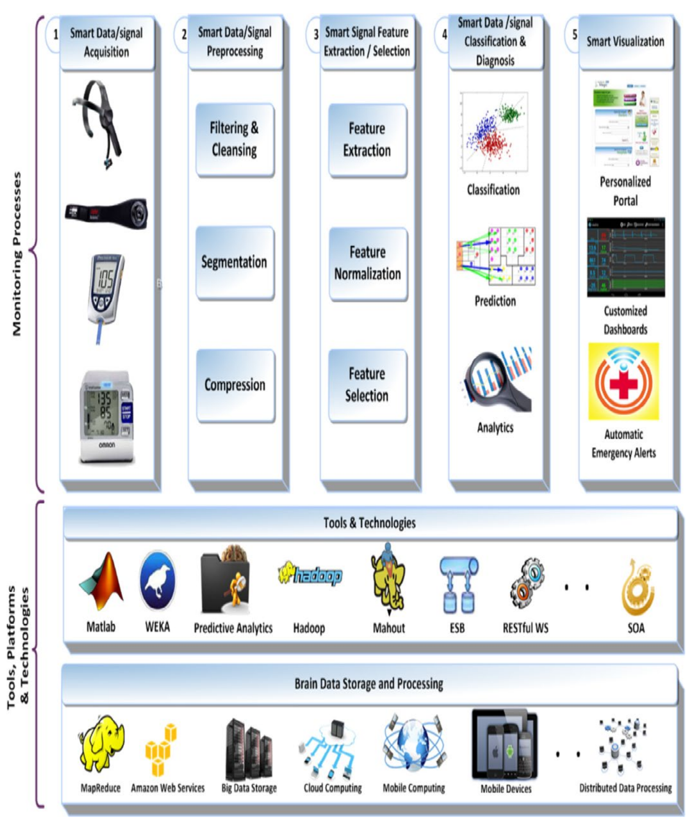

Figure 16. Smart end-to-end architecture for health monitoring and visualizing [35].

4.2 Remote Monitoring and DAQ at CERN: EDUSAFE case study

In this chapter, a case study of a Remote Monitoring and Data Acquisition System used for assistance of personnel in extreme environmental conditions such as radioactive environments, is presented.

EDUSAFE European project [36], is a 4-year Marie Curie ITN program involving 15 European academic and industrial institutions, providing training for 10 Early Stage Researchers and 2 Experienced Researchers and funded by FP7-2012. The project focuses on research into the use of Augmented Reality (AR) during planned and emergency maintenance in extreme environments such as nuclear installations, space, deep sea and other harsh industrial scenarios.

Advanced AR technologies for a personnel safety system platform, including features, methods and tools, are used in a way that remote monitoring of the personnel's activity and health status can be achieved. Previous technology was not acceptable because of significant time-lag in communication and data transmission, missing multi-input interfaces, and simultaneous supervision of multiple workers who are working in the extreme environment. The aim is to technically advance and combine several technologies and integrate them as integral part of a personnel safety system to improve safety, maintain availability, reduce errors and decrease the time needed for scheduled or sudden interventions.

The research challenges lie in the development of real-time (time-lags less than human interaction speed) data-transmission, instantaneous analysis of data coming from different inputs (vision, sound, touch, buttons), interaction with multiple on-site users, complex interfaces, portability and the goal of developing a wearable prototype. The goal is to provide the technicians and the engineers with visual instructions through an Augmented Reality head mounted display (HMD) in order to reduce their stress, and to monitor their health status while they perform maintenance tasks in this complex and radioactive environment of ATLAS. The LHC at CERN and its existing Personnel Safety System, requirements and protocols are used as a test and demonstration platform.

The EDUSAFE first prototype was consisted of the following independent systems also shown in Figure 17:

- The Supervision System placed on the helmet of the worker (Wireless Personnel Safety System or WPSS).
- The Augmented Reality (AR) Prototype, a hand-held device including a server for computer vision algorithms.
- A Gamma camera is a standalone device which connects wirelessly to the AR prototype.

Figure 17. The EDUSAFE first prototype

The final goal is to develop AR prototype glasses for the user, to provide instant information on the status of the user's health, guidance through complex tasks and most importantly protect the person from exposed radiation by showing radiation hot spots in the real environment. For a desired AR visualization where display contents must be visually fused in the real world from a user's viewpoint correctly (see Figure 18), HMDs need to be mounted stably on the user's head.

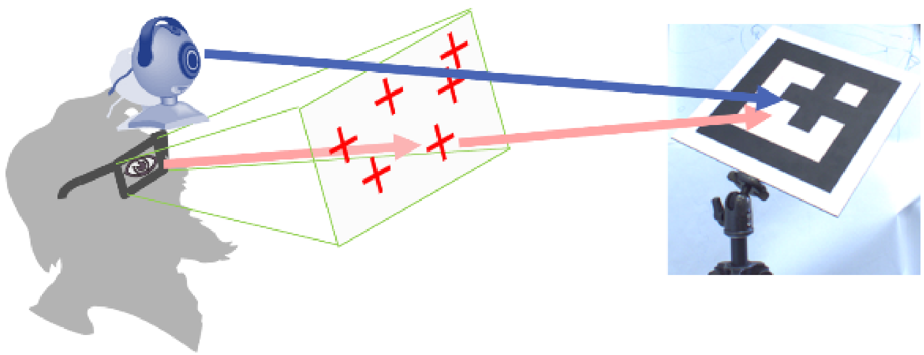

Figure 18. Schematic diagram of AR registration with an HMD.

Figure 19 shows a prototype display system. An optical see-through HMD from Vuzix, STAR 1200XL-D, which has a 1280x720 image resolution with a stereo support was employed. The display has a controller box which can take video signals via HDMI connection.

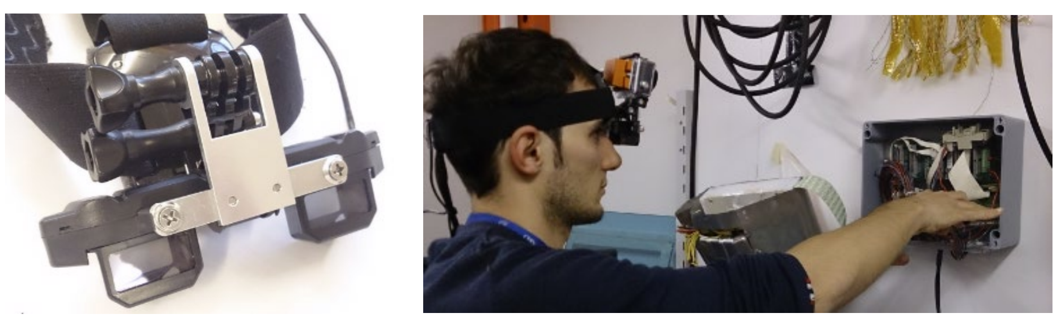

Figure 19. The HMD system. (left) The display part without the tracking component. (right) the system worn by a user with a tracking camera.

[37].

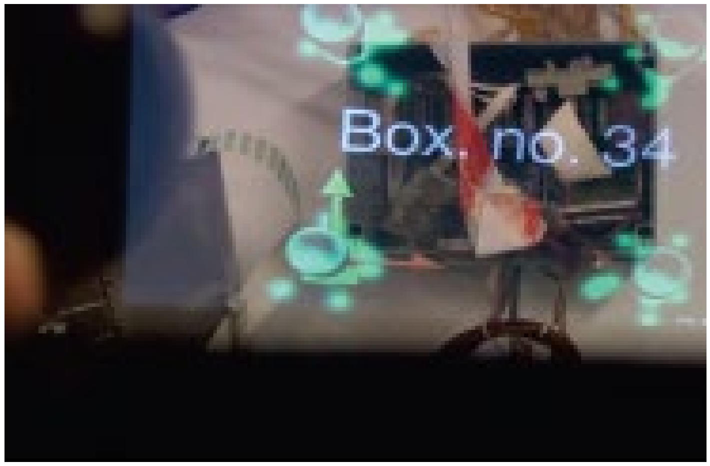

Figure 20. Augmented Reality content through the HMD.

The main tasks of these glasses are 2D object detection/3D pose determination, Pose filtering/interpolation (Sensor Fusion), visualization of the information from different sources, environmental and health monitoring data processing and wireless communication with control and data acquisition [37]. The user can see the steps he has to follow through the AR glasses as seen in Figure 20. This guidance helps the user to reduce his anxiety during the perform of the various work tasks and avoids possible human errors.

Environmental sensing acquisition is accomplished by Oxygen, Carbon dioxide, Humidity and Gamma dose radiation sensors. All these sensors are connected to the same microcontroller. The calibrated measurements are stored in the registered database. A list of health sensors is operating in this group. Some of these are body temperature, ECG, blood pressure, Oxygen saturation, Accelerometer, Air flow monitoring. The sensors interface is implemented with the MSP430F1611 microcontroller by Texas Instruments. The various sensors are connected to the microcontroller to either its analog or digital inputs. The main responsibility of the Sensing uController is reading and sampling the measurements from the sensors and forwarding them to the main processing group through UART by the Modbus Protocol [37].

Chapter 5

5 Control System and Data Acquisition System

In this chapter, the development and the optimization of the Control System (CS) and the Data Acquisition (DAQ) System for acquiring various bio-signals of the personnel, will be presented. This research was mainly carried out within the EDUSAFE research project. A Remote Monitoring System was also developed to supervise and communicate with the personnel in ATLAS cavern for assistance to maintenance activities or sudden interventions.

5.1 System development requirements

Real-time communication

Rich Internet Application (RIA) technologies provide richer, faster and more interactive experiences by updating data without reloading the entire page. AJAX (Asynchronous JavaScript and XML) is one of the most popular RIA technologies [26]. The software of the Control System and the Data Acquisition system is based on the ATLAS Personnel Visualizer System (APVS) code available on GitHub. This Software is written in Java programming language and uses the Google Web Toolkit (GWT) framework [38], which is an open source Java Software Development framework, in order to develop and maintain the complex Web application (front-end development) needed in EDUSAFE project. GWT was chosen because it is a powerful tool that facilitates the development of our complex application, providing a mechanism that simplifies the communication from our web application to our web server. GWT takes a strong approach to OO (Object Oriented) architecture, hence proper software architecture (as applied in java, since GWT is Java-based) can be applied in GWT as well. This offers a lot of potential for maintaining and scaling up this application.

In a classic web application model, users on the browser (client) side trigger an HTTP request to the server side. The server-side deals with the request and returns the updated page to the user. In order to achieve a real time client-server framework in our architecture we need a more complex model than the classic web application model. For that reason, we are using the Atmosphere Framework [39], which is the most popular asynchronous application development framework for Java. The Atmosphere Framework provides the features required to build a massive scalable and real time asynchronous application using transports like Web Socket, Server Sent Events and traditional Ajax Techniques. GWT cross-compiles this client-side code into optimized JavaScript (browser-compliant JavaScript and HTML) that automatically works across all major browsers. The architecture of the system that was designed to allow real-time communication, will be described in the following sections.

5.2 Mobile Personal Supervision System

In the framework of the EDUSAFE research a Mobile Personnel Supervision System (MPSS) module was developed by Novocaptis [40] and AUTH [41], to serve an automated, remote operation of a large number of sensing and mobile-autonomic application domains.

The MPSS module is the host for the communication module, the safety sensors and the local intelligence for data-video treatment such as Augmented Reality features. It is an essential node for propagating the data from the cameras and sensors to the control system (providing supervision services to administrators and users) and vice versa from the control system to the Head Mounted Displays (HMDs). All these services are managed by a Control System (CS) and Data Acquisition System (DAQ).

The MPSS is a wearable mobile safety device (see Figure 23_1), which combines several technologies and integrates them in a personnel safety system where data are received from different input sources (gamma camera, vision cameras, and sensors) interacting with multiple on-site users. A Raspberry Pi 2 single board computer and a dedicated camera were used for the implementation of the MPSS. It is used for wireless transmission of data for supervision purposes and it is mounted on the safety helmet that all the personnel use in the ATLAS cavern. The data transmitted from the MPSS are video, bidirectional audio and radiation sensor data received from the dosimeter that will be explained in this chapter. The video transmission latency that is achieved is under 214ms, transmitting 30fps of 640x480 pixels resolution.

5.3 Sensor Board

Sensor and power Printed Circuit Boards (PCBs), based on electronic miniaturization process, were developed by Prisma Electronics SA [42], to acquire the various sensor data, both biological and environmental. We can refer to the developed PCB's from now on as Personnel Transmitting Unit or PTU, since it collects sensor data for the personnel that uses this device. All the sensor values are acquired through the DAQ System described later on in this chapter. The DAQ System accepts various types of sensors such as Radiation, Temperature, Humidity, O_2, CO_2, Barometric Pressure, Body Temperature and Accelerometer. These sensor data are acquired from the developed sensor board through the DAQ Server in JSON messages over Wi-Fi and are properly persisted into the Database. The JSON messages that are sent to the PTU should be formatted to be parsed in the correct way. The sampling rate of each JSON message that the server receives is 5s. Moreover, in case a measurement exceeds a certain threshold, the control system is activated, and an alarm is created. These thresholds are set as shown in Table1.

The processing unit that was used manages all the complex tasks such as mobile sensing and data transfer. It is a Gumstix based OMAP30 Digital Video Processor in ARM Cortex-A9 architecture from Texas Instruments. This is a good combination of processing power and power consumption in the smallest and lightest board that is currently available, ideally for wearable systems with high processing requirements. It supports Linux Ubuntu/android operating system.

Gumstix COMs are fully functional Linux computers with a wide range of open-source software applications available. The Gumstix board has 2x70 pins connector by which it can be easily mounted to the carrier PCB. The unit can run with 5.0 Volts with less than 1 Ampere. From electronic miniaturization, modularity, portability, and low power point of view the selection of this Gumstix based processor is a good choice for the modular design.

Sensor	**Unit**	**Up Threshold**	**Down Threshold**
CO_2	ppm	600.0	100.0
O_2	%	22.0	18.5
Temperature	deg;C	40.0	10.0
Body Temperature	deg;C	39.0	35.0
Barometric Pressure	bar	1.2	0.8
Humidity	%	90.0	10.0
Dose Accumulation	mSv	0.0	0.0
Dose Rate	mSv/h	0.0075	-1.0E-6

Table 5. Sensor thresholds

The design of sensor board is based on the MSP430F1611IPM Microcontroller from Texas Instruments. It is a low power processor with many analog and digital input and output options. The main features of this microcontroller are:

a. The Texas Instruments MSP430 family of ultralow power microcontroller consisted of several devices featuring different sets of peripherals targeted for various applications.
b. The architecture, combined with five low power modes, optimized to achieve extended battery life in portable measurement applications.

c. The device features a powerful 16-bit RISC CPU, 16-bit registers, and constant generators that attribute to maximum code efficiency.

The Sensing controller operating system is based on Intelligent Sensors Operating System ISOS and written in C. It was developed by Prisma Electronics SA to support various sensor interfaces and provides a reliable platform to easily develop new applications. The SW development environment of microcontroller unit is based on Code Composer Essential V2. The various sensors are connected to the microcontroller to either its analog or digital inputs. Different sensing parameters are accommodating by different types of sensors. There are two types of signals, analog and digital. Digital signal is directly used for Microcontroller processor where Analog to Digital Converter (ADC) is used from the analog sensing device. Different bus communication (I2C, SPI) is used as a peripheral device of these sensors whereas Xbee based wireless module can be used for radiation sensors. The microcontroller unit can be used for various sensor devices. It can be separated by many sensor blocks 1, 2, 3…. N. Some of these blocks can be: gas sensors (O_2, CO_2, CO, H2S etc.) block, environmental sensors (humidity, temperature, air pressure, etc.) block, human sensors (body temperature, heartbeat monitoring, etc.) block, worker position awareness block etc.

The main responsibility of the Sensing controller software is reading and sampling the measurements from the sensors and forwarding them to the main processor through UART0 using the Modbus Protocol. The sensor data forwarder application is composed of a program written in C (main program). The UART-to-TCP forwarding utility is configured to run in the Advantech board and implements the communication with the Sensing controller (MODBUS over UART) and the server-side program (JSON over TCP/IP). The main program performs the actual communication forwarding between the server-side program and the Sensing controller. Specifically, it requests sensor measurement values from the Sensing controller via UART. This sampling happens whenever the respective sensor timer, held by the main program, expires. The main program polls the serial port and when a response with a measurement or event is received, it is forwarded to the server-side program over an established TCP connection, to which the main program connects as a client.

The power supply module is responsible for providing the power to the PTU. The input voltage for sensor board is +5.0 V and +3.3V with approximately 0.5 Amp whereas the input voltage for the main processor board is 5.0 volt with less than 1 amp. In total, the system consumes approximately 5 watts. The charging circuit which has 12-volt input and will charge the Li-I battery with 8.7 volts. The capacity of the battery is 3000 mAH by which system can run approximately 5 to 6 hours. The charging circuit will then provide the voltage to the converter circuit to reach different voltage levels for example +5.0-volt, +3.3 volts. The dimension of the PCB is 60x36 mm (see Fig. 21).

Figure 21. PCB of the power module sensor board.

5.4 Gamma Camera

Localization of radioactive sources is a major issue for radiological safety of operators in radiation facilities. For this issue, a gamma radiation imaging system capable of superimposing a gamma image with a visible image is a powerful technique, especially for 3D mapping and radioactive source localization, as desired for the ATLAS environment. With this system, “fast” dose rate algorithms are needed, using the latest technologies of ray tracing and radiation shielding codes to calculate the volumetric dose rates over the entire ATLAS environment, along with the 3D localization of all hotspots and dose rates of concern. A gamma camera was designed, fabricated and successfully tested in the LHC Experiments environment (see Figure 23_3). This camera superimposes gamma radiation detected on top of the physical environment. It is lightweight (~2.5kg), easy to use and remotely controllable. The platform is based on a prototype system developed by the CEA in 2011, along with a CANBERRA prototype developed in 2012. The system utilizes a single cable for power and communication and has more than five (5) hours of autonomy if the internal battery pack is used. Specialized software has been developed with the iPIX system to process the raw data from the iPIX head unit, delivering an all in one solution for the measurement and localization of hotspots.

5.5 Dosimeter

Exposure to ionizing radiation (gamma, beta, and particle radiation) takes place while working at a particle accelerator and in the associated experimental facilities. Legal dose limits assure the safety of personnel working under these conditions. The dose received by individuals working with ionizing radiation at CERN is monitored with personal dosimeters. Every person working at CERN in Radiation Areas or with sources of ionizing radiation must wear a CERN dosimeter [24]. The CERN dosimeter registers the personal dose from sources of ionizing radiation around particle accelerators. It combines an active detector for gamma and beta radiation based on the Direct-Ion Storage (DIS) technology and a passive detector for quantifying neutron doses [24].

The gamma/beta dose registered by a CERN dosimeter can be read out as frequently as deemed necessary, but it must be read at least once per month on one of the approximately 50 reader stations which are installed CERN-wide. The monitoring period for the neutron dosimeter is, in principle, one year. It must then be returned to the supplier for evaluation [24].

For work in Controlled Radiation Areas, where the radiological risk and the dose rate are above 50 μSv/h, the additional use of an operational dosimeter is required. CERN provides all staff who may work in Limited Stay Radiation Areas or High Radiation Areas with a system for active dosimetry with an alarm, in the form of a dosimeter, model DMC-2000 from MPG instruments [24]. This dosimeter is used to acquire the personal dose for every user of our system.

5.6 Control System and Data Acquisition System Architecture

The basic goal of the CS and DAQ System, is to design and optimize a modular architecture, in order to firstly acquire data from safety sensors, gamma camera (device to image gamma radiation and provide localization of radioactive sources), as well as other vision cameras, secondly store them in a reliable Data Base, and thirdly display them in a User Interface to enable remote supervision of workers in the ATLAS cavern, for security purposes. The CS and DAQ architecture must be designed to accept various types of sensors for testing in the ATLAS environment, hence the adaptation of this service-based application is a crucial goal that needs to be defined and fulfilled and will be thoroughly analyzed in chapter 6.

The CS and DAQ system that is developed and optimized, provides the following services:

a. Sensor Data Acquisition
b. Video Streams Acquisition
c. Gamma camera Data Acquisition
d. Data-Base storage
e. User Interface development for remote supervision of workers in the ATLAS cavern for security purposes.

The Architecture design of the Control System and Data Acquisition System of EDUSAFE is graphically shown in Figure 22.

The Control System (CS) and the Data Acquisition (DAQ) System that was developed and optimized, is based on the ATLAS Personnel Visualizer System (APVS) [43] of the Wireless Personal Supervision System (WPSS) research project [44] of CERN. This Control system is a Google Web Toolkit (GWT) [45] application that contains the Java-server and the web-client responsible for the Graphical User Interface of EDUSAFE.

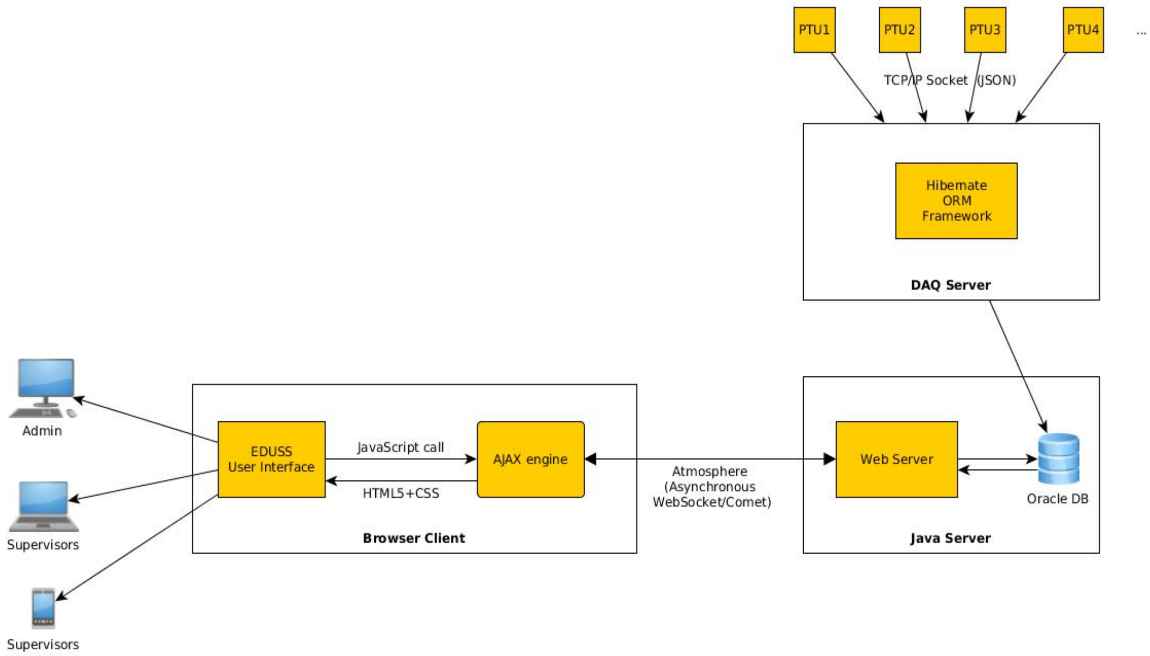

Figure 22. Architecture Design of the Control System and Data Acquisition System

This interface is completed using web development tools and it is named EDUSAFE GUI or EDUSS. Through this interface the supervisors can monitor with video streaming the activities of the personnel in the ATLAS cavern and communicate with them to guide them when needed. The functionalities of this interface are described in the next subchapter.

In this design an AJAX engine runs within the browser on the client-side, communicates with the server, performs interactions, and displays requested information in the browser. If the AJAX engine requires more data, it sends requests asynchronously to the server in the background to retrieve updated data (and potentially additional code) without interfering with the user's interaction with the application. Instead of loading a webpage, at the start of the session, the browser loads an Ajax engine, written in JavaScript. This engine is responsible for both rendering the interface the user sees and communicating with the server on the user's behalf. The Ajax engine allows the user's interaction with the application to happen asynchronously, independent of communication with the server. This process enables the real time server-client communication, a requirement described in the beginning of this chapter.

