A Methodology for Automated Interoperability Testing of Healthcare Information Systems based on an Actor Emulation Approach

vorgelegt von
Diplom-Ingenieurin
Diana Elena Vega

Von der Fakultät IV - Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften
Dr. Ing.

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr. Ing. habil. Thomas Magedanz

Berichter: Prof. Dr. -Ing. Ina Schieferdecker
Berichter: Prof. Dr. phil.-nat. Jens Grabowski

Tag der wissenschaftlichen Aussprache: 14.07.2011

Berlin 2011

D 83

Abstract

Over the last two decades, the number of healthcare services at the edge of the traditional medical care and computer technologies has increased dramatically, making eHealth infrastructure-related services ubiquitous. Services such as telemedicine, telehealth, Electronic Health Record (EHR) systems are common terms and practices in the actual medical-care sector. A main characteristic of Healthcare Information Systems (HISs) is that they are very *data-intensive* systems. In this respect, a major problem is the lack of product interoperability. Many vendors provide solutions, which are rather provider-centric approaches (i.e., proprietary protocols and message formats), hence, interoperability is not regarded.

The interoperability can be evaluated by means of interoperability testing. The publication of standards such as the Health Level Seven messaging standard (HL7) for defining a common message-structuring scheme for message exchange between medical information systems, or the adoption of the Integrating the Healthcare Enterprise (IHE) integration profiles for specifying use cases that implementers should follow, is an important step in enabling interoperable HISs. These standards and recommendations are the basis for interoperability testing.

This thesis develops an interoperability testing methodology and its realisation concepts for coping with the aforementioned issues in HISs. The main problem addressed in this thesis is how to design a test system that can deal with very data-intensive systems and, at the same time, is capable of emulating the interacting parties. The challenge in this approach is how to automatically customise and configure the test platform to simulate an interoperability scenario by instantiating test components programmed in advance to simulate the behaviour of particular interacting entities as required by the test scenarios. The methodology consists of three main parts: 1) the test design process 2) the message event patterns which are used to derive test simulators and 3) the conceptual architecture for a test framework.

The developed interoperability testing methodology is instantiated in a test framework based on the TTCN-3 test technology. An important component of the realization of the test system is based on the semantic mapping of HL7 version 2.x message structures to TTCN-3 message types that preserves the ontology. A set of derivation algorithms for providing TTCN-3 test behaviours and configurations completes the abstract definition of test scripts. The framework also provides means of communication with System under Test (SUT) and is capable to dynamically adapt to any test configuration as required by test scenarios.

The effectiveness of the developed interoperability testing methodology is demonstrated throughout two case studies. The first case study investigates the interoperability of systems from the IHE Patient Care Devices (PCD) domain. The second applies the methodology to another domain, namely IHE IT Infrastructure (ITI). While the first case study realised within a research project shows the suitability of the developed methodology, the second case study demonstrates its feasibility directly in an industrial context.

Zusammenfassung

In den letzten zwei Dekaden hat sich die Anzahl von medizinischen Diensten in Ergänzung der traditionellen medizinischen Versorgung erhöht. Die auf eHealth-Infrastrukturen bezogenen Dienste wie Telemedizin, Telehealth, Electronic Health Record (EHR) sind allgegenwärtig und gehören mittlerweile zum medizinischen Alltag. Eine Hauptcharakteristik der Healthcare Information Systems (HIS) ist, dass sie *datenintensive* Systeme sind. Eines der größten Probleme bei HIS-Lösungen ist die fehlende Interoperabilität der Produkte. Viele Hersteller liefern Lösungen, die eher anbieterzentrierte Ansätze verfolgen (z.B. proprietäre Nachrichtenformate) und damit wenig herstellerübergreifend bzw. interoperabel sind. Die Interoperabilität kann durch Interoperabilitätstesten überprüft werden. Dabei ist für die Interoperabilität von HISs die Etablierung und Einhaltung von Standards wie z.B. der Health Level Seven (HL7) Standard, der die Nachrichtenstruktur für den Austausch von Nachrichten zwischen medizinischen Informationssystemen festlegt, oder von sogenannten IHE Profilen (Integration Healthcare Enterprise) für die Spezifikation von Anwendungsfällen, wichtig. Diese Standards und darauf aufbauende Empfehlungen werden als Basis für das Testen der Interoperabilität herangezogen.

Diese Dissertation führt eine Interoperabilitätstest-Methodologie ein und präsentiert die Konzepte für deren technische Realisierung unter Beachtung der Besonderheiten von HIS Systemen. Eine wesentliche Herausforderung dieser Dissertation bestand in einem flexiblen und skalierbaren Entwurf eines Test-Systems, welches auf der einen Seite mit den sehr datenintensiven Systemen umgehen kann und auf der anderen Seite fähig ist, die interagierenden Komponenten zu simulieren. Eine weitere Herausforderung bestand in der automatisierten Anbindung und Konfiguration des Test-Systems bzgl. der Verschiedenartigkeit der Anwendungsszenarien, so dass das Test System in der Lage ist, Interoperabilitätsszenarien durch die Instanziierung der im Voraus programmierten Testkomponenten zu simulieren. Die Methodologie besteht aus drei Teilen: 1) ein Prozess für den Testentwurf, 2) Muster für den Austausch der Nachrichten, die für die Ableitung des Verhaltens der Tests verwendet werden und 3) ein Konzept für eine Testarchitektur eines Test Frameworks.

Die entwickelte Interoperabilitätstest-Methodologie wird in einem auf der TTCN-3 Test Technologie basierten Test-System realisiert. Ein wichtiges Element der Realisierung ist das semantische Mapping der HL7 Version 2.x Nachrichtstrukturen nach TTCN-3 Typen, das die Struktur und Semantik der Nachrichten bewahrt. Eine Sammlung von Ableitungsalgorithmen, die passendes TTCN-3 Test-Verhalten erzeugen, vervollständigen die Definition der Testskripts. Das Framework unterstützt auch die Kommunikation mit dem System unter Test und ist in der Lage, sich dynamisch an beliebige Testkonfigurationen, die die Test-Szenarien erfordern, anzupassen. Die Effektivität der entwickelten Interoperabilitätstest-Methodologie wird anhand von zwei Fallstudien demonstriert. In der ersten Fallstudie wird die Interoperabilität der IHE PCD Systeme (Patient Care Devices) untersucht. In der zweiten Fallstudie wird die Methodologie auf IHE ITI (IT Infrastructure) konforme Systeme angewendet. Während die erste Fallstudie in einem Forschungsprojekt durchgeführt wurde, welches die Angemessenheit der Methodologie nachweist, demonstriert die zweite Fallstudie ihre Anwendbarkeit in einem industriellen Kontext.

Acknowledgements

Working on this thesis was for me like a long journey. When I reached the end of this journey I understood that travelling to the destination is more important than reaching the destination itself. This effort would not have been possible without the support of some people whom I would like to thank.

First of all, I want to express my deepest gratitude to my parents for their love and continuous encouragements along these years. They were always there to motivate me to walk till the end, to help me get up when I felt down and to appreciate any little success I achieved.

Along my professional path, I especially thank my first coordinator, Prof. Dr. Ina Schieferdecker, for bringing me to the healthcare IT field and for creating the proper research environment for me to learn, create and work with experts in the field. I appreciate her a lot for helping me keeping my faith, for the many valuable discussions we had and for guiding me over the years.

I would also like to thank Prof. Dr. Jens Grabowski for being my second coordinator and for reviewing my thesis.

I consider myself fortunate to have been involved in a very challenging research project. I enjoyed working with Dr. Armin Metzger, Georg Götz, Stefan Holpp, Hilbert Koetsier, Cristian Taslitchi and Prof. Dr. Florica Moldoveanu during the ReTeMes project. Also in this context, I want to thank Testing Technologies for providing me with the tool development environment for doing the experimental work.

For the possibility to demonstrate my results in an industrial context, I want to express my gratitude and my respect to Milan Zoric from ETSI for inviting me as a consultant to both Connectathon events 2010 and 2011 and for supporting my work and my ideas over the two years.

Sincere and special thanks to Dr. George Din, my former colleague at FOKUS, for many fruitful discussions, suggestions, support and valuable comments during my research.

MOTION research group at Fraunhofer FOKUS provided me with the well-equipped environment in both technical and non-technical sense that made this work not just possible but even enjoyable.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

To know the road ahead, ask those coming back.
– Chinese proverb

Healthcare services provision and healthcare improvement is a current major concern for governments all over the world. More than never, current scientific and technological findings try to meet the needs of society for enabling a better and healthier life. In this context, the healthcare sector has to be aligned to the actual society context where globalisation tendencies and changing demographics, such as an aging population and enhanced habitat dynamics thanks to cheaper travelling possibilities or open borders, tremendously affect how healtcare services are offered.

Like in many other domains, in the healthcare area, the *eHealth shift paradigm* [Jos05] occurred too, replacing traditional paper-based patient records with electronic-based standardised data formats. Hence, the quality assurance became more challenging due to the complexity of the data records, products, processes, heterogeneous architectures, etc.

Exchange of digital data has a long history in the health sector. Although it appears that health institutions in general have been comparatively slow in embracing digital technologies in medical applications, the trend towards computerisation is now gaining momentum. Data in the health sector is increasingly being collected, stored, refined, evaluated and exchanged in digital form, displacing paper-based systems.

The development of Electronic Health Record (EHR) solutions represents the base of information systems in the medical industry. A key issue of EHR systems is their diversity in terms of proprietary protocols, supported interfaces and messaging formats. This fact leads to poor interoperability between different vendors' solutions available on the healthcare market [EAR+05]. The lack of interoperability limits healthcare professionals in combining different software components to provide better care services.

According to many sets of recent statistics [Fun06], [SOD+09], EHR systems are very spread today. In many developed countries such as Holland and Norway they have even almost integrally replaced paper-based patient records. Besides daily care support, another benefit of storing patient data electronically is to serve additional usages within secondary disciplines such as healthcare research, demographics statistics, quality assurance, education, etc. Although the adoption of electronic records was advantageous, EHR systems still have a long path towards maturity until interoperable and safe healthcare information systems are realised.

1.1 Motivation

Quality assurance of today's healthcare information systems is motivated by the rapidly increasing complexity of products and processes, heterogeneous architectures and limited budgets. Increased attention has to be given to a common methodology for assuring the quality of Healthcare Information System (HIS) systems by means of testing. HIS are based on standards for structuring, rendering, storing and exchanging of patient data between different systems from different healthcare areas, e.g., radiology, laboratory.

The adoption of standards such as Health Level Seven Messaging Standard (HL7)[1] [HL787b] published by Health Level Seven Standards Developing Organisations (HL7 SDOs) [HL787a] used for data representation or such as integration profiles published by Integrating the Healthcare Enterprise (IHE) [IHE97] for describing interactions between actors, is an important step in enabling interoperable healthcare systems. HL7 defines a common message-structuring scheme for messages used in medical information systems. IHE defines the use cases which HIS implementers should follow. These standards should serve as a basis for interoperability assurance activities.

Unifying test procedures and realising an intelligent interoperability test design adaptable to different configurations and to various equipments, are real challenges along the testing process of HISs. This thesis addresses these challenges by developing an interoperability test methodology including a test framework based on the test language Testing and Test Control Notation, version 3 (TTCN-3) [ETS07a] standardised by the European Telecommunications Standards Institute (ETSI) [ETS10c] to automate the testing process of HISs. At the same time, this work is the first attempt to define a general test framework for testing HISs systems.

1.2 Objectives of the Thesis

Existent approaches for interoperability testing simply require that two systems are tested for interoperability by running interoperability test scenarios which involve interactions across the interfaces of the two systems. This is a necessary approach because, in the end, only this way can one show that two systems interwork.

When applying this approach one has to consider constraints such as:

- Interoperability testing is frequently constrained by development schedules and synchronisation between teams. For example, the interoperability of a system against other systems cannot be started when depending subsystems are still under development.
- Another problem is the initialisation of the test data on all subsystems, which is sometimes done manually. Interoperability scenarios across the subsystems is time consuming due to data inconsistencies, limited data access, etc.
- Developing and testing against other HISs can result in costly usage fees and can be time constrained.
- The availability of counterpart systems can also be an impediment in running interoperability tests according to internal deadlines.

[1] In the rest of the thesis, HL7 abbreviation will refer to HL7 Messaging Standards

- In general, interoperability tests are performed during specially organised interoperability events which take place at a remote location. This is a huge impediment in terms of limited hardware resources, which can be transported to the event location.

This thesis questions the possibility of revealing interoperability issues of a system prior to putting the system interwork with another system.

Therefore, this thesis proposes a different approach to design interoperability test systems for HISs by emulating the actors and their interfaces in a workflow, and by running the very same interoperability test scenarios one would execute during plug-in events where HISs are checked for interoperability against each other. This way, the interoperability test scenarios can be performed in-house before attending interoperability test events and, thus, eliminating the aforementioned constraints.

This method is not a substitution for the traditional interoperability testing methods based on the analysis of traces obtained by running two systems against each other, but, on the contrary, helps the discovery of interoperability issues at lower costs in advance. This method allows an interoperability check by running the same interoperability scenarios as during an interoperability test event, and even a more thorough investigation by running an enhanced set of such scenarios.

In order to construct such test systems based on the proposed method, in terms of dynamic adaptable test configurations when the workflow setting is frequently changing, as in the case of HISs, the development of methods and algorithms at various levels in the design of the interoperability test framework is required.

This thesis introduces a *methodology for designing and implementing interoperability test systems for HISs* based on the method of emulating actors' behaviours. The thesis presents a process for interoperability checking of various interacting HISs. The process is completed by a set of methods and design rules to automatically derive test configurations from specified interaction workflows and to enable transparent switching of the test system units responsible for the transport layer. Additionally, it discusses how these tests can be technically realised and executed by introducing a general test framework architecture.

There are many possibilities to architect such a test configuration, but the main concern is to find the one which requires the minimum effort and the minimum number of changes (desirable none), when the target system's configuration has suffered modifications. This is a concern, especially when it comes to healthcare informational workflows, because they allow for great flexibility in terms of the number of actors involved within a workflow. The greater the study of the system at an abstract level as a whole with all its interacting components, the better the practicability, re-usability and user transparency aspects will be captured when designing interoperability tests.

1.3 Structure of the Thesis

In this section an overview of the chapters of this thesis is presented. This chapter gives an overview of the scope of the research topics of this thesis. It introduces the problems that the work is dealing with, its objectives, contributions, structure, and roadmap.

Chapter 2 provides the literature review focused on the topic of interoperability and how it is approached in different domains. In particular, the techniques and methods used so far for achieving interoperability of eHealth informational systems define the integral components of the research work. Besides presenting the panorama of the eHealth field as it is seen today, this chapter intro-

duces the standards and recommendations chosen for the research in this thesis.

Chapter 3 introduces the developed methodology for interoperability testing applied to healthcare information systems. The particular challenges to be considered while testing for interoperability of healthcare information systems, in general, or during specially organised interoperability test events are firstly identified. The proposed interoperability test design methodology is based on an interoperability test design process, which is also proposed in this thesis. Furthermore, a number of identified message exchange patterns between different application roles enhances the set of concepts contributed in this thesis.

Chapter 4 provides design guidelines for implementing an interoperability testing platform based on the TTCN-3 test technology. An important component of the realisation of TTCN-3 test system is based on the semantic mapping of HL7 v2.x message structures to a *TTCN-3* type system that preserves the ontology. Additionally, a set of derivation algorithms for providing TTCN-3 test behaviours and configurations is presented.

Chapter 5 introduces two case studies which serve as the basis for experimental work and prove the feasibility of the proposed methodology and test design concepts.

Finally, Chapter 6 concludes the thesis with a summary of the work performed and discussed in the preceding chapters. It also provides suggestions for future work that will enable the research work to be continued. The chapter revises the contributions and objectives stated in this thesis and discusses whether the claims have been successfully achieved.

1.4 List of Publications

Parts of the presented work have already been published in conference proceedings and project reports. The work presented in this thesis is original work undertaken at the Technical University Berlin (TUB), the Entwurf und Testen kommunikationsbasierter Systeme (ETS) chair and at the Fraunhofer FOKUS Institute, Modelling and Testing for System and Service Solutions (MOTION) department.

The list of research papers published during the Ph.D. degree are:

- [VSD10]: D. Vega, I. Schieferdecker, and G. Din. Design of a Test Framework for Automated Interoperability Testing of Healthcare Information Systems. In eTELEMED 2010: Proceedings of the Second International Conference on eHealth, Telemedicine, and Social Medicine, St. Maarten, Netherlands Antilles, pages 123-130. IEEE Computer Society, 2010. ISBN 0-909925-88-7.
- [VSD08]: D. Vega, I. Schieferdecker, and G. Din. A TTCN-3 based Test Automation Framework for HL7-based Applications and Components. In CONQUEST 2008: Proceedings of the Conference on Quality Engineering in Software Technology, 2008.
- [VDS10]: D. Vega, G. Din, and I. Schieferdecker. Application of TTCN-3 Test Language to Testing Information Systems in eHealth Domain. In MCIT 2010: Proceedings of the International Conference on Multimedia Computing and Information Technology, University of Sharjah, U.A.E., pages 123-130. IEEE Communications Society, IEEEXplore, 2010. ISBN 0-909925-88-7.
- [DVS08]: G. Din, D. Vega, and I. Schieferdecker. Automated Maintainability of TTCN-3 Test Suites based on Guideline Checking. In SEUS'08: Proceedings of the 6th IFIP WG

10.2 international workshop on Software Technologies for Embedded and Ubiquitous Systems, pages 417-430, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 978-3-540-87784-4.

- [ZVS+07]: Benjamin Zeiß, Diana Vega, Ina Schieferdecker, Helmut Neukirchen, and Jens Grabowski. Applying the ISO 9126 Quality Model to Test Specifications - Exemplified for TTCN-3 Test Specifications. In Software Engineering 2007 (SE 2007). Lecture Notes in Informatics (LNI) 105. Copyright Gesellschaft für Informatik, pages 231-242. Köllen Verlag, Bonn, March 2007.

- [Veg10]: Diana Vega. Towards an Automated and Dynamically Adaptable Test System for Testing Healthcare Information Systems. In Proceedings of the 2010 Third International Conference on Software Testing, Verification and Validation (ICST '10), pages 331–334. IEEE Computer Society. ISBN 978-0-7695-3990-4.

- [DV11]: Diana Vega, Ina Schieferdecker. Automated Interoperability Testing of Healthcare Information Systems. Chapter in book series Advances in Computers, Elsevier Inc., Editor Atif Memon. To be published in 2012. ISSN 0065-2458.

Chapter 2

Fundamentals of Interoperability Testing in eHealth Domain

Science is organized knowledge.
– Immanuel Kant

Interoperability testing is a complex conceptual and technical activity within the sphere of software testing in general [AO08], [KFN99] and answers questions related to the capability of a system[1] to communicate with other systems and to use the transmitted information. Quality labelling of IT products for interoperability is a topic intensively debated today in many domains. Similar to telecommunications, eGovernment, automotive, avionics, etc., where interoperability gained increasingly in attention until it now plays an essential role, in the eHealth[2] context, the concern of how to make systems even more interoperable, likewise, received great interest from different standardisation bodies and key industry players.

Furthermore, numerous governments worldwide and even the European Commission have seen the stringent and urgent need for exchanging patient information at a national or even cross-border dimension. Consequently, considerable budgets were made available in the last few years to achieve the goal of interoperability in the eHealth arena.

Irrespective of the domain, one of the typical questions that interoperability testing is trying to answer is the following: does the system understand data from another system it is supposed to communicate with? Interoperability testing requires not only concepts, methods and tools but also a broad understanding of interacting systems, the underlying communication protocols and messaging formats.

This chapter reviews the main concepts related to interoperability testing. The main focus is however on how these concepts are understood and applied to the healthcare IT context.

[1] The notion of system is used here in a broader sense: it can mean a software application subsuming several functionalities or only one role of a system, a service, etc.

[2] The term *eHealth* is used in this thesis interchangeably with *healthcare IT*, *medical IT* and denotes the applicability of Information and Communications Technology (ICT) in the healthcare field.

2.1 Interoperability and Interoperability Testing

The need for interoperability[3], even at a borderless country dimension, was recognised, agreed and the requirement for interoperability support, was urged by many governments in the last few years. The growing urgency to electronically exchange and handle information was considered by the EC in the European Journal of ePractice [KC09] the "critical success factor to forge ahead in the online provision of public services".

In 2004, the European Interoperability Framework (EIF) version 1 [Com04] differentiated itself in the context of delivering pan-European electronic government services, three layers of interoperability: the *technical*, *semantic* and *organisational* interoperability. The technical layer handles all issues concerning software, systems and data. The semantic level deals with issues concerning data content, common directories, ontologies, etc. The organisational level aggregates political and human issues, aspects regarding inter-communities, legal matters, and international scale of differences. In the draft constituting the basis for the second version of EIF [Com08] published in 2008, the layers of *legal* interoperability and the *political* context were added.

In 2006, in a white paper [VW06] published by ETSI a fourth layer was introduced; this layer is the *syntactic* interoperability. This level is usually associated with data formats, e.g., high-level transfer syntaxes as XML, HTML, of the data transferred. The distinction between the technical interoperability and the syntactical interoperability was adopted in 2009 also by the EC and was announced in the European Journal of ePractice [KC09].

Aligned to the increasing demand for interoperable systems targeting compliance with one or the other interoperability stages, interoperability testing[4] has become a real challenge given the multitude of standards, heterogeneous systems, specific environments, hence the development of testing tools has to cope with a great deal of test diversity.

After briefly presenting the interoperability definitions given by various organisations or adopted within different frameworks, an overview is given of the most pregnant concepts, methods and techniques employed for achieving and ensuring quality labelling for interoperable systems.

2.1.1 Interoperability Definitions

The interoperability topic is of a highly complex nature, consequently, different definitions of interoperability co-exist. In the following subsections, interoperability definitions are outlined and grouped based on the discussed layers of interoperability. This analysis was conducted to motivate how the interoperability goals are regarded and vary from level to level. Additionally, the definitions help in understanding the facets of interoperability from technical aspects up to human aspects including culture, security, privacy, etc, details.

2.1.1.1 General Interoperability Definitions

Table 2.1 shows the definitions provided by different standardisation bodies and organisations worldwide.

[3] In the literature, the term *interoperability* is usually abbreviated as *IOP*

[4] In the literature, the term *interoperability testing* is usually abbreviated as *IOT*

Table 2.1: General Interoperability Definitions

Organisation Name	General Interoperability Definitions
International Organisation for Standardisation (ISO)/International Electrotechnical Commission (IEC) (2002:12)	Interoperability refers to the ability of two or more systems (computers, communication devices, networks, software and other information technology components) to interact with one another and exchange information according to a prescribed method in order to achieve predictable results.
Institute of Electrical and Electronics Engineers (IEEE) (2000)	There are four definitions for interoperability. Interoperability means: • the ability of two or more systems or elements to exchange information and to use the information that has been exchanged. • the capability for units of equipment to work together to do useful functions. • the capability, promoted but not guaranteed by joint conformance with a given set of standards, which enables heterogeneous equipment, generally built by various vendors, to work together in a network environment. • the ability of two or more systems or components to exchange information in a heterogeneous network and use that information.
EIF (2004:5) (IDABC 2004)	Interoperability means the ability of information and communication technology (ICT) systems and of the business processes they support to exchange data and to enable the sharing of information and knowledge.
European Committee for Standardisation (CEN)/Information Society Standardisation System (ISSS) (2005)	Interoperability is a state, which exists between two application entities when, with regard to a specific task, one application entity can accept data from the other and perform that task in an appropriate and satisfactory manner without the need for extra operator intervention[5].
Healthcare Information and Management Systems Society (HIMSS)	In the most fundamental sense, interoperability is the ability of two or more systems or their components to exchange information and to use the information that has been exchanged.
National Alliance for Health Information Technology (NAHIT) (2005)	Interoperability is the ability of different information technology systems, software applications and networks to communicate, to exchange data accurately, effectively and consistently, and to use the information that has been exchanged.

Continued on Next Page...

[5] A clear distinction is made to the terms *interface* and *integration*.

Table 2.1 – Continued

Organisation Name	General Interoperability Definitions
ETSI (2005:1)	Interoperability is the linking of systems, networks or services so that they can work together successfully.
European Parliament Source: FIII (2005:2)	Interoperability means the ability of a computer program to communicate and exchange information with other computer programs and mutually use the information which has been exchanged, including the ability to use, convert or exchange file formats, protocols, schemas, interface information or conventions, so as to permit such a computer program to work with other computer programs and with users in all the ways in which they are intended to interact.
TMA (2004:2-1)	Interoperability in general can be defined as the state of having sufficient power, skills or resources to mutually perform a function or produce an appropriate effect.

2.1.1.2 Technical Interoperability Definitions

Table 2.2 overviews the understanding of interoperability from a technical point of view.

Table 2.2: Technical Interoperability Definitions

Organisation Name	Technical Interoperability Definitions
EIF (2004:16)	Technical interoperability covers the technical issues of linking computer systems and services. Key aspects include open interfaces, interconnection services, data integration and middle-ware, data presentation and exchange, accessibility and security services.
HIMSS	Technical dimension of interoperability includes uniform movement of healthcare data, uniform presentation of data, uniform user controls, uniform safeguarding data security and integrity, uniform protection of patient confidentiality, uniform assurance of a common degree of service quality.
TMA (2004:2-3)	Technical interoperability consists in being able to communicate and interact between two systems coming from different manufacturers. The functional goal is to allow data to be exchanged between different projects in multiple countries using different equipments, software, etc., from multiple manufacturers or vendors (p.2-3).

2.1.1.3 Semantic Interoperability Definitions

The subject of semantic interoperability addresses the capability of computer programs to combine the exchanged information with other information resources in a meaningful manner. This implies an agreement on ways to discover or to give a context to the information allowing the data exchange and processing between tools that are designed independently. Table 2.3 lists how different organisations express the term semantic interoperability.

Table 2.3: Semantic Interoperability Definitions

Organisation Name	**Semantic Interoperability Definitions**
EIF (2004:16)	Semantic interoperability is concerned with ensuring that the precise meaning of exchanged information is understandable by any other application that was not initially developed for this purpose. (EIF 2004:16). It thus enables systems to combine received information with other information resources.
World Health Organisation (WHO)/EC	The initial considerations on semantic interoperability were addressed during the workshop on interoperability of eHealth systems (V4): Using a "holistic" (p.12) definition the goal of semantic interoperability is to "improve communication" on medial and health related aspects both among humans and machines" (p.11). In order to achieve this, a two-pronged approach is necessary: achieving a unified health ontology (longer term) and tackle concrete and clearly delineated issues (short term). Within semantic interoperability various dimensions, such as medial/administrative or human/machines levels can be distinguished (p.12).
CEN/ISSS (2005:39)	CEN/ISSS stresses that semantic interoperability is not an "all-or-nothing" concept. That is "the degree of semantic interoperability depends on the level of agreement between sender and receiver regarding the terminology, and the content of archetypes and templates to be used".

2.1.1.4 Organisational Interoperability Definitions

Table 2.4 presents different definitions of organisational interoperability. While the first and the last definitions address a more general cross-domain understanding of the organisational interoperability, the second definition highlights its meaning in the context of the healthcare IT sector.

Table 2.4: Organisational Interoperability Definitions

Organisation Name	**Organisational Interoperability Definitions**
EIF (2004: 16)	Organisational interoperability is concerned "with defining business goals, modelling business processes and bringing about the collaboration of administrators that wish to exchange information and may have different internal structures and processes".
TMA (2004:2-4)	Organisational interoperability is defined as the state where the organisational components of the health system are able to perform seamlessly together. The vision is "an integrated health system that provides efficient, effective and holistic citizen-centred services based on the principles of Health for All-Access, Equity and Solidarity".
ETSI	Organisation interoperability, as the name implies, is the ability of organisations to effectively communicate and transfer (meaningful) data (information) even though they may be using a variety of different information systems over widely different infrastructures, possibly across different geographic regions and cultures. Organisational interoperability depends on successful technical, syntactical and semantic interoperability.

2.1.1.5 eHealth Interoperability Definitions

Table 2.5 captures the main interoperability definitions used in an eHealth context.

Table 2.5: eHealth Interoperability Definitions

Organisation Name	**eHealth Interoperability Definitions**
EC, Communication on a European eHealth Area Com (2004) 356; Source: TMA bridge (2005:2)	Interoperability should enable the integration of heterogeneous systems, allow secure and fast access to comparable public health data and patient information located in different places over a wide variety of wired and wireless services.
TMA-bridge (20052-2)	The ultimate objective of eHealth interoperability is to allow different people from different countries (meaning having different habits, traditions, cultures, languages) to easily communicate different data and interact with different systems coming from different manufacturers or vendors with the same results (p.2-2).
US Department of Health (RFI 2004:2)	Interoperability is the ability to exchange patient health information among disparate clinicians and other authorised entities in real-time and under stringent security, privacy and other protections.