The final integrated wearable system was tested with the DAQ and CS. The following Figure 23 provides an overview of the Remote Health Monitoring and DAQ System processes and functionalities. The DAQ System can acquire various types of data from subsystems:

1. The Mobile Personal Supervision System (MPSS), video/audio/radiation data.
2. The sensor board, environmental and biological parameters.
3. The gamma camera, radiation hot spot localization images.
4. The Operational Dosimeter (DMC) used for the radiation measurements is also shown in 16.

Once the data are acquired they are shown on the developed EDUSS GUI and stored in an Oracle Database for off-line analysis.

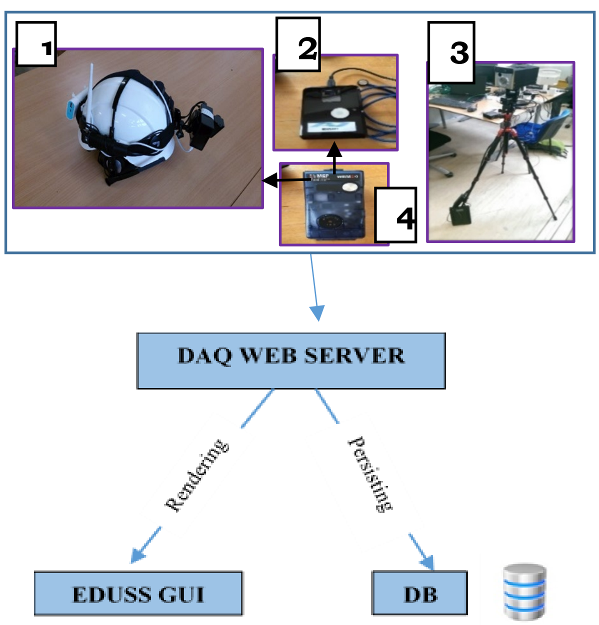

Figure 23. DAQ System sources: (1) Mobile Personal Supervision System (MPSS) (2) Sensor Board (PTU) (3) Gamma radiation camera (4) Operational Dosimeter (DMC) measuring radiation data.

5.7 Remote Monitoring System (RMS)

In this chapter we present the Remote Monitoring System (RMS) that was developed, which we will refer to as EDUSAFE Supervision System User Interface (EDUSS GUI). It is developed to monitor remotely the various worker sessions and guide the personnel if needed through the communication of the supervisor and the workers in the ATLAS cavern. The EDUSS GUI enables the health monitoring of each worker by monitoring the various sensor data like gamma radiation measurements, that correspond to the dosimeter of a specific worker. For the development of the User Interface we are using HTML5. The Styling is done in CSS (Cascading Styling Sheets). Apache Maven [46], is also used as a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information.

Through the EDUSS GUI the supervisor can:

1. Create multiple sessions with the workers in the ATLAS cavern.
2. Monitor every worker with video streaming.
3. Communicate with audio connection.
4. Monitor the sensor values, such as gamma radiation measurements, that correspond to the dosimeter of a specific worker.
5. Access the graphical interface of the gamma camera that is set in the cavern.
6. Generate and save plots of the complete session for each sensor type resulting in faster offline analysis of the acquired data.
7. Additionally, the supervisor can dynamically generate plots using GWT Highcharts [47], which is a comprehensive API within the GWT application.

The video acquisition from the prototype of the supervision system is successfully tested in order to firstly acquire the video from the helmet camera and then display the stream in the “Camera” interface of the EDUSS GUI. The results are properly shown in EDUSS supervision GUI in Figure 24.

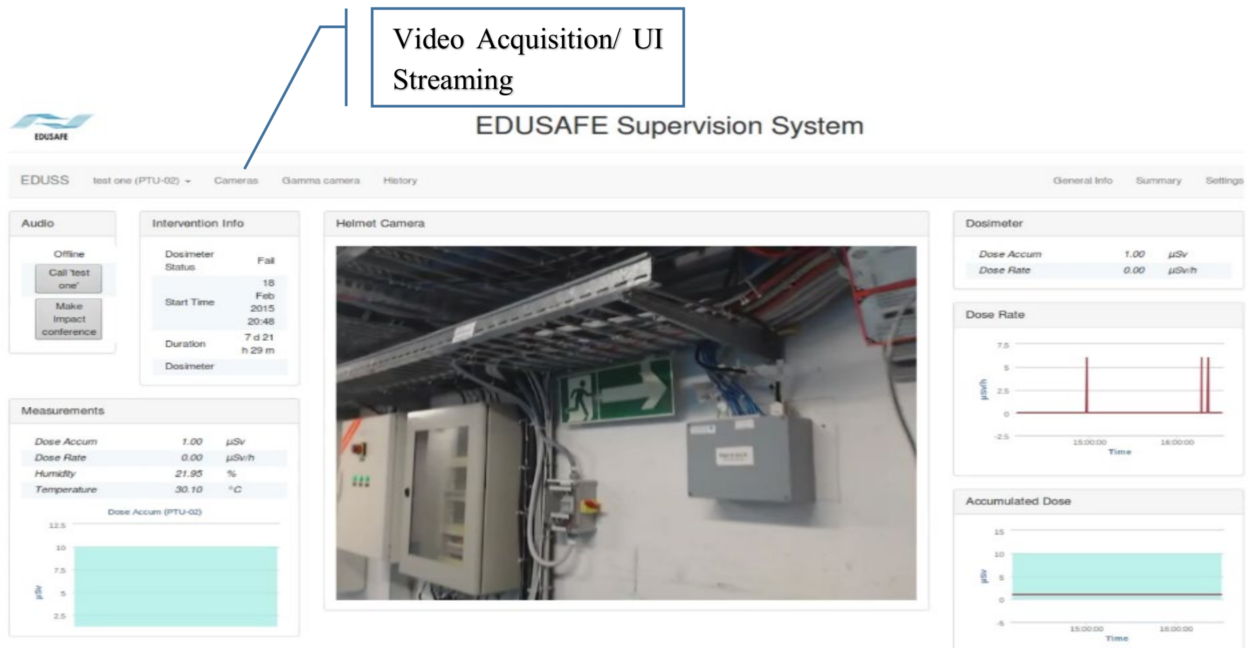

Figure 24. EDUSS UI video streaming.

The Dose Rate and Dose Accumulation measurements are acquired from the prototype in JSON format, are stored in the Oracle Database, and are displayed in the EDUSS UI. The rest of the sensor data are also acquired from the sensor board with JSON formatted messages and are stored in the Oracle DB for offline use. The gamma camera User Interface described in section 8 is integrated in the EDUSS UI providing the current view of the gamma ray measurements on the field.

As mentioned, the supervisor can dynamically generate plots using GWT Highcharts, a comprehensive API within the GWT application. Highcharts is a charting library written in pure JavaScript, offering intuitive, interactive charts to this web application. Supports line, spline, area, area spline, column, bar, pie and scatter chart types. Since Highcharts is written in JS we are using a java wrapper from the moxie group to deliver the jar file. It is a freely available open source library that provides an elegant and feature complete approach for including Highcharts and Highstock visualizations within apvs GWT application using pure Java code, including GWT widget libraries, such as SmartGWT or Ext GWT (see Figure 25).

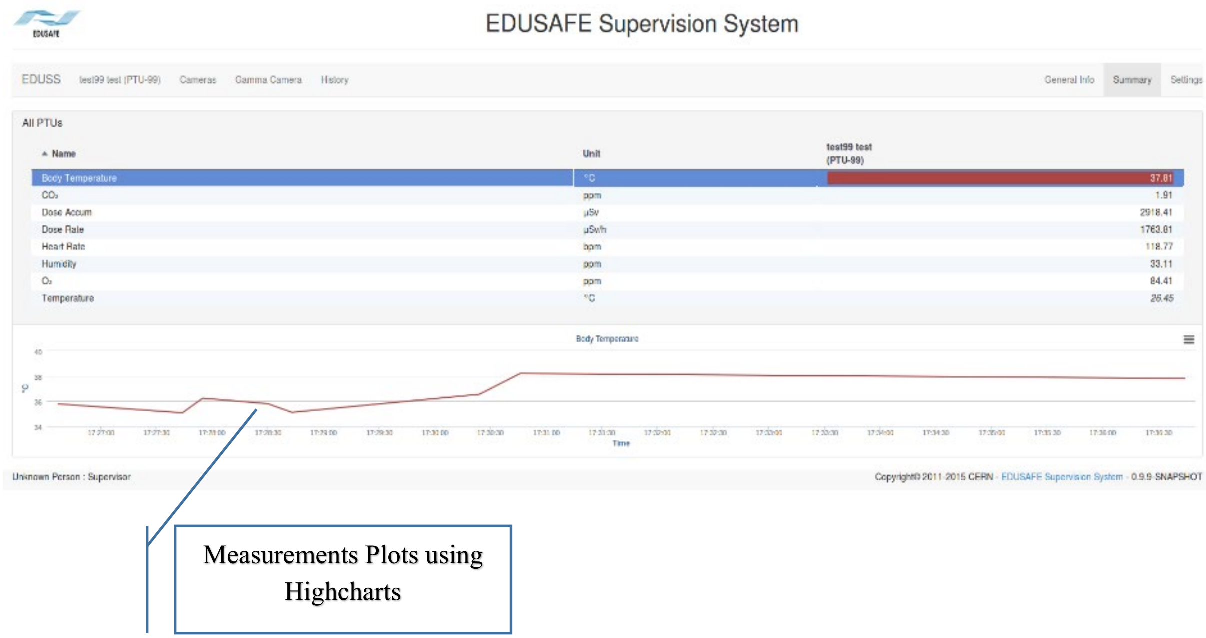

Figure 25. Highcharts plots for data measurements.

GWT interfaces, Java, JavaScript, HTML5, CSS, Highcharts along with several web-developer tools such as Bootstrap were used to create a lighter interface for both Desktop and iPad interfaces. The Dose Rate and Dose Accum measurements are acquired from the prototype in JSON format, are stored in the Oracle Database, and are displayed in the EDUSS UI. The rest of the sensor data are also acquired from the sensor board with JSON formatted messages and are stored in the Oracle DB for offline use. The gamma camera User Interface described in section 8 is integrated in the EDUSS UI providing the current view of the gamma ray measurements on the field.

The EDUSAFE Supervision System (EDUSS) GUI is the web client application of apvs, it is written in Java, converted in JS, and it connects to the java-server via Atmosphere [39], the Asynchronous Web Socket/Comet Framework. The user interface is providing the supervisor with the following information among other:

1. Sensor Data from the sensor board manufactured by Prisma: O_2, CO_2, Barometric pressure, Accumulation Dose, Dose Rate, Temperature.

2. Gamma Ray Data acquired from the standalone gamma camera.

3. Video streams from the worker's view (hand-held or helmet camera while he works at the ATLAS cavern, for supervision purposes).

4. Plots of the measurements that are received from PTUs.

Detailed instructions on how to run and use the EDUSS GUI are given in the next chapter.

5.7.1 Running the SW for the EDUSS GUI

The source code for the GUI is stored in the apvs code in GitHub (https://github.com/CERN/apvs/tree/edusafe). Follow the next steps to run the SW:

1. Execute the following command to run the SW for the EDUSS Graphical User Interface that is developed for the supervision purposes.

   cd apvs

   /run-prod.sh

   The contents of the shell script that will be executed are:

```
#!/bin/sh
    export APVSpwd=yourPassword
    java -Xmx1024M  -Dlogback.configurationFile=logback.xml -jar apvs-
    jetty/target/apvs-jetty.war
```

2. Once you notice the following output in the terminal "APVS started on http://localhost:8095/index.html" as shown in the following Figure 26, you can access the EDUSS GUI with any browser by typing the following address: http://127.0.01:8095/index.html (start page in Figure 27).

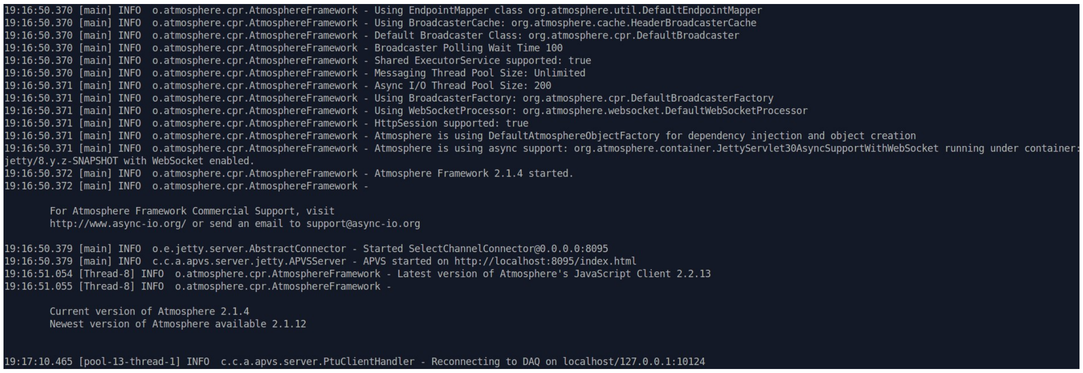

Figure 26. Building the GUI.

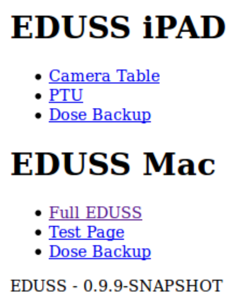

Figure 27. Start page of the EDUSS GUI.

Choose Full EDUSS link to access the EDUSS GUI for desktop displays. In Figure 28, there is an example of the EDUSS GUI:

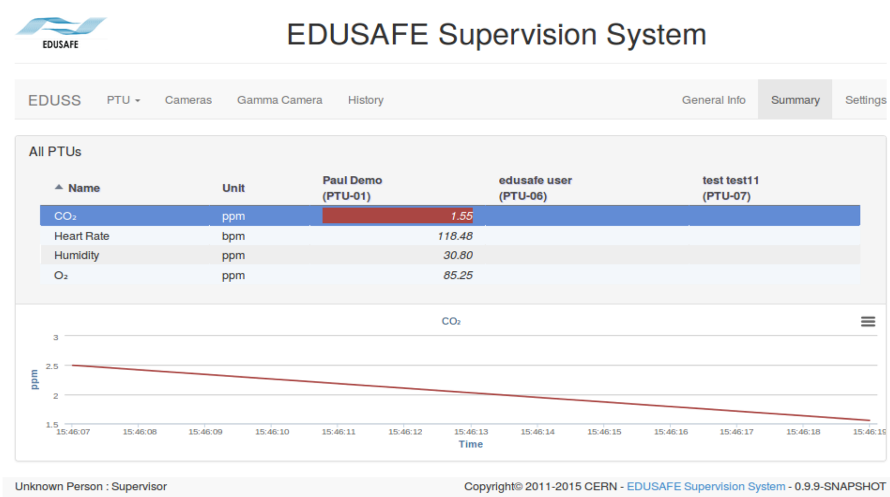

Figure 28. EDUSS GUI

5.7.2 How to run the CS and DAQ Server

1. Follow the step by step installation instructions for the CS and DAQ System SW from the Technical Manual on the EDUSAFE Git repository:

   ESR5/ documentation / Technical Manual

2. Configure the CS and DAQ System. Set the DB connection settings: Go to the daq-server directory

   cd apvs / apvs –daq

   3. Edit the hibernate.cfg.xml file as described in the Technical Manual with the proper username and password for the Database connection.
   4. Initiate the daq-server:

      cd apvs/apvs-daq-server

      ./start.sh

An optional check that JSON messages are properly received from the daq-server can be achieved with the following command:

telnet localhost 10123

The output will be similar to the following Figure 29.

Figure 29. JSON formatted messages example.

Before receiving any sensor data, the devices must be setup in the Oracle DB. There are two ways to setup a device in the DB, either through the Oracle SQL Developer interface either directly through the EDUSS GUI.

5.7.3 How to set up a device through Oracle SQL Developer:

In order to set up a device through the SQL Developer interface follow the next steps:

a. Open SQL Developer and go to the connection created for the DB. In this case it is called edusafe-superdaq, see Figure 30.

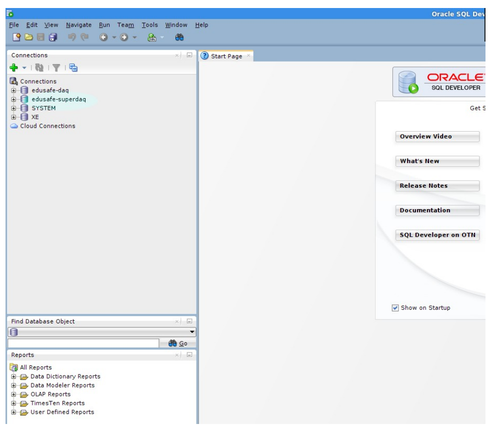

Figure 30. Oracle SQL Developer connections.

b. Under the user (edusafe user) select" Tables".

c. Under" Tables" select TBL DEVICES. When you open the" Data" tab you will see the table that contains all the devices that are already saved in the DB (see example in the Figure 31).

d. Add a new device by clicking the" plus" icon or with the short key" Ctrl+I" and insert the following information: ID, NAME, IP, DSCR, MAC ADDR and the HOST NAME of the device.

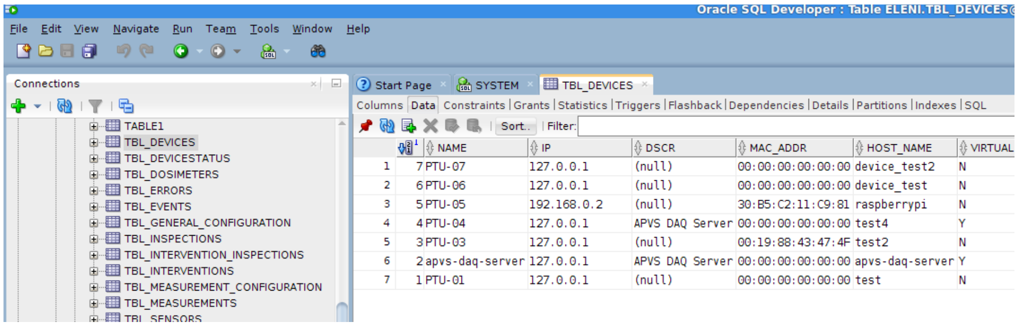

Figure 31. Setting up the devices in Oracle SQL Developer

5.7.4 Set up a new device through EDUSS GUI

In this chapter instructions on how to setup a new device through the GUI are given.

Sensor data acquisition and video acquisition can start once a device is set up. In order to set up a new device, for example a personal transmitting unit (PTU) which acquires data from various environmental and biological measurements, follow the next steps:

1. Choose the" History" tab of the EDUSS GUI.
2. Choose the" New Device" tab on the bottom of the page.
3. Fill the needed information in the pop-up window that will appear as shown in Figure 32 and click" Ok".

The fields are:

(a) Device Name: the name of the device.

(b) IP: the IP of the device.

(c) MAC Address: Media Access Control (if this option is not used type 00:00:00: 00:00:00).

(d) Host Name: the host name you will give to the device.

(e) Description: Any comments you would like to note (optional).

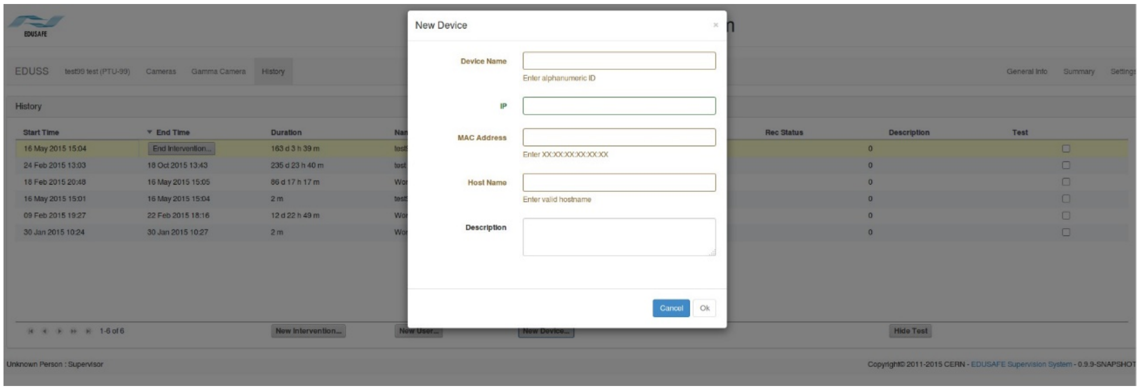

Figure 32. Setting up a new device through EDUSS GUI.

5.7.5 Set up a new user through EDUSS

Follow the next steps in order to set up a new user, for example the name of the worker that will be supervised and guided:

1. Choose the" History" tab of the EDUSS GUI.
2. Choose the" New User" tab on the bottom of the page.
3. Fill the needed information in the pop-up window that will appear as shown in Figure 33 and click" Ok".

The fields are:

(a) First Name

(b) Last Name

(c) CERN ID (optional)

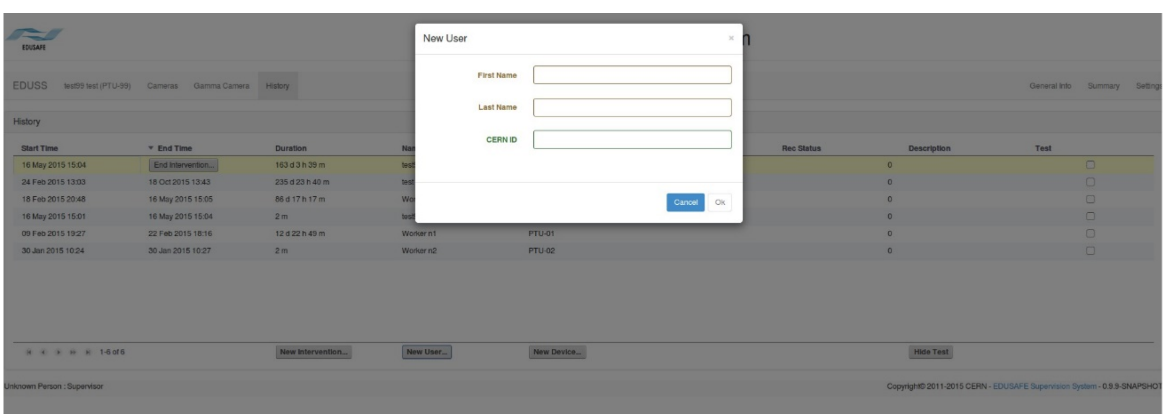

Figure 33. Setting up a new user through EDUSS GUI.

5.7.6 Set up a new intervention through EDUSS

Once the devices and the users as set up you can initialize a new intervention by allocating the device to its user as shown in Figure 34 with the next steps:

1. Choose the" History" tab of the EDUSS GUI.
2. Choose the" New Intervention" tab on the bottom of the page.
3. Fill the needed information in the pop-up window that will appear as shown in Figure 34 and click" Ok". The fields are:

(a) User: A list of option should appear if you have successfully set up the users, scroll to choose the one you need.

(b) Device: A list of option should appear if you have successfully set up the devices, scroll to choose the one you need.

(c) Impact Number (optional)

(d) Description (optional) There is an additional option of selecting the box" Test Intervention" in case you want to check the intervention before initializing it. Click" Ok" to initialize the intervention.

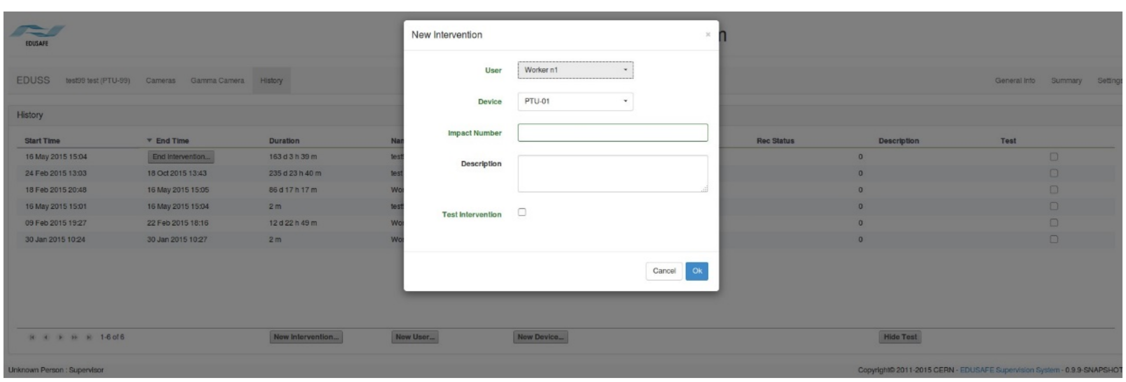

Figure 34.Setting up a new intervention through EDUSS GUI.

5.7.7 How to end an intervention through EDUSS

To end an intervention that is created go to the" History" tab and click" End Intervention" at the" End Time" column of the displayed table as shown in Figure 35.

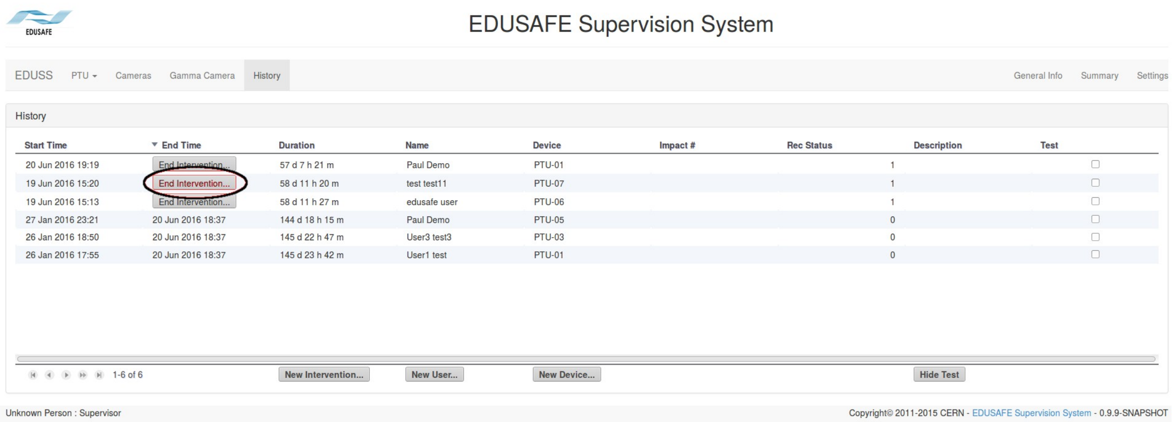

Figure 35.End an intervention through EDUSS GUI.

5.7.8 PTU view

Once you have initialized a new intervention with a user and a device, you will see a new tab created between the" EDUSS tab" and the" Cameras" tab. This new tab will have a scroll option in case you have set up more than one active interventions. Choose the one you want to supervise. In Figure 36 you will see an example of the new interface that will appear. This page is the main" PTU-view" of the current intervention. It is divided in 7 parts:

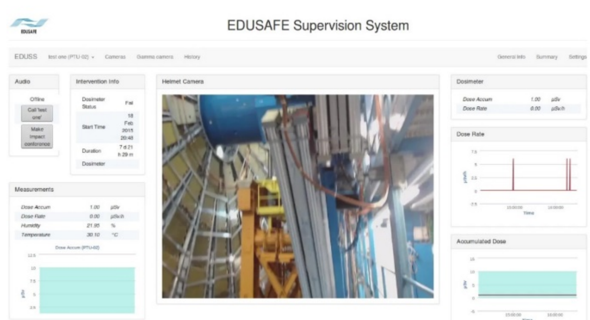

Figure 36.Main PTU view for each intervention.

1) **Audio**: the audio communication with the worker.

2) **Intervention info**: Provides information about the following:

(a) Dosimeter Status

(b) Start Time

(c) Duration

(d) Dosimeter

3) **Measurements**: this table is updated every 5 seconds or in the sending- receiving rate of the sensor data from the PTU. There is an arrow decorator to indicate the increase, decrease, or stability of the value that is measured, in order to provide a visually faster overview of the worker status. There is an option of creating instantly a plot just by clicking the sensor you are interested in. These plots are created using Highcharts SW as explained in the previous chapter and they are dynamically formed according to the changes in the values from the start of the intervention initialization until the ending point.

4) **Helmet Camera**: This is the central part of the page where the video acquisition from the camera of the MPSS is displayed. To understand how the video is transmitted to the server check the documentation of ESR3. The video acquisition settings are available in the" Settings" tab that is described in the next chapters.

5) **Dosimeter**: in this block the values of the Dose Rate and Dose Accumulation are separately displayed. These two measurements are separately displayed due to their big importance, especially in extreme environmental conditions such as the ATLAS cavern where radiation is crucial.

6) **Dose Rate** Plot box over time.

7) **Accumulated Dose** Plot box over time.

5.7.9 Cameras

In this tab all the ongoing video sessions are displayed. This interface is dynamically updated with the addition or deletion of a video communication, using Bootstrap grid system. It can support multiple video sessions simultaneously providing a broad overview to the supervisor.

5.7.10 Gamma Camera

The Gamma camera interface connects with the gamma camera that is set up in ATLAS cavern and provides the hot spot localization of the radiation.

5.7.11 History

History tab displays information about the history of the interventions (Start time, End Time. Duration, Name (of the user/worker), Device, Impact, Rec Status, Description, Test). This tab is very useful to the supervisor due to the bottom-button options where the supervisor can set up the name of the user (worker), the name of the transmitting sensor device and the initialization of each intervention. Additionally, there is the button option of" Hide Test" to choose whether you want to display the Test mode of an intervention. See the following Figure 37.

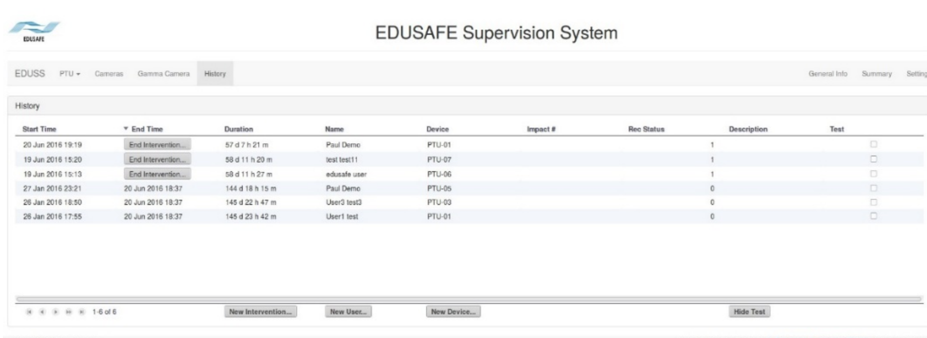

Figure 37.History of all the PTUs and the interventions.

5.7.12 General Info

In the" General Info" tab the supervisor can monitor the following information as shown in Figure 38:

a. Server Status
b. Audio Status
c. Video Status
d. DAQ Status
e. DB Connect Status
f. DB update Status

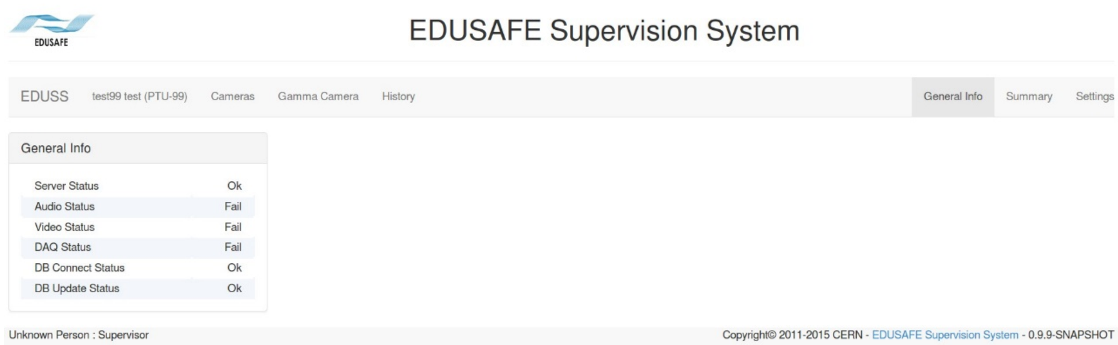

Figure 38.General Information.

5.7.13 Summary

This tab displays a summary of all the measurements for each active PTU device as shown in the following Figure 39:

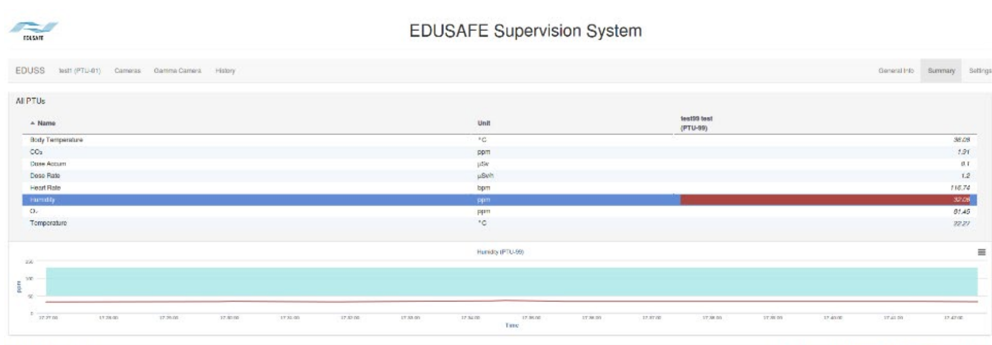

Figure 39.Summary of all PTUs.

By clicking a sensor type you can create instantly a plot underneath using Highcharts SW as shown in Figure 40. The supervisor can export the current plot in various types of files such as PNG, JPEG, pdf and svg. This option is very useful especially for offline meta-analysis of the sensor data over time.

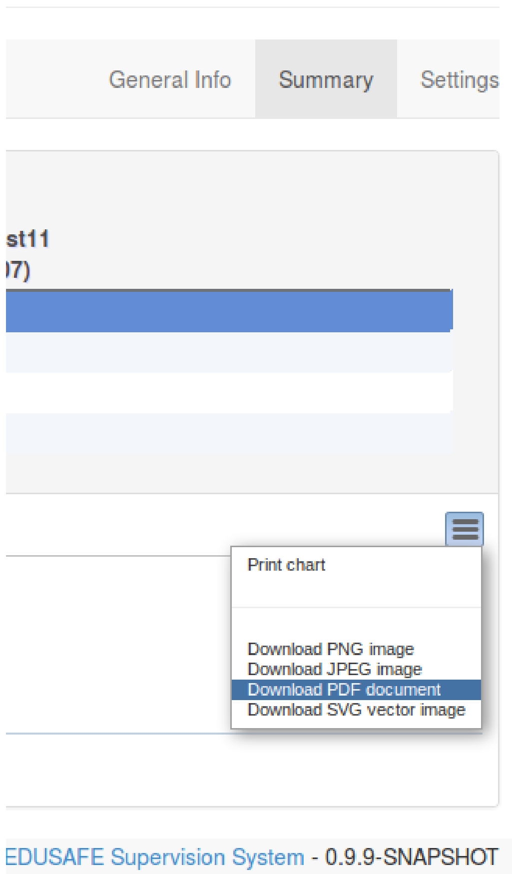

Figure 40.Exporting plots of sensor data for offline analysis.

5.7.14 Settings

This page is separated in 3 parts:

1) PTU Settings: set up here the URL of the helmet camera of the MPSS that is wireless transmitting the video.

2) Server Settings. The server settings that appear here are

configured in the APVS.Properties file in the apvs directory of the apvs code. See the screenshot provided as an example in Figure 41.

3) Audio Supervisor Settings for audio connection settings.

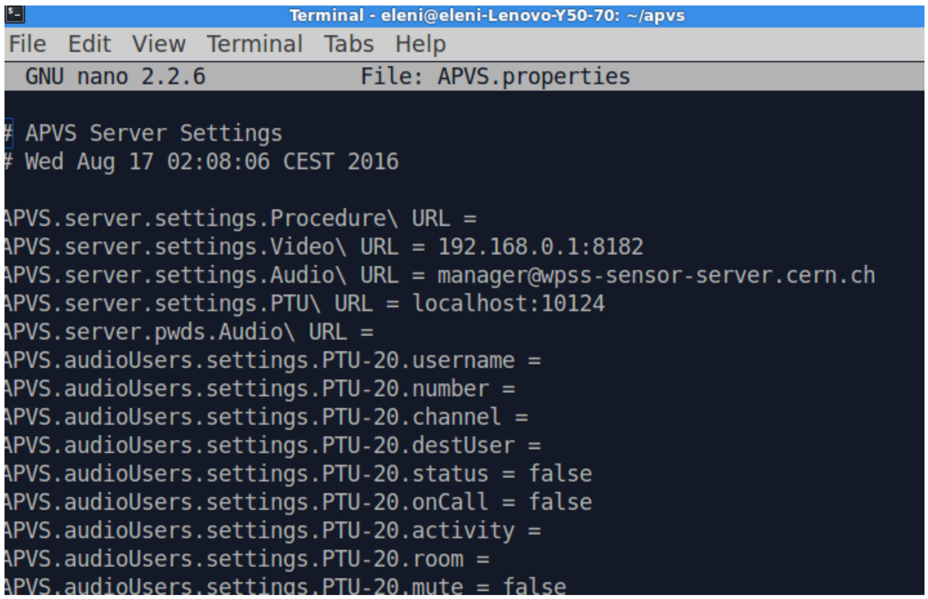

Figure 41.Example of the APVS.Properties configuration for server settings.

5.8 Proposed Safety System

We propose a Safety System consisted of the three sub-systems we described in this chapter: a Data Acquisition System (DAQ), a Control System (CS) and a Remote Monitoring System (RMS). These sub-systems are integrated to provide real-time guidance to the users and enhance human radioprotection. There are four external systems that interact with the Safety System, namely PTU, MPSS, Gamma Camera and AR System. These hardware sources were described in this chapter. The architecture design of the proposed safety systems and the interdependencies with the external systems are shown in the following Fig.41.

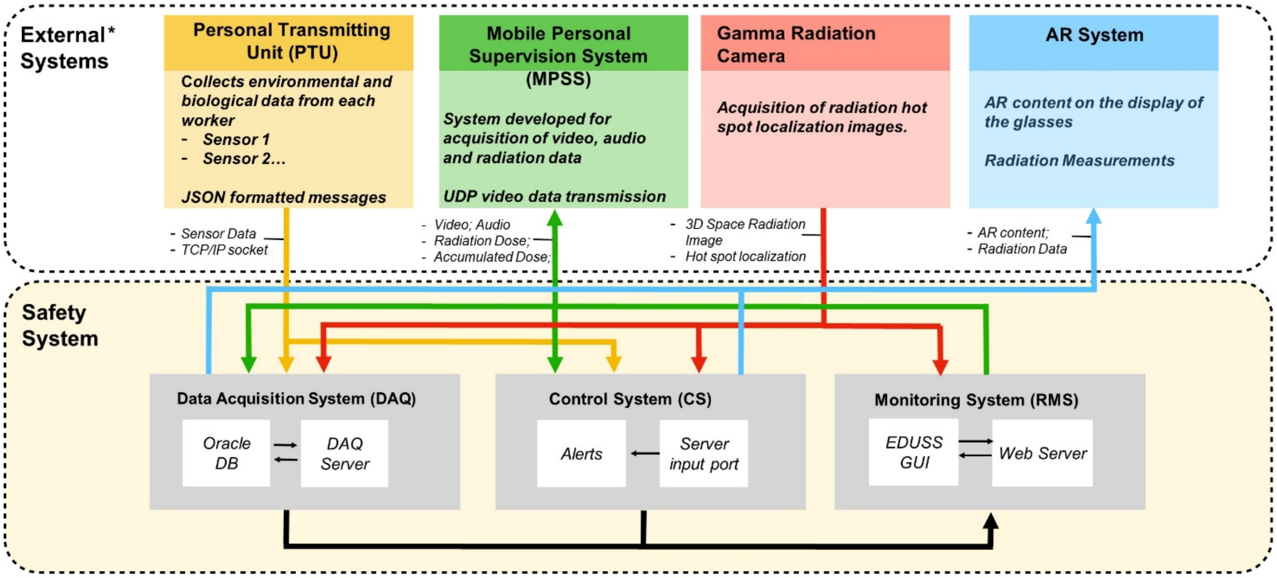

Figure 42. Proposed Safety System Architecture Design.

* the systems mentioned here were developed under the EDUSAFE project [36] and interact with the proposed Safety System.

A Filtering algorithm was also developed to filter all the data streams that the DAQ server is receiving and extract only the measurements that are related to radiation. Dose Rate and Dose Accumulation are mined through the json streams and forwarded to a dedicated filter port. The AR glasses connect to this filter port to receive the current radiation measurements and display them in real time.

Chapter 6

6 DAQ System Database

The data that are acquired from the various sources described in the previous subchapter are properly stored in a dedicate Oracle Database to enable the continuous monitoring of the personnel and the offline analysis. Once the JSON data are acquired from the DAQ Server, Hibernate [48], an Object Relational Mapping (ORM) tool, is used to map the java classes to the DB tables and the java datatypes to SQL datatypes. Hibernate then persists the acquired data in the Oracle Database (DB).

6.1.1 Relational Diagram

The relational diagram of the database is shown in the following Figure 43.

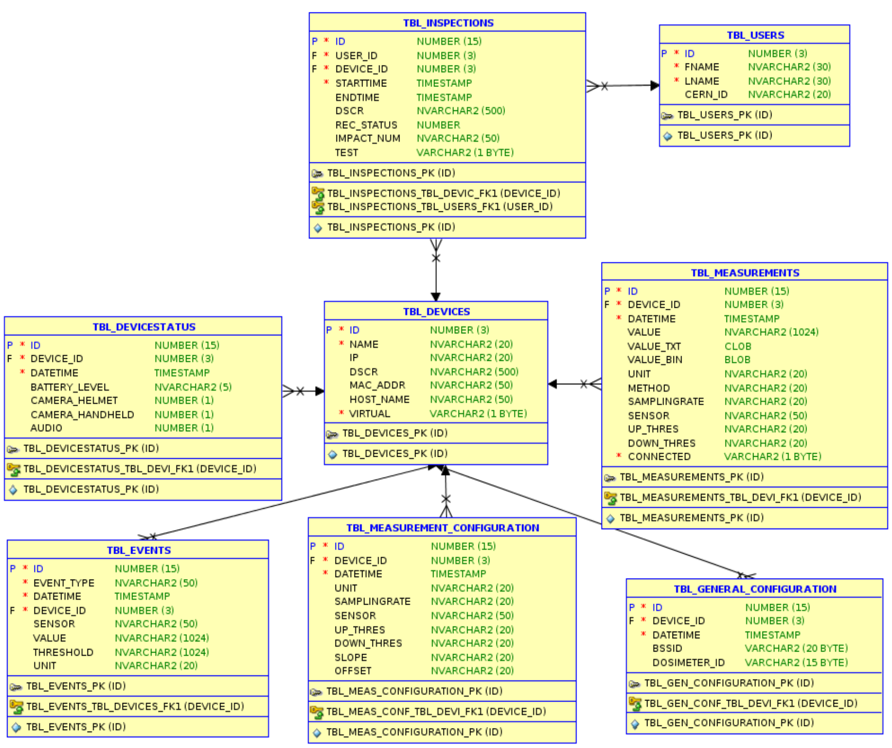

Figure 43. Relational Diagram of the DB

6.1.2 DB Tables

The tables that are used in the DB are described in this subchapter.

6.1.2.1 TBL_INSPECTIONS

Each session that occurs with a user consists an inspection that has a starting and ending time at a particular date, a description and a video recording status among other features. Every session corresponds to one user with its own ID and the device this user is assigned to. The inspections are stored in the TBL_INSPECTIONS of the DB with the following columns:

TBL_ INSPECTIONS		
COLUMN	**TYPE**	**DESCRIPTION**
ID (PK)	NUMBER (15,0)	The inspection ID
USER_ID (FK)	NUMBER (3,0)	The ID of the user to who the device was assigned
DEVICE_ID (FK)	NUMBER (3,0)	The ID of the assigned device
STARTTIME	TIMESTAMP (6)	The date and time when the inspection started
ENDTIME	TIMESTAMP (6)	The date and time when the inspection ended
DSCR	NVARCHAR2(500)	Description of the inspection
REC_STATUS	NUMBER	The video recording status
IMPACT_NUM	NVARCHAR2(50)	The impact number
TEST	VARCHAR2(1 BYTE)	Test

6.1.2.2 TBL_USERS

The users of the devices are stored in the TBL_USERS with the following columns providing information about their First and Last Name as well as their CERN ID:

TBL_USERS		
COLUMN	**TYPE**	**DESCRIPTION**
ID (PK)	NUMBER (3,0)	The user ID
FNAME	NVARCHAR2(30)	The user’s first name
LNAME	NVARCHAR2(30)	The user’s last name
CERN_ID	NVARCHAR2(20)	The CERN ID of the user

6.1.2.3 TBL_DEVICES

The data concerning the various devices are the device name, the IP address of the device, the description, the MAC address of the device and the hostname. These data are stored in the TBL_DEVICES of the DB with the following columns:

TBL_ DEVICES		
COLUMN	**TYPE**	**DESCRIPTION**
ID (PK)	NUMBER (3,0)	The device ID
NAME	NVARCHAR2(20)	The device name
IP	NVARCHAR2(20)	The IP address of the device
DSCR	NVARCHAR2(500)	Description of the device
MAC_ADDR	NVARCHAR2(50)	The MAC address of the device
HOST_NAME	NVARCHAR2(50)	The hostname of the device

6.1.2.4 TBL_DEVICESTATUS

The status of the devices is providing information about the date and time of the report, its battery level, the camera status of the helmet and possible handheld camera as well as the audio status. This information is stored in the TBL_DEVICESTATUS of the DB with the following columns:

TBL_ DEVICESTATUS		
COLUMN	**TYPE**	**DESCRIPTION**
ID (PK)	NUMBER (15,0)	The device status ID
DEVICE_ID (FK)	NUMBER (3,0)	The device ID
DATETIME	TIMESTAMP (6)	The date and time of the report
BATTERY_LEVEL	NVARCHAR2(5)	The battery level of the device
CAMERA_HELMET	NUMBER (1,0)	The helmet camera status
CAMERA_HANDHELD	NUMBER (1,0)	The handheld camera status
AUDIO	NUMBER (1,0)	The audio status

6.1.2.5 TBL_EVENTS

The data concerning the various events are stored in the TBL_EVENTS of the DB with the following columns:

TBL_ EVENTS		
COLUMN	TYPE	DESCRIPTION
ID (PK)	NUMBER (15,0)	The event ID
EVENT_TYPE	NVARCHAR2(50)	The type of the event
DATETIME	TIMESTAMP (6)	The date and time when the event occurred
DEVICE_ID (FK)	NUMBER (3,0)	The ID of the device that reported the event
SENSOR	NVARCHAR2(50)	The sensor that is related to the event
VALUE	NVARCHAR2(1024)	The value of the sensor measurement related to the event
THRESHOLD	NVARCHAR2(1024)	The threshold of the sensor measurement related to the event
UNIT	NVARCHAR2(20)	The measurement value unit

6.1.2.6 *TBL_MEASUREMENTS*

The measurements are stored in the TBL_MEASUREMETNS of the DB with the following columns:

TBL_ MEASUREMENTS		
COLUMN	**TYPE**	**DESCRIPTION**
ID (PK)	NUMBER (15,0)	The measurement ID
DEVICE_ID (FK)	NUMBER (3,0)	The ID of the device that sent the measurement
DATETIME	TIMESTAMP (6)	The date and time when the measurement was taken
VALUE	NVARCHAR2(1024)	The value of the measurement
VALUE_TXT	CLOB	The text value of the measurement (for large objects)
VALUE_BIN	BLOB	The binary value of the measurement (for large objects)
UNIT	NVARCHAR2(20)	The measurement unit
METHOD	NVARCHAR2(20)	The measurement method
SAMPLINGRATE	NVARCHAR2(20)	The sampling rate of the sensor measurement
SENSOR	NVARCHAR2(50)	The sensor type
UP_THRES	NVARCHAR2(20)	The configured up level threshold

DOWN_THRES	NVARCHAR2(20)	The configured down level threshold
CONNECTED	VARCHAR2(1 BYTE)	The connection status

6.1.2.7 TBL_MEASUREMENT_CONFIGURATION

The configurations of the various measurements are stored in the TBL_MEASUREMENT_CONFIGURATION of the DB with the following columns:

TBL_ MEASUREMENT_CONFIGURATION		
COLUMN	**TYPE**	**DESCRIPTION**
ID (PK)	NUMBER (15,0)	The measurement ID
DEVICE_ID (FK)	NUMBER (3,0)	The ID of the device that sent the measurement
DATETIME	TIMESTAMP (6)	The date and time when the measurement was taken
UNIT	NVARCHAR2(20)	The measurement unit
SAMPLINGRATE	NVARCHAR2(20)	The sampling rate of the sensor measurement
SENSOR	NVARCHAR2(50)	The sensor type
UP_THRES	NVARCHAR2(20)	The configured up level threshold
DOWN_THRES	NVARCHAR2(20)	The configured down level threshold
SLOPE	NVARCHAR2(20)	
OFFSET	NVARCHAR2(20)	

6.1.2.8 TBL_GENERAL_CONFIGURATION

General configurations are stored in the TBL_CONFIGURATION of the DB with the following columns:

TBL_ GENERAL_CONFIGURATION		
COLUMN	**TYPE**	**DESCRIPTION**
ID (PK)	NUMBER (15,0)	The measurement ID
DEVICE_ID (FK)	NUMBER (3,0)	The ID of the device that sent the measurement
DATETIME	TIMESTAMP (6)	The date and time when the measurement was taken
BSSID	VARCHAR2(20 BYTE)	The BSSID
DOSIMETER_ID	VARCHAR2(15 BYTE)	The ID of the dosimeter that is used

Chapter 7

7 Adaptability and scalability of the CS and DAQ System

It is very crucial to explore the adaptability and the scalability of the developed CS and DAQ System in other environmental conditions. To achieve this goal, a research has been conducted and presented in this chapter. Firstly, we present an introduction to the main concepts of adaptation in service-based applications, such as the adaptation design techniques, adaptation triggers and adaptation strategies. Through the study and review of the literature, the need of designing adaptable service-based applications is obvious. In chapter 7.10 we will form a case study of the developed CS and DAQ System from a service-based application perspective in order to investigate the scalability and adaptability of the System on other markets and different environmental conditions.