2.1.2 Interoperability Testing

In the following, the approaches employed in different domains for interoperability testing are reviewed. Some representative interoperability test frameworks are discussed. Most of these frameworks, in their pursuit of interoperability testing, target a specific corresponding standard, hence, they generally have limited scope and provide insufficient testing services. The purpose of this analysis is to discover the methodological reference points, and thus, highlight how interoperability testing is achieved across various domains.

2.1.2.1 Conformance vs. Interoperability Testing

In ETSI's vision, different than in conformance testing [ISO92], [BG94], where the focus is to validate specific components within a system against requirements from a base specification, "IOT concentrates on a complete device or a collection of devices. It is system testing rather than unit testing. It is most commonly applied to end-to-end testing over networks. It shows, from the user's viewpoint, that functionality is accomplished (but not how)" [VW06]. Additionally, the most common usage of this term is associated with a semi-formal testing performed at multi-vendor events in order to obtain valuable information about the capability of communication of similar products [ETS10d].

Even though ETSI's main standardisation activities in the field of testing were mainly focused on developing conformance testing specifications, the relevance of conformance testing for interoperability testing was outlined back in 2003 [Wil03]. In this presentation, the author stated that the telecom industry performs by tradition conformance testing but it foresees that the "IP world tends to go for interoperability testing". Reasons for this emerging tendency are complex technologies, increased interest in branding and certification programs that encourage a combined approach, etc.

2.1.2.2 CEN Global eBusiness Interoperability Test Bed Methodologies (GITB)

GITB [CEN09] is a global initiative hosted by CEN and supported among others by ETSI, EIC, NIST, KorBIT[6] which looks into organisational interoperability level with a strong emphasis on the political dimension. The GITB's main motivation comes out of the observation that current businesses are transforming toward networked enterprises and shared service centres. Therefore, the interoperability also becomes a main concern.

GITB addresses eBusiness and enterprise interoperability and develops, under EU support and guidance, a comprehensive and global eBusiness interoperability test bed in a global collaboration of European, North American and Asian partners.

The main outcome of the project is a concept for a global eBusiness interoperability test bed which has as main purpose to outline a global testing methodology. The practicability of the concept is analysed in a feasibility study which demonstrates that the objectives of defining a global interoperability methodology, though very ambitious, are plausible.

An overview on an initial idea of the test bed was captured by NIST in a presentation [NIS03] in 2003. Besides the idea of a global collaborative test bed spread over a huge consortium (USA, Europe, Asia), NIST also introduced several concepts such as test interoperability stack, semantics checkers, syntax checkers, grammar checking as well as business choreography checking. Part of their test framework architecture, the *Reflector* component maps to the concept of *Monitor* introduced later in other approaches presented next.

Another contribution to this project is provided by [Woo07] which presents a new approach to assess business2business (B2B) interoperability. The approach is named agile test methodology for B2B/B2C interoperability and emphasises re-usability and test efficiency, which seem to be missing in other existing testing technologies.

2.1.2.3 ETSI Generic Approach to Interoperability Testing Methodology (GAIT)

The first view on interoperability testing at ETSI was conceived in the context of TIPHON^{TM}[7] project of Methods for Testing and Specification (MTS) group, published in [ETS03] and also presented in [Wil03]. The main target was that, using the experience from conformance testing and the practices in interoperability at that time, to develop a generic methodology for NGN interoperability testing that will be acceptable to all ETSI technical bodies.

Initially published in 2007, the current ETSI GAIT [ETS10d],[Ber08] (currently version 1.2.1 is available) was designed on purpose for interoperability testing. The methodology, as the name itself suggests, claims not to focus on a particular software domain. From the perspective of coverage of interoperability levels described in the previous section, GAIT addresses *technical* interoperability testing.

As summarised also in [QPS09], GAIT defines two main processes covering the complete testing lifecycle: a) the process of developing an interoperability test specification and b) the technical interoperability testing process. The methodology assumes the participation of two types of actors: *Test Driver* (many instances per test system), responsible for running the tests, and *Test Coordinator* (only one instance per test system), responsible for coordinating the *Test Drivers*.

Additionally, the methodology also presents an interoperability testing architecture, whose main

[6] Korean B2B Interoperability Testbed (ForBIT) Consortium
[7] Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON)

characteristics are: 1) considers Upper and Lower points of control and observation, 2) is suitable for active testing, 3) does not consider passive testing, and 4) considers involvement of monitoring components. This architecture can be a diagram or textual description and describes the testing equipment and the communication paths between the equipment parts.

Even though the ETSI approach is very popular and has won recognition, the performed analysis and experiments demonstrated that this approach is too complex to be used by non-professional testers and additionally is oriented towards telecommunication area, as stated in [GCS09]. Hence, a simplification of the GAIT that aims to be straightforwardly applicable for testing software interoperability is presented in [Gli10]. Furthermore, this approach neither addresses the interoperability testing needs in eHealth nor has it been previously applied to healthcare IT and in particular to HISs.

The proposed ETSI testing interoperability methodology has been employed in different telecommunication areas such as IPv6 [ETS08], [TPI10], IP Multimedia Subsystems (IMS) [ETS09]. Recent activities regard its applicability to testing grid and cloud infrastructures [ETS10b], [ETS10a], [RNG08], [RIN09]. However, as stated in [RGS10], the specification of executable test cases (including the specification of test configurations, test components and their behaviours) is still considered as future work. ETSI also encourages vendors to bring together their products and test their interworking by organising a series of interoperability events mainly from telecom and IP world [ETS].

2.1.2.4 OASIS Web Service Interoperability (WS-I)

OASIS Web Services Interoperability organisation (WS-I) [WSI04] is an open industry organisation that addresses the interoperability needs of Web services across platforms, operating systems and programming languages. The main target is to create and publish use cases and test tools to help the deployment of web services compliant with these interoperability use cases. The use cases are grouped into profiles. WS-I published and, currently, maintains 12 profiles which provide implementation guidelines for how related Web services specifications should be used together for best interoperability.

The method operated by WS-I to evaluate interoperability is to use a set of two tools which monitor, record and validate the interactions between services [Bri03]. The WS-I Monitor tool is used to capture and log interactions between web services. The monitor works like a proxy which routes the communication between the services. The log is processed by the WS-I Analyser to verify that the monitored interactions conform to the WS-I profiles.

The tools are free to download and the tests can be executed on site. The testing tools do not interact with the web service, but only intercepts existent communication, which can only be triggered from outside of the test system. Therefore, the WS-I tools do not ensure the interoperability but evaluate whether a Web service is compliant with the requirements for interoperability.

2.1.2.5 OASIS ebXML Implementation, Interoperability and Conformance (IIC)

The main target of OASIS ebXML IIC group [IIC11] is to specify Electronic Business using Extensible Markup Language (ebXML) interoperability testing specifications for the eBusiness community. The main output of this group is a framework that provides means for software vendors to create infrastructure and applications, which adhere to the ebXML specifications and are able

to interoperate. OASIS ebXML IIC Test Framework is a test framework to verify interoperability of ebXML eBusiness systems.

The IIC test framework can address both the technical aspect of ebXML testing related to transport layer and the interaction flows. Similar to ETSI's methodology, ebXML IIC test framework consists of two components: Test Driver and Test Service. The Test Driver interprets testcase data and drives testcase execution. The ebXML IIC Test Service is equivalent to the Test Driver from ETSI methodology and it implements some test operations (actions) that can be triggered by received messages. These actions support and automate the execution of testcases.

2.1.2.6 Academic Approaches

The topic of interoperability testing was also captured in many scientific publications. In the following, relevant attempts to define a method or a methodology on how interoperability testing should be approached in various sectors are presented.

About a decade ago, the research topic of deriving interoperability tests for communication protocols was investigated in [KK97] [KSK00]. Within the context of communication networks, the interoperability is defined as the degree of interoperation, and measuring the interoperability, i.e., interoperability testing, shall be performed at two levels: specification and implementation. The authors introduced an interoperability test derivation method and demonstrated the feasibility of their interoperability testing approach applied to the ATM signalling protocol. Additionally, and executable test system based on TTCN[8] test technology was provided in order to test the actual ATM equipment. An important point outlined in this paper is that the work of interoperability assurance should come at the stage of specification development and then be augmented through interoperability testing. In this approach, the specifications and implementations of communicating entities by means of ATM Signalling Protocol are modelled as some sort of Finite State Machines (FSMs) - Input Output State Machines (IOSM). The two IOSM models constitute the input for a procedure for systematic derivation of interoperability test suites which consists of: a) construct an interaction graph out of the two communicating IOSMs, b) decide on target test coverage and test architecture, and c) derive interoperability test cases from the interaction graph. With respect to the selection of the test architecture, even though often in practice a monitoring point, known as Point of Observation (PO) is set between the interworking entities (IUTs), this approach does not consider this interoperability test architecture based on proxy or monitoring component from the following reasons: 1) it would be more costly to generate test suites, 2) test generation would be more costly, and 3) interoperability testing would have much overlapping with conformance testing. However, as the authors themselves stated that, in order to achieve full automation of interoperability testing, their approach needs to be further extended to consider the data, content of the messages and operation on them.

One year later in 2001 [BCKZ01], one of the authors contributing to the work presented above, was also involved into the work and publication of an article describing two methods for interoperability test generation for different entities implementing the same protocol. Both methods are applied to a case study involving TCP/IP protocol. The specification of the TCP-Reno protocol is provided in Specification and Description Language (SDL). SDL is widely used for specification of protocols being based on the semantic model of Extended Finite State Machines (EFSM). Different to the previous approaches existent up to that time, the focus in this work was on test selection

[8] Tree and Tabular Combined Notation (TTCN) is a programming language used for testing of communication protocols. Beginning with version 3 TTCN was renamed to Testing and Test Control Notation (TTCN-3).

criterion that guarantees the coverage of faults related to the interoperability of the involved components. The employed test architecture bases on the idea of having *Points of Observation* (POs) to capture the communication between entities at the side of each component and *Points of Control and Obeservation* (PCOs) on each entity. After comparing the test sequences obtained with the two methods, the author found that they are complementary. While the first method is more useful in deriving the test control starting with the interactions and addressing the coverage of particular interactions, the second method permits an easier data differentiation by analysing branches with embedded decisions. The outcome of this work was used later in 2004 in [CMZ04] for developing a formal interoperability testing methodology of protocols and services with applicability to wireless access to Internet: WAP systems but it can be extended to other telecom systems such as GPRS - General Packet Radio Service and UMTS - Universal Mobile Telecommunication System. The proposed methodology applies to test architectures that integrate concepts of PCOs and POs available in different combinations.

Another approach proposing a test interoperability methodology, from the context of telecommunications, with applicability to the MANET routing protocol is presented in [MGM07]. Similar to the works introduced above, the specification of the MANET protocol is designed in SDL. In order to test the interactions between the two implementations, test scenarios are automatically derived based on the specification and on the requirements. The methodology consists of four steps: 1) construction of a formal specification, 2) test purpose definition, 3) test scenario derivation using test generation algorithms, and 4) formalisation of the test scenarios using Message Sequence Charts (MSC[9]) and Tree and Tabular Combined Notation (TTCN) for test specification. According to the accessibility to the interfaces of the two involved entities and their communications, different possible test architectures are described. The test system architectures are based also on POs and PCOs concepts. More formal frameworks for interoperability testing are presented in [VBT01] and [DV05].

In an article published in 2008 [PPP08], the authors presented their view on interoperability testing for eGovernment Web services. Their motivation is given by the evolution of eGovernment Web Services. In this context, although the same standards are adopted, the Web Services interworking capabilities are doubtful. Even though the target seems to be a complex one - the achievement of interoperability in a complete heterogeneous environment - the need is actually for interoperability testing of common Web Services. The authors argued that the existent testing methodologies lack in a clear-stated definition of specific steps that should be followed for the precise definition of interoperability testcases. Even though the TTCN-3 test technology does offer the syntax and semantics needed to express interoperability test steps in an adequate representational form, the conclusion of the authors after investigating this test technology was that, besides the fact that it is difficult for a human to read and understand, there are no open source tools for compiling and interpreting TTCN-3 test scripts. To work on this problem, an Web services interoperability testing process (WSIT) is proposed. It consists of two parts: a) the specification of a sequence of definitions of interoperability test steps and b) their representation in XML form based on the XML Requirement and Test (XRT) language. The feasibility of their proposal was demonstrated by evaluating the interworking capability of two fully operational Web services based eGovernmental solutions for eInvoicing. Even though the XRT schema was extended by the authors in order to provide all required information for representing interactions among different participants, the chosen XML based test language does not seem to be able to cover all interoperability facets that were introduced in the previous section. For example, the XML test language gives the possibility to refer the involving entities, referred standards (e.g., SOAP, WSDL) for each entity, the

[9]MSC is a formalism widely used in industry to describe message exchanges.

"actions" for each participant entity (the business process implemented by that particular entity) and the testing result and verdict, but it is not shown how the messages can be correlated within a sequence of test steps. The later work of the authors referred to the integration of the proposed WSIT process in an existing testing methodology.

In [DK03] and [DK06] a model for interoperability testing named Model for Automated Interoperability Test (MAIT) is presented. MAIT uses TTCN-3 test language to describe tests. This model offers a framework for automated interoperability test suites and means on how the interoperability tests can be reused as conformance tests. A MAIT test suite consists of two main parts: a) *MAIT-static* being protocol independent and the same in all MAIT test suites (in the TTCN-3 test language it corresponds to creating the PTCs, Main Test Component (MTC)) and b) *MAIT-dynamic* containing the part to check the interworking of the IUTs (in TTCN-3 it corresponds to the functions used for configuration of the implementations and executing the testcases). The architecture of the test system bases on the Monitor concept for analysing the communication between the two implementations. Additionally, the test system has to accomplish the tasks of remote control and communication with the two implementations. This methodology was not defined or standardised by a standardisation organisation, therefore it can not reach a certain maturity [MAI09].

2.1.3 Adopted Definitions of Interoperability and Interoperability Testing

This section summarises the definitions adopted in this thesis for interoperability and interoperability testing.

The term *interoperability* will designate in the rest of this thesis "*the capability of the software product to interact with one or more specified components or systems*". This definition is the definition adopted by the International Software Testing Qualifications Board (ISTQB) [IST11] in the *Standard glossary of terms used in Software Testing* [IST10] (version 2.1, 2010) and was originally introduced in the standard ISO 9126 [ISO04]. In alignment with the view on interoperability of ISTQB, the worldwide-recognised software testing body in its software testing certification programs, similarly in this thesis, the interoperability is approached from the perspective of *only one* system. This is different compared to the other interoperability definitions where the focus is on two interworking systems.

Interoperability testing will be understood in this work as *"the process of testing to determine the interoperability of a software product"*. The adopted interoperability testing definition is also the one used along the testing certification programs conducted by ISTQB [IST11] and included in the ISTQB's published glossary [IST10]. This perspective of understanding interoperability testing addresses again the focus on *only one* system, without specifying the test method. This is again different to the previously analysed interoperability testing definitions, which assume the involvement of two systems. As the definition itself suggests, the final target of the testing process is to derive conclusions about the interoperability capability of one software product, without mentioning or imposing that the testing process requires interacting systems.

2.2 Healthcare IT Domain

The applicability of ICT to the healthcare domain is the promise and the solution for improving healthcare quality and safety while reducing costs. The healthcare IT domain is as vast as the space of healthcare activities, being a substitution in a virtual world, usually designated as Health

Information Technology (HIT), of various daily healthcare practices.

2.2.1 eHealth Shift Paradigm

The main goal of Health Information Technology (HIT) is to align the healthcare domain to the requirements of the current world's tendencies towards globalism and mobility while widening the spectrum of benefits for the patient. This means to ensure a patient-centred virtual world which enables more flexibility through the availability of telehealth and telecare services, increased accessibility to healthcare data over different geographical coordinates by assuring interoperable EHR systems, enhanced data security and privacy, reduction of risks by avoiding handwriting mistakes thanks to ePrescribing systems, automated help for physicians in clinical and medication decisions, progress in healthcare research and education by employing the advances in data mining and artificial intelligence fields, and many more. Furthermore, actual eHealth systems are not just about replacing paperwork with EHRs, smartcards, etc., but HIT also enables healthcare to be personalised. This not only makes treatments more effective, but it enables doctors to diagnose problems more quickly, and even predict them before they occur.

This shift, from traditional healthcare paper-based records to a digitalised healthcare domain, plastically named in [GS90] as eHealth shift paradigm, represents the new lens through which the healthcare view has to be perceived today. Furthermore, the shifting process as such is not enough, given the increasing demand for healthcare, ageing population and decreasing healthcare workforce, equally important is how fast and under which quality parameters it occurs.

Another aspect of this shift paradigm is pointed out by Joseph in 2005 in his book [Jos05] and is related to the usability of the eHealth system. He considers that the success of the employment of these technologies consists in providing a successful design for a human-computer interface. Furthermore, "in the context of the e-business revolution, e-health is seen as a paradigm shift from a physician-centred care system to a consumer-driven care system. In other words, e-health systems place the e-consumers rather than the caregivers at the centre" [Jos05].

2.2.1.1 Medical Informatics Discipline

As a side effect of the digital transformation, a new and evolving field and academic discipline emerged: medical informatics [GS90], also known as biomedical informatics [SC06]. As the term "informatics" suggests, the medical informatics term refers to the applicability of computer technology to the medical domain, but keeping as a central aspect not the technology but the medical information: management and modelling of healthcare and biological information systems, knowledge-based decisions support systems, ontological-based data representation systems, teaching and learning systems in the medical field, etc.

Many definitions of this disciple are available. The idea presented in 2004 that medical informatics is the science and art of processing medical information [Sar04] did not change much over the time. Already in 1984, Van Bemmel [GS84] foresaw the significance of the field and defined medical informatics as comprising "the theoretical and practical aspects of information processing and communication, based on knowledge and experience derived from processes in medicine and health care". A similar view for that period was shared by Shortliffe [Sho84]: "Medical information science is the science of using system-analytic tools to develop procedures (algorithms) for management, process control, decision making and scientific analysis of medical knowledge." In Greens' view, 20 years ago [GS90], "medical informatics is the field that concerns itself with the

cognitive, information processing, and communication tasks of medical practice, education, and research, including the information science and the technology to support these tasks." All these definitions outline basically one idea: medical informatics lies at the interface between the subject domain of medicine and the science and technology of computing [Sar04].

All these views address a data-centric approach concentrating on the representation of the healthcare concepts and their relation: 1) *data* are pure observation values for different parameters, e.g., blood pressure; 2) *information* is the result of interpreting the data based on rules, e.g., hypertensive; and 3) *knowledge* is the relation between different information and can be used as input for other or other knowledge databases. Differently, in [HHA96], the perspective of medical informatics is stressed as a processes modelling and design and their implementation as computer systems.

Regardless of the perspective, some of the challenges that medical informatics addresses are: what standards are desirable for human-computer interactions, which protocols and data formats should be used for linking disparate systems to facilitate the interchange of data between applications, how should knowledge be used to make decisions most effectively, etc. In other words, in this modern age of information, similar to other emerging fields, specialists from medical informatics attempt to solve the fundamental issues in data and knowledge management that underlie decision making in healthcare.

2.2.1.2 eHealth Panorama

This revolutionary new paradigm for healthcare, concretised in eHealthcare shift [Jos05], describes the actual status of the evolution of healthcare system. This move arose naturally from many needs. Firstly it was a need for processing efficiently massive health data by automating the routine management [Jos05]. Secondly, there was a need for intelligent medical information systems or office automation to reduce time and effort consumption on the part of health knowledge workers such as physicians. Then the access to healthcare services in a flexible manner by employing networking technologies and electronic transmission of data was necessary. And now, there is a need for making all those systems within eHealth world interoperable.

There are many views and definitions for the eHealth paradigm. Simply defined by the World Health Organisation (WHO), "eHealth is the combined use of electronic communication and information technology in the health sector". Regardless of the actual perception of eHealth, a future for healthcare without an eHealth solution, services and infrastructure is not possible, as reported in a recent study [Sul09].

A panorama of the eHealth as a whole system of subsystems, was described in Joseph's book [Jos05] (Figure 2.1, page 44) published in 2005 and reproduced in this thesis in Figure 2.1. This perspective indicates the complexity of the eHealth system and how various eHealth care concepts such as eHealth clinical services, teleconsultation, eHealth decision support, remote patient monitoring, eMarketing, etc. relate one to each other.

A recent snapshot of the eHealth sector is perfectly captured by the international business research and consulting company Frost & Sullivan in a study conducted in 2009 [Sul09] and reproduced in Figure 2.2. As the figure indicates, the eHealth field can be regarded from two main perspectives: *applications* and *services*. While *services* refer to telehealth services including telemedicine[10] and

[10] *telemedicine* is the deployment of information and telecommunication technologies to allow remote sharing of relevant information or medical expertise regardless of the patient's location [Jos05] (page 44)

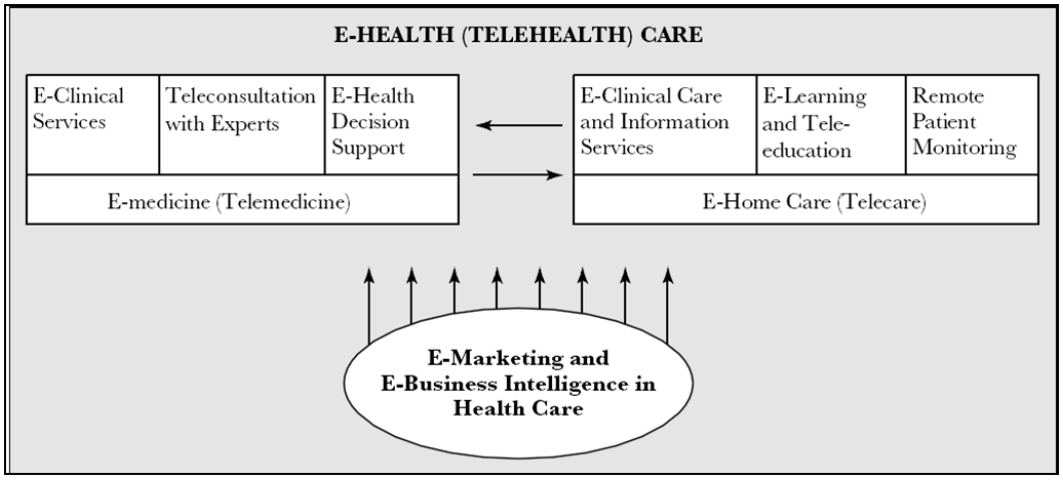


Figure 2.1: eHealthCare Systems and Subsystems

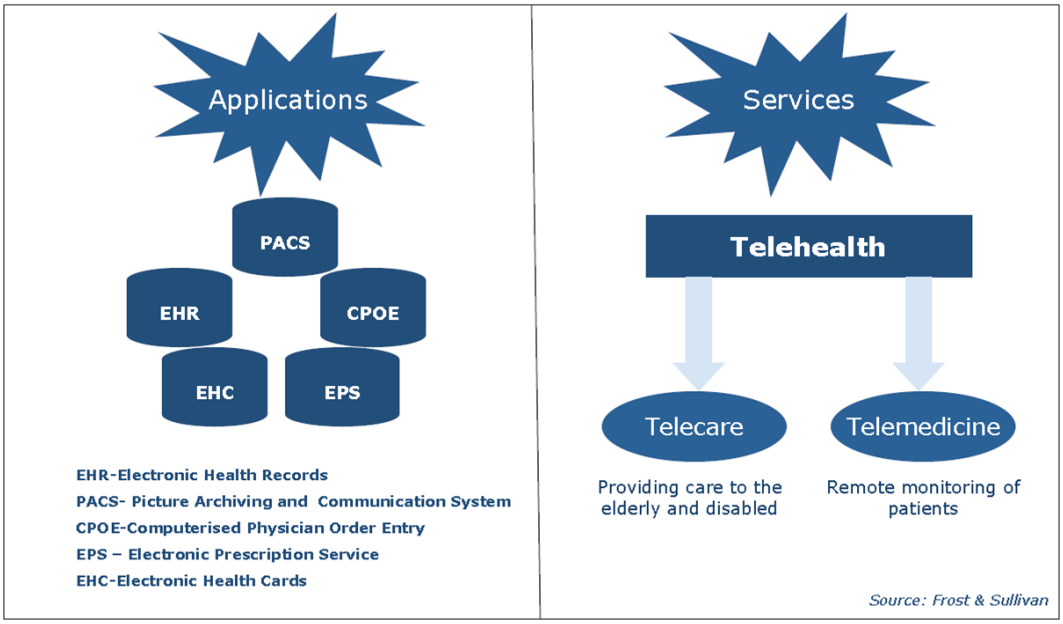


Figure 2.2: eHealthcare Major Components

telecare[11], *applications* target a larger scope covering various information systems that deals with different data and processes along healthcare practices: Electronic Health Record (EHR), Picture Archival and Communication System (PACS), Electronic Prescription System (EPS), Electronic Health Card (EHC), Computerised Physician Order Entry (CPOE). This perspective over is adopted in this thesis, with the difference that, what the authors called eHealth *applications*, in this thesis is designated by the term Healthcare Information System (HIS) and can be extended with other possible information systems.

2.2.2 Healthcare Information Systems

As already described above, eHealth is a complex interdisciplinary field situated at the interface between healthcare and computer technology. As stated [Jos05, Chapter 3, p. 60], HISs, and in particular EHR systems, are parts of GTS (General Thinking Theory) theory. This theory applies to any kind of system in general, irrespective of the domains it belongs, and bases on the idea that systems are formed by interlocking and connecting many other systems (subsystems). Similarly, the HISs are subsystems inside the larger information system in healthcare, which, in its turn, is also subsystem of the whole medical field system that includes not only medical care but also healthcare education, research, etc.

The author in [Fer04] considered that the HISs "comprise the entire infrastructure, organisation, workforce and components for the collection, processing, storage, transmission, display, dissemination and disposition of information in the healthcare industry."

There is a number of differences between HISs and telehealth services including telemedicine or Telehealth (see Figure 2.2). While the HISs address modalities and methods to model and electronically store, exchange, process, etc., medical data from various healthcare sectors including patient personal information, history of decreases, laboratory information, billing aspects, administrative data, etc., the later rather relate to the transfer of medical information and expertise over different telecommunication technologies including phone, internet, and sometimes even network. As defined in [Jos05], Telemedicine regards the transfer of the medical information between a healthcare professional and a patient with the goal of consulting and sometimes remote medical examinations. In addition, Telehealth also consists of preventive aspects besides the therapeutic aspects. Telehealth means delivery of various health-related services and information using different telecommunications technologies. The delivery could be very simple such as a discussion about a patient case on the phone between two health professionals, or sophisticated, e.g., using videoconferencing between providers at facilities located in different countries, or even more complex such as using robotic. In this thesis, this way to perceive HISs is adopted.

2.2.2.1 Terminology and Definitions

One of the most referred HIS in eHealth world is the EHR system. In the IT healthcare literature, the concept of electronic representation of a patient's record is presented using different terms. The collected literature was reviewed in order to place the state of the art research and definitions in this field and the outcomes were too large to be cited comprehensively here. The EHR term is often used interchangeably with other abbreviations, but the term is given slightly different or similar definitions. Even though it is impossible to cover all understandings of EHR, in the following

[11] *telecare* is the application of e-technologies to assist patients who choose to be located at home rather than at a health care facility [Jos05] (page 44)

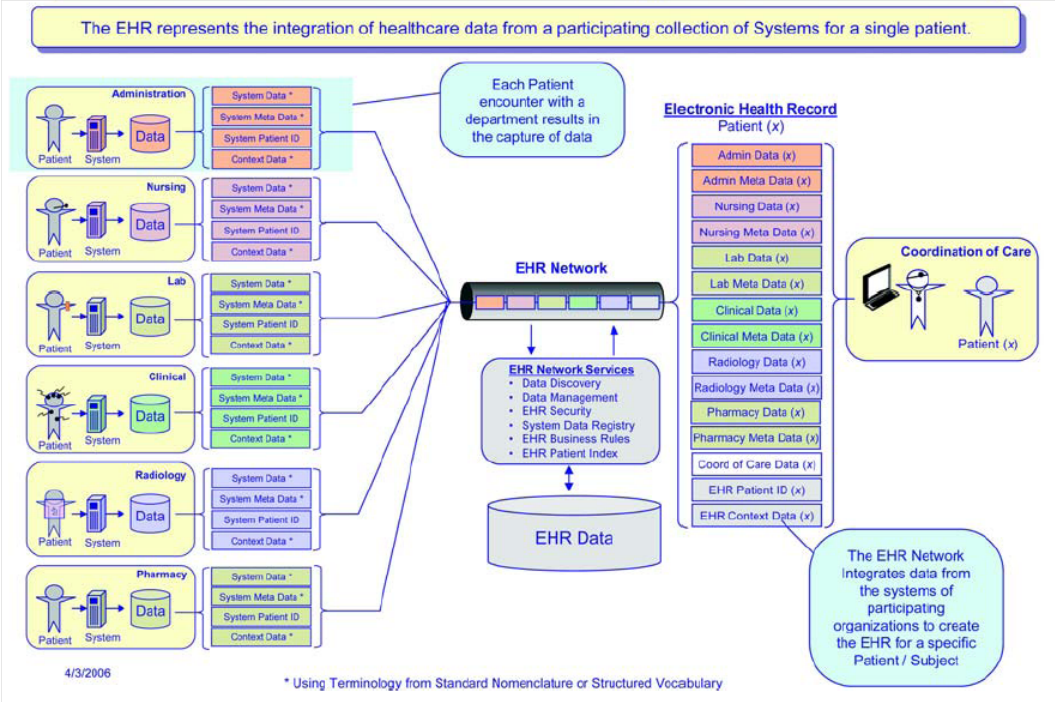


Figure 2.3: Electronic Health Record - Concept Overview

paragraphs, some relevant definitions regarding the EHR concept and associated terminology are introduced. At the end, the interpretations adopted along this thesis are introduced.