7.1 Introduction to adaptation in Information Systems

Information Systems need to operate correctly in a highly dynamic world regardless of any unexpected changes in factors such as environmental conditions, user requirements, technology, legal regulations and market opportunities. Systems must continuously adapt themselves to quickly adjust to those changes. Therefore, the service concept has emerged as a suitable abstraction mechanism, allowing new technologies, standards and methods to create service-based applications (SBA) [49]. Service-based applications are software applications that provide several services which may be developed and provided by third party application developers. These applications offer complex and adjustable functionalities by composing different types of services. However, SBA's themselves must be able to dynamically adapt to every requirement of any user. Hence, mechanisms that enable adaptation should be introduced in the life-cycle of applications, both in the design and in the runtime phases [50].

Service-Oriented Architecture (SOA) is an application architecture, in which all functionalities and services that may originate from different sources and underlying technology environments, are assembled together in a single approach to improve the implementation of a business system. From a Software Design perspective, SOA is an approach that suggests reusing components and services to achieve a business system that will be adaptable and scalable to more than one environment. An even broader interpretation is that SOA indicates the end of monolithic enterprise applications and the beginning of more flexible and adaptable business process centric applications.

The majority of the proposed approaches for the life-cycle of service-oriented applications, assume human interventions, for example ASTRO (http://www.astroproject.org/) focus on the possibility to monitor and intervene on SBAs in order to recovery from unwanted and unexpected behavior [50]. However, it is interesting to mention the approach of Lane et al. [51], who propose a life-cycle supporting self-adaptation of the service-based applications, even if they lack explicit guidelines for the design of adaptable service-based applications. Various frameworks supporting adaptation have been defined in the literature, each of them addressing a specific issue.

Bucchiarone et al. [50] propose a coherent, holistic, and easy to apply design for adaptation that will support developers in the usage of the available mechanisms. The main idea is that adaptation strategies can be programmed at design and implementation time and it will be presented in this essay.

An introduction to the adaptation of service-based applications, what is adaptation of these kind of applications, where it is used as well as an overview of the main components that are needed to build and operate an adaptable SBA, are presented in subchapter 7.2. In subchapter 7.3 I focus on the several types of adaptation that a system may need to comply with or could use to achieve adaptability. The next subchapter 7.4 describes the existing research roadmaps related to services and software as they were presented in the survey of Metzger et al. [49]. These roadmaps are divided in two categories, industry-led roadmaps and European level roadmaps. A proposition of a European level roadmap, the European Network of Excellence in Software Services and Systems, called S-Cube, is presented in subchapter 7.5. It has established an integrated, multidisciplinary, vibrant research community, enabling Europe to lead the software-services revolution and helping shape the software-service based Internet which is the backbone of our future interactive society. Subchapter 7.6 illustrates how to design an adaptable SBA, describing firstly the design techniques (subchapter 7.6.1), secondly the adaptation strategies that could be used (subchapter 7.6.2) and finally the identification of possible adaptation triggers (subchapter 7.6.3).

To facilitate the comprehension of subchapter 7.8 that is going to introduce an adaptation framework through possible software challenges, I firstly describe in Chapter 7.7 the software architecture life cycle. Chapter 7.9 focuses on the different scenarios from various industries or environments where the design for adaptation activities can be performed, as they were presented by Bucchiarone et al, [50]. The factors that trigger adaptation, the types of adaptation realization that are suitable in each scenario and the appropriate strategies for adaptation, are presented in this chapter accordingly to the various characteristics of every case study. EDUSAFE, a Marie Curie ITN project that focuses on research into the use of Virtual Reality (VR) and Augmented Reality (AR) during planned and emergency maintenance in extreme environments (nuclear installations, space, deep sea etc.), is examined and presented as a case study in subchapter 7.10. Finally, in subchapter 7.11 the conclusions of this research are presented.

7.2 Adaptation of Service-Based Applications

Service-based applications are composite applications which are created and based on multiple software services, as Lane et al. mention [52]. Software services or services are computational components that use computer networks to function and operate. The service-based applications developers may or may not be in control of the various services, due to the fact that all these kinds of services can be independent. This fact generates the need to link the self-contained services in an effective way, and to produce adaptable applications, but it also enhances the potential build of SBA from unrelated services which consequently increases and enriches the functionalities of a system.

Towards the better understanding of what is a service-based application in the first place (before proceeding to the definition of what is adaptation of SBAs), we present an example derived from the aerospace manufacturing domain and operation business. Apart from developing their products, many manufacturing companies, aim to provide high quality services and integration of the product development with maintenance operations. To achieve this goal, Zhu et al. [53] propose a product service system (PSS) focusing on maintenance, repair and overhaul (MRO) services in the aerospace industry. This application is combining the information between aviation manufacturers, airlines and MRO service providers, by linking the databases and IT systems each enterprise has, through web-based services.

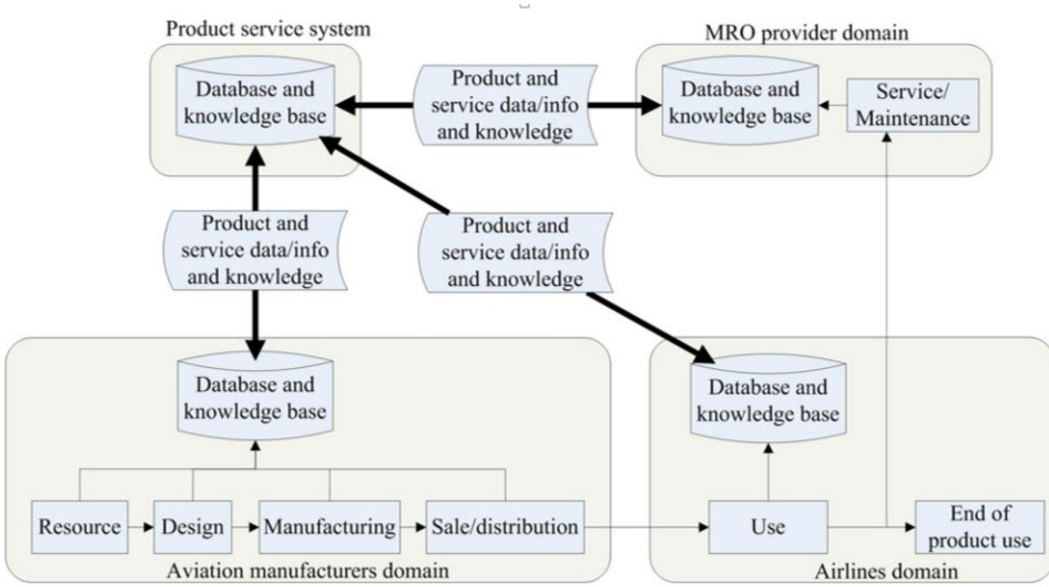

Figure 44. Information flow through various application services *[53]*.

Zhu et al. mention "For example, if an aviation manufacturer upgrades its products information, but MRO providers do not upgrade their product databases, they cannot provide the customers with a quality service in such situation. Therefore, the product and service databases are not only independent but also interrelated in the proposed system". Fig. 44 shows the information and knowledge provision and exchange among the various services of an application or different applications [53].

Adaptation of Service-Based Applications is a process in which a system adapts its behavior to tackle problems, such as service evolution (e.g., a new version may be available), hardware volatility (e.g., network quality changes may be needed), and varying user demands and new requirements (e.g., a new functionality or a different level of quality of service) [54]. System and software engineering are fields that provide research paths and methodologies for current service-oriented applications. The ability of an application to be adaptable is strongly linked with the design principles and guidelines that enable adaptation. There are various facets of adaptation and evolution in SBA's. The definition and operation of adaptable service applications is schematically shown in Figure 45, as shown by Bucchiarone et al. [50].

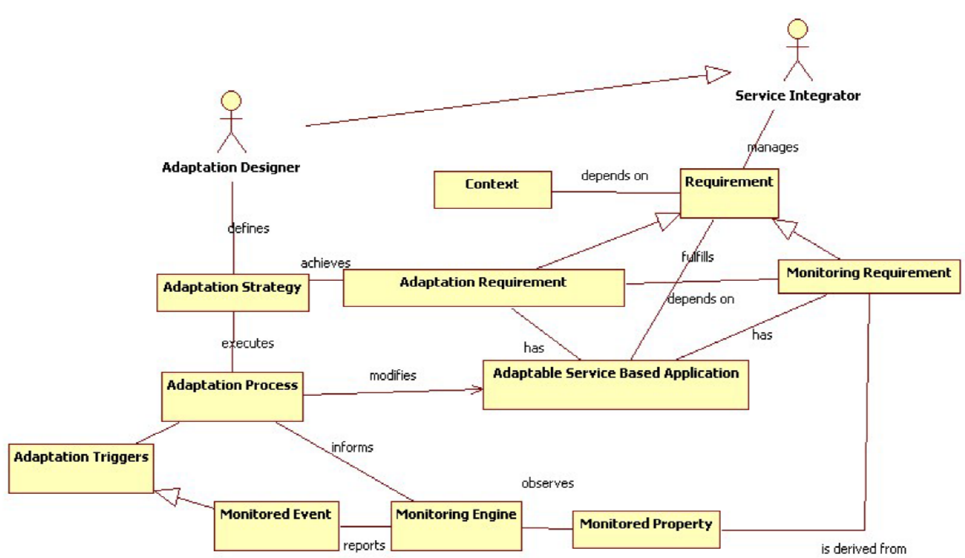

Figure 45. Main ingredients of building and operating an Adaptable SBA.

In the above figure two main developers, the *Adaptation Designer* and the *Service Integrator*, manage and define crucial components of the adaptation process of the application. Their role and all the components that help in building and operating an adaptable service-based application, are explained in further detail below, based on the definitions of Bucchiarone et al. [50].

One of the adaptation aspects, is the *Monitoring Requirements*, which refer to the need of detecting properties and situations that should be monitored to design an adaptable SBA. Those properties should be defined from the *Adaption Designer* and should be observed at runtime by a *Monitoring Engine* which depending on its values, it delivers, and reports monitored events. The latter events trigger the *Adaptation Process*, where various strategies are performed and fulfill the *Adaptation Requirements*, the requirements that are needed to build and operate an adaptable SBA.

Context is an important part in this perspective, involving users and execution properties. There are several ways to acquire the user's information and preferences. For instance, this information can be acquired through the use of pre-existing profiles for each user, or each user can be profiled during the run-time of the application and in that case the information about the user is acquired only during the operation of the application. The context is depended on the requirements that the *Service Integrator* manages. Monitoring can be used to obtain further information and data for the user, such as geographical position, user's points of interest and the actions that a user performs when interacting with the surrounding environment. The conditions under which the service-based application and its services execute, are the execution properties. In general, adaptation requires temporary alterations, to allow to change according to the modifications in the requirements and/or the context of the SBA or to respond to potential errors. Adaptation for a service, could be for example, the re-execution of an inaccessible service or the replacement of a faulty service. Adaptation is called evolution, when different circumstances, may require the re-design and re-engineering of the application with permanent changes. Additionally, evolution could be required if a defective situation is repeatedly noticed. In this case, instead of repeatedly applying the required adaptation strategies, a modification of the application logic would be preferred.

Wanghu Chen et al. [55] mention: “In some domains such as Bioinformatics and the Industry of Chinese Traditional Medicine, constructing applications through assembling services shared by third parties is widely adopted. e.g., in Bioinformatics, several autonomy research institutes have provided much more than 3000 Web services to share their rare resources such as genes databases and sequence analyzing software for users”. This is an example of the large set of cases where applications that require assembling of various services need to be adaptable to maintain the high quality of their multiple service and their combination. The need for the adaption of service-based applications may concern the component services or the context of an SBA, derived from the following changes that may occur as they were mentioned by Lane et al. [52]:

i. Changes in the functionality of a service due to variation of the service interface (e.g., signatures, data types, semantics), variation of service interaction protocol (e.g., ordering of messages) or possible failures of the application.
ii. Changes in the service quality due to service availability, degrade of QoS parameters, violation of SLA (Service Level Agreement) or decrease of service reputation (e.g., black lists), etc.
iii. Changes in the service context that may occur due to changes in the business context, changes in agile service networks, or new business regulations and policies.
iv. Changes in the computational context such as different devices, protocols, and networks.
v. Changes in the user context such as different user groups and profiles, social environment, physical settings (e.g., location/ time), and different

user activities.

All the above changes consist a small introduction to the factors that indicate the need for adaptation in service-based applications. In chapter 7.3, the several types of adaptation triggers, which could address either to the component services or the context of SBAs, are described in further detail, confirming that adaptation in that kind of applications is indeed needed.

7.3 Types of Adaptation

Lane et al. (2011), refer to two types of adaptation, static adaptation and dynamic adaptation. "With static adaptation, adaptation mechanisms are hard coded into the application at development time and the adaptation logic cannot be changed without recoding. Dynamic adaptation on the other hand, allows the adaptation logic to be modified or replaced during runtime without shutting the system down. Dynamic adaptation is more flexible than static adaptation, but it requires some process to guide the manual intervention during runtime" [52].

From the perspective of Bucchiarone et al., there are different types of Adaptation that they distinguish as follows: Perfective Adaptation, Corrective Adaptation, Adaptive Adaptation, Preventive Adaptation and Extending Adaptation. These types of adaptation are described here:

- *Perfective adaptation* aims to better achieve the users' requirements. The goal is to improve and optimize the quality attributes of a SBA even if it runs correctly.
- *Corrective adaptation* focuses on repairing and preventing failures. This type of adaptation removes any faults in the behavior of the application.
- *Adaptive Adaptation* executes modifications in a SBA when its execution environment changes.
- *Preventive Adaptation* focuses in preventing possible future faults before they occur.
- *Extending Adaptation* extends a SBA by adding new functionalities as

required.

It is interesting at this point to mention the following techniques for determination of the demand for an adaptation of an SBA (techniques for determination of an adaptation trigger) [56]:

- Corrective adaptation based on online testing: To define possible failures of the SBA's constituent services, online tests of services are performed during run-time (operation) of the SBA. A corrective adaptation trigger is a potential failure of such test trigger, which could possibly be satisfied by replacing the failed service with an alternative service.

- Perfective adaptation based on requirements engineering (RE): Usually, enterprises have contract relationships with other business partners. This fact is reflected in the set of services that may be used in SBAs. These partner services usually meet the requirements specified by the requirements engineers in the enterprise. In some cases, however, due to the dynamic nature of the service provision, new relationships are established with other (previously unknown) business partners. If the newly introduced service is better and/or more appropriate (e. g. cheaper or faster), the requirements engineer could recommend the use of this new service and thereby issue a perfective adaptation trigger. Moreover, these new services may introduce additional and/or enhanced functionalities, which may generate new requirements.

The above adaptation processes are missing from the software engineering literature and the ability to be self-adaptable is an important research topic in the service development community [57].

7.3.1 Integration of Perfective and Corrective Adaptation

The adaptation of a service-based application can address different types of objectives, such as correcting possible errors in the SBA (corrective adaptation) or adapting the application to the various changes and requirements that may occur (perfective adaptation), as it was mentioned in the previous chapter. During the development of such service-based applications, which need to fulfill the afore mentioned objectives or even more, it is crucial to consider the interplay and the interactions among the several types of adaptation in order to prevent possible conflictions between those types of adaptation that are used. As an example, to address the goal of corrective adaptation, the service-based application might aim at replacing a failed service A with a service B, while at the same time the SBA, in order to address the aim of perfective adaptation, aims to replace service A with a service C, which promises to provide a better quality of service than service A [56, p. 3]. Therefore, the alignment and synchronization of the needs for applying corrective and perfective adaptation in a service-based application is leading to the production of solutions for the problems that arise (conflicts among types of adaptations).

Exploiting the techniques introduced in the previous chapter (Corrective adaptation based on online testing, Perfective adaptation based on requirements engineering (RE)), one can determine the need for adaptation of the SBA and how to avoid the sometimes inevitable conflicts between the diverse adaptation objectives. Both above techniques share the characteristic that they are pro-active in nature, i.e., both techniques lead to "predictive" adaptation triggers. In the case of online testing, the failure of a service could point to a problem of the SBA (which involves this service) in the future. In the case of recommendations from RE, this provides the possibility to improve the SBAs and to anticipate future requirements. Thereby, both of those adaptation drivers, which are the core building blocks of our proposed solution, share a fundamental commonality. This simplifies addressing the problem of synchronizing the two adaptation goals [56, p. 4].

This approach is a solution proposed by the S-Cube Network of Excellence that also proposes a further key idea, to use a *central enterprise service registry*. This registry contains references to in-house services, e. g. those services provided by the enterprise itself, and to external services, e. g., those services, that are provided by external service providers. Only these services can be used in the enterprise's service-based applications. Each reference to one of the services is accompanied by a service description. Since the enterprise service registry is a private registry, it can be administrated solely by the enterprise. On the one hand, this enterprise service registry constrains possible adaptations and, thereby, reduces the flexibility of the SBA. On the other hand, it enables the use of techniques (such as testing techniques) that require a certain level of stability (see Fig. 46) [56].

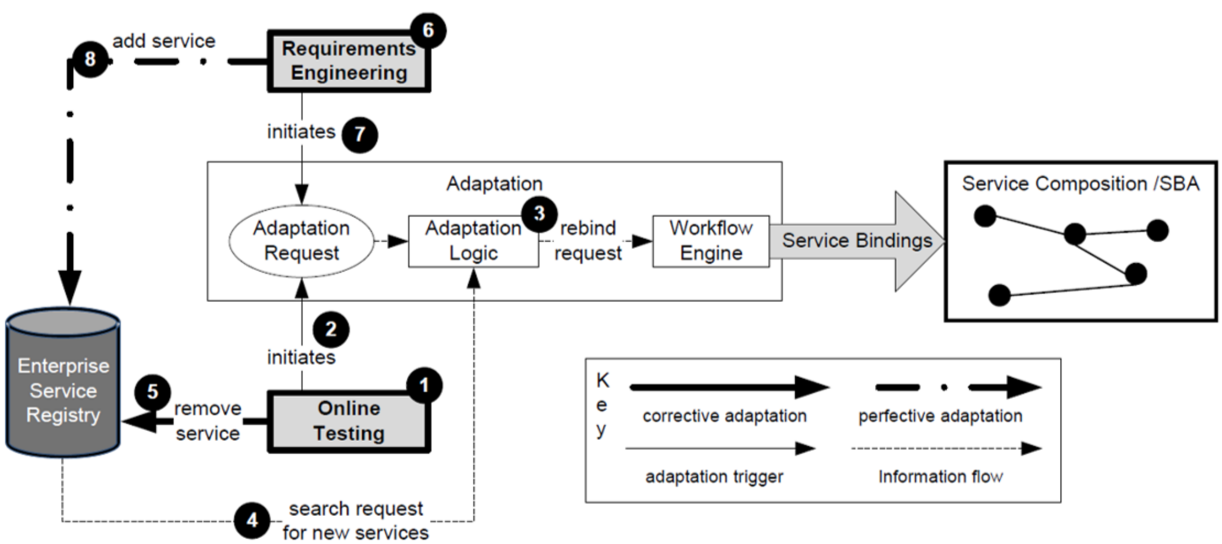

Figure 46. Overall approach

Gehlert et al., propose the approach shown in *Figure 46*, to avoid conflicting adaptations (perfective and corrective adaptation) in the service-based application. This approach that is graphically shown in the previous *Figure 46*, is explained below [56, p. 6]:

- Corrective Adaptation: Assume that online testing uncovers a failure of one of the SBA's services (block 1 in *Figure 45*). In this case, the online testing activity issues an adaptation trigger (block 2). Based on this adaptation trigger, the actual adaptation component of the SBA (block 3) needs to determine how to adapt the SBA, in our case this means it needs to find an alternative service (e. g., by searching the service registry). The adaptation component then must notify the workflow engine to bind this alternative service to the service composition in place of the failed service (block 4). To avoid using the failed service in other SBAs of the enterprise, the online testing activity removes it from the enterprise service registry (block 5).

- Perfective Adaptation: Assume that a new service was discovered during requirements engineering and that this service is cheaper or better than the previously used one (block 6). If the requirements engineer decides to use this service in the SBA, this engineer needs to add it to the enterprise service registry (block 8) and to trigger an adaptation (block 7). This adaptation trigger will eventually notify the workflow engine h3 and, like above, the workflow engine rebinds the new service to the service composition instead of the old one (block 4).

In the above approach, an assumption has been made that the service-based application is already running and therefore uses services for the enterprise service registry. Online testing activity constantly tests the services of the SBA for possible failures during the run-time phase. In the case of a failure of a service, that particular service is going to be removed by the online testing activity. The requirements engineering activity scans repeatedly for new and inventive services and in the case such services are noticed then they are added to the enterprise service registry. By taking into consideration that this methodology, that is described so far, requires that each and every service-based application will be using services from the enterprise service registry, we can conclude that this registry, together with the workflow adaptability, serve as tools and mechanisms to achieve the synchronization of the activities of requirements engineering and online testing. For example, it is not possible in this process, that online testing and requirements engineering simultaneously replace services in the workflow. There are no possible conflictions noticed in this workflow, since the allowed actions are adding or removing the services to or from the registry [56].

7.4 Existing Research Roadmaps

This Chapter is an introduction to the existing research roadmaps related to services and software as they were presented in the survey of Metzger et al. [49].

Existing Research Roadmaps

A. Industry-Led Roadmaps.

The Industry-Led Roadmaps involve approaches that were applied from European small and medium enterprises as well as key industrial players. These industry-oriented roadmaps have been released in the context of European Technology Platforms (ETPs2).

Those roadmaps include:

- NESSI (http://www.nessi-europe.com/), the European Technology Platform (ETP) dedicated to Software, Services and Data. NESSI provides input to the EU Institutions on research actions and technology matters of particular importance to the software domain, and the overall aim is to enable the software and services sector help vitalize the great potential of the European economy and society. NESSI gathers partners and members from all over Europe, both from industry and academia, and engages in close dialogue with the European Commission and other stakeholders on several topics of specific relevance to NESSI - such as Big Data Value, Cloud Computing and Software Engineering.

NESSI takes an active role in addressing future challenges of Europe, by working for ensuring that enough resources are invested in leading-edge industrial and academic research for innovative technologies in the software and service domain. Europe must turn its outstanding R&D potential, its infrastructure and its technological environment, into successful product development and marketable products in order to maintain its competitive edge.