The European Committee for Standardisation (CEN 13606, 2000) simply introduced EHR as "a healthcare record in computer readable format"[CEN00].

Amatayakul stated in a book [Ama04] on EHRs published in 2004, that the vision of EHR emerged more than two decades ago. The original term was Computer-based Patient Record (CPR), then later in the mid-1990s the term used was Electronic Medical Record (EMR). The target of CPR is to avoid dangerous medical mistakes, reduce costs and improve patient care and health system operations. Though the name has changed over the years, the vision of EHR barely changed: it helps not only to improve the quality of healthcare, but also to support research and education. However, the feasibility of this dream confronted healthcare specialists and IT engineers with each implementation attempt meant to technically capture the complexity of *EHR* systems. Only when put in practice, it was revealed that EHRs systems are not only a simple computer application but a collection of interworking systems, combining administrative and clinical workflows. The EHR system is not only a billing system or laboratory information system, it is more "an information system framework that accomplishes a set of functions".

As published in 2006 in a report by the National Institutes of Health (NIH) in US [NIH06], the complexity of an EHR system is illustrated in Figure 2.3, reproduced in this thesis. The central idea is that EHR systems are designed to combine data from the large ancillary services, such as pharmacy, laboratory, and radiology, with various clinical care components such as nursing plans, medication administration records, physician orders.

According to a draft technical report published by ISO/TC 215[12] in 2003, EHR is defined as a repository of information regarding healthcare episodes of a subject of care (patient), in a format that a computer can process. Additional aspects such as data privacy also have to be encompassed by the EHRs systems, e.g., secure storage and transmission, accessibility by multiple authorised users. This information usually refers to a patient's demographics, medical history, laboratory report, billing information, etc. The definition given by ISO TC215 in ISO/TR 20514:2005 is: an EHR is "a longitudinal collection of personal health information concerning a single individual, entered or accepted by healthcare providers, and stored electronically. The information is organised primarily to support continuing, efficient and quality healthcare and is stored and transmitted securely." [ISO05] Consequently, a medical software based on EHR must prove its long-term viability.

A similar view was provided by HIMSS, which defined an EHR as follows[13]: "The Electronic Health Record (EHR) is a longitudinal electronic record of patient health information generated by one or more encounters in any care delivery setting. Included in this information are patient demographics, progress notes, problems, medications, vital signs, past medical history, immunizations, laboratory data and radiology reports. The EHR automates and streamlines the clinician's workflow. The EHR has the ability to generate a complete record of a clinical patient encounter - as well as supporting other care-related activities directly or indirectly via interface - including evidence-based decision support, quality management, and outcomes reporting."

Likewise, the authors in [WLG09] outlined the fact that the EHR has a broader definition in that it implies a longitudinal collection of information about a patient across more than one healthcare organisation.

Joseph Tan in his book [Jos05] considered that EHR is "the lifeblood of eHealth care" and that the, EHR concept has different meaning than EMR or CPR. The difference between the semantics hidden behind the terminologies EHR, EMR, PHR, CPR is currently perceived also by many organisations and authors. In 2002, the authors in [DC02] stated that, the terms EMR and CPR are used interchangeably. They considered EMR as "solely an electronic representation of data that makes up a medical record" while CPR as "more of completely searchable representation of a patient and their care. It is almost a complete model of the patient". In a recent study (2010) on current status and problems concerning the interoperability of healthcare information systems [BH10], the authors clearly stated the principal characteristic of an EMR: it represents the record of health-related information maintained for each patient within a single healthcare organisation. An EMR is a source of data for the EHR.

Another type of HIS in eHealth is a PHR or electronic Personal Health Record (ePHR) system. In the following paragraphs, definitions of a PHR and the difference between a PHR and an EHR are provided. HIMSS defines an ePHR (PHR) as follows[14]: "An electronic Personal Health Record (ePHR) is a universally accessible, layperson comprehensible, lifelong tool for managing relevant health information, promoting health maintenance and assisting with chronic disease management via an interactive, common data set of electronic health information and e-health tools. The ePHR is owned, managed, and shared by the individual or his or her legal proxy(s) and must be secure to protect the privacy and confidentiality of the health information it contains. It is not a legal record unless so defined and is subject to various legal limitations." In other words, the PHR contains

[12] The ISO/TC 215 is the International Organization for Standardization's (ISO) Technical Committee (TC) on health informatics. TC 215 works on the standardization of Health Information and Communications Technology (ICT), to allow for compatibility and interoperability between independent systems.

[13] www.himss.org/ASP/topics_ehr.asp

[14] www.himss.org/ASP/topics_phr.asp

the same health information the EHR includes, with the essential difference that the maintainer of the PHR is the patient themself, while the EHRs are designed to be controlled and used by one or many healthcare professionals. The PHR may also provide valuable services to the individuals, such as drug-drug interaction checking or electronic messaging. There is a large number of PHR providers that enable the individuals to store their health information in a digital, secure format.

Regardless of the multitude of the perspectives of HISs such as EHR, PHR, EMR systems, the most important aspect is that their common goal is to improve the accuracy and clarity of health-care records reducing medical errors. All of them consist of similar interactions among interfaces which can be comprehended in the same abstract communication model. This abstraction will serve as a basis for the development of the methodology in this thesis.

2.2.2.2 EHR Usage in Various Countries

Different surveys [SOH+06], [SOD+09], [Fun06] pointed out that the current adoption of EHRs at a national level is characterised by great discrepancies within different countries around the globe. As reported by a recent Frost & Sullivan analysis [Sul09], the United Kingdom has been the pioneer in introducing e-healthcare initiatives among major Western European countries.

According to the Commonwealth Fund, International Health Policy Survey conducted in 2006 [Fun06], the adoption of EHRs in the United States is poor among other developed countries. Only about a quarter of primary care doctors in the U.S. (28%) and Canada (23%) use EMRs, in contrast to a large majority of primary care doctors in the Netherlands, (98%), New Zealand (92%), the U.K. (89%) and Australia (79%).

Another study conducted in 2009 and published in the *Health Affairs* journal [SOD+09], despite the fact that it indicates an increase of EHR' usage in different countries compared to 2006, still outlines the different readiness levels of NIH when it comes to applying the results of scientific advances at the edge of healthcare and computer technology. According to responses from 10,000 doctors in 11 countries, the United States and Canada lag behind several other countries in the use of EHRs. Only 46 percent of physicians in the United States and 37 percent of physicians in Canada use EHRs compared with 97 percent in New Zealand and 95 percent in Australia. Europe is situated at the top of the EHR usage rate with 99 percent in the Netherlands, 97 in Norway, 96 in the United Kingdom, 94 percent in Italy and Sweden. The rate of EHR use was 72 percent in Germany and 68 percent in France.

2.2.2.2.1 European Union

The European Commission has invested and guided eHealth research and development for over 20 years and thus it has contributed to the emergence of new and advanced technologies in several areas of healthcare.

Since 2004, the European Commission has been playing an official role in supporting the eHealth deployment, with the adoption of the *eHealth Action Plan* [15] to facilitate a more harmonious and complementary European approach to eHealth. The eHealth Action Plan has provided a tool to promote core ideas, organise working groups for discussion, raise awareness of the importance of eHealth among users, patients and health care professionals, and to foster collaboration among industry players by addressing technical issues, interoperability and benchmarking. The Action

[15] http://ec.europa.eu/information_society/activities/health/policy/index_en.htm

Plan sets out the steps needed for widespread adoption of eHealth technologies across the EU by 2010.

The eHealth Stakeholders' Group was created to enable healthcare providers, their suppliers and users to share their ideas and concerns. One focus of these exchanges has been to improve interoperability between eHealth services and tools.

According to the Action Plan, the majority of European health organisations and regions (communities, counties, districts) need to be able to provide on-line services, such as teleconsultation, e-prescription, e-referral, telemonitoring and telecare. The Commission's focus is also on supporting the deployment of health information networks based on both fixed and wireless broadband, mobile infrastructures and grid technologies. As many tools are based on the Internet, the faster roll-out of high-speed Internet access is a crucial development necessary to exploit the widespread benefits of eHealth, especially because those who need health services the most such as the elderly, disabled or unemployed, are often those who have the least Internet access.

Furthermore, in order to be aligned with the EC's requirements, all Member States across Europe should adopt procedures for testing and accreditation of eHealth tools and services. Additionally, the Action Plan defines another important aspect concerning the availability of a pan-EU electronic health insurance card for enabling European citizens to benefit of treatment all over the Union.

The 2008 *European Commission Recommendation on cross-border interoperability of electronic health record systems*[16] encourages the creation of means whereby authorised health professionals can gain managed access to essential health information about patients, with respect to patients' consent and taking into consideration appropriate data privacy and security requirements. The health information should be accessible from any place in the EU, but, at the same time, must be preserving fundamental rights of the individual, such as the right of protection of personal data. The health data could include parts of a patient's electronic health record, patient summary and emergency data.

The Recommendation is aimed at contributing to the achievement of overall European eHealth interoperability by the end of 2015. A set of guidelines for Member States is proposed, addressing the following objectives: to agree upon the principles of shared and interoperable eHealth information, to enable interoperability between health information shared among different healthcare systems, based on existing approaches and standards in use in different Member States, to resolve the various challenges of achieving cross-border EHR interoperability by building networked systems and services that cover the entire continuum of care, and to assess the benefits of and also the barriers in achieving eHealth interoperability and to identify the necessary preconditions and relevant incentives for overcoming them.

The Recommendation asks Member States to undertake actions at multiple levels: at political level by harmonising the Member States legislation and considering means of incentives for attracting eHealth investments; at organisational level by creating an organisational framework for interoperability between autonomous Member States' eHealth infrastructures; at technical level, by promoting technical standards and architectures; at semantic level, by agreeing on common priorities and specific applications; and at the level of education and awareness raising by analysing and monitoring the intended developments, and considering various education mechanisms.

In 2010, the European Commission unveiled *Europe 2020*, the EU's economic growth strategy. Part of the EU strategy, the Digital Agenda represents a 5-year plan for encouraging information and communication industry growth. The action plan includes measures to use technology for

[16] ec.europa.eu/information_society/newsroom/cf/itemlongdetail.cfm?item_id=4224

addressing the rising healthcare costs and the population's ageing issue. The Digital Agenda also outlines plans for providing effective interoperability between IT products and services.

2.2.2.2.2 United States

A national interest of the United States is the increase of the adoption of interoperable EHRs. This goal was expressed by the President Bush' statement in 2004: "every American should have an electronic health record by 2014"

In 2009, the US government adopted the HITECH Act[17] which was aimed at encouraging healthcare providers to convert their medical records into EHR systems. The US investments in this direction are aiming to stimulate the development and adoption of EHR, of strict and open standards and certification of software. Additionally, the government sustains and rewards the healthcare providers for making a "meaningful use" of certified EHR systems by covering different aspects such as improving care coordination, reducing healthcare disparities, engaging patients and their families, improving population and public health, and ensuring adequate privacy and security.

Under this Act, beginning with 2011, US medical providers that met certain EHR criteria could qualify for up to $107,750 over a five-year period, this amount being paid by Medicare and Medicaid, two governmental programs that provide medical and health-related services to specific groups of people in the United States.

Moreover, by 2015, when the rewards program will end, healthcare providers could be fined for not storing private patient health information in EHR systems that meet the government's encryption and security standards. Healthcare facilities who do not adopt an EHR by 2015 will be penalised 1% of Medicare payments, increasing to 3 over 3 years. The US government stated that out-dated technologies ended up costing more money and time to manage those EHR databases. In addition to that, EHR systems would significantly contribute to ensuring ready access to data for both doctors and patients, and also to enabling software-based interoperability between different healthcare facilities.

2.3 Interoperability of Healthcare Information Systems

The vision of a connected and interoperable healthcare infrastructure is one of the most significant healthcare industry efforts of the 21^{st} century[18]. The first step in transforming this vision into reality has been concretised thanks to the efforts of the standardisation bodies which published healthcare IT standards that would enable interoperability in a multi-vendor and multi-service HIS environment. Some of the important standardisation activities regard data model standards focused on data interoperability, terminology standards that help in providing precise, well-documented semantics to the data, workflow standards that provide support to the process of patient care, and finally, clinical care and clinical guidelines standards. However, regardless of the excellent work that has been done in the standardisation field, the provision of healthcare IT interoperability standards shall not be confused with interoperability.

In this section, significant standardisation efforts towards enabling interoperable HISs together with the main interoperability testing initiatives in this context are introduced.

[17] www.hipaasurvivalguide.com/hitech-act-text.php
[18] www.himss.org/ASP/topics_ihe.asp

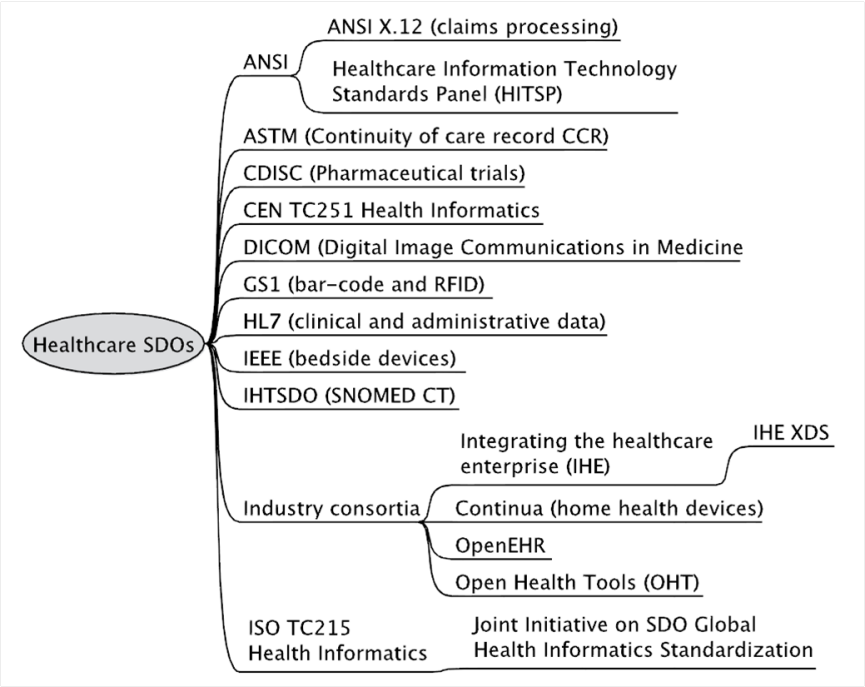


Figure 2.4: Healthcare Standard Development Organisations

2.3.1 International Organisations

A movement towards interoperable health records and health information exchange has started worldwide and relevant interoperable eHealth initiatives are ongoing in the United States and in the European region.

The landscape of organisations and groups involved in standardisation activities aiming at the achievement of the interoperability of HISs is diverse, large and spread all over the world. Figure 2.4 published in the book by Benson [Ben09] (page 77, Fig. 5.1) summarises the most relevant healthcare Standards Developing Organizations (SDOs) as by 2009. Even the common goal is to gain applicability worldwide, these organisations actually spilt their influence sphere in US - Healthcare Information Technology Standards Panel (HITSP) - and in EU - CEN. There is also a strong industry consortia supporting the standardisation activities. Furthermore, groups such as IHE are pushing the development of interoperability profiles to reduce areas of ambiguity to stronger interoperability. Nevertheless, it goes beyond the scope of this thesis to present these healthcare organisations.

2.3.2 Most-Referenced Messaging Standards and Specifications

Most healthcare IT centred SDOs produce standards, sometimes called specifications or protocols, for a particular healthcare domain such as administration, pharmacy, medical devices, imaging or insurance transactions (claims processing). As a general characterisation of current healthcare IT standards, the Tannembaum's statement - "the nice thing about standards is that there are so

many of them to choose from."[19] - captures perfectly the current state of healthcare IT standards landscape.

A basic primer on healthcare standards is presented in [IV07]. A good overview of EHRs standards available in 2005-2006 is presented in [EAR+05], [EAR+06]. A more recent snapshot from 2009 of the healthcare IT standardisation bodies is presented in [Sul09]. The current status, problems and research issues regarding the standards and interoperability in HISs is comprehensively captured in [BH10]. The authors identified issues such gaps in data standards, too many and often changes in standards, lack of content in terminology standards, and maybe the most important problem characterising healthcare IT standards is their overlapping.

2.3.2.1 HL7 - Health Level 7

HL7 is an American National Standards Institute (ANSI) accredited international standard developed by the Health Level Seven Standards Developing Organisation (HL7 SDO). It represents an EHR communications messaging scheme for exchange of information between applications. Data focused, the family of HL7 standards are designed to model healthcare information at a conceptual level that is platform or system independent. The HL7's aim is to improve care delivery enabling the interoperability between different HISs such as EHR, Patient Administration System (PAS), Electronic Practice Management (EPM), Laboratory Information System (LIS), etc.

There are many HL7 standard messaging versions [HL787b] coexisting, known as versions 2.x family [HL787c] and version 3 [HL705]. Currently, the version 2 of HL7 messaging standard[20] [HL787c] is supported by every major medical information systems vendor in the United States[21]. According to [Ben09] (page 106), this version is the most widely used and implemented healthcare interoperability standard in the world. Its coverage represents over 90% of all hospitals in the USA and is widely supported by healthcare IT suppliers worldwide. The next version, HL7 version 3 [HL705], is being refined and is expected to gradually replace HL7 v2.x based EHR implementations.

HL7 messaging standard version 2.x was originally created in 1987 and defines a series of electronic messages to support healthcare processes from different areas. According to Benson's book [Ben09] (page 94, Figure 5.1), HL7 standard covers messages that exchange information in the general areas of patient demographics, patient charges and accounting, patient insurance, clinical observations, encounters (registration, admission, discharge, transfer) orders for clinical service (tests, procedures, pharmacy, dietary and supplies), observation reporting including test results, scheduling of patient appointments and resources, patient referrals - specific messages for primary care referral, medical records document management, synchronisation of master files between systems. Over time, the standard has been updated regularly, resulting in versions 2.1, 2.2, 2.3, 2.3.1, 2.4, 2.5, 2.5.1, 2.6 and recently 2.7. The HL7 v2.x electronic messages are mostly expressed in textual non-XML encoding syntax, based on delimiters, notation known as Vertical Bars (VB) format.

These new generation HL7 v3 standards [HL705], developed in parallel with v2.x since 1996, differ from v2.x family in that all standards developed under v3 arise from an underlying RIM [HL706] - a data architecture introduced by HL7 Standards Developing Organisation (SDO)

[19] Tanenbaum A, Computer Networks Second Edition. http://en.wikiquote.org/wiki/Andrew_S._Tanenbaum

[20] In this thesis *HL7 version 2.x messaging standard* is used interchangeably with *HL7 v2.x*

[21] http://en.wikipedia.org/wiki/Health_Level_7

and intended to become a common functional framework for healthcare systems. RIM itself is an ANSI-approved standard and represents the basis for the specification of a messaging standard and for the semantic specification of message elements. HL7 RIM expresses the data content needed in a specific clinical or administrative context and provides an explicit representation of the semantic and lexical connections that exist between the information carried in HL7 messages, documents or services [Hin07].

Developed as syntax-independent models using the XML encoding syntax, HL7 v3 standards have as main objective to produce consistency in definition of different healthcare information objects and their representation in messages.

The **HL7 v3 Clinical Document Architecture (CDA)** is part of the HL7 v3 standard and defines an exchange model for clinical documents. It specifies the encoding, structure and semantics of the exchanged clinical documents.

A CDA document consists of two parts: a) mandatory human readable part and b) optional XML encoded structured part for automated processing. The structured part is based on the HL7 RIM, the HL7 v3 Data Types, but also relies on coding systems such as Logical Observation Identifiers Names and Codes (LOINC). CDA documents can be transported using HL7 v2.x messages, HL7 v3 messages, as well as by other mechanisms such as Digital Imaging and Communications in Medicine (DICOM).

The **HL7 Clinical Context Object Workgroup (CCOW)** represents a standard protocol designed to enable the visual integration of different healthcare applications. CCOW specifies technology-neutral architectures, component interfaces and data definitions, with the goal of sharing user context and patient context in real-time at the user-interface level.

2.3.2.2 DICOM - Digital Imaging and Communications in Medicine

The modern image display in healthcare bases on DICOM standard which is the equivalent to the film in the pre-digital era [Pia08]. DICOM [DIC85], is known as the de-facto standard for medical image communication. It is being developed by medical industry and medical professional organisations under the umbrella of the National Electrical Manufacturers Association (NEMA). The standard, consisting of 16 volumes in 2009 [Pia08], defines data structures and services for exchanging medical images and related information in a vendor independent manner.

The main goal of DICOM is to assure compatibility between imaging systems from different healthcare fields as cardiology, dentistry, radiology, endoscopy, surgery, and other information systems in healthcare [Sul09]. This way, the issue of how to integrate imaging patient data in the patient's EHR can be addressed.

Additionally, DICOM not only that controls parts of imaging workflows within a Picture Archival and Communication System (PACS) such as image acquisition (e.g., from devices like as Computed Tomography - CT - scanners) image storing/archiving, transfer, distribution, printing but without DICOM, image post-processing demanded by computer-aided diagnosis would not be possible.

2.3.2.3 CCR - Continuity of Care Record

The Continuity of Care Record (CCR) represents a standard describing the patient's health summary and is developed jointly by American Society for Testing and Materials(ASTM) Interna-

tional, the Massachusetts Medical Society (MMS)[22], the HIMSS[23] and other health informatics vendors.

The CCR includes data of the most relevant administrative, demographic, and clinical information facts about the healthcare of a patient, covering one or more healthcare encounters [AST]. The goal is to provide to the healthcare providers the means to aggregate all of the pertinent data about a patient and forward it to another practitioner to support the continuity of care.

In the opinion of HL7 and its members, the CDA CCD combines the benefits of the CCR and the HL7 CDA specifications. HL7 CCD is not a competing standard but rather a CDA-based implementation of the CCR.

2.3.2.4 ANSI ASC EDI X12 and UN/EDIFACT

American National Standards Institute (ANSI) engaged the Accredited Standards Committee (ASC) X12[24] to develop a new data format standard in the Electronic Data Interchange (EDI) area. Firstly published in 1982, the ANSI ASC EDI X12 [ETI10] standard was aimed to uniform standards for electronic exchange of business transactions (eCommerce) between different industry players (trading partners) in a system and provider independent manner. This standard is mainly used in U.S.

In healthcare IT, the EDI X12 standard defines message types used in healthcare billing systems such as invoice, purchase order, healthcare claim, etc. Each message type has a specific number assigned to it instead of a name. For example: an invoice is 810, purchase order is 850 and healthcare claim is 837.

In 1986, the United Nations Economic Commission for Europe (UN/ECE)[25] approved the standard United Nations Electronic Data Interchange for Administration, Commerce and Transport (UN/EDIFACT) [FAC]. UN/EDIFACT is an international EDI standard designed to meet the needs of both government and private industry. In healthcare IT, similar to the EDI X12, the UN/EDIFACT standard defines message structures for financial purposes such as healthcare claim, encounter request. While the EDI X12 is used in the US, most of the rest of the world uses the EDIFACT transaction sets.

An issue identified in [BH10] is that ASC X12 and HL7 have some duplications in standards used for reporting of clinical data associated with the claims process.

2.3.2.5 NCPDP SCRIPT

Another information system in healthcare IT is ePrescribing[26] which replaces a paper prescription. ePrescription[27] improves the patient's safety by preventing medication errors due to poor handwriting or ambiguous nomenclature. The ePrescription is then accessed in a pharmacy through the network infrastructure.

[22] www.massmed.org
[23] www.himss.org
[24] www.x12.org
[25] www.unece.org
[26] www.ncpdp.org/eprescribing.aspx
[27] www.epsos.eu - ePrescription means a medicinal prescription, i.e., a set of data like drug ID, drug name, strength, form, dosage and/or indication(s), provided in electronic format.

In US, the National Council for Prescription Drug Programs (NCPDP) is an ANSI-accredited SDO which creates and promotes data interchange standards for the pharmacy services in the healthcare industry. Even though the adoption of the electronic prescription is slowly growing, the majority of U.S. physicians in 2009 still write prescriptions by hand, as presented in a progress report about interoperable electronic prescribing [FSB09]. The authors of the same article argued this fact due to the confusion about standards for data exchange.

In Europe, according to a study conducted in 2007 [Sul07], the trends towards acceptance of the concept of ePrescription were encouraged, being popular in Germany, by the introduction of the Electronic Health Card (EHC). In UK, the terminology used for the ePrescribing (eRX) is Electronic Prescription System (EPS) and was part of the National Health Service (NHS) IT modernisation program. EPS shall not be regarded as an entity in isolation, but as a combination of EHR, Clinical Decision Support (CDS) systems and electronic transmission [Sul09].

The HIS used for a physician to order an electronic prescription is Computerised Physician Order Entry (CPOE). This system is used also to communicate orders via network to other medical stuff or departments such as laboratory, radiology. Furthermore, besides the fact that CPOE reduces time and provides error-checking, it is usually enhanced with support for other functionalities like CDS systems.

The overview of current ePrescribing standards published in 2009 in [FSB09] indicates that NCPDP SCRIPT 8.1 [SCA11] standard was preferred in US to other standards. However, in 2010, as stated in [SCR11] the SCRIPT 10.6 standard was selected in US as effective for use since July 2010. This version continues to support SCRIPT 8.1. The SCRIPT standard permits quick and accurate communication between the physician and the pharmacist, enabling security and tracking capabilities. It describes messages in a format choice: XML or EDI, and in order to implement services based on this standard, other components are required such as Data Dictionary (contains the actual field descriptions, sizes, formats, comments, and usage instructions), External Code List (a list of value codes that may contain links to other terminologies such as RxNorm, SNOMED, etc.).

2.3.2.6 Clinical Terminologies and Code Systems

Besides the variety standardised messaging structure schemes used to describe various stages and aspects regarding the healthcare of a patient, the healthcare IT field is also characterised by a large set of other types of standards used for defining various codes, terminologies, nomenclatures, drugs IDs, etc. Terminologies and classifications provide a framework to facilitate the storage, retrieval, analysis and interpretation of data. Enabling universal standardised nomenclatures in healthcare IT not only increases patient' safety and allows for more interoperability between HISs, but, theoretically, it provides also borderless compatibility and interworking between various EHR systems. In the following, the most referenced HIS vocabularies standards are briefly reviewed.

Systematised Nomenclature of Medicine - Clinical Terms (SNOMED CT)[28] is considered to be the most comprehensive, multilingual clinical healthcare terminology in the world. It has been owned and maintained since 2007 by the International Health Terminology Standard Development Organisation (IHTSDO)[29], a not-profit association in Denmark. SNOMED represents a general terminology for use in electronic health records and contains more than 311,000 unique concepts as of the January 2008 release.

[28] www.ihtsdo.org/snomed-ct
[29] www.ihtsdo.org

Logical Observation Identifiers Names and Codes (LOINC)[30] represents a database and a standard for identifying terms targeting the laboratory domain and other clinical observations, and is maintained by the Regenstrief Institute, an international healthcare and informatics research organisation. The latest LOINC release (version 2.32, June 2010) contains nearly 59,000 terms related to laboratory findings, e.g., chemistry, haematology, microbiology and clinical observations, e.g., vital signs, electrocardiograph (ECG).

International Classification of Diseases and Related Health Problems (ICD 10)[31], 10th revision, is the international standard diagnostic classification released by the WHO, targeting all general epidemiological, health management and clinical use. ICD 10 is currently available in 40 languages. In some countries, modified versions of the ICD 10 classification are used to fulfil specific national needs.

WHO Family of International Classifications (WHO FIC)[32] includes complex classifications of health-related terms in multiple languages. The WHO FIC is comprised of reference classifications for disease, functioning and disability, health intervention, as well as other derived classification.

The **WHO's International Classification of Functioning, Disability and Health (ICF)**[33] includes terms from body, individual and social perspectives, being structured in two lists a list of body functions and structure and a list of domains of activity and participation.

The **International Classification of Health Interventions (ICHI)**[34] aims to be a common tool for reporting and analysing the distribution and evolution of health interventions for statistical purposes.

Unified Medical Language System (UMLS)[35] represents a set of controlled vocabularies targeting the biomedical sciences, developed by the US National Library of Medicine (NLM).

2.3.3 Most-Referenced Workflows Recommendations

A step forward in enabling and assessing the interoperability of different healthcare applications was to propose vendor-independent interworking scenarios within particular HISs or between many HISs involving various applications. This way, when business workflows are available worldwide, as standards or as recommendations, the process of evaluation of organisational interoperability for applications adhering to a specific workflow, became much clear and applicable in unique ways.

2.3.3.1 IHE Integration Profiles

Integrating the Healthcare Enterprise (IHE)[36] represents an international organisation that includes healthcare professionals and industry partners, and is aimed at improving the information exchange methods between healthcare systems. IHE promotes the coordinated use of established

[30] http://loinc.org/
[31] www.cdc.gov/nchs/icd/icd10cm.htm
[32] www.who.int/classifications/en/
[33] www.who.int/classifications/icf/en/
[34] www.who.int/classifications/ichi/en/
[35] www.nlm.nih.gov/research/umls/
[36] www.ihe.net

standards in the context of specific healthcare settings for achieving the goal of deploying interoperable IT healthcare systems [IHE97].

IHE Integration Profiles[37] offer an implementation guide to specific integration problems, by documenting the system roles (Actors), communication standards and design details. The IHE integration profiles provide exact information on how standards such as DICOM, HL7 and security standards can be implemented to meet specific clinical needs. IHE covers multi-domains with integration profiles for radiology, cardiology, laboratory and Information Technology (IT) Infrastructure enabling interoperability both within and across multiple healthcare organisations. The IHE role is to provide the framework in this puzzle by connecting or integrating all of the healthcare-based standards and build an inoperable healthcare system for the future.

The specifications for the integration profiles, inclusive the technical details, are published in **IHE Technical Frameworks (TF)**. Vendors, claiming that their products comply with an IHE TF, publish a so called IHE **Integration Statement** document, which states which functionality from the profile is supported.

An approach towards the formal modelling of IHE integration profiles considering also the human interaction is provided in [AMA+09]. For modelling IHE workflows, the authors propose the BPEL for People Language (BPEL 4PPL) for defining business processes that also considers people as another type of participants by introducing people activities and tasks along the process.

2.3.3.2 Other Specifications and Approaches

Many healthcare organisations define the patient data time-motion and how it is collected. Successfully defining workflows requires knowing each step or stage of the processes, that the processes are structured correctly and that the integration to other supporting systems can be achieved.

In a recent article [dCJB+10] from 2010, the authors propose to standardise the representation of clinical trials workflows in UML in order to enable an international site comparison. The result of their analysis was that they managed to formalise in UML the workflows corresponding to two Brazilian clinical trials sites in rheumatology and oncology.

2.3.4 International IOP Research Projects

There are many projects addressing and covering at an international level the problematic of interoperability between different HIS systems. In the following, the most relevant initiatives in this direction are presented.

The HITCH Project: Healthcare Interoperability Testing and Conformance Harmonisation (HITCH) [HIT11]. The goal of the project goal is to develop the European Union's roadmap for interoperability and conformance testing of information systems in the field of healthcare. HITCH analyses different aspects of eHealth interoperability testing, focusing on the organisation, performance and quality management, and proposes how they can be further developed in order to be more complete and effective. Aspects analysed by HITCH are the interoperability testing tools, interoperability quality management systems, interoperability testing-based quality labelling and certification of eHealth products, etc. HITCH was launched in 2010 and is expected to conclude in 2011, its results consisting of the state-of-the-art analysis and roadmap for eHealth interoperability testing domain. The analysed list of the available interoperability testing tools contains the

[37] www.ihe.net/profiles/index.cfm

TTCN-3 interoperability tool presented as the second case study of this thesis.

The epSOS Project: Smart Open Services for European Patients [EPS11]. The project is an European project organised by 27 beneficiaries covering 12 EU member states. epSOS aims at developing an eHealth framework and an Information & Communication Technology (ICT) infrastructure that will enable secure access to patient health information, particularly with respect to basic patient summaries and ePrescriptions between different European healthcare systems. As previous efforts had limited targets, such as regional or national health networks, epSOS advances interoperability across all EU member states.

The infrastructure and tools used for the IHE-Europe Connectathon [Con10] testing process have been selected as the foundation for testing among the epSOS partners. epSOS benefits from IHE-Europe support in organising IHE Connectathon-inspired events for testing health systems (named epSOS Projectathons). In November 2010, an epSOS Projectathon was held in Slovakia to test whether the interoperability of different countries' healthcare systems met epSOS specifications. Nine countries (Austria, Czech Republic, Denmark, Spain, France, Greece, Italy, Sweden and Slovakia) successfully tested cross-border patient data exchange. As of 2011, 23 European countries will cooperate in the epSOS project on cross-border transfer and sharing across multiple member states of fully coded patient summary and fully coded ePrescribing data, the information being translated into the respective language.

The eHealth ERA Project [ERA09] aims at establishing an effective European Research Area (ERA) in eHealth. The objective of the project is to contribute to the coordination of member states' eHealth strategy definition and implementation, as well as eHealth-related research and technology development. The project analyses eHealth roadmaps and research programmes across Europe, identifies common priority issues, and develops a roadmap for joint actions. eHealth ERA recommends sustainable mechanisms for effective transnational cooperation between all or several of the participating states for their mutual benefit, thereby supporting the eHealth action plan adopted by the European Commission.

The CALLIOPE Project: Creating a European coordination network for eHealth interoperability implementation [CAL10] represents a European thematic network aimed at the deployment of interoperable eHealth solutions. CALLIOPE is supported by the European Commission's ICT for Health Unit. The project was launched in June 2008 and had its closing event in November 2010. CALLIOPE represented 22 different EU and European Free Trade Association countries.

CALLIOPE comprises a dedicated forum where stakeholders such as decision makers, implementers, professionals and patients can share knowledge and good practices on establishing interoperable eHealth services. A successful collaborative platform has been established within the CALLIOPE network to include many actors in eHealth interoperability in Europe. CALLIOPE delivered an EU eHealth interoperability roadmap which is aimed at accelerating the deployment of eHealth services and identifying common ways to reach interoperable solutions. CALLIOPE also delivered an eHealth interoperability recommendation review which consists in the analysis performed by the CALLIOPE Network members of the 2008 European Commission recommendation on cross-border interoperability of EHR systems.

The SemanticHEALTH Project: Sharing knowledge in eHealth Information Systems [SEM08] represents a specific support action funded by the European Union 6th R&D Framework Programme (FP6) and is aimed at developing a European and global roadmap for research in the health ICT area, focused on semantic interoperability issues of eHealth systems and infrastructures. SemanticHEALTH objectives include the identification of short-term and medium-term research needs for achieving semantic interoperability in eHealth systems, the

analysis of unsolved interoperability research issues, the consideration of non-technological aspects such as health, social, economic, legal policies and the integration of results from other FP6 studies.

The RIDE Project: A Roadmap for Interoperability of eHealth Systems in Support of COM 356 with Special Emphasis on Semantic Interoperability [RID07b] is a FP6-funded Coordination Action aimed at providing a semantic interoperability roadmap for eHealth systems at European level which has been finished at the end of 2007. As the final outcome of the RIDE project, RIDE roadmaps for semantic interoperability in eHealth domain covered semantic interoperability in EHRs, patient identifiers, eHealth messaging systems and clinical guidelines business processes, thus conveying implementation guides for an interoperable European health network.

2.3.5 Certifications Bodies

In the pursuit of interoperability of EHR systems, activities targeting interoperability labelling and measuring start to play an important role. In the following, the most relevant organisations that seek to measure the interoperability of various EHR systems are introduced.

2.3.5.1 Certification Bodies in United States

CCHIT®- Certification Commission for Healthcare Information Technology [CCH04] is an independent, non-profit organisation which has the public mission of accelerating the adoption of health IT. CCHIT was founded in 2004 and is certifying EHR systems since 2006. More than 200 EHR products had been certified by mid-2009, representing over 75% of the marketplace. This certification commission is approved by the Office of the National Coordinator (ONC), US Department of Health and Human Services (HHS) as an ATCB (ONC-ATCB). CCHIT offers two main certification programs:

- CCHIT Certified ®program: an independently developed certification that includes a rigorous inspection of the functionality, interoperability and security of an EHR and its compliance with the test criteria developed by the experts of the Commission.
- ONC-ATCB certification: tests complete EHRs or modules of an EHR against the Final Rule issued by the Office of the National Coordinator, US Department of Health and Human Services in July 2010 to qualify EHR technology for the American Recovery and Reinvestment Act (ARRA).

Other US ONC-ATCB are **DGI - Drummond Group Inc.** [DRU10] - which offers EHR testing services and **InfoGard Laboratories** [Inf10].

2.3.5.2 Certification Bodies in Europe

EuroRec - European Institute for Health Records [Eur03] is an independent not-for-profit organisation, aimed at promoting in Europe the use of high quality EHR systems. As a European certification body, one of its main missions is to support the quality labelling for EHRs and to define functional and other criteria. Founded in 2003, EuroRec is organised as a network of ProRec centers, which are national non-profit organisations with the goal of spreading the use of EHR systems across Europe.

EuroRec participated in a series of projects. EuroRec was involved in Quality Labelling and Certification of EHR systems (Q-REC), a European project completed in 2008 and aimed at creating an efficient, credible and sustainable mechanism for the certification of EHR systems in Europe. Currently, EuroRec is also involved in the project EHR-IMPLEMENT, which has as general objective to collect, analyse and compare broad-scale electronic health record implementations among European countries and to provide best practice, policy and strategic recommendations for facilitating EHR implementation initiatives throughout Europe. Other ongoing projects with EuroRec participation are: HITCH [HIT11] and ARGOS (Transatlantic Observatory for Meeting Global Health Policy Challenges through ICT-Enabled Solutions).

2.3.5.3 Guidelines and Best Practices

Microsoft Connected Health Framework (CHF) - Architecture and Design Blueprint[38] represents a guideline for eHealth software best practices. Based on a "Knowledge Driven Health" vision, CHF offers an architectural approach for developing patient-centred health information networks. CHF establishes a business and a technical framework that provides application integration and technical interoperability. The requirements developed using Connected Health Framework include aspects regarding the integration of applications through the use of open industry standards, process orchestration, clinical messaging such as HL7 v3, standardised terminology coding such as SNOMED CT, ICD, plug-and-play application and module integration. CHF also addresses privacy aspects through requesting patient consent, encryption and so on, as well as business intelligence support for data analysis, best-practices identification, forecasting health needs, and decision support. The technical framework offers support for ensuring cross-platform interoperability, identity management, authentication and authorisation, data recovery and auditing access, data synchronisation, use of personal devices, and scalability.

IBM Health Integration Framework[39] provides healthcare-specific reference architectures and also a suite of tools, transformation engines and application adapters built on healthcare standards, such as The Health Insurance Portability and Accountability Act (HIPAA) EDI, HL7 and IHE integration profiles. The framework provides means to build and extend infrastructures in order to allow multiple systems integration and interoperability. The framework is based on a Service-Oriented Architectural (SOA) approach, focused on the consumption and reuse of business services.

2.3.6 Interoperability Testing of HISs

As outlined in [SL09], there is a need for conformance testing, interoperability tools and techniques in all healthcare IT domains in order to facilitate and assure the integration of healthcare enterprises. The healthcare IT domain is characterised by dynamism, because of the continuous development and improvement of standards [SL09]. Indeed, standards are necessary, without them interoperability being simply impossible [NR10]. Standards describe the message syntax, but they do not give additional information on how the messages generated by healthcare IT systems can be combined in a workflow. Hence, to address the interoperability testing, firstly, the selection and the evaluation of interoperability scenarios are necessary [MT08].

Furthermore, there are significant differences between even two versions of the same standard,

[38] www.microsoft.com/industry/healthcare/technology/HealthFramework.mspx
[39] www.ibm.com/software/industry/healthcare/framework.html

e.g., HL7. This fact enhances the potential for semantic interoperability issues and even more, it conducts to a very limited provision of interoperability testing tools on the healthcare IT market as a consequence of the unwillingness of the test investors to commit effort and resources in order for their testing tools to keep up with the constant changes in the healthcare standards. Moreover, as identified in a recent publication from 2010 [NR10], "interoperability testing in healthcare is very new". By initiating the first IHE Connectathon interoperability test plug-in event about 10 years ago, IHE became pioneer in the healthcare testing [NR10]. Also, as the authors highlight in the cited article, the uniqueness of this event and the large number of the already tested systems have conducted to transforming Connectathon in de-facto testing in healthcare IT.

Additionally, achieving interoperability between heterogeneous components in an EHR system requires various types of testing [NR10]. In the following, the requirements for interoperability testing of HISs identified in the literature together with the main approaches regarding interoperability testing of HISs are introduced.

2.3.6.1 Interoperability Testing Requirements

The interoperability of HIS was not very often concerned in the literature. The majority of the existent testing solutions are rather in-house test tools instead of neutral open tools. In 2010 the authors in [NR10] also came to the same conclusion when stating that very few projects exist for interoperability testing in healthcare.

One challenge that the HISs, and implicitly the testing tools for HISs, have to face is to cope with the extremely diverse clinical information covering diagnostic images laboratory or cardiology, orders and results, etc., as well as with the multitude of healthcare specific standards [NR10].

As identified in [KAP09], another particular challenge is that "interoperability has several different levels including technical, workflow, privacy and semantics". Consequently, the test systems have to support all these levels and be able to correlate information from different levels in order to discover more complex interoperability issues.

2.3.6.2 Interoperability Test Approaches and Tools

The authors in [L. 08] propose and examine testing strategies for the HL7 version 2.x messaging standard which provides particular testing challenges due to the many options it allows. They introduce two testing methods for evaluating the conformance. The first method bases on the Upper Tester (UT) that takes place of the user or of the business application supported by the SUT and Lower Tester (LT) a to replace a peer application of the SUT, approaches introduced also in section 1 of this chapter. The second method employs actors to interact with the applications being tested. As the authors themselves identified, a limitation of the first method is that, the usage of this form of black box testing can not be applied to an environment composed of several communicating applications. Additionally, it is not suitable if there are multiple systems to be tested simultaneously. The second method based on the actor approach assumes that the testing framework employs HL7 applications to provide the operational environment in which the SUT functions are tested. The advantage of this method is the extendibility of the test system, i.e., regardless of how many applications are employed in the operational environment, actors can always be employed to replace them in the testing environment.

In [RJHSD10] the authors addressed the problem of semantic interoperability. The authors examine whether the provision of XML-based standards in eHealth such as Clinical Document Ar-

chitecture (CDA) Continuity of Care Document (CCD) or ISO/EN 13606 supports the semantic validation of of standard-based EHR documents. An approach that semantically validates the EHR documents is described. In the authors' view, this implies that the EHR documents are checked for conformance with the underlying archetypes and the reference model by means of XML Schema, without requiring an additional validation language. The original XML Schema corresponding to the EHR documents is this way enhanced with semantic elements and the actual validation of EHR document is done against the resulted XML Schema. A tool to automate the different steps of the semantic validation approach is provided.

An interoperability and conformance test framework called TestBATN is presented in [NAD09], [Dog10]. The framework's goal is to design and execute tests for HL7 version 3 based healthcare systems such that the interoperability at communication, document and business can be assessed. The scenarios regarded at the business layer, usually involve exchanging of messages between two actors and are published by IHE. Similarly to the methodology proposed in this thesis, the TestBATN framework tests for HL7 v3 communication, document syntax and business level and the tests are described using a test description language developed by the authors. The test system can also act as a proxy between the two interacting actors. From the architecture point of view, the test framework bases on the Upper-Lower Tester model introduced in cite [L. 08] and tests only HL7 interoperability. This is different from the interoperability test architecture introduced in this thesis which supports interoperability checking of various types of messaging formats, versions, etc. within the same workflow.

In [PRR+09] the TAXI testing tool is introduced. It supports the document validation of XML and HL7 v3 based healthcare documents for the PICASSO healthcare platform. The tool uses techniques to generate XML instances from an XML Schema. It provides automated support for the validation of transformations performed within PICASSO platform, this way addressing only the validation of syntactic interoperability.

The conception of a model for the HL7 version 3 messaging standards by using the Eclipse Modelling Framework (EMF) technology introduced in [BUT+09] was motivated by the fact that, currently the interoperability of HL7 version 3 healthcare systems, is very difficult to achieve given that the knowledge defining the model is spread over many Model Interchange Format (MIF) schemas, specifications, or it is scattered around domain experts. At the core of the EMF framework is the Ecore meta-model that describes models in EMF. One can either build these models from scratch or generate them out of available XML schemas, Unified Modeling Language (UML) models or annotated Java interfaces. A model would allow modelling the main HL7 core concepts making use of a high-level UML, Java, Extensible Markup Language (XML) representation.

However, these approaches are difficult to apply in the early stages of testing as they require complex set-ups and, especially, the presence of all interacting components. Compared with these approaches, the methodology introduced in this thesis proposes a different technique, in which the test system emulates the components that the system under test (SUT) needs to interact with. This way, the capability of a system to be interoperable is checked always against a reference implementation emulated by the TS.

2.3.6.2.1 LAIKA - EHR Testing Framework

LAIKA [LAI10], [McC08] represents an open source EHR testing framework developed with Ruby on Rails and Java programming languages. It was developed by Certification Commission

for Healthcare Information Technology (CCHIT) and the MITRE[40] corporation, being firstly released to the general public in March 2008. Laika is currently used to support CCHIT certification for interoperability in health information technology product. LAIKA performs testing of the input and output of EHR data for HITSP C32[41] specification and IHE XDS-MS integration profile.

The continuity of care was the initial focus of Laika v1.0 in supporting of interoperability testing [DGM+08]. CCHIT requires that EHR vendors need to have their product compliant with the HL7/ASTM Continuity of Care Document (CCD) data standard, that has been constrained by the HITSP C32 v2.1 specification (CCD/C32) [DGM+08]. The data entered in the LAIKA user interface will be provided to an EHR user via a single CCD/C32 XML document. LAIKA verifies that a CCD/C32 document produced by an EHR system is valid with respect to the standard as specified by HL7, ASTM, and HITSP. This way, the framework as it is, is able to cope only with the mentioned XML based data standards.

2.3.6.2.2 IHE Gazelle

The IHE Gazelle [Gaz10] project is aimed at developing tools for testing the interoperability of healthcare systems and the compliance of the exchanged messages with IHE TFs. Gazelle is used for participant registration and for both pre-Connectathon testing and testing during the Connectathon itself. The Bordeaux Connectathon [Con10] (April 2010) was the first one run entirely with Gazelle. At Connectathon, Gazelle acts as a test management platform and defines additional modules such as Demographic Data Server (DDS) to provide demographic data to be used during test session, Proxy which allows the capture of the messages exchanged by the participants to a test instance, etc. For the Connectathon 2011, Gazelle was extended recently with an HL7 v2.x Message Validation Service developed within the Gazelle project. This service is a Web Service as well as a Web User Interface, which allows the validation of HL7 v2.x messages. It is integrated in a more comprehensive service called External Validation Service (Gazelle EVS) that verifies the syntax and semantic (by having access to various code sets) of the exchanged messages within an interaction flow. The external validation services support the validation of HL7 v2.x messages, CDA documents, Audit Trail and Node Authentication (ATNA) log messages.

[40] www.mitre.org
[41] www.hitsp.org/ConstructSet_Details.aspx?&PrefixAlpha=4&PrefixNumeric=32

Chapter 3

Interoperability Test Methodology for Healthcare Information Systems

You must be the change you wish to see in the world.
– Mahatma Gandhi

This chapter introduces a methodology for developing and executing interoperability tests for HISs including the methods to design and derive those interoperability tests. Although interoperability is a common topic among people and organisations in the healthcare IT world, the research does not address interoperability testing of HISs at a general level from an automated perspective, including all facets as test derivation, test parameterisation, test adaptability, test execution, test evaluation, etc., but rather encourages the industry vendors to participate with their systems at interoperability plug-in events as a primary and ultimate interoperability certification. However, the introduced methodology shall not replace the existent interoperability verification strategies used at plug-in events, but rather complete these strategies with a low cost and in-house pre-evaluation phase. Systems passing this interoperability testing pre-evaluation phase are more prone to be interoperable also in a real setup.

3.1 Challenges in Interoperability Testing at Plug-in Events

In practice, interoperability testing of HISs is done in an ad-hoc manner by plugging systems together. An example for interoperability testing events are IHE Connectathon events. However, this process of interoperability checking by directly plugging various vendor systems suffers for many drawbacks:

- huge costs: organising events at such dimensions needs a lot of effort, time and money on both sides: participants and organisers. A significant issue is the synchronisation and coordination of systems to interact during the event. This requires additional platforms for time scheduling, tests and results management, monitoring data, etc..
- restricted time for debugging: systems identified to be faulty need some time for investigations and fixes of software bugs. Usually the plug-in events last about one week and the developers have to interact with developers from many other vendors at the same time. De-

bugging around an identified problem may be an impediment in running other interaction scenarios.

- limited team participation: it is very expensive and almost impossible given the space constraints at the event (most plug-in organisers allocate two chairs per system) for a vendor to participate at the plug-in event with the whole team involved in the development of a system. Faulty pieces of software can be best debugged by developers responsible for those pieces of software. Hence, the remotely synchronisation of the team may slow down considerably the problem analysing and fixing process.

- restricted systems configuration for the plug-in event: the systems brought to the plug-in event differ from real-world configurations. Usually, the systems are installed on a minimal hardware, e.g., the whole system is installed on a laptop while in normal functioning it may access a distributed database. This could hide interoperability issues which will appear when the system is fully deployed.

- overhead introduced by the interoperability test environment: all systems participating at the event have to share network resources, e.g., the same DNS, router, proxy. This may result in loss of messages, mixed up ports, etc.

3.2 Challenges in HIS Interoperability Testing

As outlined in Chapter 2, the world of HISs is a very data-intensive domain where heterogeneous components built on top of various technologies and from different vendors have to interact each other. A critical need for the HISs is interoperability, defined by the Healthcare Information and Management Systems Society (HIMSS) as "the ability of health information systems to work together within and across organisational boundaries in order to advance the effective delivery of healthcare for individuals and communities" [oD09]. Different to applications from other domains, e.g., telecommunication, eGoverment, the healthcare information systems are characterised by a set of aspects that make the process of interoperability testing more challenging. These aspects are the subject of this section.

The result of the integration of two systems with different internal healthcare data information models, e.g., EHRs are dependent on the common semantics shared between the two models. Substantial efforts on standardisation, research and policy making have been invested to tackle the issue of organising healthcare data in a common and, if possible, unique structuring format. However, with the evolution and improvement of standards, coexistent implementations of HIS complying with different versions of standards have been in use. On the other side, irrespective of the underlying data standard version, HISs still have to interact and exchange healthcare data whose semantic meaning is used further by other HISs. Consequently, in practice it is possible that a component of HIS should support multiple versions of the same data messaging standard within an interaction scenario for the communication with various systems. With respect to interoperability testing, this typical situation encountered in the interconnecting HISs, especially when it comes to HL7 messaging standards [HL787c], [HL705], translates in a very challenging requirement that has to be addressed when designing and implementing an interoperability test system. A concrete example illustrating exactly this challenge is presented in Chapter 5 where the second case study on IHE Patient Identifier Cross-Referencing (PIX) integration profile is presented.

Another complexity facet of healthcare IT comes from the way the HIS semantically equivalent

information is represented at messaging infrastructure layer. Many healthcare messaging standards allow multiple data encodings, e.g., the HL7 messaging standards support XML (v2.x/v3), ER7[1] ("pipe notation" or "vertical bar" syntax), Simple Object Access Protocol (SOAP), etc., formats. This freedom in implementing and supporting various encoding formats might become an impediment for interoperability. Consequently, an interoperability test system should support all of them.

Information systems in healthcare are becoming increasingly more distributed and diverse. Part of the diversity is also the variety of transport protocols necessary to communicate healthcare data between or/and within HISs. For example, the patient data from one legacy system within a hospital, e.g., laboratory systems, needs to be correlated with patient information stored in another system, e.g., Admission Discharge and Transfer (ADT) system. The communication between the two systems happens over a message transport layer such as Minimal Lower Layer Protocol (MLLP) [MLL09], DICOM [DIC85], Web Services, etc. Figure 3.1[2] shows an example of messaging infrastructure and message transport layers used to transmit HL7 messages between two HL7 systems. Very often an interaction flow between such systems demands support for multiple message transport layers. This aspect constitutes another essential requirement that needs to be covered when designing interoperability test architectures for HISs.

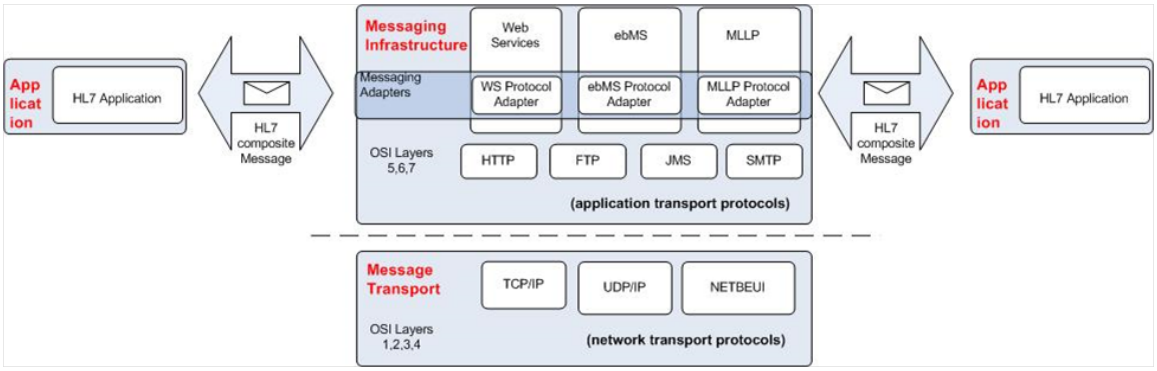


Figure 3.1: HL7 Reference Messaging Architecture

Furthermore, when deploying HISs, they still have to be adapted to healthcare enterprise specific configurations and set-up information. This information may not only be configuration parameters for different components of HISs such as IP address or ports, but it also must be included in the semantic of messages along an interaction flow. For example, the sending and receiving facilities identifiers are saved in each message header (MSH) of HL7 messages. From this point of view it is not straightforward to plug and play HISs. Hence, the demand for configuration parameters associated to different real set-ups must be carefully covered and reflected in the architecture of each interoperability test system.

Besides maintaining the organisation technical and operational environments and technology lines, there is also a need for HISs to continuously accommodate to the complexity and changeability in terms of clinical protocols, administrative processes and messaging standards underlying their interfaces. This changeability is the response of a maturing process in the IT healthcare world while interoperability remains the greatest demand. This calls for another challenge when developing and executing an interoperability testing strategy: aspects related to rapid changeability of

[1] http://wiki.hl7.org/index.php?title=ER7
[2] http://wiki.hl7.org/index.php?title=Application_Architecture

standards for describing HIS interfaces have to be carefully addressed. The test system needs to be easy to change itself while preserving the capability of rapid localisation and reveal of faults.

3.3 HIS Testing Methodology Process

The participation to interoperability plug-in events remains a must in the process of interoperability check of HISs. However, applying systematic interoperability test methods prior to involving the system in direct communication with counterpart systems will help considerably in speeding up the process and deploying more reliable interconnecting systems.

For interoperability testing of HISs, there is a coming out need for a more general methodology which focuses on the requirements introduced in the previous section. The methodology should address in parallel different sides of the problem: functionality aspects such as message semantic, actor behaviour, etc., and non-functional aspects, e.g., degree of test system re-usability, degree of automation, etc. To address these aspects, a number of new concepts have been introduced. The interoperability test design methodology is based on an interoperability test design process described in Figure 3.2.

Step1: Integration Profiles Modeling. The process starts with the identification of the interacting systems and their required activities, i.e., behaviours. Many organisations or standardisation bodies already identified such complementary workflows and processes and published them either as standards, e.g., for clinical processes, for healthcare processes, healthcare provider research processes, healthcare provider educational process, or as recommendations, e.g., IHE Technical Frameworks (IHE TFs). Even more, healthcare enterprises define their own in-house specifications of workflows. The selection and identification of referred messaging standards and data types follows naturally.

In this thesis we adopt the term *integration profile* to refer to a healthcare workflow and its human readable description. This term has been introduced by IHE, which is the main international organisation publishing healthcare workflows, but along this methodology it refers also to non-IHE integration profiles. An *integration profile* contains detailed description about:

- *actor types*: the involved types of participant applications
- *transactions*: the data types and reference to messaging standards, structure and content constraints of the interchanged messages
- *sequence of transactions*: the order in which the transactions are performed

The goal of this step is to bring the specification of an *integration profile* from a human-readable format into a formal format which can be processed in an automated fashion. This means that the resulting integration profile model should contain the same information as the *integration profile* description, but no particularities of the healthcare environment should be reflected yet. For example, an interoperability interaction can be described between two *actor types*, but in a healthcare enterprise, several instances of the same actor types may exist. This should not be included in the integration profile model, but it will be regarded at Step2.