- Net! Works (http://www.networks-etp.eu/) is the European Technology Platform for communications networks and services. Communications networks enable interaction between users of various types of equipment, either mobile (e.g. mobile phones) or fixed (e.g. PCs); they are the foundation of the Internet. The Net!Works European Technology Platform gathers more than 700 players of the communications networks sector: industry leaders, innovative SMEs, and leading academic institutions. The mission of Net!Works is to strengthen Europe's leadership in networking technology and services so that it best serves Europe's citizens and the European economy. Net!Works members and collaborators have released a strategy recommendation on networking and telecommunications for cloud computing and service platforms [58].

B. EU-Level Roadmaps.

In the European landscape of roadmaps, the following approaches are presented:

- The European Future Internet Assembly (FIA: http://www.future-internet.eu/) is a collaboration between projects that have recognized the need to strengthen European activities on the Future Internet to maintain

European competitiveness in the global marketplace. The goal is to help consolidate Europe's efforts in shaping the Internet of the future.

- EUDAT (http://eudat.eu/eudat-service-road-map) Service Road Map: EUDAT's mission is to design, develop, implement and offer "Common Data Services" to all interested researchers and research communities. These services will be offered through the Collaborative Data Infrastructure (CDI) which is being identified by many data different initiatives at community, research organization and cross-border level (disciplines and countries). Common data services obviously must be relevant to several communities and be available at European level and they need to be characterized by a high degree of openness: (1) Open Access should be the default principle; (2) Independent of specific technologies since these will change frequently and (3) Flexible to allow new communities to be integrated which is not a trivial requirement given the heterogeneity and fragmentation of the data landscape. The roadmap includes three overlapping phases of brainstorming, selecting and defining the services to be offered. Development and improvement of services is a continuous process.

Considering the landscape of nowadays market and business in Europe it is worth mentioning a very recent roadmap published by the European Commission (May 2015), The EU Digital Single Market Roadmap (http://webershandwick.be/).

- The Objectives are:
    - Do away with intra-EU differences to create a strong single market for business and consumers
    - Improve consumer trust in ICT technologies and online services
    - Enable innovation and development of next generation infrastructure and technologies
    - Ensure the EU regains leadership in ICT innovation

7.5 The Proposition of S-Cube

S-Cube (http://www.s-cube-network.eu/) is a European Network of Excellence in Software Services and Systems and it was funded by *the European Commission (from 01.03.2008 to 29.02.2012)*. It has established an integrated, multidisciplinary, vibrant research community, enabling Europe to lead the software-services revolution and helping shape the software-service based Internet which is the backbone of our future interactive society.

The quick advancement of information and communication technology creates an increasing number of research paths for better approaches of processing and computing. In this setting, one vital open door is the advancement of creative frameworks through the synthesis of software services, which are accessible over distributed computing infrastructures. Software services have the ability to provide software functionality to clients in a more dynamic and adaptable manner than conventional software systems.

The challenge for the creation of S-Cube was driven from the fact that in the past years the majority of European research teams were focusing on the development of individual exploration on SBA's without investigating the possibility of generating a more dynamic approach of services responsive to a broader set of information systems and their needs. Even though multidisciplinary research was often required, it could not be conducted by the single research groups because it was usually overpassing the existing means or even their potential goals. Therefore, the fragmentation of all the pre-existing approaches was inevitable, resulting to a scattered development of methods, strategies and mechanisms.

This problem was particularly obvious when the orientation of the high-priority research was conditioned to focus more on the functional synthesis of service-based applications. During this effort of defining these functionalities, it is necessary and required, to combine the several sub-disciplines that form the result. These disciplines may be software engineering, service infrastructure, service composition and coordination as well as business process management. The lack of the interaction between those disciplines (for example the software engineering community with the business process management) lead to the exacerbation of the problem. Detachment and isolation of some communities such as service composition and business process management were setting the goal of an optimized result difficult to achieve. Because of this fragmentation, the approaches that were followed before this project, tended to address particular issues and specific research obstacles, with the absence of a basic theoretical and methodological framework that could be used in services. Exchange and aggregation of the ideas and the various research approaches were left uninvestigated, as well as the lack of knowledge transfer having an ultimate goal of developing and forming a focal concept.

The solution to this problem inspired S-Cube research team to investigate the design of a coherent and adaptable approach. This Software and Systems Network had set as a general goal to build a linking layer between different research areas, that are, in the vast majority of cases, interdependent with each other. The final purpose of S-Cube was to develop a holistic and coherent design approach for every recognized core problematic areas and issues in services research. To achieve that goal, the basic approach is to join and combine experiences, results and insights from distinctive research teams. S-Cube aimed to achieve the following goals as they were documented in the final report of S-Cube Network of Excellence:

a. *Integration of Research Communities*: This target included the integration of the various key European research groups and organizations, by re-adjusting and re-shaping their goals or orientation in their approaches and finally by merging and synthesizing all the different research blocks. This integration leads to accomplishing huge advancement driven from the state-of-the-art research.

b. *Research on Engineering and Adaptation Methodologies and Service Technologies*: This goal encompassed the research on engineering, quality assurance and adaptation concepts and techniques for service-based applications, and also research on service technology foundations to realize service-based applications. In this context, S-Cube intended to combine all the interdependent parts and develop quality assurance strategies, adaptation techniques and new methods in engineering. Merging the various distinctive capabilities of each subfield S-Cube aims to create a holistic view and trigger the form or the reinforcement of service integrators, services providers and other relevant stakeholders to compose, evolve and adapt service-based applications. Moreover, this Network was targeting the development and design of innovative mechanisms and methodologies that are going to be used in the up and coming era of service-based applications that support consistent engineering design and adaptation over the business process and the multiple service functionalities.

c. *Integration of Education*: This objective intended to initiate a Europe-wide common program of education and training for analysts, researchers and industry that will reinforce the integration and the combination of European knowledge increasing its effectiveness and impact through time.

d. *Bonding of Research Staff*: This objective encompassed the exchange of European researchers and doctoral students, to encourage a research framework, where an optimal transfer and exchange of knowledge, information and experience would be successfully accomplished.

e. *Community Outreach, Spreading of Excellence and Collaboration*: Concerning the outreach goals of S-Cube Network, this project focused on organizing and establishing not only European but also international conferences, in order to increase the community awareness about its work and objectives and most importantly to enhance the integration of research communities. S-Cube Network also targeted in setting up a solid and durable connection with industry in Europe mostly through ETPs (European Technology Platforms) like NESSI (Networked European Software & Services). Building solid links with universities and research institutions was also part of S-Cube objectives aiming to create and enhance collaboration and knowledge exchange that will impact the scientific field's future.

The research framework of S-Cube (shown in Figure 47), as it was presented in their final report, is a key component in promoting the development of multi-disciplinary research integration. This structure supports a solid and well-defined division of interests, encouraging the optimal and successful integration of research. These interests are visible on this high-level 6-block representation of the framework. These blocks are sorted in two main categories: the "service technology layers" and the "service techniques and methods planes".

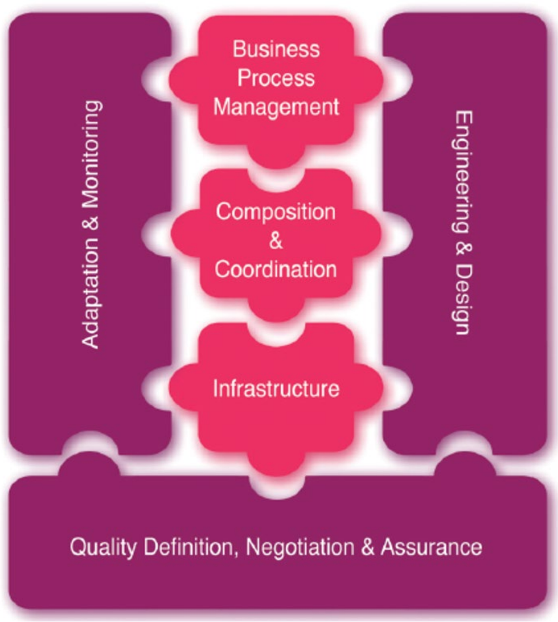

Figure 47. S-Cube Integrated Research Framework

Metzger et al. have clustered the research challenges into four topic areas along the above S-Cube Integrated Research Framework (IRF) to guide the research activities of the network [59]. These topics as they were introduced in this research are described below:

i. **Service Life-Cycle and Software Engineering**

This topic area refers to the right-hand pillar of the IRF. It includes all challenges related to engineering and design of service-oriented systems, focusing on life-cycle aspects and specific needs for software engineering of services.

ii. **Service Technology Foundations**

This topic area refers to the middle pillar of the IRF. It includes all challenges related to the services technology stack, including service infrastructures (such as grids and clouds), service composition & coordination, and business process management & service networks.

iii. **Multi-Layer and Mixed-Initiative Monitoring and Adaptation for Service-Oriented Systems.**

This topic area relates to the left-hand pillar of the IRF. Adaptation is considered a key capability of service-oriented systems to address the highly dynamic setting in which those systems need to live. This topic area includes challenges related to monitoring and adapting service-oriented systems across the whole technology stack as well as considering context and humans as triggers for adaptation.

iv. **Online Service Quality Prediction for Proactive Adaptation**

This topic area relates to the bottom pillar of the IRF. It includes all challenges related to assessing and predicting service quality during run-time to proactively respond to imminent problems and issues. Due to the highly dynamic nature of services, predicting the quality during design time does not suffice, thus calling for online techniques.

7.5.1 The S-Cube Life-Cycle Model

The S-Cube Life-Cycle Model, is a dynamic perspective of characterizing the core actions of adaptation and evolution of service-based applications, facilitating in visualizing as well the dependencies between those adaptation and evolution procedures. A general view of the S-Cube Life-Cycle Model is presented in the following Figure 5. This model concentrates on the dynamic adaptation strategies, which are or can be applied during run-time of the service-based applications. The S-Cube Life-Cycle sets the two loops (shown in Figure 48), aiming to address both viewpoints, design time as well as run-time. Those loops can be performed repeatedly.

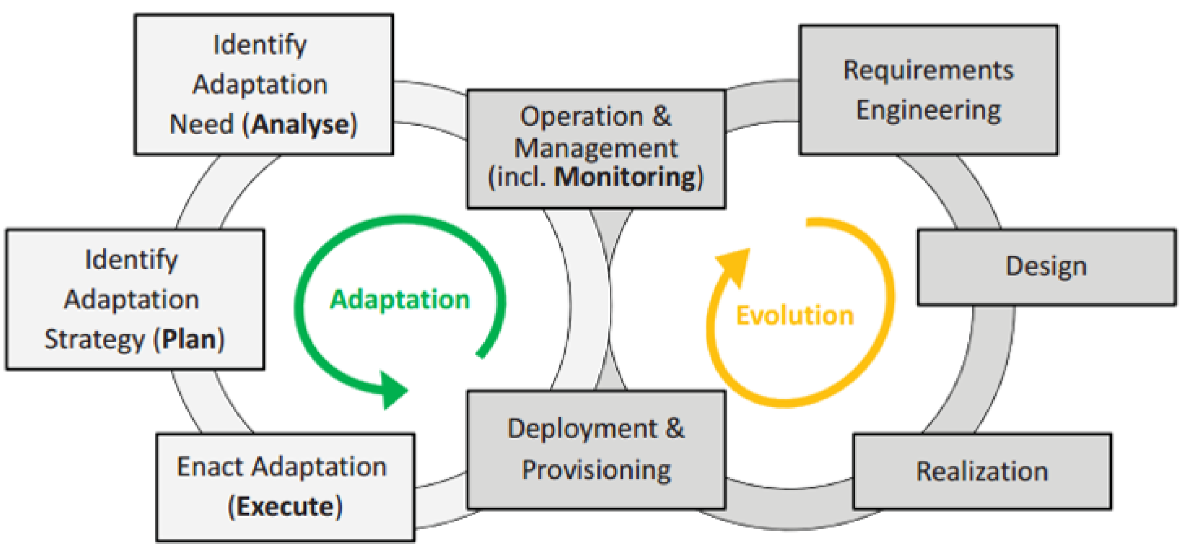

Figure 48. The S-Cube Life-Cycle

The Evolution loop. On the right-hand side of the S-Cube Life-Cycle, there is the Evolution loop that sets the development and the development actions, along with the engineering requirements, the design, the realization, and the deployment and provisioning. Then again, it expands them with "design for adaptation" steps, to define and actualize the self-monitoring and the changes that the service-based application itself has to perform when entering the left-hand side of the life-cycle.

The adaptation loop. During the adaptation loop, all the activities linked to the process of autonomously addressing modifications in run-time of the SBA, are defined. The actions, taking place in this adaptation loop, are inherited and collide with the activities followed in the MAPE loop (Monitoring-Analyze-Plan-Execute) that is traditionally met in the autonomic systems. The MAPE model consists a standard reference that facilitates the division of the various adaptation phases [60].

In the evolution cycle, shown on the right-hand side of the figure, the software engineer concentrates on the development of the SBA through the traditional stages of requirements engineering, design, construction and deployment, while also focusing on quality assurance. However, as adaptation is a desirable feature in SBAs, the software engineer must also consider how the application will adapt during its lifetime. Within the complete life-cycle, there must also be a focus on Operation and management and Deployment and provisioning [51].

The framework of the S-Cube Life-Cycle, is based on the methods that are used in software engineering. This model follows a continuous and iterative process for achieving the development, the implementation as well as the maintenance of the system's services. Across this lifecycle, all the important information is continuously cycled between the various phases, with the repetition of redefinition and adaptation steps. The technique that is used, helps in developing solutions that are a combination of services, which subsequently enhance the ability to examine, reevaluate and modify the solution itself. Persistent alterations and changes take place through all the layers and steps of the two loops of adaptation and evolution.

Service Level Agreements (SLAs) and quality objectives, affect the need of changes in the system, during monitoring and measuring of services. As a result, the following three functionalities of the S-Cube Life-Cycle are created (as shown in the left-hand side in *Figure 46*):

i. Detect new issues, changes and requirements for adaptation.
ii. Identify possible adaptation strategies.
iii. Enact the adaptation strategies.

When service-based applications have been adapted, they will be put into operation once again and be re-assessed.

7.6 Designing adaptable SBAs

As presented in the previous chapter, one of the goals of the S-Cube Network of Excellence in Software Services and Systems, is to develop a life-cycle for the development of adaptable SBAs. To provide reliable and optimally designed adaptable applications, the services that are included in this application, should always be adaptable to all the potential changes and unexpected needed modifications that may take place at any time. The identification of possible alternatives should be conducted during the design time. To gain a deeper understanding of the several adaptation strategies, the S-Cube Life-Cycle (Figure 47) is more thoroughly presented in Figure 48 [50], showing the multiple criteria that must be considered when trying to design and develop an adaptable service-based application.

On the right-hand side of the adaptable S-Cube Life-Cycle (Figure 48), the Evolution loop sets the following development actions:

a. Define adaptation and monitoring requirements during the early requirements of engineering.
b. Design monitoring and adaptation during the requirements of engineering

and design step.

c. Construction of the monitors and the adaption mechanisms during the construction step.

On the left-hand side of the adaptable S-Cube Life-Cycle (Figure 48), the Adaptation loop involves the following design artifacts:

a. Adaptation requirements, context, monitored events during the identification of the adaptation needs step.
b. Suitable strategies during the identification of adaptation strategy step.
c. Strategy instance and adaptation mechanisms during the step of enacting adaptation.

The decision between adaptation and evolution is a development action that takes place during the identification of adaptation needs step. Additionally, when operating and managing the application the action that takes place is the run-time monitoring and the design artifacts are the monitored properties. Finally, during deployment and provisioning, deployment of adaptation and monitoring mechanisms as well as deployment-time adaptation take place.

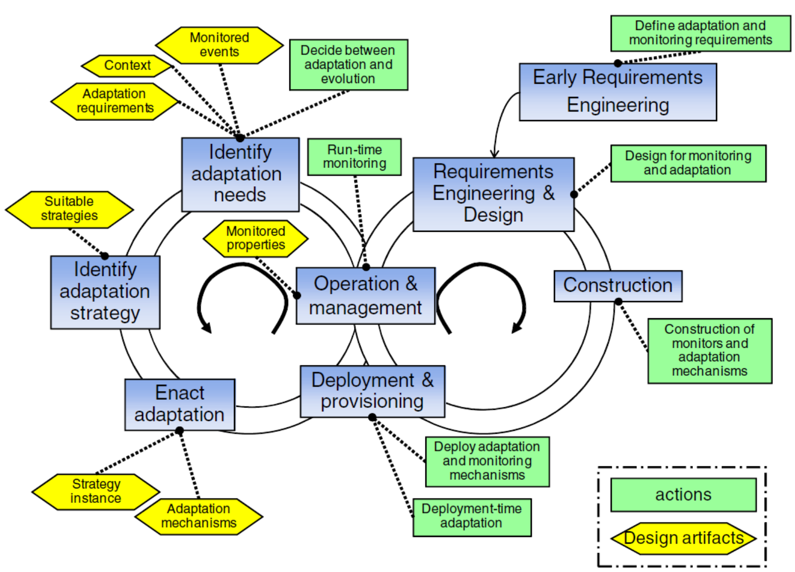

Figure 49. The life-cycle of adaptable SBAs

In the following subchapters the design techniques for adaptable SBAs, the adaptation strategies and the identification of the adaptation triggers, are presented. All of these consist useful guidelines to design adaptable service-based applications.

7.6.1 Design techniques

It is crucial to specify the principals and analyze the techniques that can be used for the following key steps, to be able to achieve optimal design for adaptable service-based applications [50]:

i. **Modeling adaptation triggers.** In this step we need to be able to examine when the adaptation is needed (monitored property) and the need for the specific adaptation that will take place.
ii. **Realizing adaptation strategies.** To accomplish this step, it is necessary to model the adaption strategies, their properties as well as their aggregation, and associate them with the underlying mechanisms and run-time environment.
iii. **Associating adaptation strategies to triggers.** There are several factors that influence the association between adaptation strategies and adaptation triggers, like the scope of the change and the impact of the change. Additional requirements may address to autonomy (for example when the adaptation is set to function without human involvement) or performance (to measure for expel how fast the adaptation strategy is).

Before proceeding with the presentation of the design techniques, it is important to clarify that the diversity of adaptation need is different from the requirements the adaptation strategies rely on. Following the exact terminology and description that was used by A. Bucchiarone et al. [50], the design approaches and techniques that can be used are:

a. **Built-in adaptation.** When potential adaptation needs, and potential adaptation configurations are fixed and known a priori, it is possible to completely define them at design time. The specification may be performed by extending the standard SBA notations with the adaptation-specific tools using ECA-like (event condition- action) rules, or aspect-oriented approaches. Typical strategies suitable for such adaptations are: service substitution (by using a predefined list of alternative service), re-execution, compensation, re-composition (by using predefined variants) and fail.

b. **Abstraction-based adaptation.** If the adaptation needs are fixed, but the possible configurations in which adaptation is triggered, are not known a priori, the concrete adaptation actions cannot be completely defined at design time. In such a case, a typical pattern is to define an abstract model of a service-based application and a generic adaptation strategy, which are then made concrete at deployment/run-time. For example, it is possible to use the abstract composition model in which concrete services are discovered and bound at run-time based on the context. Alternatively, it is also possible to define at design time only the final goal or utility function and then it is achieved or optimized by dynamic service re-composition at run-time based on the specific environment and available services. Strategies that may be used for such adaptation, are service concretization, service substitution (by dynamic discovery), re-composition (based on predefined goal and utility function), and re-negotiation.

c. **Dynamic adaptation.** It is possible that adaptation needs that may occur at runtime are not known or cannot be enumerated at design time. In such a case, it is necessary to provide specific mechanisms that select and instantiate adaptation strategies depending on a specific trigger and situation. The scenarios in which such adaptation is needed may include modifications or corrections of business process instances via ad-hoc actions and changes performed by business analyst, changes in the user activities that entail modification of current composition and creation of new ones. At run-time, these mechanisms are exploited to firstly identify one or more suitable adaptation strategies depending on a concrete situation, secondly define concrete actions and parameters of those strategies, and thirdly execute them using the appropriate mechanisms. This type of adaptation may be built on top of the others to realize specific adaptation needs; the focus, however, is on the mechanisms for extracting specific adaptation strategies and actions at run-time. Accordingly, different strategies may apply here: re-composition, service substitution, and compensation, re-execution, evolution, and fail. The realization mechanisms, however, are different; they may require active user involvement (e.g., for making decisions, for performing ad-hoc changes, etc.).

7.6.2 Adaptation Strategies

Different strategies and techniques can be applied to achieve optimal design and adaptation of SBAs. To reach the highest and better performance of the system, it is essential to identify the most suitable adaptation strategy that will be able to adjust the application's operation to the diverse requirements and contexts. Between the various adaptation strategies there are some strategies that are domain-independent and others that are domain-dependent. Domain-independent adaptation strategies can be applied in nearly every service-based application. Domain-dependent strategies are constrained to particular execution cases. The following description of the most common domain-dependent adaptation strategies are presented as they were introduced by A. Bucchiarone et al. [50] who also described the afore mentioned distinction between domain-dependent and domain-independent adaptation strategies and the identification of adaptation triggers (chapter 7.6.3).

The most common domain-dependent adaptation strategies are [50]:

a. **Service substitution.** Service substitution is the reconfiguration of the service-based application providing a dynamic substitution of a service with another one.

b. **Re-execution.** Re-execution strategy involves the possibility of going back in the process, to a point defined as safe and reliable. When reaching that point the goal is to redo the same set of tasks that were previously performed or to follow an alternative path.

c. **(Re-) negotiation.** The re-negotiation strategy is a simple termination of the service that is used on the requester side as well as a re-negotiation of the service level agreements properties to complex management on reconfiguration activities on the provider side.

d. **(Re-) composition.** This strategy targets to reorganize and rearrange the control flow that links the different service components in the business application.

e. **Compensation.** The compensation strategy is used to define the ad-hoc activities that can undo the effects of a process that fails to be completed.

f. **Trigger evolution.** During the trigger evolution strategy, the insertion of workflow exception that will be able to activate the application evolution, is followed.

g. Log/update adaptation information. This strategy involves storage of all the information concerning the adaptation activities for different goals (e.g., service reputation, QoS analysis, outcome of adaptation).

h. **Fail.** In this case the system reacts to the changes by storing the system status, causing the failure of the service and then re-executing that service.

7.6.3 Identification of Adaptation Triggers

There are several different types of factors and triggers that indicate the need for adaptation in service-based applications. These triggers could address to the component services or the context of SBAs. Concerning the **component services,** changes in the service functionality may be identified, such as variation of the service interface (e.g., signatures, data types, semantics), variation of service interaction protocol (e.g., ordering of messages), and failures. Moreover, changes could also take place in the service quality: service availability, degrade of QoS parameters, violation of service level agreement, decrease of service reputation (e.g., black lists), and so on. When it comes to **contextual triggers**, one can distinguish the following three categories of changes:

i. Changes in the business context, such as changes in agile service networks, new business regulations and policies.

ii. Changes in the computational context, such as different devices, protocols and networks.

iii. Changes in the user context, such as different user groups and profiles, social environment or physical settings (e.g., location/time) as well as different user activities.

Some of these aspects may be interleaved. For example, if a user moves to a new location (i.e., change in the user context), new set of services may be available (i.e., change in the business context) with different bandwidth (i.e., change in the computational context) [50, p. 472].

Every trigger can be associated with several types of the adaption strategies (presented in the previous chapter) that can offer the re-alignment of the application within the framework of the system and/or the context needs. This association is described in the following Table 6 [50].