According to Ammann and Offutt [AO08], the necessity of introducing models along the process of designing tests is motivated by the argument that raising the abstraction level makes test design much easier. A high level of abstraction, known as Model-Driven Testing (MDT) [BDG+07], is

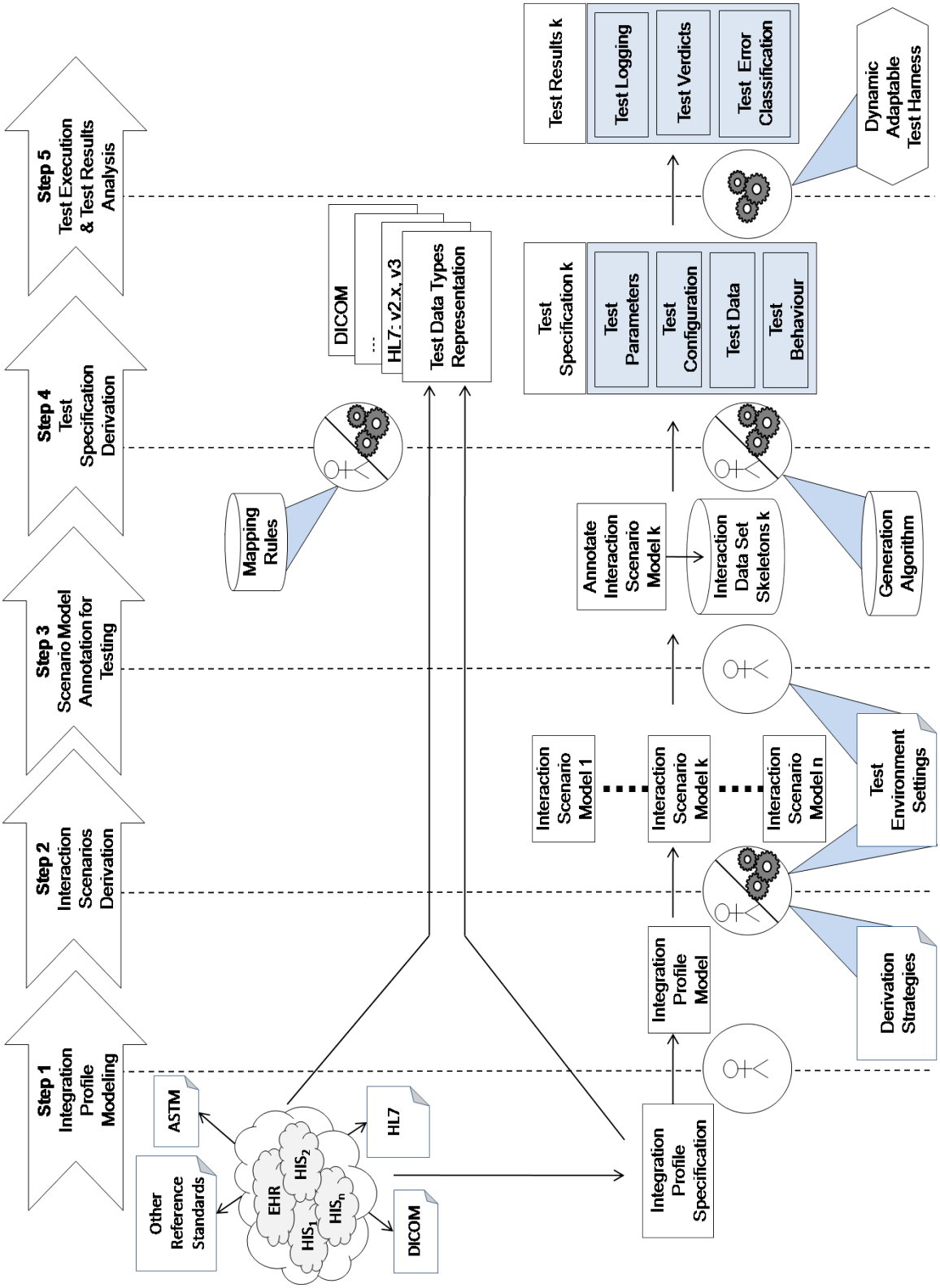


Figure 3.2: Test Methodology Process

obtained when modelling techniques are employed not only for the system but also to abstract the test design.

Building such a behaviour model on a formal basis not only that it helps in better understanding and clarifying the workflow requirements but also it can reveal inconsistencies appeared during the specification phase. Then, a formal specification of interactions constitutes a rigorous input in the process of collecting and correlating healthcare data used for research or educational purposes, which, in the end, will lead to better knowledge that assures a better health care. Furthermore, accessing a formal and rigorous specification for healthcare integration profiles may be the starting point for risk analysis or similar loss estimation systems.

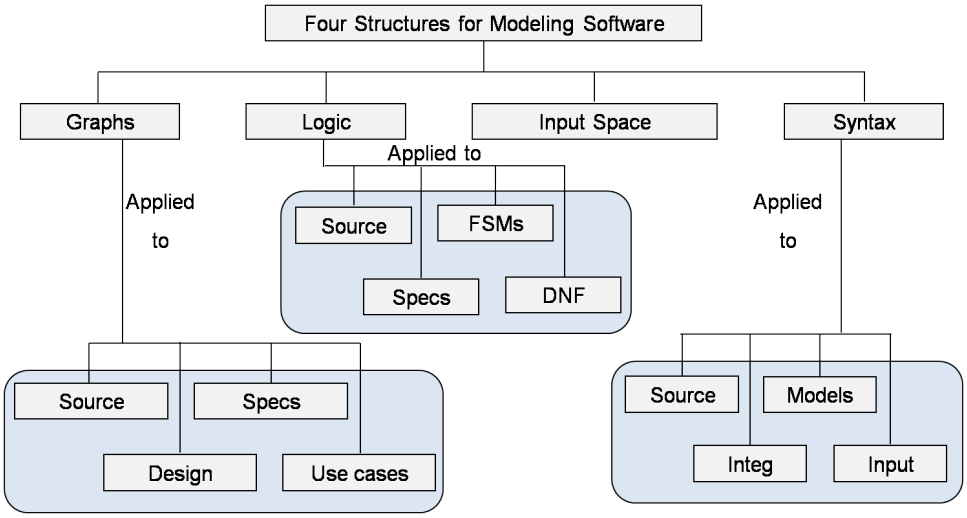


Figure 3.3: Coverage Overview

With respect to the choice of a modelling formalism, there are various techniques to model the set of interrelated healthcare activities within a workflow. Figure 3.3 shows four groups used to represent the known software testing techniques, according to the book by Amman and Offutt [AO08]: graphs, logical expressions, input domain characterisation, syntactic structures. These techniques are not presented here, but several criteria to select an adequate modelling formalism are discussed. The first criterion for this selection is based on the possibility to express at a higher degree of completeness these activities, tasks, conditions, etc., described by a healthcare workflow. Adequate methods to achieve that are based on graphs techniques (the first group of techniques in Figure 3.3), which are the most commonly used for testing. For example, a graph is conceptually the same, no matter whether it comes from source code (i.e., control flow graph), design documents, specifications, requirements models, use cases, FSMs or state-charts. UML is the most widely used modelling language, but approaches based on other languages such as Specification and Description Language (SDL) were also already applied in telecommunication domain [MGM07] to specify communicating systems and protocols. Within the scope of the RIDE project [RID07a], a description is given of how to express the IHE integration profiles through eBusiness eXtensible Markup Language (ebXML) and Business Process Specification Schema (ebBP) languages [ebX07]; the reasoning behind is that, concise and machine-processable configuration information can be used in an automated way [DKG+06]. The second essential criterion for choosing a modelling language is the technical back-end for model diagrams, i.e., how the models are stored and how the information from the models can be accessed in a programmable

way.

This modelling step calls for domain-knowledge and therefore it cannot be done in an automated way. However, sometimes, parts of the knowledge can be derived automatically. For example, the referred messaging standards within an integration profile may be given in a formal description, e.g., HL7 v2.x messaging standard is delivered also as a database, hence, an automatic process can be conceived in order to derive test data type specifications in a target testing language. Details about this aspect are presented when describing Step4 of the methodology process.

Step2: Interaction Scenarios Derivation

The output of Step1 is a complete system model which represents the input for Step2. In Step2 different *derivation strategies* are applied in order to obtain the interaction scenarios of interest for interoperability. The output of Step2, namely the interaction scenarios, is the basis for Step3 where the interaction scenarios are annotated, and the basis for Step4 where the annotated interaction scenarios are used for test specification derivation.

Various *derivation strategies* can be applied. Figure 3.4 (from the book in [AO08]) correlates different *criteria* to be considered when deriving tests according to the technique employed for modelling the system. The *derivation strategies* try to fulfil these *criteria*. In the case of graphs, the most commonly used structure for testing [AO08], tests usually are intended to *cover* the graph in some way according to selected *criteria*. Here, the key idea is that tests are based on the selection of execution paths of the system. These execution paths correspond actually to the interaction scenarios from the methodology process. Hence, it is essential to follow some criteria, based on which, the derivation strategies will deliver interesting interaction scenarios from interoperability point of view.

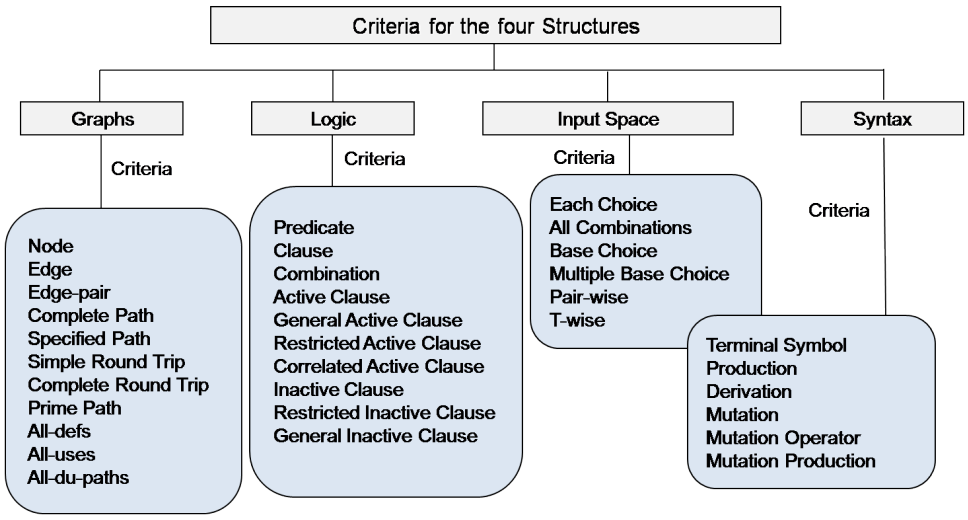


Figure 3.4: Coverage Criteria

Since the input for this step is a formal description of an integration profile in a machine readable format, automation of this step can be conceived. Algorithms implementing the *derivation strategies* referred above have been already proposed and are supported by a variety of academic or commercial tools. However, human intervention and domain knowledge about the healthcare environment may still be needed to some degree, as for instance:

- selection of coverage criteria parameters
- setting of configuration parameters, e.g., number of instances of actor types
- extend the interaction scenarios for different data consistency checks, which are not specified as "activities" in the integration profile specification (input for Step1).

The .getmore tool [SEP09] is a relevant example of a tool which applies different derivation strategies such as full path coverage, full edge coverage, full node coverage, named path, prioritising strategies on system models formalised with UML activity diagrams. This tool has been used within the ReTeMes European project [ReT09], the research context where the concepts of this thesis have been elaborated, and it has been further extended for the needs of automation of Step2 of the process. Other examples of such tools whose aim is to generate tests out of functional and behavioural system models specified in UML are Conformiq Tool Suite (former Qtronic) commercialised by Conformiq [QTR10], Test Designer™ from Smartesting ® [STD08], TestCast Generator (used to be called MOTES) from Elvior [ELT08]. Different than these tools, the Spec Explorer tool from Microsoft [SPE10] has as input the intended behaviour model encoded in Abstract State Machine Language (AsmL) [ASM10]. Even more tools are presented in Utting and Legeard book [UL06] on model-based testing.

This thesis does not try to propose an automation technique for this step since many such approaches already exist. An example of such a contribution is covered in doctoral thesis of Z. R. Dai [Dai06] where methods and tools for system model transformation to the test model are investigated. This thesis rather attempts to outline that various automation techniques can be employed in order to obtain the most interesting interaction scenarios for interoperability.

Step3: Scenario Model Annotation for Testing

The process continues with the preparation, i.e., annotation, of each interaction scenario for testing. The methodology presented in this thesis assumes that some of the actors, i.e, their whole behaviour, set of activities, configuration parameters, etc., are substituted by an interoperability test system. Therefore, some extra information about these aspects has to be encapsulated in the interaction scenario models in form of annotations. At this annotation step, different types of information to be annotated can be distinguished:

- test set-up information: which actors play the role of the test system (TS) or the SUT.
- configuration parameters: ports, IP, host names, etc. Additionally, these configuration parameters can be assigned default values. They should be easy to modify later on in the Step4.
- information to indicate which concrete messages should be used or how concrete messages can be built up from skeleton messages. The required concrete messages exchanged within an interaction scenario are derived from the so called *interaction data set skeletons*. The advantage of providing such a set of data skeleton messages is that they may be re-used in other interaction scenarios, doing only the required tuning.
- semantic information (e.g., refer to name of functions), i.e., how to use configuration parameters in message creation preparation. For example, a correct acknowledge message coming as a reply to a request should contain information (e.g., message ID) from the request.

The result of this step is a set of *annotated interaction scenarios* which serve as input to the test specification derivation step. Each *annotated interaction scenario* should contain the necessary

information to derive a complete interoperability test specification. Since it is at the latitude of the tester to decide how to use and how to associate these annotation to different HIS components, the activities at this step can not be done automatically.

Step4: Test Specification Derivation

As a consequence of the previous three steps, the model of an annotated interaction scenario contains all the information needed for deriving interoperability tests. Step4 deals with the derivation of interoperability test specifications to a target testing language.

The derivation consists of two parts. The first part deals with the representation of message types in the chosen testing language. The information about the referenced messaging standards, e.g., HL7, is obtained from the *integration profile specification*. As any transformation demands, a set of *mapping rules* is required. The transformation itself may be automated as long as the messaging standards containing the type definitions are available in a formal description as well. However, in case only an informal description is available, the automation is not possible.

The second part involves *generation algorithms* applied to the *annotated interaction scenarios* obtained at the previous step. While the types representation in the testing language is common for all interaction scenarios, a separate test specification is generated for each interoperability test scenario. Throughout this work, a *test specification* is understood as the collection of all elements needed to abstractly define an interoperability test scenario:

- *test parameters*: the configuration parameters identified at the previous step, which influence the test data and test behaviour execution.
- *test configuration*: representation of actor types and actors in test language elements. The test configuration should also concern the possibility that an actor can exchange message over multiple protocols.
- *test data*: concrete messages expressed by using elements of the target testing language and built on top of the *test data types* defined in the first part of this step and using the *interaction data set skeletons* introduced at Step3.
- *test behaviour*: representation of the sequence of *transactions* for the test scenario in the target testing language.

Step5: Test Execution and Test Results Analysis

The process ends with the test execution and test results analysis. In order to execute the abstract test specification which is the output of the previous step, adequate *test harness* is mandatory.

The *test harness* is the central part of a test execution platform which supplies the functionality to execute and validate tests [ORLS06]. According to Binder [Bin99], a *test harness* is a software just like an application system. Among the elements of the *test harness* are: stubs, test drivers and test control systems. These components together build the system that starts the SUT, sends messages to it and then evaluates the responses. In this thesis we refer to the *test driver* to the part of the *test harness* that is responsible for handling the communication means between the TS and SUT.

The SUT architecture influences the design of the test system and especially the design of the *test driver*. Implementers of HISs playing the role of the SUT have a high degree of freedom with regard to system configurations, number of actors, interfaces, protocols, etc., involved in an

workflow. With respect to interoperability testing, it becomes very time consuming and effort demanding to adapt the test platform every time the system configuration changes and accordingly the test specification. This problem has been tackled in this thesis by looking into the concept of the *dynamic adaptable test driver*. The main benefit of this approach is that the *test driver* does not require further changes and, consequently, the test scripts are ready to run against the SUT. Further details about this concept are presented in the next section.

The result of the execution is captured in the form of a logging trace and a verdict whether the test succeeded or failed. Given the particular case of HIS systems which are very data-intensive systems, a concept for different levels of interoperability verdicts and a concept for the classification of interoperability problems is needed. These concepts are detailed in the next section.

In order to easily track the interaction flows or identify interoperability and conformance inconsistencies, the logging framework should offer the possibility to store any interchanged message, time stamps and information about who originated and received those messages. However, this information would not be enough without an adequate presentation format, to help the tester to quickly understand the potential issues. Furthermore, logging management information, e.g., systems which have been tested, execution identification, time stamps, shall be encapsulated in a logging trace as well.

3.4 HIS Interoperability Testing Concepts

The focus of the interoperability methodology presented in this thesis is to provide test design and test driver guidelines for automated test execution. The automation is possible mainly because of a suitable test design which comes along the test process.

The design covers the requirements and challenges outlined in sections 3.1 and 3.2. The process introduced earlier copes with these requirements and provides a five steps method to come from a specification to the final executable test. While the first two steps which concern the modelling and the *interaction scenarios* derivation, independent on testing, can be approached with existing methods such as Model-Based Testing (MBT), the last three steps call for specific design concepts related to interoperability testing of HISs.

In the following sections, the functional architecture of a test system capable of automated execution and the concepts introduced along this architecture are explained in more detail. To understand the motivation for the design decisions, a thorough discussion about possible interactions between SUT actors and TS actors is demanded.

3.4.1 Perspectives of Interoperability Testing

Interoperability tests have to be performed at the end points and at functional interfaces of the participating systems. Even though for interoperability testing a system needs to communicate with another system, and therefore the interoperability can be regarded from both sides (both systems are interoperable with each other), from the testing point of view, interoperability tests are targeted to one of the systems. For example, Figure 3.5 illustrates two systems which need to interoperate. The *SUT System* is one of those systems, while the second is used to test the interoperability of the *SUT System*. All interoperability tests are executed having as target the *SUT System*. However, the successful tests for the *SUT System* imply also that the *Another System* is interoperable with the *SUT System* for those particular tests, not only vice-versa. This way of regarding interoperability

only from one side is used at IHE Connectathon events where a vendor brings a system and runs only the interoperability tests designed for its system type. The test results apply to the tested system, but they validate the interoperability status of the counterpart system as well. However, IHE Connectathon may require for the counterpart system further tests; therefore, the counterpart system needs to be regarded separately as *SUT System*.

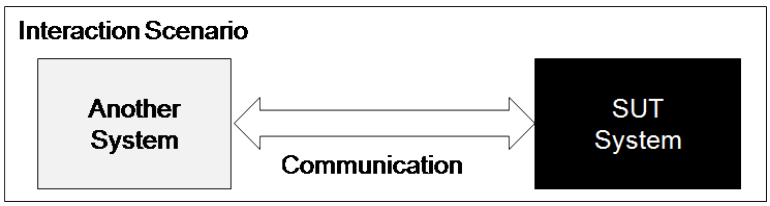


Figure 3.5: SUT System communicating with Another System

Out of this simple view from Figure 3.5 which shows two systems which are put together to run *a particular interaction scenario*, important conclusions can be derived. By running the interaction scenario between *SUT System* and *Another System* one can learn about **interoperability** capability of **both systems**.

Methodologically, the *Another System* in Figure 3.5 can be a real system or a test system simulating the behaviour of the real system, or a *reference implementation*[3] of the *Another System* which can be encapsulated in the test system. These ways are discussed in the following. These different ways of carrying out interoperability testing should be rather considered *stages of interoperability testing* than interchangeable types of interoperability testing. They should be operated all together in sequential order such that all aspects of interoperability are regarded. Unfortunately, in practice, two or even only one approach are used. The three different ways of caring out the interoperability testing are:

a) *Interoperability testing by replacing some components of the system with test simulators.*

b) *Interoperability testing by using reference implementations of some components and test drivers associated to all components.*

c) *Interoperability testing by using monitoring and proxy components*

The three stages are generally applicable to distributed systems and can be used in different domains. With respect to interoperability testing of HISs, the literature that has been regarded so far, outlines only options b) and c). The approach c) is used during plug-in events, such as Connectathon [Con10]. The approach a) has been firstly elaborated in this thesis. In order to understand the reasoning process, the approach a) will be introduced after explaining the approaches b) and c).

3.4.1.1 Approach b) - IOT with Reference Implementation

Figure 3.6 presents the second approach which regards interoperability testing by using a reference implementation instead of *Another System* and a *Test System Driver* associated to that system. The

[3] A reference implementation of a system is a real system which is considered to be working correctly.

reference implementation interacts with the *SUT System* and deals with all aspects of communication such as state machine, message flow, encoding/decoding. The *Test System Driver* triggers the functionality of the reference implementation and, whenever the reference implementation reaches an erroneous state, the verdict of the test is assigned to fail value.

Additionally, another *Test System Driver* can be attached to the *SUT System* (either automated by using a proprietary interface or manually by human). This approach can be generalised to the case that more than one *Another System* needs to interact with the *SUT System*. In this case, each *Another System* is replaced by a reference implementation and an associated *Test System Driver*. These *Test System Drivers* are coordinated by a *Test Coordinator*.

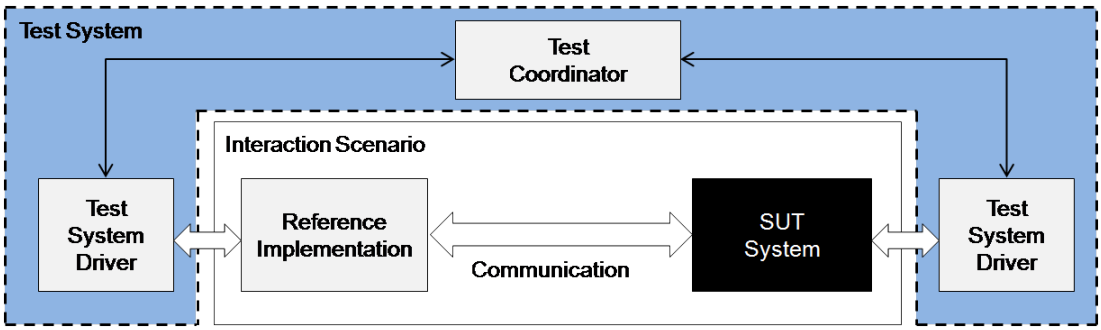


Figure 3.6: b) Interoperability Testing by Using Reference Implementation of Counterpart System

Reference implementations are easier to integrate and use. Nevertheless, this approach has the disadvantage that the reference implementations are not always functionaly correct and it is very difficult for many systems to have access to such reference implementations. Additionally, the communication between the reference implementation and its associated *Test System Driver* requires a proprietary interface.

This method can be considered a particular case of the Generic Approach to Interoperability Testing (GAIT) [ETS10d] recently published by ETSI and reproduced in Figure 3.7. In the figure, the *Qualified Equipment (QE)* system can be considered the *reference implementation* system and the *Equipment under Test (EUT)* is the equivalent of the *SUT System*. *The main difference is that, in the approach b) only one particular interaction scenario is regarded*, while GAIT covers all possible interaction scenarios. Additionally, in ETSI's approach the SUT is considered as the combination of the two systems: QE and EUT.

There are earlier approaches on testing interoperability by using *reference implementation* available [aCRSSH90], [Hog90]. The difference between the ETSI approach and those earlier approaches is that the later consider the *reference implementation* as part of the test system and not of the SUT.

Figure 3.8 shows Gadre et al.'s approach introduced in 1990 [aCRSSH90] to test the interoperability of a system (in the figure, IUT) by using a *passive interoperability test architecture*. Very important to mention is that the word *passive* used in the article had at the time of publishing the article a different meaning than it has today in the context of *passive testing*. By term *passive* the authors referred to running interaction scenarios only with valid data. This approach introduced the first idea of regarding interoperability testing with a reference implementation (RI) as part of the test system. Regarded overall, this proposed test architecture targets the interoperability capability of a system (SUT) by running it against a test system. Additionally, in this test architecture,

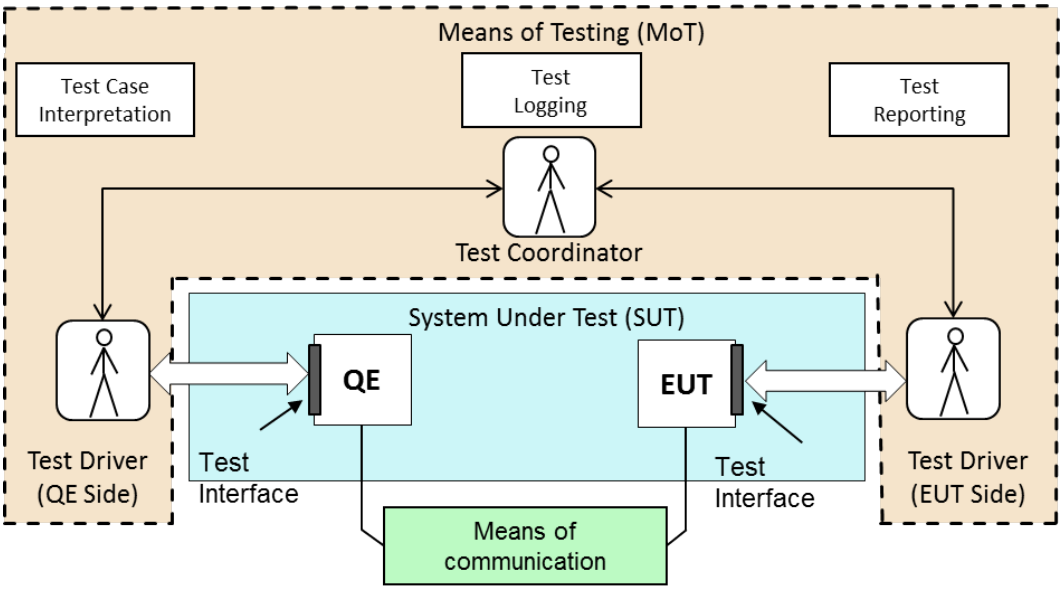


Figure 3.7: ETSI Generic Approach to Interoperability Testing (GAIT)

it is assumed that the TS, which includes the reference implementation, is able to perform the whole functionality required by any system supposed to interwork with IUT.

3.4.1.2 Approach c) - IOT with Monitoring and Proxy Component

Figure 3.9 shows the third option for interoperability testing. The test system in this approach is similar to the approach b) but, instead of using a reference implementation, it uses a real system called *Another System* in Figure 3.9. The *Another System* and *SUT System* are controlled by *Test System Drivers*. Additionally, a *Monitor* is monitoring the communication between the two systems and sends the monitored data to a *Test System Monitor Driver* which is capable of validating the correctness of the interaction from an interoperability point of view.

A first attempt to test the interoperability by using a monitoring component has been foreseen more than one decade ago and published in 1998 in [TW98]. The proposed interoperability test architecture is reproduced in Figure 3.10. As the figure indicates, the TS consists of the monitor (TC) and two additional components (MTC and TC) to control the interacting systems SUT1 and SUT2. TC stands for Test Component while MTC stands for Main Test Component. This is actually the model still used today at plug-in events to test the interoperability of two systems, denoted in Figure 3.10 SUT1 and SUT2.

The use of a monitor gives the advantage of testing the interoperability of two real systems. The two systems can be controlled automatically by the test system and/or by human. Although, the approach works very well for two systems in an isolated environment, it is technically difficult to realise such a monitor for testing a complex distributed system with many systems. It is even more difficult to handle encrypted data. In eHealth context, this approach was experimented in [NAD09] for HL7 version 3 based HISs.

A simplification of this approach is to have the monitor as part of the logging service of one of the tested systems. In large systems, this simplification is practicable only when the monitoring

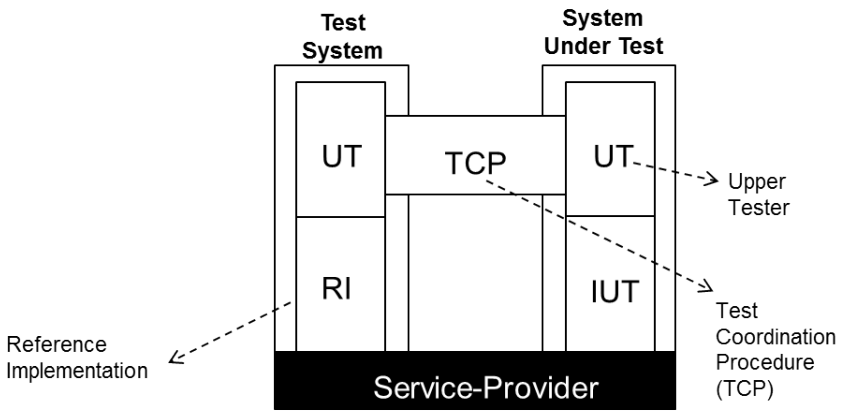


Figure 3.8: Passive Interoperability Test Architecture - Gadre et al., 1990

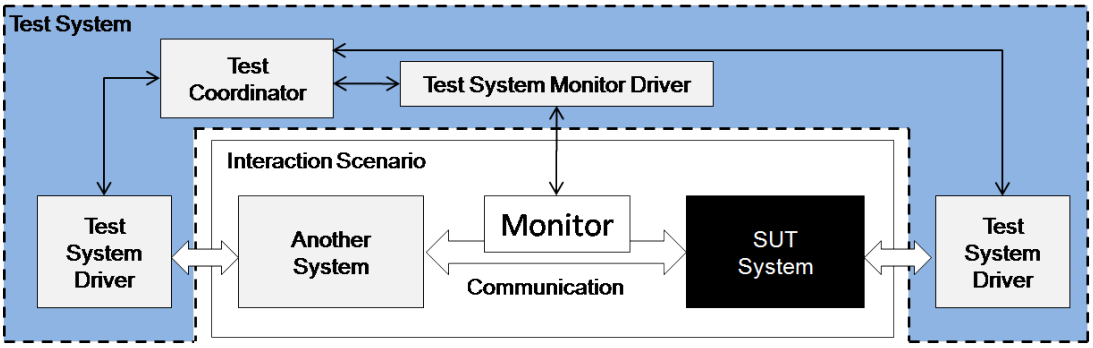


Figure 3.9: c) Interoperability Testing by Monitoring the Communication

feature is supported by the *SUT System*. Even so, adding this functionality to the *SUT System* may not be wished or possible as long as the functionality of the *SUT System* is altered. This approach is used during Connectathon events [Con10] in the healthcare domain.