Adaptation Trigger	**Adaptation Strategy**
Changes in the service functionality	Service Substitution, Re-execution, Re-negotiation, Re-composition, Compensation, Fail
Changes in the service quality	Service Substitution, Re-Negotiation
Changes in the business context	Service Substitution, Re-Negotiation, Re-composition, Trigger Evolution, Log/update relevant adaptation information
Changes in the computational context	Service Substitution, Re-negotiation, Re-composition, Trigger Evolution, Log/update relevant adaptation information
Changes in the user context	Service Substitution, Re-negotiation, Re- composition, Trigger Evolution, Log/update relevant adaptation information

Table 6. Association between Adaptation Triggers and Adaptation Strategies

There are several additional requirements that might be linked with the adaptation triggers, which need to be taken into consideration when trying to choose the most suitable adaptation strategy to apply. Two examples of this kind of requirements, that may affect the decision of the adaptation strategy, are given below:

a. The Scope of the change. When we consider this requirement, it means we must evaluate whether the change affects only a single running instance of the service-based application or this particular change has an impact in the whole model. In the case where the scope of the change influences the whole application system, the "trigger evolution" strategy is suitable to be applied.
b. The Impact of the change. It is essential to consider this requirement as well because it indicates the capability of the application to accomplish its current task. In the case of the impact of the change, such strategies as "re-execution" or "substitution" may apply when the service-based application state did not change, and the task still can be fulfilled. Then again, "compensation", "fail", or "trigger evolution" may be used in the case where there is no alternative to complete the current task.

7.7 The software architecture life-cycle

To facilitate the comprehension of the next chapter that is going to introduce an adaptation framework through possible software challenges, it is preferable to briefly describe first the software architecture life cycle and focus later one of the life cycle's components which is the architectural maintenance.

Software architecture is an abstraction of a software system, which is enables the communication, reasoning, and learning about important system components and properties. Weinreich et al. [61] mention that recent definitions of the term software architecture also tend to include the decisions, knowledge and rationale that lead to a software architecture to conclude in the following definition, that "software architecture is in essence knowledge about a software system, about its elements, its properties, and about the main decisions that lead to its current form” [61]. Hofmeister et al. (2005, 2007 [62], [63]) present a general model for architecture design graphically shown in the following Figure 50.

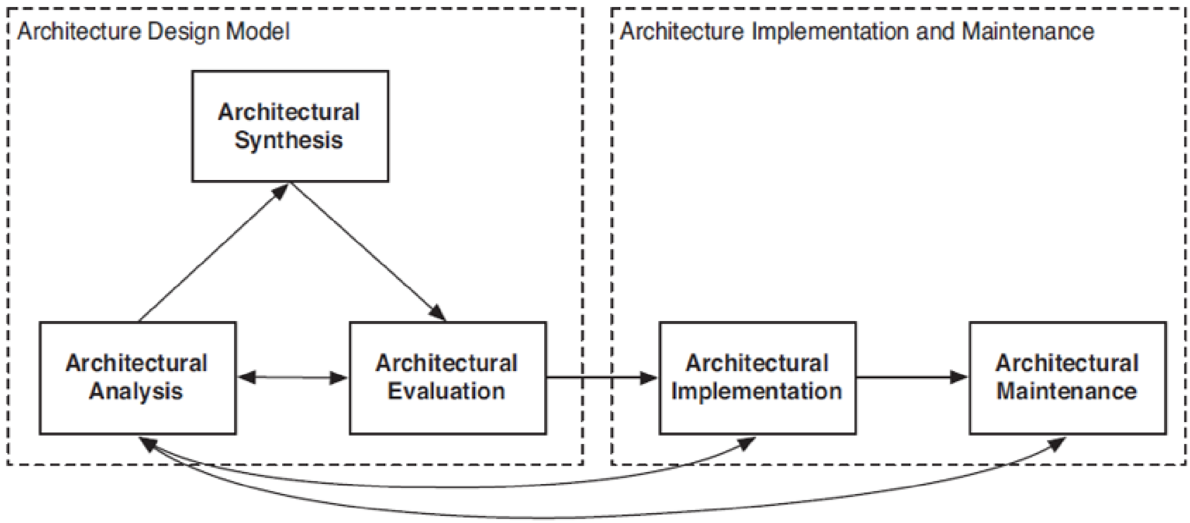

Figure 50. Software Architecture life-cycle

The model introduced by Hofmeister et al. [62] is consisted of three main processes: the architectural analysis, the architectural synthesis, and the architectural evaluation (see left part of Figure 50). These three main processes are briefly described in this chapter as demonstrated in [62] [63] and [61] to offer an overview of the architecture design model structure. The issues that may be needed to be solved, i.e. the architecturally significant requirements (ASRs), are identified from given user requirements and context concerns, during the **architectural analysis**. Potential solutions for these problems are developed during the process of the **architectural synthesis**. Architecturally significant requirements that have been identified during architectural analysis are the basis for the architecture synthesis and therefore the design of the architecture of the software. During architecture synthesis, architectural design decisions (ADDs) are made and solution structures are created. **Architectural evaluation** is a validation step, where the identified solutions are checked against the identified ASRs. This evaluation is a special kind of architecture analysis; whose goal is to check whether the architecture design has a reference to all of the requirements that have been defined during the architectural analysis step. Architecture evaluation is typically performed after architecture design and before architecture implementation. In agile settings, however, architecture evaluation also needs to be performed incrementally and intertwined with the other activities of the architecture lifecycle. The result of the whole process is a validated architecture, which is used as input for the next steps in the software life cycle. As Hofmeister et al. point out [63], these three activities are not performed sequentially but in "small leaps and bounds as architects move constantly from one to another", since it is simply too complex to address all concerns at once.

In the right part of Figure 49 the architectural implementation and maintenance are shown. The **architectural implementation** is the process of realizing the architecture in the system implementation. The main task during this activity is to ensure that the implemented architecture conforms to the intended architecture, which can be reduced to the problem of checking that the implementation conforms to the provided architecture description as indicated in the research of Weinreich et al. [61].

7.8 Adaptation framework through software challenges

As in all other systems, software systems are consisted of various components that interact with each other within the application. The growth of the complexity and the space of those applications lead to the adoption of developing methods that re-use previously developed components and extend already existing functionalities of the previous system, rather than building from scratch a completely new application. Of course, in order to re-use those interacting system parts towards the development of new applications, it is crucial to investigate and examine each individual part and understand the way every part is linked, and it is interacting with the rest of the system parts (see Kell et al.) [64].

Assembling services shared by third parties' developers, is widely adopted in service-based applications since those systems are software applications and are composed of software services. As explained in previous chapters, it is needed, and it is crucial to have adaptable service-based applications and be able to choose the most suitable services in every specific case. Since services in SBAs are shared by third parties, it is not ensured that those services that are provided and required from the application will be accessible when needed. It is important to highlight as well that functional and non-functional parameters, such as the cost or the quality of services, may change without notice. SBAs may be required to adapt for many reasons such as business agility or failure recovery. While adapting, it may be needed to replace or reconfigure services within an SBA through self-adaptation or through manual adaptation [51].

The challenges that arise when developing a service-based application that needs to be adaptable, are both technical and software process oriented. On one hand, the technical challenges may arise during the implementation of the mechanisms that are suitable for adaptation. On the other hand, the software process challenges, relate to the development methods required for adaptable service-based applications. Some of the software process challenges are very briefly mentioned here: interoperability, security, performance install-ability, and maintainability. Additionally, the risks to the safety and reliability of the software, due to software requirements and requirement changes, shall be assessed when considering these software challenges. The focus of this chapter is on the software process challenges following the S-Cube life-cycle model from Lane et al. research view. The major difference between traditional Software Development Life Cycles (SDLC), such as waterfall or spiral [65], and the S-Cube life-cycle model, is the adaptation cycle of the latter [51].

The maintenance in traditional software engineering, usually involves redeveloping parts of an application, hence it is a more expensive process compared to the adaptation of service-based applications process that usually involves the straight substitution of component services. On the other hand, there are commonalities between adaptation and maintenance driven from the fact that both involve the modification of an application. The contribution of this research view (Lane et al., 2013 [51]) to the S-Cube life cycle is to elicit adaptation activities from existing service-based development approaches. This research team has noticed that “many software engineering reference life-cycles and assessment models do not make direct reference to software maintenance” and they observed that “there is little or no coverage of the maintenance process in the major assessment models despite the fact that software maintenance can take up to 60% of the time [66] and 70% of the budget [67] of a software project. April et al. [68] propose a Software Maintenance Maturity Model (SMMM) that can be used as an add-on to the CMMI™”.

Among the several definitions of software maintenance this research approach agrees that it is the process of changing software after initial delivery and outlines the five most recognized types of software maintenance:

i. *Perfective Maintenance* is performed to improve performance or maintainability.
ii. *Corrective Maintenance* is carried out in response to system failures.
iii. *Adaptive Maintenance* is carried out in response to a change in operating environment or in response to new functionality requirements.
iv. *Preventive Maintenance* is maintenance carried out in a system to detect future errors in a software product.
v. *Emergency Maintenance* is an unplanned maintenance that is carried out to keep a system operational.

As mentioned before, the maintenance process of software engineering has resemblances with the adaptation process of SBAs. In more detail, during the maintenance process the following actions take place: the identification of enhancements and the resolution of software errors, faults, and failures. The requirement for software maintenance initiates Software Life Cycle Process (SLCP) changes.

The SLCP is re-executing the following maintenance activities [65]:

a. Identify Software Improvement Needs
b. Implement Problem Reporting Method
c. Reapply SLC

It is interesting at this point to present as well another adaptation framework for SBAs. Using the OASIS/OSOA standard *Service Component Architecture*(SCA: http://www.oasis-opencsa.org/sca), to represent service-oriented software architecture models, Mirandola et al. [69] propose an optimization process that supports the adaptation of service based applications including both static and dynamic aspects at runtime and at re-design time. SCA defines the way the heterogeneity of the various services affects the assembly, description and deployment of those services in a graphical (through visual notation developed with the Eclipse-EMF environment) and XML meta-data driven way. This graphical representation can be accomplished independently of the implementation languages and the deployment platforms and it is shown in Figure 51 [69].

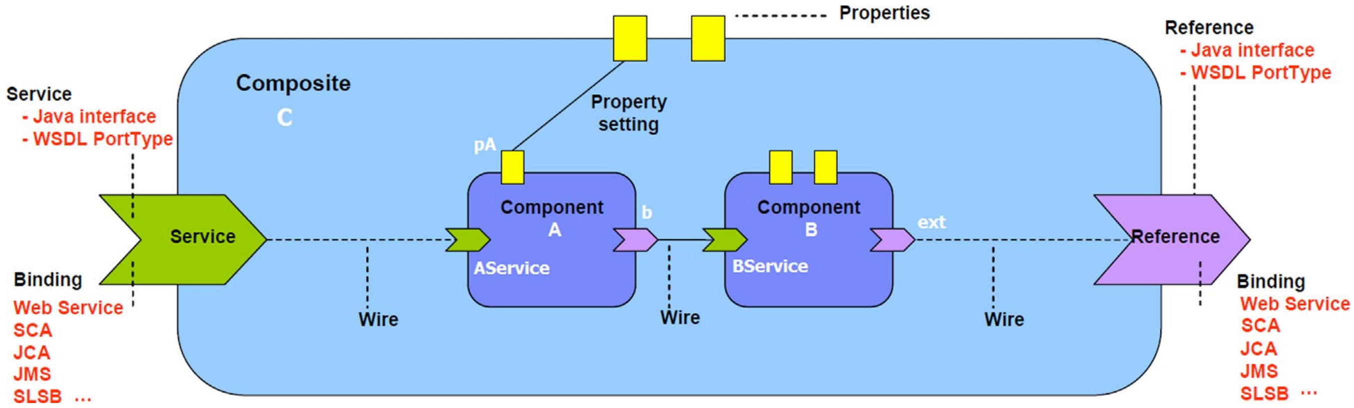

Figure 51. Service Component Architecture

The adaptation mechanism that is followed by this SCA-based framework to achieve adaptation, involves the following adaptation actions [22]:

i. Adding or removing components, component services, references, properties, and wires (component interactions).
ii. Changing a component implementation (but keeping its shape).
iii. Changing component properties values.
iv. Changing SCA domains (components re-deployment).
v. Changing the component interaction style in synchronous and asynchronous, unidirectional or bidirectional.

The optimization process that is proposed in this research (Mirandola et al. [69]) suggests how to adapt the architecture of a service based application by choosing the right equilibrium between functional and non-functional requirements, creating the so called "adaptation space exploration" which is interesting for further study when service-oriented applications may require adaptation to tackle the changing user needs, system faults, changing operational environment, resource variability, etc.

7.9 Case studies

In this chapter the focus is on the different scenarios from various industries or environments where the design for adaptation activities can be performed, as they were presented by Bucchiarone et al, [50]. The factors that trigger adaptation, the types of adaptation realization that are suitable in each scenario and the appropriate strategies for adaptation, are presented in this chapter accordingly to the various characteristics of every case study.

I. **Automotive case study.** In this case study, the complex supply-chain business processes in the automobile production domain are examined. The activities of the processes we meet in the automotive production include ordering and importing automobile body parts from suppliers, manufacturing activities, customization of the specific products according to the needs of the customers, etc. These processes are usually time-consuming with possible long duration and they involve a wide range of enterprise services provided by organizations such as various suppliers, logistics providers, warehouses, and regional representatives. All these participate in an Agile Service Network (ASN) where they rely one on each other services in a dynamic way. The critical changes that require adaptation in this scenario range from instance-specific problems (e.g., failures and SLA violations, specific customers) to the changes that affect the whole SBA (e.g., changes in business context). In the former case, it is possible to apply built-in adaptation and define the reactions at design-time by completely describing the corresponding strategy (compensation activities, process variants for different customers) or its parameters (for SLA re-negotiation, for service substitution). In the latter case, the specific adaptation strategy is chosen at run-time as the effect of changes on the system is not known. In the business settings, such a choice can hardly be automated; the business requirements and decisions require human involvement. In particular, business analysts make decisions on triggering evolution and/or on how the running process instances should be changed (i.e., ad-hoc process modifications).

II. **Wine production case study**. In the wine production application domain, the activities of vineyard cultivation handling, the control of grapes maturation, their harvesting and fermentation rely on extensive use of a service-based application which is realized on top of a Wireless Sensor and Actuator Network (WSAN). In this context, sensors and actuators are seen as service providers able, respectively, to report information regarding the state of the vineyard and to execute some specific actions. These devices are not fully reliable. They may crash, run out of battery, or provide incorrect information. This may happen due to changes in physical context (e.g., humidity) or to the activation of new measurement activities (e.g., depending on the season). In this scenario the dynamically changing state of the WSAN network requires continuous monitoring and optimization of the resource usage. For this purpose, the adaptation should include the following steps: (i) re-arrange the sensor network in order to minimize the sensor energy consumption, and (*ii*) optimize the modes, in which the sensors operate, e.g., by optimizing the data transfer frequency. While the latter solution requires domain-specific realization mechanisms, the former may be achieved by dynamic re-composition of services to minimize of the utility function corresponding to the energy consumption.

III. **Mobile user application case study.** In the mobile industry domain, an increasing number of modern applications aims to give end users access to various services through their mobile devices. Such services include SMS service, voice service, email service, pay-per-view movie service, route planning, transport ticket booking, services for accessing social networks, and a wide range of information services mashed up by those applications. These several services are functional and non-functional aspects of the SBA and it is necessary to be able to identify available services that can be used to support the functional and non-functional aspects of the system, and to develop design models of the system based on the characteristics of existing services [70]. In this scenario, the SBA should adapt firstly to the changes in its context (e.g., changing location and time, different user settings), and secondly to the changes in the user activities and plans. The former may be very dynamic. This means that different services may apply for different locations or user settings. Abstraction-based adaptation is indeed required in this case: the abstract activities (e.g., buy a ticket for local transportation) are defined at design-time and made concrete at run-time using service concretization techniques (e.g., buy a ticket using online service of public transport company). To deal with the changes in user activities and plans, it is necessary to understand the impact of those changes on the current processes and state of the SBA (i.e., perform dynamic adaptation). Depending on the outcome, different adaptations may apply (e.g., compensate or re-compose some tasks, fail). By comparing this mobile user application case study with the automotive case study, one can easily locate a main difference between them. In the case of automotive scenario, the business analysts are high-level domain experts but in the mobile user application scenario the decisions cannot be delegated to the mobile user, as they may have no expertise on the low-level technical details of the service-based application. Therefore, it is necessary to design such decision mechanisms that at run-time may reason on the specific situation in order to reveal an appropriate strategy and its parameters (e.g., to decide whether re-composition may be done, to derive concrete composition goal and the corresponding composition, etc.).

In the following table 2 [50], a summary of all the different adaptation aspects is presented in reference with the aforementioned case studies. For each scenario, this table presents the overview of the properties that characterize the main services that are provided in each case study, the possible adaptation triggers, the design approach that could be followed and the adaptation strategy that should be implemented.

Case study	**Properties**	**Adaptation Trigger**	**Design Approach**	**Adaptation Strategy**
Automotive	Stable context and potential partners, long-running SBAs, diversity of adaptation needs, decisions require human involvement	functional changes, failures, SLA violations, changes in business context	Dynamic adaptation (human-driven); built-in adaptation (for compensation or process customization)	Service substitution (selecting from ASN partners); SLA re-negotiation; re composition by ad-hoc changes of process control/data; re composition by selecting predefined process variants; compensation; trigger evolution
Wine	Fully dynamic and unreliable services, fully autonomous SBA	degrade of service (sensor) QoS	Abstraction-based adaptation	Re-composition of services (to optimize resource utility function), domain-specific actions (e.g., data transfer frequency changes)

Mobile user	Strong dependency from context and goals of users	context changes, changes of user activities	Abstraction-based (for context changes), dynamic	Service substitution (by dynamic discovery); re-composition.

Table 7. Adaptation aspects of three case studies.

The mobile user case study was used by Spanoudakis et al. [70] to describe a service discovery framework that has been developed within the EU 6th Framework projects SeCSE and Gredia. The framework supports design of service-based systems based on existing services and adaptation of service-based systems during their execution due to different situations. It assumes services described from different perspectives and uses complex service discovery queries specified in a XML-based language that was developed.

7.10 CS and DAQ case study

In this chapter we will form a case study for the developed CS and DAQ System.

7.10.1 CS and the DAQ System Architecture Optimal Design

To achieve an optimal design of CS and DAQ SBA, the key steps mentioned in chapter 6.1 are followed:

i. **Modeling adaptation triggers.** In this step the focus is on the need for adaptation of the Data Acquisition System of EDUSAFE. The DAQ system needs to be able to accept various types of sensors for acquiring the measurement data from the ATLAS environment. In this case this service of the system needs to be adaptable to the changes that may occur due to the addition of new sensors.

ii. **Realizing adaptation strategies.** To accomplish this step, it is necessary to model the adaption strategies, their properties as well as their aggregation, and associate them with the underlying mechanisms and run-time environment.

iii. **Associating adaptation strategies to triggers.** As described in the previous chapter 6.1, there are several factors that influence the association between adaptation strategies and adaptation triggers, like the scope of the change and the impact of the change. Additional requirements may address to autonomy (for example when the adaptation is set to function without human involvement) or performance (to measure for example how fast the adaptation strategy is).

The identification of possible adaptation triggers for both cases of the Control System and the Data Acquisition System is analyzed in the next chapter 7.10.2. Adaptation strategies are presented in chapter 7.10.3. The various design techniques for EDUSAFE CS and DAQ are described in chapter 7.10.4. Finally, in chapter 7.10.5 the adaptability and scalability of the CS and DAQ in other industries is explored.

7.10.2 Identification of possible adaptation triggers for EDUSAFE CS and DAQ System

7.10.2.1 DAQ System adaptation triggers

There are specific possible changes that may occur and could form the adaptation triggers for the DAQ system of EDUSAFE:

a. Changes in the computational context. Different devices such as gamma ray camera, vision cameras or sensor boards may be required to enlarge or modify the data the system acquires. Consequently, different protocols and networks could be required to comply with DAQ system requirements.
b. Changes in the business context. New business regulations and policies may occur due to possible change of environment, such as the workers position change in different radioactive environments.
c. Changes in the user context. Different user groups and profiles

could be added to the DAQ system, physical settings such as location of the worker may change, as well as different user activities both for the worker and the supervisor may be needed. All these changes could form adaptation triggers for the DAQ system.

7.10.2.2 CS System adaptation triggers

Additionally, there are specific possible changes that may occur and could form the adaptation triggers for the CS system of EDUSAFE:

a. Changes in the computational context. Different devices such as gamma ray camera, vision cameras or sensor be required to enlarge or modify the data the system acquires. Consequently, the CS needs to adapt to those changes and must be able to control all these new or modified information. The communication between the supervisor and the worker must be scaled up to the new requirements.
b. Changes in the business context. New business regulations and policies may occur due to possible change of the environment in which the worker performs his activities. In this case the CS must comply with these new regulations that could be for example to adapt the thresholds of the radiation background monitoring depending on the location and time of the worker, from a regulation and policies perspective.
c. Changes in the user context. Different user groups and profiles that supervise the worker, or physical settings (e.g., location/time of the worker) as well as different user activities may occur as it was mentioned in the data acquisition system adaptation triggers as well. In this case once again the CS must comply with these new settings that could be for example to adapt the thresholds of the radiation background monitoring depending on the location and time of the worker, from a context perspective. This means the CS must easily accept different user groups that are set to supervise the

personnel and set multiple profiles for those users. Moreover, the CS must be able to respond fast in providing all the information the supervisor or the worker may need even if the physical setting that were mentioned before, such as location and time, vary in a rapid and complex way. The complexity of the change of these physical settings is strongly linked of course with the existing complexity of the underground radioactive working environments.

7.10.3 Adaptation strategies for EDUSAFE CS and DAQ

In this chapter all the possible adaptation strategies that can be applied in the EDUSAFE case concerning the CS and DAQ system, are analyzed.

The most common domain-dependent adaptation strategies introduced in chapter 6.2 that could be applied in EDUSAFE case study are:

a. **Service substitution.** Service substitution is the reconfiguration of the service-based application providing a dynamic substitution of a service with another one. In this case the CS and DAQ system could replace the audio communication between the user and the supervisor with modified AR content or simple text instructions, in case the noises of the working environment do not allow the audio communication.

b. **Re-execution.** Re-execution strategy could be also applied in the case of the control system's functionalities involving the possibility of going back in the process, to a point defined as safe and reliable. When reaching that point the goal is that the users (workers) redo the same set of tasks that they were previously performing or to follow an alternative path (optimized instructions) to ensure safe execution of each task.

c. **(Re-) negotiation.** The re-negotiation strategy could be adopted by simply terminating the service that is used on the requester side as well as renegotiating the service level agreements properties to complex management on reconfiguration activities on the provider side. An assessment of those reconfiguration activities must be further researched.

d. **(Re-) composition.** This strategy targets to reorganize and rearrange the control flow that links the different service components in our application. This strategy could be combined with the firstly proposed strategy of service substitution, as the AR content display service and the audio communication service, which are (or may be) linked services, can be adjusted to a smart and optimized control flow.

e. **Compensation.** In case the process fails to be completed, the compensation strategy can be used to define ad-hoc activities that can undo the effects of such failure.

f. **Trigger evolution.** During trigger evolution the role of inserting workflow exceptions could be obviously crucial to activate the evolution of the CS and DAQ system.

g. **Log/update adaptation information.** One part of this strategy is already used and followed in the case of EDUSAFE CS and DAQ system. All the information concerning the DAQ process and control system processes is logged and saved in a DB. To achieve the optimum use of this strategy we could log and update a more holistic view of the application operation. Combining all the stored information could lead on an interesting and useful QoS analysis.

h. **Fail.** In this case the system can react by storing the system status, closing the service and re-executing that service.

7.10.4 Design techniques for EDUSAFE CS and DAQ

Following the terminology and description that was used by A. Bucchiarone et al. [2], that was presented in chapter 6.1, the design approaches and techniques that can be used in the case of the EDUSAFE project, are:

a. **Built-in adaptation.** There are some of the potential adaptation configurations that are fixed and known a priori in EDUSAFE case. For example, the system must be able to accept several gamma cameras or vision cameras as well as various types of sensors. The system should also support the addition of multiple and simultaneous worker sessions to provide as an additional functionality the supervision of multiple workers. In this case all of these potential adaptation configurations can be defined at design time. The specification may be performed by extending the standard service-based application notations with the adaptation-specific tools using ECA-like (event condition- action) rules, or aspect-oriented approaches. Typical strategies suitable for such adaptations are: service substitution (by using a predefined list of alternative service), re-execution, compensation, re-composition (by using predefined variants) and fail.

b. **Abstraction-based adaptation.** Let's assume that the adaptation needs for the EDUSAFE system are fixed, such as the need to accept various types of sensor devices, but the possible configurations in which adaptation is triggered are not known a priori, for example the bandwidth of the new device or even the format of the data exchange between the sensor device and the data acquisition system (for example JSON formatted messages). In this case, the concrete adaptation actions cannot be completely defined at design time. A typical pattern in this scenario, is to define an abstract model of a service-based application and a generic adaptation strategy, which are then made concrete at deployment/run-time. For example, it is possible to use the abstract composition model in which concrete services are discovered and bound at run-time based on the context. Alternatively, it is also possible to define at design time only the final goal or utility function and then it is achieved or optimized by dynamic service re-composition at run-time on the basis of based on the specific environment and available services. Strategies that may be used for such adaptation, are service concretization, service substitution (by dynamic discovery), re-composition (based on predefined goal and utility function), and re-negotiation.

c. **Dynamic adaptation.** There are cases in which the adaptation needs that may occur at runtime are not known or cannot be enumerated at the design time of the application. In such a case, it is necessary to provide specific mechanisms that select and instantiate adaptation strategies depending on a specific trigger and situation. The scenarios in which such adaptation is needed may include modifications or corrections of business process instances via ad-hoc actions and changes performed by business analyst, changes in the user activities, such as adding an additional task for maintenance in the working environment in parallel with other tasks. All of the possible corrections or changes that were mentioned above entail modification of current composition and creation of new ones. During the run-time of the application, these mechanisms are exploited to firstly identify one or more suitable adaptation strategies depending on a concrete situation, secondly define concrete actions and parameters of those strategies, and thirdly execute them using the appropriate mechanisms. This type of adaptation may be built on top of the others to realize specific adaptation needs; the focus, however, is on the mechanisms for extracting specific adaptation strategies and actions at run-time. Accordingly, different strategies may apply here: re-composition, service substitution, and compensation, re-execution, evolution, and fail.

The realization mechanisms, however, are different; they may require active user involvement (e.g., for making decisions, for performing ad-hoc changes, etc.). It may be required for example that the user (worker or supervisor) should make a decision according to each different situation and follow the changes of a dynamic adaptation.

7.10.5 Exploring the adaptability and scalability of the CS and DAQ in other industries

The first step of exploring the adaptability of the CS and DAQ system of EDUSAFE in new and different working environments is to identify the challenges that may occur in those different industries and markets.

Possible industries and markets in which the CS and DAQ System could be applied, are the following:

1. Nuclear Power Plant emergent or planned maintenance
2. Aircraft maintenance/ build
3. Aerospace maintenance activities while in operations
4. Industrial subsea activities
5. Oil extraction platforms (onshore and offshore) maintenance
6. Oil refineries maintenance
7. Ship maintenance/ build
8. Factory Fire Department
9. Mining

All the above different infrastructures are functioning on different environmental conditions that influence the parameters that are needed to design a versatile, scalable and adaptable system.

I examine three cases in which EDUSAFE Control System and Data Acquisition

System could be adapted. To achieve a better understanding of this approach I focus mainly on the changes that could form adaptation triggers in these cases depending on the existing characteristics and functionalities of our application.

Industrial subsea activities may refer to technology and methods applied in marine biology, undersea geology, offshore oil and gas developments, underwater mining, and offshore wind power industries. Complex mechatronic systems for critical applications, such as subsea production systems, require high reliability and a very low probability of failure [71]. “Subsea structures operate in very harsh conditions and their operation and lifetime may be compromised by natural or manmade hazards and accelerated aging processes. Access to these structures for maintenance or repair purposes is difficult, expensive and requires trained personnel and sophisticated equipment”, as mentioned by B. Glisic [72]. One of the systems that play an important role in providing safe working conditions for drilling activities in 3000 m ultra-deep water regions is the blowout preventer system (BOP) [71]. The DAQ and CS of EDUSAFE could be modified to be used in systems in industrial subsea activities, such as BOP, or in maintenance procedures that require communication with trained personnel and supervisors. Focusing on this case lets categorize the changes that could form the contextual triggers for adaptation:

i. Changes in the business context are obviously present. These changes can be in agile service networks, new business regulations and policies accordingly to the Industrial subsea activities that could take place when operating systems such as BOP that need to be very reliable.
ii. Changes in the computational context. Different devices, protocols and networks are required to comply with the Industrial subsea activities’ requirements, such as sensors that acquire the temperature or analyze the needed characteristics of the deep-water regions. These devices may not be fully reliable (may crash or provide incorrect information).
iii. Changes in the user context, such as different user groups and profiles,

physical settings (e.g., subsea location, deep water regions, humidity change) as well as different user activities, for example drilling activities in deep water regions.

Modeling of the above adaptation triggers is the first step for realizing the need for adaptation in the Control System and Data Acquisition system when interested in using those specific applications in the Industrial subsea activities. In this scenario the dynamically changing state of the environmental sensors require continuous monitoring. To achieve this, the adaptation should optimize the way for example the various sensors are functioning, ensuring their reliability and stability through domain-specific realization mechanisms. The adaptation could also re-arrange the workflow of the data acquisition of the sensors (to minimize for example the sensor energy consumption). The latter approach could be achieved by dynamic re-composition of services to minimize the energy consumption by promoting the services that provide the most critical sensor measurements. The abstract activities, such as the location characteristics are defined at design-time and are specified more accurately at run-time. Therefore, abstraction-based adaptation is indeed needed in this case.

In the aerospace industry, maintenance, repair and overhaul (MRO) services are normally provided during product usage, as mentioned by Zhu et al. [53]. This indicates that **Aerospace maintenance activities while in operations** is a possible case study where EDUSAFE system including the control system and data acquisition system could be applied. This scenario involves complex supply-chain for importing possible parts that need to be replaced, customizations according to the needs of the customers, etc. Built-in adaptation is possible to define the reactions at design time. There are changes that may affect the whole SBA where the requirements require human involvement. In that case, dynamic adaptation can be applied.

Thermal power stations in which the heat source is a nuclear reactor are Nuclear Power Plants (NPPs). Recently, instrumentation and control (I&C) systems in nuclear power plants have been replaced with digital-based systems due to the fact that they offer important and critical advantages over conventional analog systems [26]. Man–machine interface system (MMIS) architecture is used in NPPs to achieve technical self-reliance, based on a network communication system for both intra-system and inter-system connections [73]. Hun Lee et al. mention “As one of the systems in the developed MMIS architecture, the Engineered Safety Feature-Component Control System (ESF-CCS) employs a network communication system for the transmission of safety-critical information from group controllers (GCs) to loop controllers (LCs) to effectively accommodate the vast number of field controllers”.

The need for transmitting information for safety reasons is critical for the aforementioned Engineered Safety Feature-Component Control Systems that are used in Nuclear Power Plants stations. This need exists of course in several activities that are conducted in Nuclear Power Plants stations. A group of these activities is conducted during **Nuclear Power Plant emergent or planned maintenance**. In this case of maintenance procedures, the EDUSAFE Control and Data Acquisition System could be used to enhance the functionalities of the ESF-CSS used in NPPs. One of the most important contextual triggers for adapting our systems in this case are the changes in the safety requirements that may occur. Similarities between those two systems (Engineered Safety Feature-Component Control System and EDUSAFE CS plus DAQ system), help the adaptation process through the optimization of pre-existing systems. An example of those similarities may be the common devices that the workers use, such as dosimeters, the data of which, are critical to be acquired and monitored in both cases.

Case study	Properties	Adaptation Trigger	Design Approach	Adaptation Strategy
Industrial subsea activities	Complex mechatronic systems, high reliability, low probability of failure	New business regulations and policies, deep water sensor variation, region characteristics changes, degrade of service (sensor) QoS	Abstraction-based adaptation	Re-composition of services (to optimize data acquisition process), domain-specific actions (e.g., data transfer)
Aerospace maintenance activities while in operations	Stable context and potential partners, long-running SBAs, diversity of adaptation needs, decisions require human involvement	functional changes, failures, SLA violations, changes in business context	Built-in adaptation (design-time) Dynamic adaptation (human-driven)	Service substitution, SLA re-negotiation; re composition by ad-hoc changes of process control/data; re composition by selecting predefined process variants; compensation; trigger evolution

Nuclear Power Plant emergent or planned maintenance	Man–machine interface system (MMIS), technical self-reliance, a network communication system for both intra-system and inter-system connections	the changes in the safety requirements	Built-in adaptation (design-time)	Service Substitution, Re-Negotiation, Re-composition, Trigger Evolution, Log/update relevant adaptation information

Table 8. Adaptation aspects of three case studies possible for EDUSAFE CS and DAQ technology exploitation

Built-in adaptation could be used since the potential adaptation needs and potential adaptation configurations could be fixed and be known a priori. It is therefore possible to completely define them at design time of the new service-based application that could be provided in this case. Service Substitution, Re-Negotiation, Re-composition, Trigger Evolution are some of the adaptation strategies that could be adopted.

The adaptation aspects of the three-case studied that are presented in this Chapter, are shown in Table 8 including the basic properties and the possible adaptation triggers that exist in each case, as well as the design approach (type of adaptation that could be followed) and the adaptation strategies in each case.

7.11 Conclusions

In this Chapter we formed a case study from a service-based application adaptation perspective. This could consist a positive feature in the development of the CS and DAQ system, as it provides a guideline for further future system optimization, derived from its adaptation needs. Moreover, all the design techniques that are described and could be followed, are enhancing the adaptability and scalability of the developed system, which is crucial for exploiting this technology in other markets and different extreme environmental conditions. Further research can be conducted to investigate the adaptation of this service-based application in the rest of the mentioned markets and industries.

Chapter 8

8 Conclusions and Future Research

8.1 Conclusions and thesis contributions

This thesis presents the development of the Control System (CS) and Data Acquisition (DAQ) System that was designed to supervise in real-time the personnel in the ATLAS experimental infrastructure of CERN where the working conditions are dangerous mainly due to high radiation.

Design and optimization of the Control System (CS) and Data Acquisition (DAQ) System

In Chapter 5, the development and optimization of the Control and the Data Acquisition system is presented. The developed DAQ System can acquire various types of sensor data that measure both environmental and personal data. Most importantly the personal radiation dose can be acquired in real time and monitored. In case a value exceeds a certain predefined threshold, the Control System generates danger alarms.

Increasing the personnel safety by Remote Monitoring

This system increases personnel safety while working in extreme environmental condition such as the ATLAS cavern where radioprotection is crucial. It provides supervision of multiple users performing simultaneously various complex tasks in the cavern and can decrease their stress by guiding them in real time through audio and video communication.

Integrating the CS, DAQ and RMS to form a Safety System

The Safety System we propose consists of the three sub-systems: the Data Acquisition System (DAQ), the Control System (CS) and the Monitoring System

(RMS). These sub-systems are integrated to provide real-time guidance to the users and enhance human radioprotection. There are four external systems that interact with the Safety System, namely PTU, MPSS, Gamma Camera and AR System. This Safety System was developed, installed and tested in the ATLAS experimental infrastructure, to ensure human radiation protection, enhance the personnel's safety against anticipated hazards and provide real-time monitoring and guidance.

Adaptability and scalability of the CS and DAQ System

In the previous chapter, the adaptability and scalability of the CS and DAQ System was proven through a service-based application point of view. Through this study we can conclude that the CS and the DAQ System can be adapted in various industries and experimental infrastructures such as:

i. Industrial subsea activities
ii. Aerospace maintenance activities while in operations
iii. Nuclear Power Plant emergent or planned maintenance

Further research can be conducted to investigate the adaptation of this service-based application in all the other markets and industries, such as:

a. Aircraft maintenance/ build
b. Oil extraction platforms (onshore and offshore) maintenance
c. Oil refineries maintenance
d. Ship maintenance/ build
e. Factory Fire Department
f. Mining

8.2 Proposals for Future Research

During maintenance periods the ATLAS environment is accessible to the personnel to perform all needed actions for maintenance or upgrade of the detector. One of the most important issue that rises while accessing ATLAS in these periods, is the difficulty in localizing people due to the complexity of the environment. ATLAS is 46 meters long, 25 meters in diameter, and weighs about

7,000 tons; it contains some 3000 km of cable. The safety of the personnel can be compromised since the personnel might be isolated or even lost. This problem has been treated so far with a dedicated system called “Finding Persons Inside the ATLAS areas” that was based on a network of passive infrared sensors which are read out by specific front-end electronics [74]. The idea for future work is to develop a new localization system that will be based only on the radiation measurements that are acquired from every person on the working field. A Data Mining algorithm, the Bloom filter, could be used for achieving the goal of this idea.

The knowledge of radiation maps is a valuable input for this research since we can compare the radiation values of these maps with the real-time measurement of radiation dose that the DAQ server receives. As described in previous chapter, the DAQ system acquires every five seconds a JSON message with the radiation dose of the worker. Radiation maps provide values that are the estimated dose rates in μSv/h on a specific position (R, Z) inside the ATLAS cavern as shown in Figure 37. These data are available for specific cooling time, for ex. t=84days. While comparing each acquired measurement with the known value is simple, the challenge lies on the available memory and computational time. By using the Bloom Filter algorithm, we could overcome these issues and achieve localization of the personnel without installing any additional HW components on the field. For better understanding we describe the Bloom Filter algorithm, starting from a brief introduction to data stream mining, since the radiation measurements our system acquires, are received as streams.

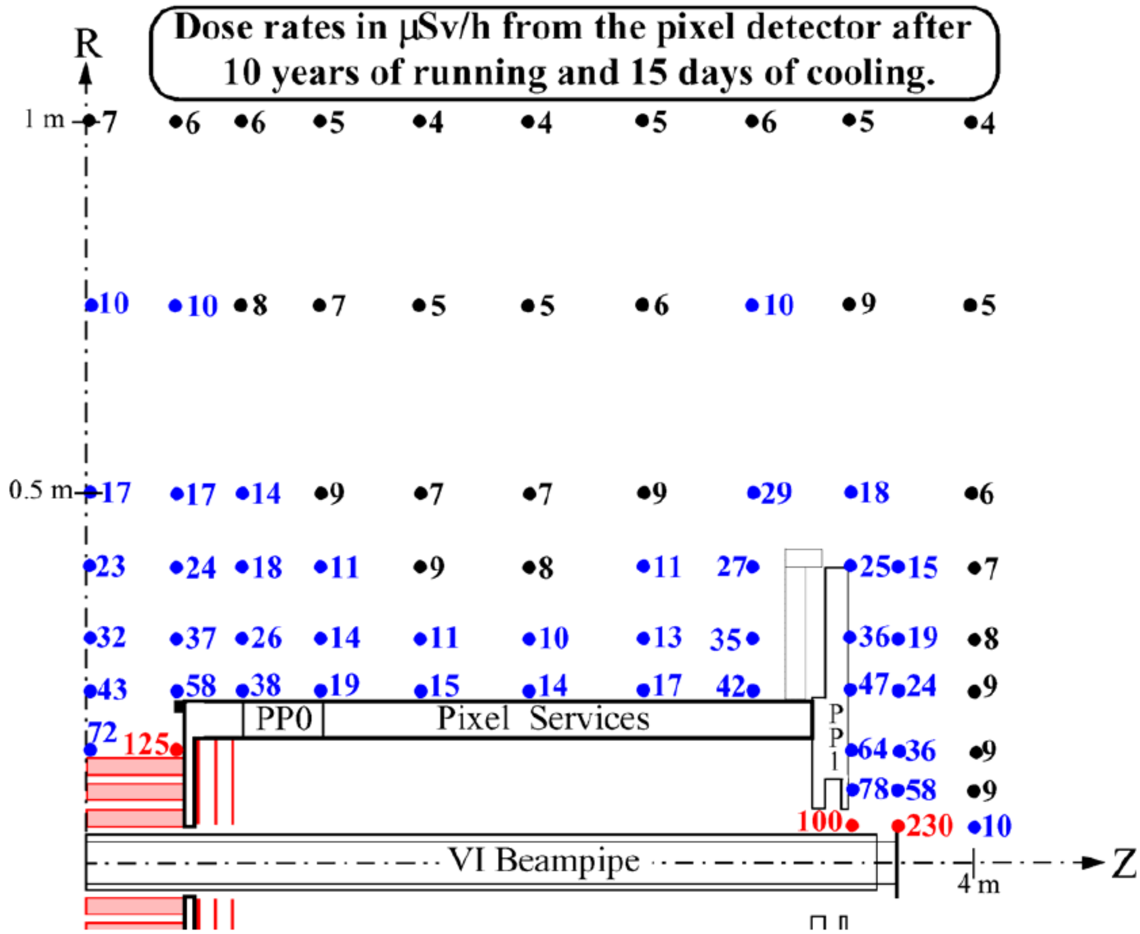

Figure 52. Dose rates in mSv/h from the pixel detector [18]

8.2.1 Mining Data Streams

Let's assume that we need to deal with data that arrive in a stream or streams and they are lost if not processed immediately. Additionally, we assume that the data rate is very high and it is not feasible to store in active storage. In order to process these data streams we need algorithms that can summarize the stream, such as creating a sample stream, filtering a stream to reduce the "undesirable" elements, estimate the number of different elements reducing the storage use, or focus on fixed-length "window" consisting of the last n elements for some typically large n [75]. Some examples of stream sources are listed below as mentioned by Ullman et al.:

i. Sensor Data. We might be interested in calculating the surface height in the ocean through GPS signals. The amount of data would be very big since the surface height varies rapidly maybe every tenth of a second. If it sends a 4-byte real number
ii. Image Data. These data could be big as well as there are millions of images acquired in a very short time in several circumstances such as Satellites and Surveillance cameras. For example, only in London there are six million cameras each producing a stream.
iii. Internet and Web Traffic. Web sites receive streams in various ways, for ex. Google receives several hundred million search queries per day, Yahoo accepts billions of clicks per day.

Stream queries can be either ad-hoc queries where questions are asked once about the current state of a stream, such as requesting the maximum value seen so far in streams or standing queries which are queries that are permanently executing and produce outputs at appropriate time, for example outputting an alert whenever a sensor value exceeds a certain threshold.

8.2.2 A Data Stream-Management System

A stream processor can be considered as a data-management system, in analogy to a database-management system and the main difference between data and stream processing is that the rate of input streams is not under the control of the system. For example, Google receives several hundred million search queries per day in the form of streams and cannot control the input rate. A high-level architecture of a data stream management system is shown in Figure 1 [75]. First, there is the stream processor which is the SW that executes the queries. This processor can store some standing queries and allow ad-hoc queries to be issued by the user. Several streams are entering the system as depicted in Figure 37 and on the right end of the stream are the most recently arrived values while time goes backward to the left. The system outputs in responding of the queries. Usually there is some archival storage that is so massive that it is not possible to do more than simply storing the input streams.

There is also a limited working storage (main memory, even disk) that we assume it stores essential parts of the input streams in a way that it supports a possible fast query [1].

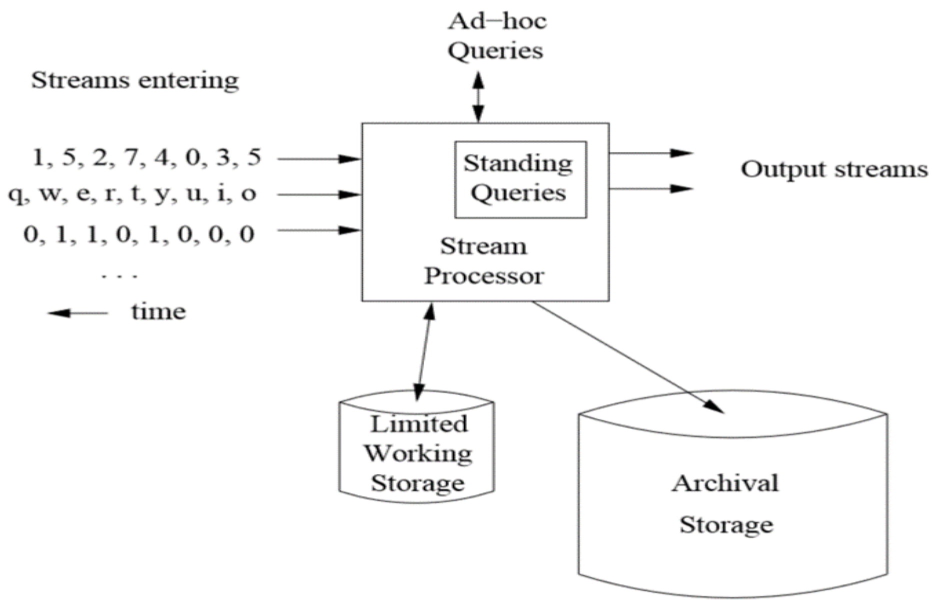

Figure 53. A data-stream management system [75].

8.2.3 Bloom Filter Algorithm

A Bloom filter is a probabilistic data structure, that is used to test whether an element is a member of a set and it was conceived by Burton Howard Bloom in 1970, *B.H. Bloom. Space/time trade-offs in hash coding with allowable errors.* [76]. One of the benefits of this technique is that by using it we can compute useful properties of an entire stream without storing it entirely. In order to understand better this algorithm, we describe this filter method step by step as it was presented by Ullman et.al [75].

A Bloom filter consists of:

1. An array of **n** bits, initially all 0's.
2. A collection of **k**-independent hash functions h_0, h_2..., h_{k-1}.
3. A set **S** of **m** key values.

The purpose of this method is to reject most of the keys that are not in S, while allowing all stream elements whose keys are in S [1]. The first step is to initialize a bit array by setting all bits to 0. Then we choose k-independent hash functions. Each hash function takes a stream element and produces a value that is one of the positions of the array. Therefore, each hash function maps "key" values to n buckets, corresponding to the n bits of the bit-array. Then, each bit that is hi(K) for some hash function hi and some key value K in S, is set to 1. The final step is to test the key K that arrives in the stream, by checking that all of the h1(k), h2(k),…h(k-1) are 1's in the bit array. If all are 1's, then the stream goes through. If one or more of these bits are 0, then K could not be in S and we reject the stream element [1]. The process of Bloom Filter is shown in Figure 54.

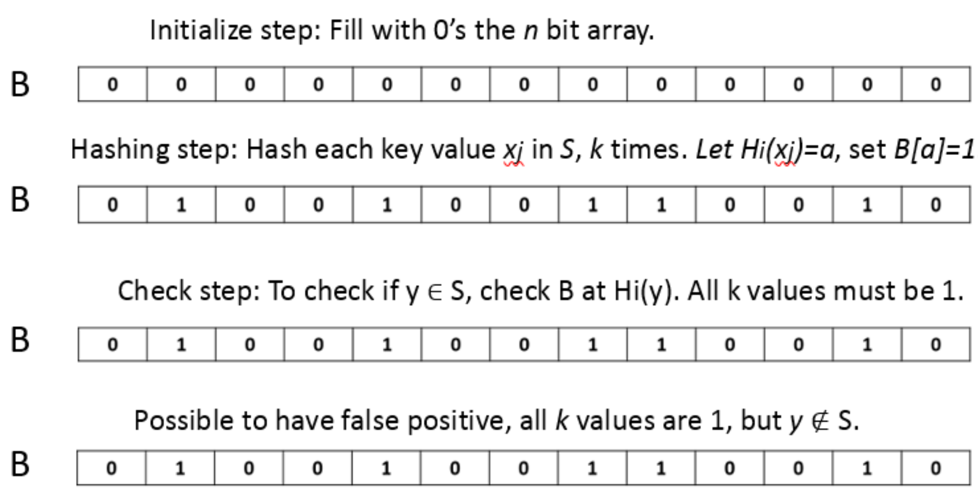

Figure 54. Bloom Filter process

8.3 Error types

There are two possible error types when dealing with set membership:

1. False Positive (FP): answering "is there" on an element that is not in the set.
2. False Negative(FN): answering "not there" on an element that is in the set.

False negative never happens when using Bloom Filters.

Design of Bloom Filtering

Bloom filter has to be designed in a way that the probability of false positive is very small.

First, the probability of an incorrect detection? Check reference 2->search for ref for math background.