Another simplification is to use a redirecting *Proxy* instead of a *Monitor*, as introduced for web service interoperability [WST03]. The *Proxy* wraps the communication ports between the tested systems. This approach is difficult to scale for large systems.

The general conclusion that can be derived when regarding this view on interoperability testing used at plug-in events is that, by running the interaction scenario between *Another System* and *SUT System* (respectively SUT1 and SUT2 in the approach from Figure 3.10 [TW98]) one can learn about *both*: conformance and interoperability capabilities of *both systems*. In other words, this test architecture, allows for investigation of compliance of exchanged messages within the IOP *interaction scenario* with the specifications of *both systems* and of capability of *both systems* to interact (to send, receive and understand exchanged) with one each other.

3.4.1.3 Approach a) - IOT with Test System Emulator

Figure 3.11 shows the approach of interoperability testing with replacement of the *Another System* in the scenario with a *Test System* which can simulate partially (e.g., only services of one interface)

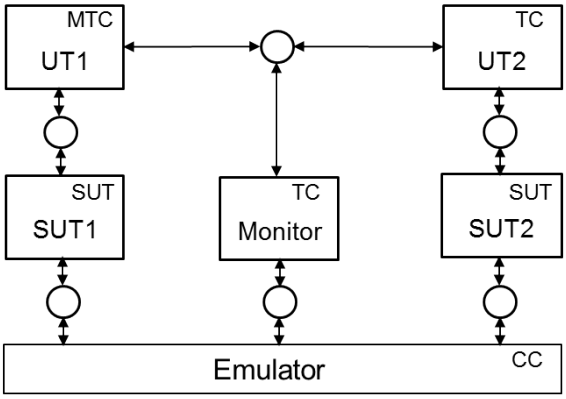


Figure 3.10: Interoperability Test Architecture - T. Walter et al., 1998

or completely (all interfaces, all services, etc.) the behaviour of the replaced system. However, with respect to the selected interaction scenario, the *Test System* supports all requirements, i.e., it emulates the functionality of *Another System* only imposed by that scenario. In the more general case, when the *SUT System* interacts with more than one *Another System* within the same interaction scenario, the *Test System* replaces all other systems.

This approach has the main advantage that the test system directly controls the messages interchanged with the SUT and can handle all states including the erroneous ones. It also has the advantage that the *SUT System* can be tested in house, not requiring the presence of the counterpart systems.

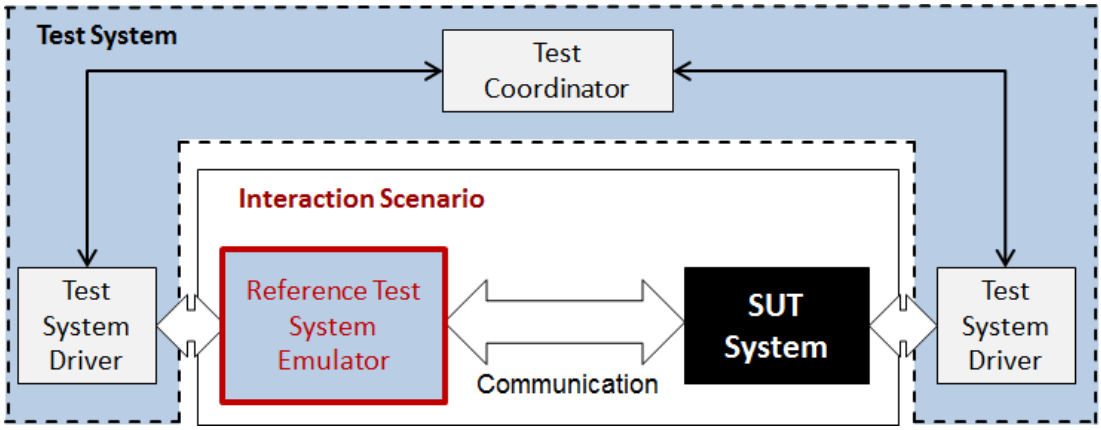


Figure 3.11: a) Interoperability Testing by Simulating Counterpart System

This test architecture can also serve for testing of *both* conformance and interoperability capabilities of *SUT System*. For a certain interaction scenario, a *reference test system emulator* is equivalent to a *reference implementation*.

This test architecture is only a mean to test a test purpose and not uniquely used for a specific type of testing, e.g., conformance, interoperability testing. A confusion that can easily arise is that, this test architecture (the TS interacts directly with a SUT system) is a pure conformance test architecture. In the following a clarification of this aspect is given. The ISO/IEC multi-part standard

9646 Open Systems Interconnection (OSI) CTMF [ISO92] is often misunderstood when it comes to conformance testing vs. test architectures. The test architectures introduced in CTMF (1995) are called *generic test architectures*, not *conformance test architectures*. CTMF even introduces the definition of a test architecture as a method of access of the SUT from within the TS, which does not impose which test purpose (conformance or interoperability) has to be validated. Indeed, this kind of test architecture proposed by CTMF (TS against SUT) has been used mainly for conformance purposes only, especially in the telecommunication world, but the conclusion that it is a conformance testing architecture is wrong. The distinction between conformance testing and interoperability testing has to be made based on the test purpose and not derived out of the test architecture. One can derive information about the interoperability capability of a *SUT system* using the same test architecture which is usually used for testing the conformance of the *SUT system*, the differentiating factor being here the test purpose.

The disadvantage of this approach is that such *reference test system emulators* are still difficult to realise, but, since they regard a particular *interaction scenario*, they are easier to implement than a whole *reference implementation*. However, this thesis presents a design method to generate and automate such test systems, thus reducing considerably the costs.

3.4.2 Triggered-based Interactions between HISs Actors

The purpose of interoperability testing is mainly to show that products from different manufacturers can work together. In general, two applications interoperate when one application invokes services from the other application which performs the task in the correct manner according to the specification and delivers the results. The delivered results must not only be received by the invoking application, but also be correctly understood, interpreted and used.

Figure 3.12 presents the basic elements involved in an interaction model. These elements are: *sender*, *receiver*, *interaction*, *trigger event*. This terminology has been introduced in the HL7 v3.0 methodology, but it is a model which actually suits to all types of interactions in healthcare IT, for any messaging standard, also non-HL7 [Spr06]. The sender is the actor who initiates the interaction with a receiver. An interaction is a single *one-way* electronic communication. In general, in any healthcare workflow, an interaction is triggered by a *trigger event* which is a set of stated conditions, which can be recognised by a computer system that initiates an interaction [Ben09]. More than one *interaction* can be initiated by the same trigger event, but each interaction is triggered by only one trigger event.

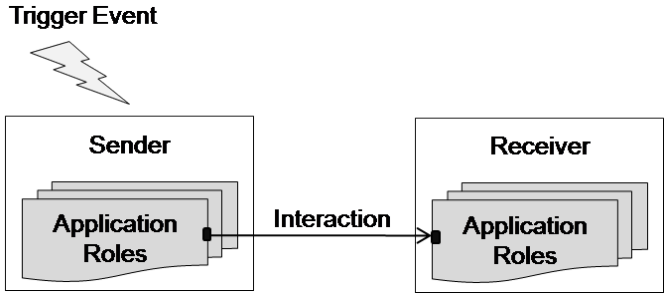


Figure 3.12: General Model of Interaction

Triggers can be grouped in the following categories:

- *interaction based*: this type of trigger event is initiated by other interactions, e.g., the response to a query (which is an interaction).
- *state-transition based*: these trigger events result from a state transition. For example, notification interactions are triggered when the sender changes its state.
- *user request based*: these trigger events occur at the request of a human user.

Application roles define the responsibilities a system should support and describe system components or sub-components that participate in interactions. A system or HIS application can have multiple application roles at the same time. For example, the HL7 methodology defines the following application roles[Ben09] (Section 10.3):

- *Placer*: an application that is capable of notifying another application about a significant event and expects the receiver to take action.
- *Fulfiller*: an application that is capable of receiving a request from a *Placer* application and starts the necessary actions in order to fulfil that request.
- *Confirmer*: an application that is capable of accepting a request from a *Fulfiller* application to send a confirmation to a *Confirmation Receiver*.
- *Confirmation Receiver*: a role implemented by a *Placer* indicating what types of confirmations it accepts.
- *Informer*: an application that is capable of notifying another application about a significant event (status changes), but does not expect any action on the part of the receiver. Paired with *Tracker*.
- *Tracker*: an application that is capable of receiving information about a significant event, but is not expected by the receiver to perform any action.

Figure 3.13 [Ste04] presents an example of different interactions between the introduced application roles of three HL7 systems: an ordering system, a laboratory system and a result reporting system. These systems implement the functionality of different actor types. Each system plays multiple application roles. The ordering system plays first of all the *Placer* application role (Observation Order Global Placer) by sending to the laboratory system a request (Observation Order Activate) demanding fulfilment of some actions on the laboratory system side. This involves that the laboratory system plays the role of a *Fulfiller* (Observation Order Global Fulfiller) when receiving that request. Additionally, the laboratory system also has a *Confirmer* application role (Observation Promise Global Confirmer) by sending back to the ordering system a confirmation upon the activation of the fulfilment task. This implies that the ordering system plays also the role of a *Confirmation Receiver* (Observation Promise Global Confirmation Receiver). The dotted arrow means a response to the original requesting system. Two further application roles of the laboratory system are of type *Informer*: Observation Event Global Informer, which informs the Tracker (Observation Event Global Tracker) about activation and completion of the initial request, and Observation Promise Global Informer, which notifies the ordering system's *Tracker* (Observation Promise Global Tracker) about the completion of the request.

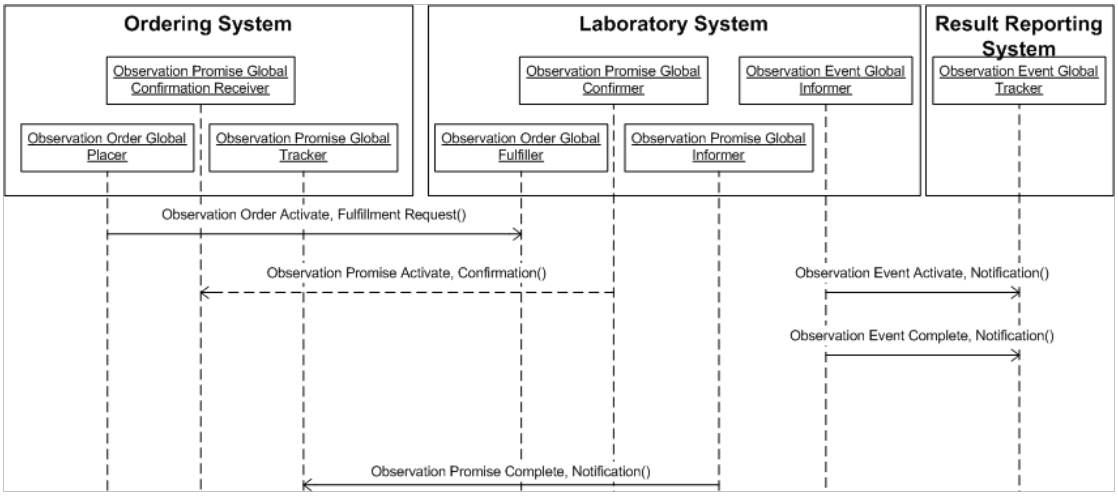


Figure 3.13: Example of Interactions between HL7 Application Roles

Even though DICOM does not explicitly name these application roles, one can recognise the same interaction patterns, as defined in HL7, also between different DICOM systems. For example, the interaction types *query* / *retrieve* images between two DICOM systems such as a viewing station system and PACS archive, associate very well to the HL7 interaction pattern *request* / *response* between a *Placer* and a *Fulfiller* systems. Similarly, a DICOM *Instance Availability Notification* service corresponds to the *Informer* - *Tracker* application roles from HL7.

3.4.3 Interoperability Message Exchange Patterns

In HIS interoperability interaction scenarios the roles of a system are not modelled as separate lifelines (UML-SD notation) but as single lifeline for the whole system. This means that an actor plays different roles and, consequently supports different interactions types. These interaction patterns are introduced in this thesis to explain 1) the different roles that the TS actors can play within an interoperability interaction scenario and 2) the derivation algorithm used to obtain an interoperability test specification.

After analysing the different application roles occurring in HISs, a number of message exchange patterns have been identified. These patterns, which are presented in the following, are the building blocks of any interaction scenario in HISs and can be combined into complex flows between many systems with multiple application roles.

Each pattern is illustrated in a figure describing the interactions between two systems. The first system, i.e., *System A* is the system which is initially triggered and, as a result, it initiates the first interaction with *System B*.

<Request - Immediate Response> Message Exchange Pattern

Figure 3.14 describes the pattern where a system requires a service of another system and receives an immediate feedback in the form of a response message. The response message includes the content required in the request. The response is always sent back, also in case of errors which are encoded or referenced in the response message.

This message exchange pattern is frequently seen between two HL7 based systems having the

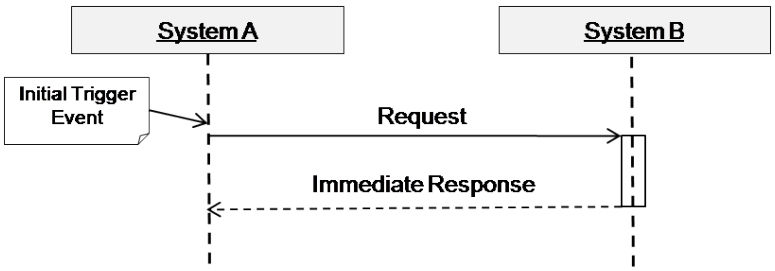


Figure 3.14: <Request - Immediate Response> Message Exchange Pattern

application roles of *Placer* and *Fulfiller*. Similarly, in DICOM based interactions, this pattern is mapped to a query/retrieve image interaction set, for example, between Radiology Information System (RIS) and PACS.

<Information - Immediate Response> Message Exchange Pattern

In case no content needs to be returned, but a simple acknowledge message, the message exchange pattern presented in Figure 3.15 is applied.

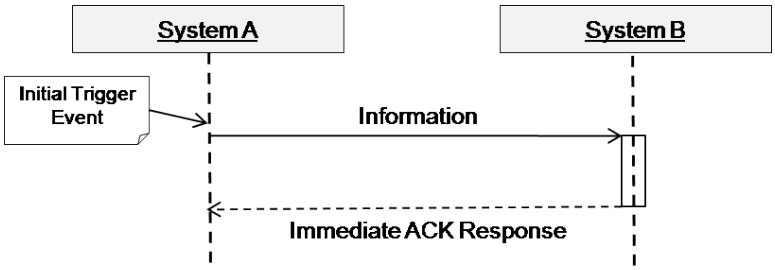


Figure 3.15: <Information - Immediate ACK Response> Message Exchange Pattern

Even though the flow of this pattern is similar to the flow of the previous pattern, from the semantic point of view, this pattern does not require the *System B* to make a certain action. Instead, the message sent to *System B* is rather informative message that has to be acknowledged upon the receive.

This pattern corresponds to the concept of acknowledged notification. For example, in HL7 based interactions, this pattern can be recognised in a dialogue between an *Informer* and a *Tracker*.

<Information - No Response> Message Exchange Pattern

In contrast to the first two patterns, this message exchange pattern illustrated in Figure 3.16 consists of only one interaction, thus neither response nor acknowledge is sent back by the *System B*.

This case is usually encountered when a system notifies another one without demanding a response or an acknowledgement. For example, this message exchange pattern occurs in HL7 based systems between an *Informer* and a *Tracker*, when the Tracker sends back no acknowledgement.

<Request - Deferred Responses> Message Exchange Pattern

This message exchange pattern illustrated in Figure 3.17 consists of two interactions and it works similarly to the publish/subscribe paradigm. *System A* plays the role of the subscriber by expressing its interest in events or information that *System B* can deliver. However, the *System B* may not

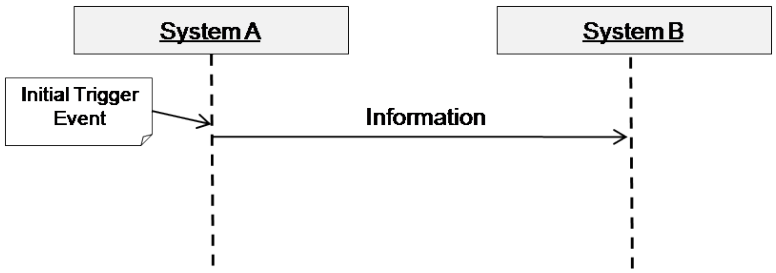


Figure 3.16: <Information - No Response> Message Exchange Pattern

necessary have the information at the moment the subscription was received. The information will be available for *System B* at a later point in time; therefore the responses to the *System A* will be deferred until the information is available. *System A* can express its interest in receiving a fixed number of responses or to get all available responses for a certain period of time.

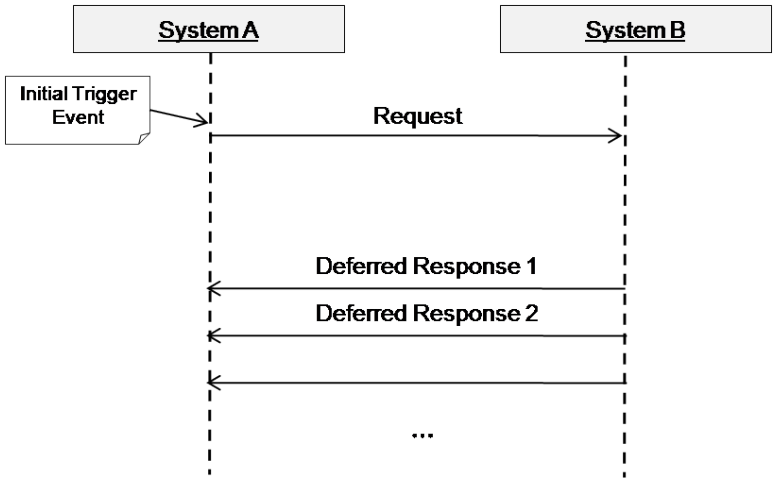


Figure 3.17: <Request - Deferred Responses> Message Exchange Pattern

A typical usage of this pattern is the model of the communication between patient care devices and consumer applications based on HL7 messaging standards. As an example, a consumer system which is interested in getting patient data, e.g., blood pressure, respiration rate, subscribes for these data to a vital signs monitoring device. The monitoring device will deliver this data in the form of several responses as soon as it is collected from the patient and it matches the filtering criteria of the consumer.

<Request - Deferred Response> Message Exchange Pattern

This pattern shown in Figure 3.18 is a particular case of the previous pattern, where only one response is expected and delivered. Similar to the previous pattern, the information will be available for *System B* at a later point in time; therefore the response to the *System A* will be deferred until the information is available.

For example, this message exchange pattern occurs in HL7 based systems between a *Placer* playing also the role of a *Tracker* and a *Fulfiller* which also plays the role of an *Informer* for the *Placer/Tracker*.

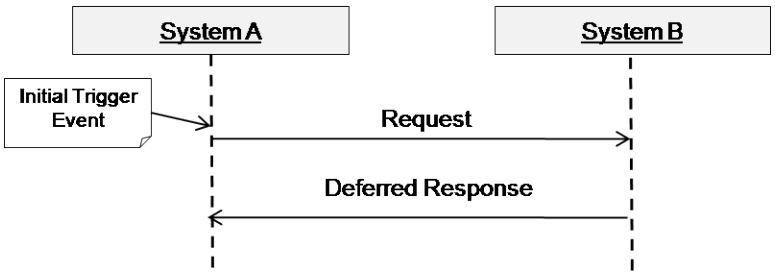


Figure 3.18: <Request - Deferred Response> Message Exchange Pattern

<Request with an Immediate Response and Multiple Deferred Responses> Message Exchange Pattern

The previously introduced pattern, *<Request - Deferred Responses> Message Exchange Pattern*, can be extended by adding an immediate response to the request message in order to acknowledge that the *System B* received the request and started the fulfilment procedure. This extension is presented in Figure 3.19.

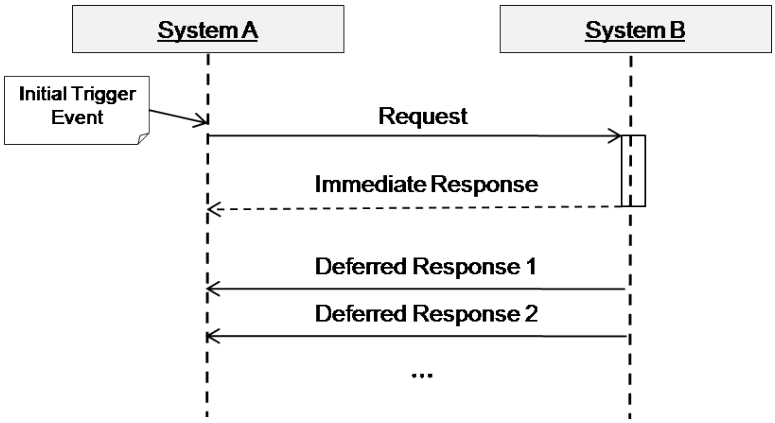


Figure 3.19: <Request with an Immediate Response and Multiple Deferred Responses> Message Exchange Pattern

In the HL7 application roles based terminology, *System A* would play the roles of *Placer*, *Confirmation Receiver* and *Tracker* for *System B* which also plays multiple roles of *Fulfiller*, *Confirmer* and *Informer*.

<Request with both Immediate and Deferred Response> Message Exchange Pattern.

The previous pattern can be particularised for an interaction with only one deferred response. This situation is illustrated in Figure 3.20.

3.4.4 Testing of Message Exchange Patterns and their Combinations

In HISs, the set of interactions between different actors are rather more complex than simple message exchanges between two applications, requiring message choreographies between more than two actors with different roles. These complex interaction scenarios consist of combinations of

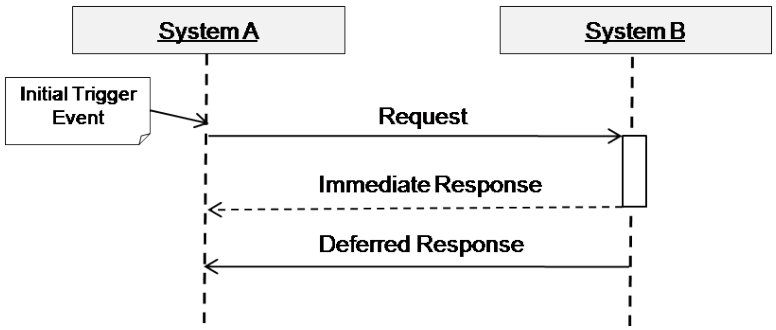


Figure 3.20: <Request with both Immediate and Deferred Response> Message Exchange Pattern

the message exchange patterns introduced in the previous section. In these combinations, a system plays several application roles at the same time. This complex set of related interactions in HISs which together perform one or more use cases is characterised by Benson [Ben09] (Chapter 10) as a *dynamic model*. This dynamic model describes the various roles in an interaction scenario and the expected behaviour of the sending and receiving components. On the other hand, the structure of messages defines a static model.

In this section, two combinations of patterns are presented. These combinations are exactly the combinations encountered in the two case studies which are discussed later in this thesis. The motivation for presenting these two examples is to show how the interoperability test methodology introduced in this thesis can be applied.

Combination of <Request - Deferred Responses> and <Information - Immediate ACK Response> Message Exchange Patterns.

Figure 3.21 shows an interaction scenario between three systems which is also met in the first case study of this thesis in section 5.1. *System A* and *System B* interact by using the *<Request - Deferred Responses>* message exchange pattern. *System B* and *System C* interact according to *<Information-Immediate ACK Response>* message exchange pattern. The interaction scenario is initiated by an initial trigger event occurring at *System A*. After this event, *System A* places a request to *System B*. *System B* defers the delivery of the responses to *System A* until *System C* sends to *System B* the information necessary to construct the responses for *System A*.

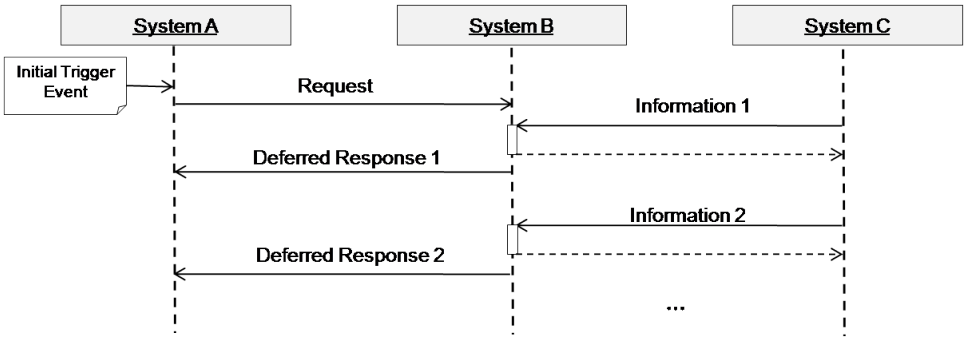


Figure 3.21: Combination of <Request - Deferred Responses> and <Information - Immediate ACK Response> Message Exchange Patterns

For interoperability testing of such an interaction scenario, the first step is to select the system to be tested. For the example in Figure 3.21, the selected system is *System B*. The next step is to group the remaining actors from the scenario to be simulated by the TS. These transformation correspond to the *Step3* in the process described in section 3.3 where the interaction scenario sequence diagram is annotated for testing. The result of these transformation is illustrated in Figure 3.22. The TS has to simulate the roles of *System A* and *System C* and has to support all the underlying message exchange patterns. Also, the sequence of interactions has to be preserved and validated by the TS. For *System B* the communication should be transparent, no extra configuration being required. In a similar way, any of the three systems or any combination of two of them can be selected as system to be tested while the remaining systems are substituted by the TS.

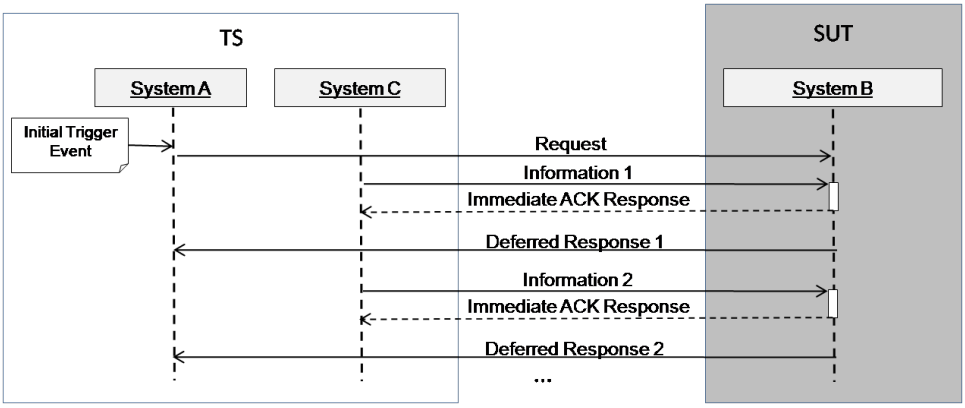


Figure 3.22: Example of Test Configuration for the Interaction Scenario from Figure 3.21

Combination of <Information - Immediate ACK Response> and <Information - Immediate ACK Response> and <Request - Immediate Response> Message Exchange Patterns.

Another example of combining message exchange patterns is depicted in Figure 3.23. This combination occurs also in the second case study of this thesis presented in section 5.2. The interaction flow between *System A* and *System B* is based on *<Information - Immediate ACK Response>* message exchange pattern. The communication between *System B* and *System C* is a bit more complex and follows two message exchange patterns, namely *<Information - Immediate ACK Response>* and *<Request - Immediate Response>*. The interaction scenario is triggered by an initial trigger event on the *System A*. As reaction, *System A* sends an information message (e.g., patient data registration, update) to *System B* which is immediately acknowledged by *System B* upon its receive. In reaction to that information, *System B* sends a further information message to *System C* which also immediately confirms the receive with an acknowledgement message. In the end, *System C* sends a request message querying for some content the *System B* which delivers that content in form of a response.

A possible test configuration for the interaction scenario depicted in Figure 3.23 is presented in Figure 3.24. In this example, the *System B* is selected as SUT while the *System A* and *System C* are simulated by the TS.

3.4.5 Conceptual Architecture of an Interoperability Test Framework

The realisation of test frameworks capable of testing interoperability of HISs using the first interoperability testing perspective presented above in section 3.4.1, demands a concept for a dynamic

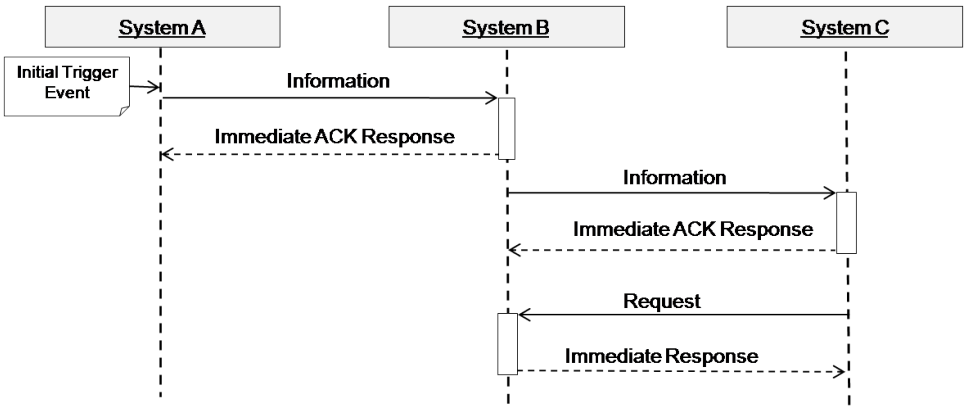


Figure 3.23: Combination of <Information - Immediate ACK Response> and <Information - Immediate ACK Response> and <Request - Immediate Response> Message Exchange Patterns

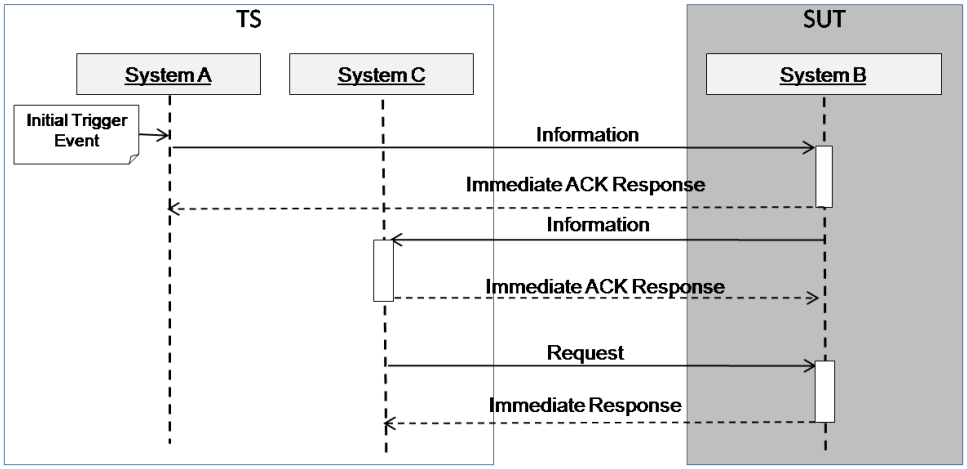


Figure 3.24: Example of Test Configuration for the Interaction Scenario from Figure 3.23

and configurable test system framework. This can be achieved by carefully designing the test system architecture and by adding the necessary elements for interoperability testing needs. A concept for a dynamic and configurable test system architecture along its elements is elaborated in this section. The possibilities of realising technically this architecture are evaluated in Chapter 4 where the elements of this architecture are instantiated by using a concrete test technology, i.e., the TTCN-3 test language.

3.4.5.1 Characterisation of Dynamic Adaptable Test Frameworks

There is a large degree of freedom for implementers of HISs with regard to system configuration, number of actors, interfaces, protocols, application identifiers, etc., involved in an workflow. All these aspects influence the configuration of the test system and its architecture as a whole. It is very time consuming and effort demanding to adapt the test platform every time the system configuration changes. Consequently, a concept for dynamic adaptable test framework is necessary.

A dynamic adaptable test framework should fulfil several needs. It should be able to generate and instantiate dynamically the internal processes to deal with the communication with as many actors as the SUT presents. It also should be able to dynamically handle the changes in the SUT architectures, e.g., new actors, multiple protocols. The architecture of the test system has to be designed in such a way that switching between protocol versions in the same workflow is realised transparently.

With respect to configurability of the test framework, further features are necessary. There are many parameters to be configured before starting a test. All information about the actors, e.g., identifiers, application domains, network addresses, need to be passed as input to the test system. Configurability calls for a mechanism to configure such parameters in an automated way, without changing the source code of the test system.

3.4.5.2 Concept for Test System Architecture

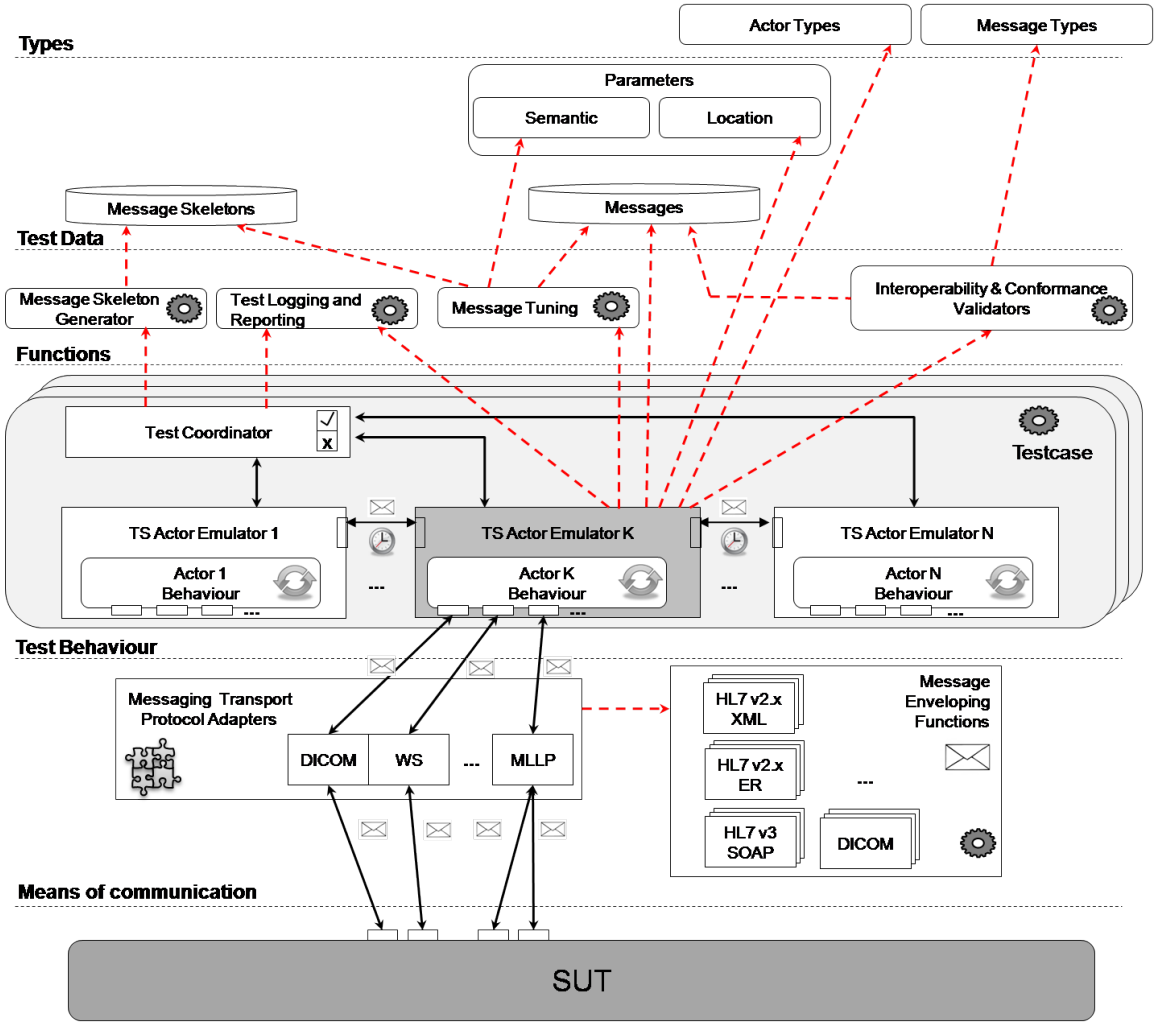


Figure 3.25: Functional Architecture of a Test System for Interoperability

The concept for a test system architecture is presented in Figure 3.25. The elements of the archi-

tecture are grouped into five layers which are described in the following. The architecture suits to the interoperability test process introduced before in section 3.3, more specifically to Step3, Step4 and Step5. This architecture is a generic architecture which can be realised with various tools and test technologies.

Types Layer. This layer contains the *actors types* and *message types*.

Message types describe the structure of patient data as indicated by the messaging protocols involved in a HIS integration profile. Especially for typed test languages, e.g., TTCN-3, the *message types* help to have a strong type checking during the test execution. This will ensure that the conformance to messaging standards will be validated simultaneously with the interoperability testing already during the interaction with the SUT.

To support a concept of emulating different types of actors within a test system, *actor types* have to be defined as well. An actor type describes the interfaces with the SUT and other emulated actors and the internal state variables.

As outlined in *Step4* of the test methodology process (in section 3.3), these types need to be written or generated only once for an integration profile and can be used for all interaction scenarios related to that profile. However, many integration profiles share the same message and actor types; therefore the type definitions can be reused for testing other integration profiles.

Test Data Layer. The test data are used as stimuli and oracles by the test system. This data layer consists of three elements: *parameters*, *message* and *messages skeletons*.

The first component, *parameters*, is used, as the name itself suggests, to define the test parameters. Conceptually, the parameters fall into two categories:

- *location parameters*: refer to networking configuration parameters such as, IP addresses and ports of actors and security settings.
- *semantic parameters*: refer to the various fields of data exchanged with the SUT and influence the meaning of the those messages. An example of a semantic parameter is the application ID which is encapsulated in all messages exchanged with the actor corresponding to that application. Such an information has to be defined as a parameter because it varies from one system implementation to another.

The second component, *messages* refers to the concrete patient data used within the test system as stimuli and oracle. They are instantiations of the *message types* from the types layer.

In order to ease the building of these concrete messages, an auxiliary component named *message skeletons* is desired. The role of this component is to provide message skeletons, i.e., incomplete messages or messages filled in with default values, which can be tuned to concrete messages. While the *messages* are specific to an interaction scenario, the *message skeletons* can be reused among multiple interaction scenarios.

Functions Layer. This layer contains functions used during test execution set-up or during the actual test execution. On one side, these functions manipulate, tune and check messages and on the other side, they provide test logging and reporting features. The components of this layer are:

- *message skeleton generator*: refer to the actual engine for generating *message skeletons* in the upper layer. These message skeletons can be generated either from existing ready to use messages or directly from a model of the data types.

- *message tuning*: to obtain the concrete *messages* introduced in the upper layer, message tuning functions help in changing or correcting the generated *message skeletons*. To achieve this, the *message tuning* functions make use of the semantic *parameters* also introduced in the layer before.

- *interoperability & conformance validators*: these are functions that are called during the test execution in order to validate the conformance of the messages received from SUT to the messaging standards and to check the correctness of the sequence of interactions, time delays, etc.

- *test logging and reporting*: they are functions used to keep track of the interactions and messages exchanges for later analysis, debugging or reporting.

While the *test logging and reporting* functions can be shared across multiple integration profiles, *message skeleton generators*, *message tuning* functions and *conformance validators* need to be implemented for each interaction scenario, but are not necessarily limited to them. In the case when the same message types and constraints are used among multiple interaction scenarios within the same integration profile, then these functions can also be reused as well. *Interoperability validators* stick to the state machines of actors involved in an interaction scenario; therefore they cannot be reused for other interaction scenarios.

Test Behaviour Layer. This layer contains multiple *TS actor emulators* and one *test coordinator* which are used to test an interaction scenario. These entities form a test configuration which creates the environment for the execution of a testcase. As introduced in the Step3 and Step4 of the interoperability test process in section 3.3, a test configuration can be automatically generated for a specific target testing language. By choosing a generation algorithm or by defining a set of mapping rules, a tool can automatically transform an annotated interaction scenario into a test configuration in the desired testing language. Since any automated transformation depends on the target testing language, a concrete set of mapping rules on how to derive TTCN-3 test configurations out of annotated interaction scenarios will be presented in the next chapter where an instantiation of the proposed generic interoperability test system architecture is presented.

On each *TS actor emulator*, an *actor behaviour* is executed. The behaviour of one actor is determined by the sequence of interactions that the actor has with the other actors and it is associated to a *TS actor emulator*. Similar to the test configurations, the *actor behaviour* can be automatically generated out of the same annotated interaction scenarios. The automated generation should be accompanied by a smart design of both *actor behaviour* and test framework. These design aspects, which again are target testing language dependent, are addressed in Chapter 4.

The creation, initiation and termination of actor behaviours are coordinated by the *test coordinator*. Additionally, based on the conformance and interoperability validator results, the *test coordinator* establishes the success/failure of the SUT. During the test execution, the actor behaviours use the types, messages, and functions from the upper layers.

For the interaction between *TS actor emulators* with the SUT and other *TS actor emulators* or *test coordinator*, communication interfaces need to be defined. These interface specify how the communication is performed in terms of protocols, encoding, etc.

Means of Communication Layer. This layer of the test system architecture deals with the transport of messages to and from the SUT. This layer consists of two components: the *messaging transport* component and the *messaging enveloping* component.

To deal with the flexibility of HIS implementations regarding the support for multiple transport protocols within the same interaction scenario, the test system architecture should allow for a plug-in-based transport component that can simultaneously deal with various transport protocols, e.g., HL7, web services, MLLP. These plug-ins are called *messaging transport protocol adapters*. They represent the communication channels used by the *actor emulators* interfaces from the upper layer to communicate with the SUT.

The messages exchanged over a communication protocol need to be enveloped into the formats understood by the SUT interfaces, e.g., HL7 v2.5 XML, HL7 v2.5 ER, DICOM. Hence, these enveloping functions have to be called before sending and after receiving a message from the SUT.

Both components of this level are integration profile and interaction scenario independent.

3.4.5.3 Levels of Interoperability Checking

The testing methodology introduced in this thesis targets the simultaneous testing of conformance and interoperability compliance of a product with the standards, which assures the readiness for plug and play interoperability with other products. The main goal is interoperability but during the validation of messages, conformance checks are performed as well. Consequently, several levels of interoperability levels can be distinguished in order to classify the failure types:

- *flow level*: at this level, the compliance with the required sequence of messages is validated. This flow level interoperability checking translates into state checking, i.e. an actor's behaviour changes its state according to the specification. This type of checking has the greatest impact in establishing whether or not HIS actors interoperate. This is also the main criterion being used during the plug-in events. Since this is the most important level, the selection of an adequate test language which supports flow and state checking, plays a crucial role.

- *semantic correlations level*: at this level the correlation of pieces of information from different messages within the sequence of messages has to be verified. An example of such a situation is the acknowledging of an interaction in HL7 based systems: the message ID corresponding to the initial interaction has to be contained in the acknowledgement content. However, semantic correlation in HISs implies much more complex situations where message content correlations have to be performed across many interactions. Similar to the validation of interoperability *flow level*, checking of the semantic correlations is also extremely important and decisive in getting an interoperability statement. In the testing process, the performance and efficiency of a validator for semantic correlation level requires test language artefacts which ease the access to the content of a message and also support saving system states for the later use in the semantic correlation of messages within the interaction scenario.

- *message type checking*: this level corresponds rather to conformance validation of message types, i.e., the purpose is to check whether a specific message from a sequence flow has the required type and whether the message structure corresponds with the type description from the integration profile. This way of validation is also critical to a smooth communication between two systems. However, the impact to the overall interoperability verdict when having this kind of failure is less important during the plug-in events. In order to achieve

this, a typed testing language and a type matching mechanism are the obvious requirements of a testing language. Additionally, the type checker has to rely on a parsing component, capable of extracting correctly the information and fill it in the corresponding message tree structure.

- *message content level*: similar to the previous level, validators at this level belong rather to the conformance compliance checkers. Different to the message type checking, here the content of messages is inspected: whether the concrete values match with the expected ones or align to different patterns, length restrictions, optionality and repetition attributes. With regard to HISs, typical message content is provided from the sets of code lists, e.g., medications codes, hence verifying whether messages are filled in with values representing those codes has patient safety implications. Even though this kind of validation is not highlighted during the plug-in events, it is extremely important for it to appear in interoperability testing of interaction scenarios. The test language requirements are similar to those mentioned for the previous kind of interoperability checking level.

- *fields conditionality level*: a common characteristic of messages in HISs, especially in the HL7 world, is the conditionally constraints across the fields within the same message. An example is the requirement regarding the presence of a field content only when another filed has a specific value or it is present. Even though this type of checking may seem to align to message content checking, it is more difficult by demanding additional content correlation within the same message. This level of interoperability checking is almost ignored during large plug-in events, but verifying its compliance may definitely help avoiding dangerous situations where, for instance, the absence of content for a specific field is interpreted as a default value. Also in this case, the test language should provide easy access to fields content and easy description of conditionality rules.

Whenever an interoperability failure occurs, it is extremely important that the logging and reporting component makes visible the level of interoperability which was faulted. This way the tester can establish the type of failure and can conclude upon the gravity of the interoperability issue. This statement is very important from the perspective that not all levels of interoperability have the same impact on the overall smooth interoperability. For instance, a flow level issue has a greater impact than issues occurring at the message content level (e.g., a name is longer than the length restriction for that field).

3.4.6 Application of the Test Methodology

The introduced methodology distinguishes itself from other approaches due to the fact that it proposes a complete process and an architecture for *testing the interoperability scenarios of an SUT by simulating the interacting parties*. Due to cost and effort factors, which were carefully analysed in this chapter, interoperability testing at plug-in events may not be as efficient as the method proposed in this thesis.

An extremely important aspect, is that the test scenarios are the same as the scenarios which are used to test the interoperability of the system against other systems. From this point of view, this approach can be considered an approach for *pre-interoperability testing*. The usage of this approach should not replace a traditional interoperability plug-in event which should be definitely used for certification purposes. The approach presented in this thesis is meant to help developers

investigate interoperability issues long before attending an interoperability event. This way, not only that many errors are detected in advance, but also more trust with respect to quality is gain.

The methodology consists of three main parts: 1) the test design process, 2) the message event patterns which are used to derive test simulators and 3) the conceptual architecture for a test framework. Parts 2) and 3) are discussed in more detail the next chapters. Along other aspects analysed along this thesis, a special attention is given to automation. Along the thesis, algorithms and techniques for automating the three parts are presented.

Chapter 4

Design and Realization of the Test Framework

I was born not knowing and have had only a little time to change that here and there.
– Richard Feynman

The methodology introduced in the previous chapter helps a test engineer in designing in a systematic way meaningful interoperability tests for HISs. The next step is to realise those tests into a concrete test framework which has to cope with the challenges and requirements for interoperability testing of HISs.

In this chapter we present how the introduced testing methodology is instantiated in a test framework based on the TTCN-3 test technology. This framework is adapted and applied to testing interoperability scenarios derived from IHE Patient Care Devices (PCD) Device Enterprise Communication (DEC) and IT Infrastructure (ITI) PIX integration profiles which are the case studies presented in Chapter 5.

4.1 Motivation for the Selected Technology

Various languages and tools can be used to design and execute interoperability tests for HISs. To put in practice the concepts presented in the previous chapter, the TTCN-3 technology has been selected. The argument in favour of this language lay on the fact that TTCN-3, as a standardised test language, is increasingly accepted in the industry as "the test specification language". TTCN-3 is the cornerstone of many complex test specifications [WDT+05].

TTCN-3 gained a great deal of attention over the last decade since ETSI made considerable efforts in sustaining large interoperability events mainly addressed to industry. This technology has been used for relevant test suites which represent the groundwork for interoperability assessment in a wide range of technologies, including GSM, UMTS, VoIP, ISDN, WiFi, and cordless telephony (DECT). ETSI not only created the TTCN-3 language but also strongly considered the use of the same test language, in testing interoperability in various domains, as the main ingredient to ensuring quality of interoperability events.

TTCN-3 is able to cope with complex testing needs thanks to it main characteristics it has been designed for. Among these features one can highlight: test specific language constructs, dynami-

cally construction of test configurations, separation of test specific aspects from execution platform details, behaviour and data validation mechanisms [WDT+05].

TTCN-3 not only copes with testing of different protocols from various OSI stack layers and different types of testing such as conformance, interoperability, performance, etc., but it becomes also an emerging testing technology applied to a cross-domain dimension range of systems, e.g., telecommunication, eGovernment, automotive, power transmission.

Furthermore, the possibility to increase as much as possible the automation process while testing interoperability scenarios is a tremendous benefit of using TTCN-3. Only with little effort it is possible to chain the TTCN-3 technology with other existent technologies, e.g., data serialisation formats and protocol parsers, test management and reporting.

Moreover, the TTCN-3 was not only addressed by the industry or pushed forward by the standardisation bodies, but it also was the research topic of many works done in academia: performance testing [Din09], model-driven testing [Dai06], distributed real-time testing [Neu04], quality assurance of test specifications [Zei10] are only a few examples.

"TTCN-3 is a living, widely established and continuously maintained testing technology" [Sch10]. Furthermore, test experts are looking into the adoption of TTCN-3 in additional domains, a fact which motivates the research on the applicability of this technology to interoperability testing in eHealth domain.

With the approach presented in this thesis, the area of applicability of the TTCN-3 technology was broadened to testing in the eHealth domain, concretely for interoperability testing of HISs.

4.2 Guidelines for Test Specifications

As a consequence of the fact that HISs are very *data-intensive* domains and the structures used for describing the patient data, especially those based on HL7 messaging standard, are very complex, the test specifications themselves become very large and complex. These specifications contain thousands of definitions. Hence, it is extremely important to be consistent when defining test specifications and to follow a set of guidelines that help in maintaining a clear, easy readable and maintainable test suite structure. Before proceeding with the presentation of concepts on how to implement the interoperability test architecture using the TTCN-3 test technology, the ideas and the works on a general approach for defining notation guidelines for TTCN-3 test specifications are introduced. This topic has been investigated in the context of this thesis and published in [DVS08] as a method to improve the maintainability of testsuites.

4.2.1 Related Work

Guidelines are designed to help the developer in writing better code. They are available for almost any programming language and have an impact on different levels such as: coding level, for instance for C++ in [Str00], design level, for instance for Java in [MIJ99], formatting level, comments level, etc.

On the testing side, the existing work focuses more on the guidelines regarding the effectiveness of tests, e.g., unit tests, integration tests, system tests, etc. The work in [Ser07] highlighted a set of 27 guideline rules for writing jUnit[Men10] tests.

With respect to TTCN-3, re-usability was explored in [MA05]. This work concentrates in great

detail on guidelines for writing reusable TTCN-3 code. Maintainability aspects, and in particular the use of refactoring techniques, have been concerned in [ZNG+06] where a catalogue of 20 refactoring rules derived from Java [Fow99] has been proposed and implemented. Refactoring is seen as a technique to systematically restructure code to improve its quality and maintainability while preserving the semantics. [NZ07]

4.2.2 Concept of Guidelines Levels

A comprehensive guideline should take into account various aspects at different levels. Three levels for guidelines for test specifications are distinguished: the *architectural level*, the *language level*, and the *physical level* (Figure 4.1). Guideline rules are defined at each level.

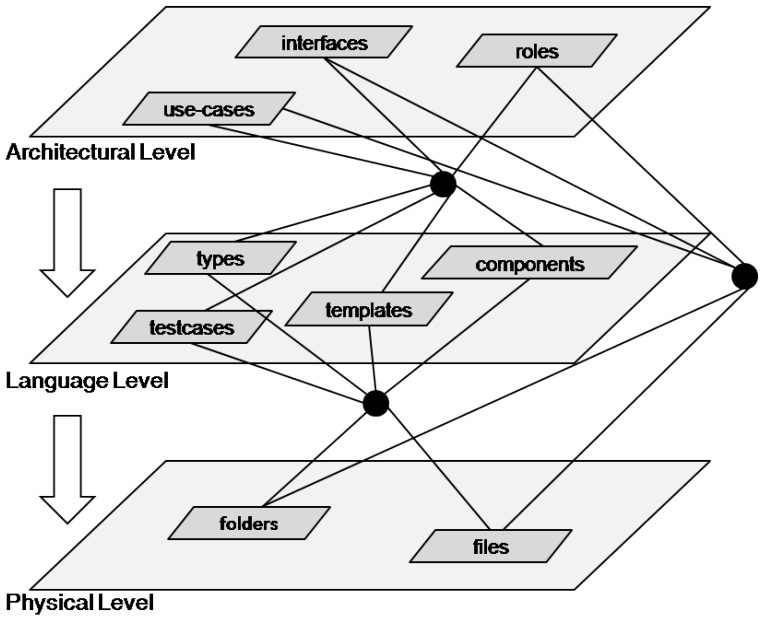


Figure 4.1: Guideline Levels

The *architectural level* refers to information related to SUT (interfaces, use-cases, roles), the *language level* refers to the definition of test constructs in the TTCN-3 language (types, components, testcases, etc.), while the *physical level* deals with file system information such as files and folders. In this classification, the information from one level may propagate only to the levels below (top-done) and never to the above ones. These three specified levels can also be seen as an instrument for measuring and improving the quality of a test system as part of the test quality model (TQM) introduced in [ZVS+07].

4.2.2.1 Architectural Level

The *architectural level* refers to information related to the SUT, e.g., interfaces, use cases, or roles. The architectural level includes guideline rules derived directly from the SUT architecture. They can be classified into:

- *interfaces*: the interaction between the TS and SUT is realised over at least one interface. To

increase the readability, a common guideline rule is to group together the definitions related to one interface.

- *roles*: the test behaviour can be designed for different roles, e.g., client, server, proxy. The test definitions defined for one role should be grouped together.
- *use cases*: a test behaviour corresponds to a type of interaction, i.e. use case, with the SUT. Multiple use cases can be treated within the same test specification. To avoid mixing the test behaviours from different use cases, a common practice is to group together the definitions related to a use case.
- *version*: a test specification may refer to multiple versions of the tested SUT' specification. A common practice is to avoid that test definitions for different SUT versions are mixed.

The information from the architectural level is used to structure the test specification and, consequently, imposes guideline rules to the two levels below.

At the *language level*, the architectural information is used to group related test definitions into TTCN-3 **modules** and **groups**. Additionally, naming conventions can also be used in order to embed architectural information into the TTCN-3 identifiers.

At the *physical level*, the architectural information is used to store the test definitions into files and folders. Also in this case, naming convention rules can be used to name the files and folders. When more than one architectural guideline rule applies, they can be combined in an arbitrary order.

4.2.2.2 Language Level

The *language level* refers to the definition of test constructs of TTCN-3, such as types, components, test cases, and similar. The language level contains guideline rules for writing the TTCN-3 code. They can be classified into:

- *formatting rules*: related to indentation, braces, white spaces, blank lines, new lines, control statements, line wrapping and comments
- *naming conventions*: related to the names of the identifiers of the TTCN-3 constructs (types, templates, testcase, components, etc.)
- *structural rules*: related to grouping the test definitions into groups and modules.

The naming conventions concern prefixing and postfixing rules which apply to all TTCN-3 elements which require an identifier: **types**, **templates**, **functions**, **altsteps**, **testcases**, **groups**, **modules**, **variables**, etc. For easier localisation, the TTCN-3 identifiers can be prefixed with a string indicating a group of definitions of the same category. For example, the message types can be prefixed by strings such as `Type`, `type`, `type_`, `T_`, etc. Multiple prefixes can occur, too. For example, type definitions can be grouped into types of messages to be sent to SUT, e.g., `Send_Msg`, and types of messages to be received, e.g., `Received_Msg`. If multiple prefixes are used, they can simply be concatenated or separated by the "_" character.

The structural rules concern the grouping of the definitions into **groups** and **modules**. This can be realised in many ways:

- *grouping by categories*: the definitions of the same category can be grouped together (e.g., **types** in a group of **types**, **templates** in a group of **templates**).
- *grouping by libraries*: the reusable definitions which are at the same time also general enough to apply to different test suites should be grouped into libraries.

4.2.2.3 Physical Level

The *physical level* deals with file system information such as files and folders. Guidelines for this level may, for example, provide information on how to structure the test suite in folders within the file system.

The physical level offers further structuring possibilities of TTCN-3 definitions:

- *files*: store particular groups of definitions in separate files
- *folders*: files can be further grouped into folders and subfolders.