Consider a particular bit j in the array, where $0 \leq j \leq n-1$. The probability that the $h_i(x)$ does not set bit j is:

$$P_{h_{i\sim H}}(h_i(x) \neq j) = \left(1 - \frac{1}{n}\right).$$

This means that this hash function does not map to j. Moreover, the probability that bit j is not set by anyone of the hash functions and anyone of the m-elements is:

$$P_{h_{1,\dots,} h_{k\sim H}}(Bit(j) = F) \leq \left(1 - \frac{1}{n}\right)^{km}.$$

But since $\left(1 - \frac{1}{n}\right)^n \approx \frac{1}{e} = e^{-1}$, then $\left(\left(1 - \frac{1}{n}\right)^n\right)^{km/n} \approx (e^{-1})^{\frac{km}{n}} = e^{-km/n}$ and therefore the probability of FP is calculated when all k bits of the new element are already set:

$$\left(1 - e^{-\frac{km}{n}}\right)^k.$$

Analysis of the FPP/ choosing the optimal k

Let's analyze the False Positive Probability in order to understand which is the needed size n of the array and how many k hash functions we should use to achieve small probability of failure, meaning small FPP.

The false positive probability upper bound is: $\left(1 - e^{-\frac{km}{n}}\right)^k$.

The FPP upper bound is minimized by choosing $k = log_e(2)(n/m)$, but $e^{-km/n} = e^{-log_e(2)\left(\frac{n}{m}\right)\left(\frac{m}{n}\right)} = 0.5$.

Then the FPP upper bound is $(0.5)^k = (0.5)^{log_e(2)n/m} = p$.

When k is chosen optimally the probability of failure is $p \approx 0.61^{n/m}$, $log_2(p) \approx \frac{n}{m} log_2(0.61)$, and $\frac{n}{m} \approx \frac{log_2(p)}{log_2(0.61)} = 0.7log_2(1/p)$. For example, to achieve $p < \frac{1}{1000}$, it is sufficient to choose $\frac{n}{m} > 0.7log_2(1000) \approx 7$.

A Bloom Filter example

Let's assume that we have an array of n=11 bits, the stream elements are integers and we choose two hash functions as follows [3]. First $h_1(x)$ is computed in these steps:

a. Take odd number bits from the right in the binary representation of x, meaning position 1 is the low order bit, position 3 is the 4th place position 5 is the 16th place and so on.
b. Looking only at the odd positions we get another binary integer, suppose that integer is i.
c. Compute i mod 11.

$h_2(x)$ is computed in the same way but from the even positions of the binary representations of x.

Suppose three integers arrive at the stream input. Initially, all bits are set to 0. If the first integer is 25, then it's binary representation is 11001 with the odd number positions counting from the right. The first hash function is formed from the odd positions: 101=5 in binary and 5mod11=5, therefore $h_1(25)$ =5. The even positions are 10=2 in binary and 2mod11=2. So, we set: positions 2 and 5 of the array to 1. We are counting positions from the left end and starting at 0. Likewise, for the second integer 159 we have 0111=7 and 7mod11=7 and even positions form 1011=eleven in decimal and 11mod11=0. All these steps are shown in the following Table 9.

Stream element	h1	h2	Filter contents
			00000000000
25 = 11001	5	2	00100100000
159 = 10011111	7	0	10100101000

Table 9. Filtering process. The odd number positions in the binary representation of the stream element counting from the right is in black and the even positions are in red color.

In conclusion, this PhD research can consist the starting point for creating a new localization system for the personnel in the ATLAS cavern, by comparing the incoming dose data streams with the existing radiation maps.

9 Appendix A

9.1 DAQ Server Configuration Files

Through the following configuration file, the connection of the DAQ server with the Database is setup.

apvs/apvs-daq-server/hibernate-example.cfg.xml

```
<!DOCTYPE hibernate-configuration PUBLIC
    "-//Hibernate/Hibernate Configuration DTD 3.0//EN"
    "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
    <session-factory>
        <!-- Database connection settings -->
        <property name="connection.driver_class">net.sf.log4jdbc.DriverSpy</property>
        <property
name="connection.url">jdbc:oracle:thin://@pcatlaswpss01:1521/XE</property>
        <property name="connection.username">xxxx</property>
        <property name="connection.password">yyyy</property>
        <!-- JDBC connection pool (c3p0) -->
        <property name="hibernate.c3p0.min_size">5</property>
        <property name="hibernate.c3p0.max_size">20</property>
        <property name="hibernate.c3p0.timeout">1800</property>
        <property name="hibernate.c3p0.max_statements">50</property>
        <!-- SQL dialect -->
        <!-- Echo all executed SQL to stdout -->
        <property name="show_sql">false</property>
```

</session-factory>

</hibernate-configuration>

9.2 OracleXE instructions

The following instructions are formed in order to install the Oracle XE DB.

```
# Step 1. Getting the required Linux packages
sudo apt-get install alien libaio1
# Step 2. Download and unzip Oracle Database Express Edition 11g Release 2 for Linux x64 from http://www.oracle.com/technetwork/database/database-technologies/express-edition/downloads/index.html
# Oracle Database Express Edition 11g (or just Oracle XE) is the free version of Oracles 11g database released in 2011 and is the second free version of Oracle’s db.
# Don't get confused with the names (release date of the paid versions#release date of the free versions).
cd ~/Downloads/
unzip oracle-xe-11.2.0-1.0.x86_64.rpm.zip
cd Disk1
# Step 3. Transform rpm to deb:
# It’ll probably take a while, but eventually, you should get the following message:
# oracle-xe_11.2.0-2_amd64.deb generated
sudo alien --scripts oracle-xe-11.2.0-1.0.x86_64.rpm
#check to see if you’ve got a chkconfig file under the /sbin directory.
#If not, then you need to create the chkconfig file as follows :
sudo nano /sbin/chkconfig
# The contents of the file are:
#!/bin/bash
# Oracle 11gR2 XE installer chkconfig hack for Debian by Dude
file=/etc/init.d/oracle-xe
if [[ ! `tail -n1 $file | grep INIT` ]]; then
   echo >> $file
   echo '### BEGIN INIT INFO' >> $file
   echo '# Provides:             OracleXE' >> $file
   echo '# Required-Start:       $remote_fs $syslog' >> $file
   echo '# Required-Stop:        $remote_fs $syslog' >> $file
   echo '# Default-Start:        2 3 4 5' >> $file
   echo '# Default-Stop:         0 1 6' >> $file
   echo '# Short-Description:    Oracle 11g Express Edition' >> $file
   echo '### END INIT INFO' >> $file
fi
update-rc.d oracle-xe defaults 80 01 #this is the last line
#set the appropriate permissions on the file
sudo chmod 755 /sbin/chkconfig
# Just to check-Go to the directory of chkconfig file in /sbin
ls -l chkconfig
```

```
# When you list the file, you should see: -rwxr-xr-x 1 root root 611 2011-11-
20 15:12 chkconfig
#Awk
sudo ln -s /usr/bin/awk /bin/awk
# Targeting your Memory – avoiding ORA-00845
#login as root
sudo su -
gedit /etc/init.d/oracle-shm #see the content on the link
chmod 755 /etc/init.d/oracle-shm
update-rc.d oracle-shm defaults 01 99
cd /path-of-the-extracted-oracle-xe.zip/Disk1/
sudo su -
# A point to note here is that su-takes you to the home directory of the user
you're switching to
# so you need to cd back to the directory containing the .deb file
cd /home/edusafe/Downloads/Disk1
dpkg --install ./oracle-xe_11.2.0-2_amd64.deb
#Lock the /etc/init.d/oracle-xe that has just been created will be looking
for a directory called
# /var/lock/subsys. On Debian distros this is located in /var/lock.
sed -i 's,/var/lock/subsys,/var/lock,' /etc/init.d/oracle-xe
#Check:
grep /var/lock/ /etc/init.d/oracle-xe
# Now you should see:
#   touch /var/lock/listener
#   touch /var/lock/oracle-xe
#            touch /var/lock/listener
#        touch /var/lock/oracle-xe
#    if [ $RETVAL -eq 0 ] && rm -f /var/lock/listener
nano /etc/rc2.d/S01shm_load
# The contents of the file
#!/bin/sh
case "$1" in
start) mkdir /var/lock/subsys 2>/dev/null
       touch /var/lock/subsys/listener
       rm /dev/shm 2>/dev/null
       mkdir /dev/shm 2>/dev/null
       mount -t tmpfs shmfs -o size=2048m /dev/shm ;;
*) echo error
   exit 1 ;;
esac
chmod 755 /etc/rc2.d/S01shm_load
/etc/init.d/oracle-xe configure
#logout from root(Ctr+D)when installation is succesfully completed.
nano ~/.bashrc
# setup the environment variables and then we can get on to the database.
#oracle-xe env variables
export ORACLE_HOME=/u01/app/oracle/product/11.2.0/xe
export ORACLE_SID=XE
```

```
export NLS_LANG=`$ORACLE_HOME/bin/nls_lang.sh`
export ORACLE_BASE=/u01/app/oracle
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
export PATH=$ORACLE_HOME/bin:$PATH
#The desktop icon
#Right-click the file on the desktop and select the Properties-> Permissions
tab.
#Now check the box to "Allow executing file as program".
#Close the window.
#You will notice that the icon has transformed into the familiar Oracle
beehive and is now called
#Get Started With Oracle Database 11g Express Edition.
#So, once more into the terminal, we need to make both ourselves and oracle
members of the DBA group that was created as part of the installation :
sudo usermod -a -G dba oracle
sudo usermod -a -G dba edusafe
```

Bibliography

[1] "About CERN," CERN, Jan 2012. [Online]. Available: http://cds.cern.ch/record/1997225. [Accessed 2017].

[2] ATLAS Collaboration et al., "The ATLAS Experiment at the CERN Large Hadron Collider," *JINST,* vol. 3, no. S08003, 2008.

[3] M.Nessi, "The ATLAS Detector at the LHC," *Nuclear Instruments and Methods in Physics Research,* vol. A 648, no. S163–S168, 2010.

[4] E. Lyndon and B. Philip, "LHC machine," *JINST,* vol. 3, no. S08001, 2008.

[5] F. Florent, "Strategies for precision measurements of the charge asymmetry of the W boson mass at the LHC within the ATLAS experiment," 2009. [Online]. Available: arXiv:0906.4260.

[6] 0. Bruning, P. Collier, P. Lebrun, S. Myers, R. Ostojic, J. Poole and P. Proudlock, "LHC Design Report, Volume I The LHC Main Ring," CERN, Geneva, 2004.

[7] CERN, "The LHCb experiment," October 2013. [Online]. Available: https://home.cern/about/experiments/lhcb. [Accessed 2016].

[8] CERN, "TOTEM, The Total, elastic and diffractive cross-section measurement," August 2014. [Online]. Available: https://home.cern/about/experiments/totem. [Accessed 2016].

[9] CERN, "The Large Hadron Collider forward (LHCf) experiment," November 2012. [Online]. Available: https://home.cern/about/experiments/lhcf. [Accessed 2016].

[10] CERN, "ALICE, A Large Ion Collider Experiment," 2016. [Online]. Available: https://home.cern/about/experiments/alice. [Accessed 2016].

[11] CERN, "The Monopole and Exotics Detector at the LHC (MOEDAL)," 2010. [Online]. Available: https://home.cern/about/experiments/moedal. [Accessed 2016].

[12] ATLAS Collaboration, "Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC," *Physics Letters,* vol. B716, p. 1–29, 2012.

[13] Burckhart, H.J et al., "The Common Infrastructure Control of the ATLAS experiment," *CERN,* p. 1, 2008.

[14] "ATLAS Fact Sheet," CERN, 2011. [Online]. Available: https://cds.cern.ch/record/1457044/files/ATLAS%20fact%20sheet.pdf . [Accessed 2016].

[15] E.Simioni et al., "Upgrade of the ATLAS Level-1 Trigger with event," *ATL-DAQ-PROC,* no. 017, 2015.

[16] R. L. Bates, A. Ciocio, S. Chouridou, P. Devran and et.al., "The ATLAS SCT grounding and shielding concept and implementation," *Journal of Instrumentation,* 2012.

[17] I. Dawson and V. Hedberg, "Radiation in the USA15 cavern in ATLAS," ATLAS Document No: ATL-COM-TECH-2004-001. [Online].

[18] S. Baranov, M. Bosman, I. Dawson, V. Hedberg, A. Nisati and M. Shupe, "Estimation of Radiation Background, Impact on Detectors, Activation and Shielding Optimization in ATLAS," CERN, 2005.

[19] CERN, "The ATLAS activation studies," 2007. [Online]. Available: http://atlas.web.cern.ch/Atlas/TCOORD/Activities/CommonSys/Shielding/Activation/activation.html. [Accessed 2016].

[20] A. Ferrari, K. Potter and S. Rollet, "Shielding for the ATLAS Experimental Region," LHC Project Note 6, 1995.

[21] A. Ferrari, K. Potter, S. Rollet and P. Sala,, "Radiation Calculations for the ATLAS Detector and Experimental Hall," CERN-EST-96-001. [Online]. Available: http://atlas.web.cern.ch/Atlas/TCOORD/Activities/CommonSys/Shielding/Activation/act_USA15.html. [Accessed 2018].

[22] "The ATLAS Shielding project," [Online]. Available: http://atlas.web.cern.ch/Atlas/GROUPS/Shielding/shielding.htm. [Accessed 2016].

[23] A. Barriuso Poy et al, "The detector control system of the ATLAS experiment," *JINST,* vol. 3, no. P05006, 2008.

[24] D. Forkel-Wirth, S. Roesler, M. Silari, M. Streit-Bianchi, C. Theis, H. Vincke and H. Vincke, "Radiation protection at CERN," *CERN,* 2013.

[25] ICRP, "Recommendations of the International Commission on Radiological Protection," *ICRP Publication 60, Pergamon, Oxford,* 1990.

[26] CERN, "Safety Code, EDMS 335729," 2006. [Online].

[27] H. Alemdar and C. Ersoy, "Wireless sensor networks for healthcare: A survey," *Computer Networks,* vol. 54, p. 2688–2710, 2010.

[28] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier and B. MacIntyre, "Recent Advances in Augmented Reality," *IEEE Computer Graphics and Applications,* vol. 21, no. 6, pp. 34-47, 2001.

[29] D. Poole and T. Ring, "The Daresbury personnel safety system," in *Accelerator Science and Technology, Proceedings of the 1989 IEEE*, 1989.

[30] A. Kumar and I. Singh, "Indoor environmental gas monitoring system based on the digital signal processing," *Multimedia, Signal Processing and Communication Technologies,* 2009.

[31] A. Pantelopoulos and N. Bourbakis, "A survey on wearable sensor-based systems for health monitoring and prognosis," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE,* vol. 40, no. 1.

[32] E. Zervas, A. Mpimpoudis, C. Anagnostopoulos, O. Sekkas and S. Hadjiefthymiades, "Multisensor Data Fusion for Fire Detection," *Information Fusion, Elsevier,* vol. 12, no. 3, 2011.

[33] E. Yoshiyuki Mizuno, K. Hirokazu and S. Nishida, "Outdoor augmented reality for direct display of hazard information," in *SICE Annual Conference in Sapporo*, Hokkaido Institute of Tecnology, Japan, 2004.

[34] S. Tsumoto Shimane and S. Yokoyama, "Mining risk information in hospital information systems as risk mining," in *IEEE/ICME Internation Confernece on Complex Medical Engineering*, 2007.

[35] M. Serhani, M. Menshawyb and A. Benharref, "SME2EM: Smart mobile end-to-end monitoring architecture for life-long diseases," *Computers in Biology and Medicine,* vol. 68, p. 137–154, 2016.

[36] "Education in advanced VR/AR Safety Systems for Maintenance in Extreme Environments," CERN, [Online]. Available: http://edusafe.web.cern.ch.

[37] F. Alam, E. Adamidi and S. Hadjiefthymiades, "Wireless Personnel Safety System (WPSS), a Baseline towards Advance System Architecture," *PCI, ACM,* 2014.

[38] GWT, "Productivity for developers, performance for users," [Online]. Available: http://www.gwtproject.org/. [Accessed 2016].

[39] "Atmosphere Framework," [Online]. Available: http://async-io.org/.

[40] "Novocaptis, Cognitive Systems and Robotics," Novocaptis, [Online]. Available: http://www.novocaptis.com/.

[41] "AUTH," Aristotle University of Thessaloniki, [Online]. Available: https://www.auth.gr/en.

[42] "Prisma Electronics SA," [Online]. Available: http://www.prismaelectronics.eu. [Accessed 2016].

[43] "ATLAS Procedure Visualizer System," [Online]. Available: https://github.com/CERN/apvs/tree/edusafe. [Accessed 2015].

[44] "Wireless Personal Supervision System," [Online]. Available: https://sites.google.com/site/cernwpss/home. [Accessed 2014].

[45] F. Kereki, Essential GWT: Building for the Web with Google Web Toolkit 2, Pearson Education, 2010.

[46] "Apache Maven," [Online]. Available: https://maven.apache.org/.

[47] "GWT Highcharts," Moxie Group, [Online]. Available: http://www.moxiegroup.com/moxieapps/gwt-highcharts/. [Accessed 2016].

[48] H. ORM. [Online]. Available: http://hibernate.org/orm/. [Accessed 2018].

[49] Elisabetta Di Nitto, Carlo Ghezzi, Andreas Metzger, Mike Papazoglou, Klaus Pohl, "A journey to highly dynamic, self-adaptive service-based applications," *Automated Software Engineering,* vol. 15, no. 3-4, pp. 313-341, 2008.

[50] Antonio Bucchiarone, Cinzia Cappiello, Elisabetta Di Nitto, Raman Kazhamiakin, Valentina Mazza and Marco Pistore, "Design for Adaptation of Service-Based applications: Main Issues and Requirements," *Proceedings of the 2009 international conference on Service-oriented computing,* pp. 467-476 , 2010.

[51] Stephen Lane, Qing Gu, Patricia Lago, et al., "Towards a framework for the development of adaptable service-based applications," *Service Oriented Computing and Applications,* 2013.

[52] Stephen Lane, Antonio Bucchiarone, Ita Richardson, "SOAdapt: A process reference model for developing adaptable service-based applications," *Information and Software Technology,* 2011.

[53] Haihua Zhu, James Gao, Dongbo Li et al., "A Web-based Product Service System for aerospace maintenance, repair and overhaul services," *Computers in Industry,* vol. 63, pp. 338-348, 2012.

[54] P. Potena, "Optimization of adaptation plans for a service-oriented architecture with cost, reliability, availability and performance tradeoff," 2012.

[55] Wanghu Chena, Jing Lia, Ji Zhangb, "An Approach to Service Adaptation for Exploratory Application Construction International Workshop on the Adaptation of Web Services," *Procedia Computer Science,* 2012.

[56] Andreas Gehlert, Andreas Metzger, Dimka Karastoyanova et al., "Integrating Perfective and Corrective Adaptation of Service-based Applications," 2011.

[57] Stephen Lane, Quing Gu, Patricia Lago, Ita Richardson, "Adaptation of Service-Based Applications: A maintenance process?," 2011.

[58] "“Net!Works European technology platform – expert working group on strategy recommendations on networking and telecommunications for cloud computing and service platforms. White paper”," 2011.

[59] Andreas Metzger, Klaus Pohl, Mike Papazoglou et al., "Research Challenges on Adaptive Software and Services in the Future Internet: Towards an S-Cube Research Roadmap," *IEEE,* 2012.

[60] Abdelkader Bouguessa, Boudjemaa Boudaa, Leila Amel Mebarki, "Adaptation of Service-Based Context-Aware Applications with FraSCAti Platform," 2014.

[61] Rainer Weinreich, Georg Buchgeher, "Towards supporting the software architecture life cycle," *The Journal of Systems and Software,* 2011.

[62] Hofmeister, C., Kruchten, P., Nord, R.L., Obbink, H., Ran, A., America, P., "Generalizing a model of software architecture design from five industrial approaches," 2005.

[63] Christine Hofmeister, Philippe Kruchten, Robert L. Nord, Henk Obbink, Alexander Ran, Pierre America, "A general model of software architecture design derived from five industrial approaches," 2006.

[64] S. Kell, "A Survey of Practical Software Adaptation Techniques, Computer Laboratory, University of Cambridge, United Kingdom," *Journal of Universal Computer Science,* vol. 14, no. 13, 2008.

[65] "IEEE Standard for Developing Software Life Cycle Processes," 1997.

[66] Conte SD, Dunsmore HE, Shen VY, " Software engineering metrics and models," *Benjamin-Cummings Publishing Co. Inc., CA,* 1986.

[67] L. MM, "Program evolution," *Info Proc Manag 20(1),* p. 19–36, 1984.

[68] April A, Hayes JH, Abran A, Dumke R, "Software maintenance maturity model (smmm): the software maintenance process model," *Journal of Software Maintenance and Evolution: Research and Practice ,* vol. 3, no. 17, p. 197–223, 2005.

[69] Raffaela Mirandola, Pasqualina Potena, PatriziaScandurra, "Adaptation space exploration for service-oriented applications," *Science of Computer Programming,* 2013.

[70] George Spanoudakis, Andrea Zisman, "Designing and Adapting Service-based Systems: A Service Discovery Framework," *Service Engineering,* pp. 261-297, 2011.

[71] Baoping Cai, Yonghong Liu, Yunpeng Ma et al., "Real-time reliability evaluation methodology based on dynamic Bayesian networks: A case study of a subsea pipe ram BOP system," *ISA Transactions,* 2015.

[72] B. Glisic, "Fiber optic sensors for subsea structural health monitoring, Subsea Optics and Imaging," *A volume in Woodhead Publishing Series in Electronic and Optical Materials,* p. 434–470, 2013.

[73] [26] Sang Hun Lee, Hee Eun Kim, Kwang Seop Son, "Reliability modeling of safety-critical network communication in a digitalized nuclear power plant," *Reliability Engineering & System Safety,* vol. 144, p. 285–295, 2015.

[74] Betramello, O; Burckhart, H.J; Franz, S; et al., "Implementation of a complex monitoring system to track people in the ATLAS detector," in *Industrial Electronics, IECON '09, 35th Annual Conference of IEEE*, 2009.

[75] Rajaraman Anand, Jure Leskovec and Jeffrey D Ullman, Mining of Massive Datasets, New York, N.Y.: Cambridge University Press, 2012.

[76] Burton H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM,* vol. 13, no. 7, 1970.

[77] M. Ying and J. Miller, "Refactoring legacy AJAX applications to improve the efficiency of the data exchange component," *The Journal of Systems and Software,* 2012.

[78] D. Forkel-Wirth, S. Roesler, M. Silari, M. Streit-Bianchi, C. Theis, H. Vincke and H. Vincke, "Radiation Protection at CERN," *CERN-2013-001.415,* 2013.

Glossary

A | B | C | D | E | G | H | I | J | L | M | O |R | P | S | V | W

A

AR Augmented Reality

APVS ATLAS Personnel Visualizer System

AJAX Asynchronous JavaScript and XML

ATLAS A Toroidal LHC Apparatus

ALICE A Large Ion Collider Experiment

ATCN ATLAS control network

ALARA As Low As Reasonably Achievable

B

BAN Body Area Network

C

CS Control System

CMS Compact Muon Solenoid

CTP Central Trigger Processor

D

DAQ Data Acquisition

DCS Detector Control System

DMC Operational Dosimeter

E

EDUSS GUI EDUSAFE Supervision System Graphical User Interface

EEG Electroencephalography

G

GLIMOS Group Leaser In Matters Of Safety

GWT Google Web Toolkit

GCS Global Control Station

H

HMD Head Mounted Display

I

ICRP International Commission on Radiological Protection

J

JSON JavaScript Object Notation

JD Disk Shielding

JF Forward Shielding

JT Toroid Shielding

L

LHC Large Hadron Collider

LHCf Large Hadron Collider forward

LEP Large Electron-Positron collider

LCS Local Control Station

M

MPSS Mobile Personnel Supervision System

O

ODH Oxygen Deficiency Hazard

OO Object Oriented

R

RMS Remote Monitoring System

RIA Rich Internet Application

P

PTU Personal Transmitting Unit

PPE Personal Protective Equipment

PDAs Personal Digital Assistants

S

SCADA Supervisory Control and Data Acquisition system

SCS Subdetector Control Station

SOAP Simple Object Access Protocol

SME2EM Smart Mobile End-to-End Monitoring

V

VR Virtual Reality

W

WPSS Wireless Personnel Safety System