Also at the *physical level* the naming conventions should appear. They are usually propagated from the upper levels and impose prefixes for the names (or even impose the name itself) of files or folders. For example, a file located as `/types/interface1/usecase1/sending.ttcn3` combines information from the architectural level, i.e., `interface1` and `usecase1` with information from the language level, i.e., **types** and `sending`. This file name means that it contains all types of messages to be sent to SUT defined for `usecase1` and for `interface1`.

4.3 Instantiation of the Test Architecture in TTCN-3 Technology

According to the HIS testing methodology process introduced in the previous chapter (section 3.3), the first step calls for the domain-knowledge within the HISs cloud and has as a result the identification of the *integration profile* in terms of its components: *actor types*, *interactions*, *sequence of interactions*.

This section explains how these elements are defined and/or derived in order to obtain TTCN-3 test specifications. The resulted TTCN-3 scripts will cover the first four layers of the proposed functional architecture of a test system for interoperability (Figure 3.25): *types, test data, functions, test behaviour layers*. To cover the whole architecture, the presented test framework based on TTCN-3 technology should contain two additional modules that will handle the messaging transport protocol adapters and message enveloping functions described in the fifth layer: *means of communication layer*.

4.3.1 TTCN-3 Types Layer

This layer deals with the representation of the test message types in the TTCN-3 testing language. The designed and realised TTCN-3 framework can cope with different messaging standards used in HISs. However, it currently supports the HL7 messaging standards from the two main families versions: v2.x[HL787c] and there is an ongoing work on supporting v3[HL705].

Any TTCN-3 Abstract Test Specification (ATS) requires a type system in accordance with the SUT interface input and output data types. Consequently, the TTCN-3 test system must provide a type

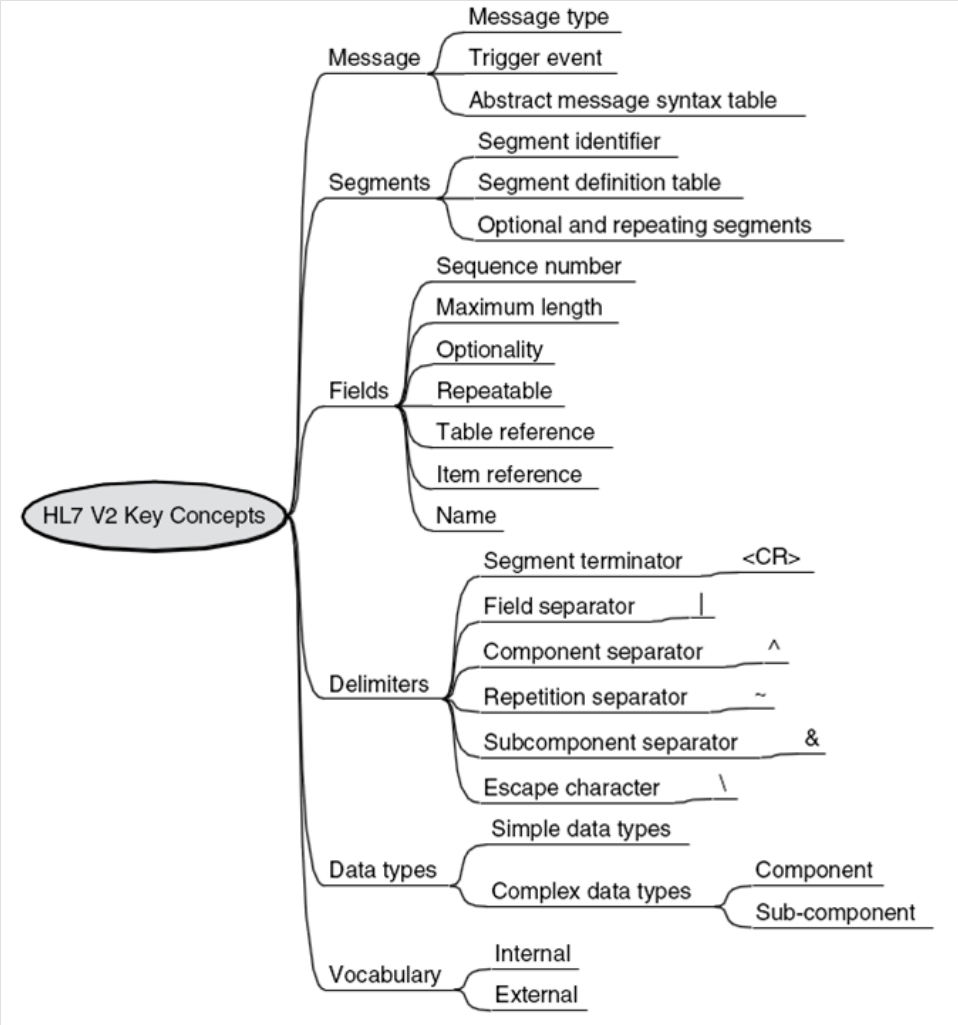


Figure 4.2: HL7 Version 2.x Key Concepts

system that can understand messages supported by HIS interfaces, e.g., HL7 messages. Therefore, a mapping from HL7 message structures to TTCN-3 types is necessary. This conceptual mapping which will be next presented is based on an initial work published in [VSD08] and conducted in 2008; however the mapping has been extensively improved and refined afterwards.

4.3.1.1 Representation of HL7 v2.x Message Types in TTCN-3

According to [Ben09] (page 106), the version 2.x of the HL7 messaging standard is the most widely used healthcare interoperability standard in the world. Its coverage represents over 90% of all hospitals in the USA and is widely supported by healthcare IT suppliers worldwide.

Even though during its long development period, the scope of HL7 version 2.x has increased significantly from version to version, the basic principles have hardly changed. Hence, despite variations from one version to another within the HL7 version 2.x family, the same mapping mechanism to the TTCN-3 types applies along all 2.x versions.

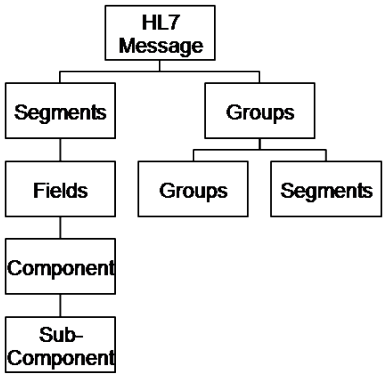


Figure 4.3: HL7 v2.x Message Structure

These facts contributed decisively as motivation to the selection of this version of the HL7 messaging standard as a first candidate for the TTCN-3 based test framework presented in this chapter.

4.3.1.1.1 Mapping of the Message Structure

The overall structure of HL7 v2.x and how the different parts are recognised, is referred as message syntax. Figure 4.2 published in the book by Benson [Ben09] summarises the key concepts related to the HL7 v2.x message syntax.

An HL7 v2.x message has an hierarchical structure of message elements. These elements can be container elements or can be primitive elements. As described in Figure 4.3, an HL7 v2.x *message* is the top-level unit of data transferred between different systems and it has a tree-like compositional structure. Each *message* consists of a set of *segments* in a specified sequence, each of which contains *fields* also in a specified sequence; these fields have specified data types. Data types are the building blocks of the *fields* and may be simple, with a single value, or complex, with multiple *components*. These components themselves have data types, which can be simple or complex, leading to *sub-components*.

The *message type* of a message, sometimes named *functional group*, is the general category into which a message fits. For example, patient administration messages are of type Admission Discharge and Transfer (ADT), laboratory orders are of type ORM, laboratory results are of type ORR, etc.

There are many message events, called *trigger events* that model real-world events which causes the messages to flow among systems. For each event associated to a *message type*, there is an unique associated *message structure*, also called *abstract message syntax* or *message definition*. The name of this *message structure* is derived from the *message type* (*functional group*), to which that message belongs, and from the identifier of the *trigger event* which triggers the message flow. For example, the *trigger event* A01 associated with the messages of ADT *message type* leads to the ADT_A01 *message structure* which is used as notification for the admission of the visit of a patient. Similarly, the *trigger event* A02 issuing the transfer of a patient conducts to the ADT_A02 *abstract message syntax*, etc.

The *abstract message syntax* also establishes which *segments* are `optional` and which can be `repeated`. Figure 4.4 describes which segments are `optional` and/or `repeatable` in an

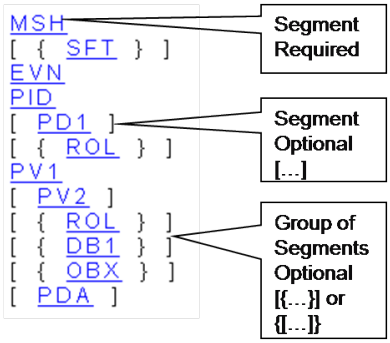


Figure 4.4: HL7 v2.5.1 ADT_A02 Message Structure

Msg. Type	Description
ACK	General acknowledgment message
ADR	ADT response
ADT	ADT message
BAR	Add/change billing account
BPS	Blood product dispense status message

...

Figure 4.5: HL7 v2.5.1 Message Types Examples

ADT_A02 *message structure*. A segment listed on its own is mandatory and may not repeat. `Optional` *segments* are marked by square brackets [. . .]. *Segments* that are allowed to repeat (for an indefinite number of times or up to a specified number) are indicated using curly braces { . . . } forming the so-called *groups*. If a segment is both `optional` and `repeatable`, it has both brackets and braces [{ . . . }]. The order is not important: [{...}] and {[...]} are equivalent.

The mapping rules that translate the above introduced HL7 message elements to TTCN-3 data types follow closely the HL7 v2.x message syntax. In the following it is presented and summarised how the composing parts of an HL7 v2.x message relate to TTCN-3 constructs.

a) Mapping of HL7 v2.x Message Types / Functional Groups to TTCN-3

HL7 Message Type / Functional Group

- defines a set of HL7 *message structures* according to a common application function, domain, etc.
- contains multiple HL7 *message structures*, each of them being associated to a *trigger event*.
- examples: patient administration messages are of type ADT, laboratory orders are of type ORM, laboratory results are of type ORR, observations of type OBX, etc. Other examples are presented in Figure 4.5.

Corresponding TTCN-3 Type

- there is no direct corresponding TTCN-3 type but a logical grouping using the **group** keyword or physical grouping by means of TTCN-3 **modules** can be used; the TTCN-3

group/module will contain definitions for all the TTCN-3 message types corresponding to all the HL7 *message structures* listed in the HL7 *message type*.

- design guideline: the identifiers for all resulted TTCN-3 message types corresponding to all HL7 *message structures* associated to an HL7 *message type* should start with the identifier of the HL7 *message type* (*functional group*).
- example: HL7 ADT *functional group* is mapped into a set of TTCN-3 data types whose identifiers follow the naming convention `ADT_*_Message_Type`, where * is replaced with the appropriated *trigger event*, e.g., `ADT_A01_Message_Type`.

b) Mapping of HL7 v2.x Message Structures / Abstract Message Syntax to TTCN-3

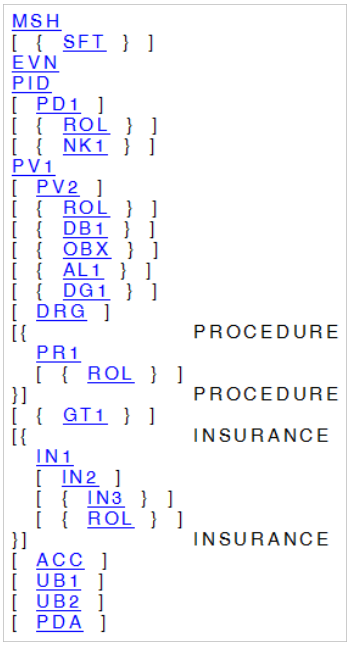


Figure 4.6: HL7 v2.5.1 ADT_A01 Message Structure

HL7 v2.x Message Structure

- describes a sequence of *segments*, each segment being characterised by attributes such as `mandatory`, `optional`, `repetitive`; these *segments* may belong to substructures which are other sequences of *segments* that belong to the root *message structure*, e.g., *INSURANCE* substructure in Figure 4.6.
- example: within ADT *message type* there are many different *message structures*, .i.e., the sequence of *segments* are differently defined according to the *trigger event* determining a *message structure*. The *trigger event* issues the HL7 message to flow, and, in combination with the *message type*, e.g., ADT, it derives the *message structure*. Figure 4.6 shows the ADT_A01 *message structure* for HL7 v2.5.1.

Corresponding TTCN-3 Type

- the TTCN-3 correspondent is a **record type** definition; the record type was preferred to the set type because the order of children fields matters.
- the fields of the TTCN-3 record are of different types: 1) `*_Segment_Type` representing the TTCN-3 type corresponding to an HL7 *segment*, e.g., `MSH_Segment_Type` 2) `*_Group_Type` representing the TTCN-3 **record of** type corresponding to a repetition of a specific HL7 *segment* within an HL7 *message structure*, e.g., `SFT_Group_Type` 3) «`substructureName`» TTCN-3 record type which is defined similarly as the root message structure record type and which corresponds to a substructure within an HL7 *message structure*, e.g., `ADT_A01_Procedure_1` 4) «`substructureName`»`_Group_Type` representing the TTCN-3 **record of** based type corresponding to a repetition of a substructure within an HL7 *message structure*, e.g., `ADT_A01_Procedure_1_Group_Type`.
- example: Listing 4.1 shows how the ADT_A01 HL7 *message structure* maps to a TTCN-3 **record type** definition.
- `optional` *segments*, *groups of segments*, *sequences of segments* (substructures) of an HL7 *message structure* map into **optional fields** in the TTCN-3 record type corresponding to the HL7 *message structure*.
- `repetitive` *segments* (i.e., *group*) and `repetitive` *sequences of segments* (substructures) defined within an HL7 *message structure* map into TTCN-3 fields of type **record of** whose identifiers follow the pattern «`segmentName | substructureName`»`_Group_Type`. For example, in Listing 4.1, the second field of the `ADT_A01_Message_Type` record is of type `SFT_Group_Type` which is defined in Listing 4.2 as a **record of** type.

Listing 4.1: Mapping Example - TTCN-3 ADT_A01_Message_Type

```
type record ADT_A01_Message_Type {                                  1
  MSH_Segment_Type MSH_Segment,                                     2
  SFT_Group_Type SFT_Group optional,                                3
  EVN_Segment_Type EVN_Segment,                                     4
  PID_Segment_Type PID_Segment,                                     5
  PD1_Segment_Type PD1_Segment optional,                            6
  ROL_Group_Type  ROL_Group_1 optional,                             7
  NK1_Group_Type  NK1_Group optional,                               8
  PV1_Segment_Type PV1_Segment,                                     9
  PV2_Segment_Type PV2_Segment optional,                            10
  ROL_Group_Type  ROL_Group_2 optional,                             11
  DB1_Group_Type  DB1_Group optional,                               12
  OBX_Group_Type  OBX_Group optional,                               13
  AL1_Group_Type  AL1_Group optional,                               14
  DG1_Group_Type  DG1_Group optional,                               15
  DRG_Segment_Type DRG_Segment optional,                            16
  ADT_A01_Procedure_1_Group_Type                                    17
      ADT_A01_Procedure_Group optional,                             18
  GT1_Group_Type  GT1_Group optional,                               19
  ADT_A01_Insurance_1_Group_Type                                    20
      ADT_A01_Insurance_Group optional,                             21
  ACC_Segment_Type ACC_Segment optional,                            22
  UB1_Segment_Type UB1_Segment optional,                            23
  UB2_Segment_Type UB2_Segment optional,                            24
```

```
    PDA_Segment_Type PDA_Segment optional                                    25
};                                                                           26
```

Listing 4.2: Mapping Example - TTCN-3 SFT_Group_Type

```
type record length (1..infinity) of SFT_Segment_Type SFT_Group_Type;         1
```

c) Mapping of HL7 v2.x Segments to TTCN-3

HL7 v2.x Segment

- provides a logical grouping of other data. Each segment is given a name, known as the *segment ID*, and it is identified by a unique three-letter code, e.g., MSH for message header *segment*.
- consists of a sequence of *fiels*.
- each *field* can be of a `simple data type` (e.g., string data (ST), sequence ID (SI)) or may be of `complex data type`, and, in this case, the *field* consists of *components* (e.g., *fields* encapsulating the extended person name (XPN) or the version identifier (VID)).
- each *field* is characterised, similar to the *segments* within a *message structure*, by attributes such as `mandatory` or `optional`, `repetitive` or not. Additionally, a *field* may have `length` restrictions or may indicate a `table number` meaning the `referenced table` which contains a set of so-called *coded values*, i.e., restricted set of values that a field can have. Any combination of all attributes for a *field* of a *segment* is possible.
- example: Figure 4.7 shows the structure of the message header (MSH) *segment* for HL7 v2.5.1. For example, the first *field* of the *segment* is of a `simple data type` (string data - ST); it is `mandatory` within the MSH *segment*, it is not `repetitive` and has a `length` restriction to 1. The last *field* of the MSH *segment* is of a `complex data type` (entity identifier - EI); it is `optional`, it is `repetitive` without number of repetition restrictions (marked with an "Y" in the "Rep" column).

Corresponding TTCN-3 Type

- the TTCN-3 correspondent is a **record type** definition named «`SegmentName`»`_Segment_Type`, a structured TTCN-3 type for which the order of the children is considered.
- the children of the TTCN-3 record, i.e., TTCN-3 **record fields**, are of different types, depending whether the corresponding *field* of an HL7 *segment* is of a `simple data type` or `complex data type`, is `repetitive` or not, has a `reference to a table number` or not, has a `length` restriction or not, or other similar combinations.
- example: Listing 4.3 shows how the HL7 MSH *segment* shown in Figure 4.7 maps to a TTCN-3 **record type** definition.
- `optional` *fields* of `simple data type` or `complex data type` within an HL7 *segment* map into **optional record fields** in the root TTCN-3 record based type definition that corresponds to the HL7 *segment*.

Seq	Description	Length	Table	r/o/c	Rep#	Item	Data Structure	Section
1	Field Separator	1		R		00001	ST	2.15.9.1
2	Encoding Characters	4		R		00002	ST	2.15.9.2
3	Sending Application	227	0361	O		00003	HD	2.15.9.3
4	Sending Facility	227	0362	O		00004	HD	2.15.9.4
5	Receiving Application	227	0361	O		00005	HD	2.15.9.5
6	Receiving Facility	227	0362	O		00006	HD	2.15.9.6
7	Date/Time Of Message	26		R		00007	TS	2.15.9.7
8	Security	40		O		00008	ST	2.15.9.8
9	Message Type	15		R		00009	MSG	2.15.9.9
10	Message Control ID	20		R		00010	ST	2.15.9.10
11	Processing ID	3		R		00011	PT	2.15.9.11
12	Version ID	60		R		00012	VID	2.15.9.12
13	Sequence Number	15		O		00013	NM	2.15.9.13
14	Continuation Pointer	180		O		00014	ST	2.15.9.14
15	Accept Acknowledgment Type	2	0155	O		00015	ID	2.15.9.15
16	Application Acknowledgment Type	2	0155	O		00016	ID	2.15.9.16
17	Country Code	3	0399	O		00017	ID	2.15.9.17
18	Character Set	16	0211	O	Y	00692	ID	2.15.9.18
19	Principal Language Of Message	250		O		00693	CE	2.15.9.19
20	Alternate Character Set Handling Scheme	20	0356	O		01317	ID	2.15.9.20
21	Message Profile Identifier	427		O	Y	01598	EI	2.15.9.21

Figure 4.7: HL7 v2.5.1 Message Header (MSH) Segment

- *fields* of `simple data type` and with `length restriction` within an HL7 *segment* map into TTCN-3 **record fields** that have as type a **charstring** based type named «`simpleDataTypeNameRestriction`». For example, in Listing 4.3, the first child of the record is of type ST1, a **charstring** based TTCN-3 type with `length` restriction (defined in Listing 4.4).

- *fields* of *simple data types* of an HL7 *segment* that have a `reference to a table number` translates into TTCN-3 **record fields** that are of **charstring** based types named «`TableName`»`_Table`; this type imposes a restriction with respect to the set of allowed values. For example, the 15^{th} *field* of the HL7 MSH *segment* (Figure 4.7) maps into a TTCN-3 **record field** of type `Acc_App_Acknowledgement_Table` which is a TTCN-3 **charstring** based type with values restrictions defined as in Listing 4.5.

- `repetitive` *fields* of *simple data types* that have a `reference to a table number` defined within an HL7 *segment* map into TTCN-3 **record fields** that are of **record of** based type named using the pattern «`TableName`»`_Group_Type`. For example, in Listing 4.3, the **record field** `Character_Set` of the TTCN-3 «`SegmentName`»`_Segment_Type` record (line 21) corresponding to the 18^{th} *field* of the HL7 MSH *segment* (Figure 4.7) is of type `Character_Set_Group_Type`, a TTCN-3 **record of** based type as defined in Listing 4.6. The elements of this **record of** type are of type `Character_Sets_Table`, a **charstring** based type having restrictions for the allowed values, as defined in Listing 4.7.

- `repetitive` *fields* of HL7 `simple data types` or of `complex data types` that have no `reference to a table number` defined within an HL7 *segment* map into TTCN-3 **record fields** that are of **record of** based types named using the pattern «`simpleDataTypeName | complexDataTypeName`»`_Group_Type`. For example, in Listing 4.3, the last child of the `MSH_Segment_Type` record based type is of type `EI_Group_Type` defined in Listing 4.8.

Listing 4.3: Mapping Example - TTCN-3 MSH_Segment_Type

```
type record MSH_Segment_Type {
        ST1 Field_Separator
        ST4 Encoding_Characters,
        HD_0361 Sending_Application optional,
        HD_0362 Sending_Facility optional,
        HD_0361 Receiving_Application optional,
        HD_0362 Receiving_Facility optional,
        TS DateTimeOfMessage,
        ST40 Security optional,
        MSG Message_Type,
        ST20 Message_Control_ID,
        PT  Processing_ID,
        VID Version_ID,
        NM15 Sequence_Number optional,
        ST180 Continuation_Pointer optional,
        Acc_App_Acknowledgement_Table
                Accept_Acknowledgment_Type optional,
        Acc_App_Acknowledgement_Table
                Application_Acknowledgment_Type optional,
        Country_Code_Table Country_Code optional,
        Character_Set_Group_Type Character_Set optional,
        CE Principal_Language_Of_Message optional,
        Character_Set_Handling_Scheme_Table
                Alternate_Character_Set_Handling_Scheme optional,
        EI_Group_Type Message_Profile_Identifier optional
};
```

Listing 4.4: Mapping Example - TTCN-3 ST1

```
type charstring ST1 length (0..1);
```

Listing 4.5: Mapping Example - TTCN-3 Acc_App_Acknowledgement

```
type ID Acc_App_Acknowledgement_Table
(
        "AL", //Always
        "NE", //Never
        "ER", //Error/reject conditions only
        "SU"  //Successful completion only
);
```

Listing 4.6: Mapping Example - TTCN-3 Character_Set_Group_Type

```
type record length (1..infinity) of Character_Sets_Table
Character_Sets_Group_Type;
```

Listing 4.7: Mapping Example - TTCN-3 Character_Sets_Table

```
type ID Character_Sets_Table(
        "ASCII" , //The printable 7-bit ASCII character set.
        "8859/1", //The printable characters from the ISO 8859/1 Character set
        "8859/2", //The printable characters from the ISO 8859/2 Character set
                ...
);
```

Listing 4.8: Mapping Example - TTCN-3 EI_Group_Type

```
type record length (1..infinity) of EI EI_Group_Type;
```

d) Mapping of HL7 v2.x Complex Data Types to TTCN-3

HL7 v2.x Complex Data Types

- are used to specify the HL7 types of the *fields* within an HL7 *segment* that can hold more than one data element. Each data element child may be of a `simple data type` or of a another `complex data type`.
- example: Figure 4.8 shows the structure of the `complex data type` Entity Identifier (EI) for HL7 v2.5.1 which consists of a sequence of four data elements. Any HL7 *segment field* of EI type consists of four components, all of them being in this case of `simple data types`.
- similar to the *fields* within a *segment*, each data element child of a `complex data type` is characterised by attributes such as `mandatory` or `optional` or `conditional`, it can have `length` restrictions or may indicate a `table number` which indicates the `referenced table` with *coded values*, i.e., restricted set of values that a data can have. Any combination of all these attributes is possible.

Seq.	Comp. No.	Index	Sec.	Table (this use)	Table (Comp.)	Opt.	Length	Data Type	Data Structure
1	114	Entity Identifier				O	199	ST	ST
2	115	Namespace ID		0363	0363	O	20	IS	IS
3	116	Universal ID				C	199	ST	ST
4	117	Universal ID Type		0301	0301	C	6	ID	ID

Figure 4.8: HL7 v2.5.1 Entity Identifier (EI) Complex Data Type

Listing 4.9: Mapping Example - TTCN-3 EI

```
type record EI {                                                          1
        ST199 Entity_Identifier optional,                                 2
        Assigning_Authority_Table Namespace_ID optional,                  3
        ST199 Universal_ID optional, //conditional                        4
        Universal_ID_Type_Table Universal_ID_Type optional //conditional  5
};                                                                        6
```

Corresponding TTCN-3 Types

- the TTCN-3 corresponding element is a **record type** definition, i.e., a structured TTCN-3 type where the order of the children is considered. As a naming convention, the new TTCN-3 type shall be named following the pattern «`ComplexDataTypeName`».
- Listing 4.9 shows the TTCN-3 corresponding type of the HL7 EI `complex data type` presented in Figure 4.8.
- the TTCN-3 **record fields** can be either of TTCN-3 **record** type or of TTCN-3 **basic data types**. They can also be defined as `optional`. Unfortunately, there is no direct correspondent in TTCN-3 to map the `conditionality` (e.g., an element is present when another element is also present) concept from HL7, instead, the HL7 `conditional` children of a `complex data type` are defined in TTCN-3 as `optional` and the `conditionality` property is verified using additional checking functions.

- HL7 data elements of a `complex data type` that are of `simple data type` and refer to a `table number` map into TTCN-3 **record fields** of type «`TableName`»`_Table` which defines value restrictions for those **record fields**. For example, in Listing 4.9, the last **record field**, `Universal_ID_Type` is of type `Universal_ID_Type_Table` because in the HL7 definition, the corresponding data element of the EI `complex data type` refers to the table number 0301. The TTCN-3 `Universal_ID_Type_Table` type is a **charstring** based type defined similar to the TTCN-3 type from Listing 4.7.
- HL7 data elements of a `complex data type` that are of `simple data type` and also have length restrictions map to TTCN-3 **record fields** of TTCN-3 **basic data type** named using the pattern «`SimpleDataTypeName`»`LengthRestrictionNumber`. The definition and the identifier of this type should reflect the specified length restriction; for example, the first **record field** (`Entity_Identifier`) from Listing 4.9 is of a TTCN-3 **charstring** based type whose definition is similar to the one defined in Listing 4.4, the only difference being the length restriction.

e) Mapping of HL7 v2.x Simple Data Types to TTCN-3

HL7 v2.x Simple Data Types

- are used to designate types of the leaf data levels of an HL7 *message structure*.
- the instances of these `simple data types` are defined as character strings.
- support `length restrictions` or may restrict the set of values when they refer to a `table number`.
- examples: ST string data type with different length restrictions, ID data types for defining coded values for HL7-defined *tables*, IS data types for defining coded values for user-defined *tables*, etc.

Corresponding TTCN-3 Types

- the TTCN-3 corresponding element is a **charstring** based type definition. The adopted naming convention is «`SimpleDataTypeName`»`[`«`LengthRestrictionNumber`»`]` or `TableName_Table`. The first alternative is used for the case where `length restriction` is applied to a TTCN-3 **charstring** based type (e.g., Listing 4.4) while the second case is adopted when restrictions concerning coded values has to be regarded, e.g., Listing 4.7.
- when different patterns are specified in HL7 to restrict the values of a string (e.g., the timestamp must follow the format YYYY[MM[DD[HH[MM[SS[.S[S[S[S]]]]]]]]][+/-ZZZZ]), in TTCN-3 a corresponding pattern is specified when defining the corresponding TTCN-3 **charstring** based type.

4.3.1.1.2 Mapping of the Message Semantic

A special role in the HL7 v2.x messaging standard is played by the so called *tables*. An HL7 *table* attribute specifies the HL7 identifier for a set of *coded values*. Many HL7 elements within a *message structure* such as *fields*, *components* or *sub-components* are (or can be) `coded`.

HL7 defines table values in three ways with respect to the acceptable content:

- *user-defined tables*: a set of values that are locally or site defined.
- *HL7 tables*: a set of values defined and published by HL7.
- *external tables*: A set of coded values defined and published by other standard organisations.

This semantical aspect is also regarded in the mapping to TTCN-3 elements. By defining TTCN-3 **charstring** based types (i.e., «`TableName`»`_Table` types as introduced in the examples before, e.g., Listing 4.5) with values restrictions, not only that the ontology of the HL7 messages is translated and preserved in TTCN-3 too, but also the process of conformance checking will be considerable eased, improved and taken over by the TTCN-3 tool as this feature is supported by the language itself. This way, the checking of acceptable values for different HL7 *fields*, *components*, *sub-components* is done automatically when running the TTCN-3 scripts and no additional content validation is necessary. Additionally, in the case of HL7 *user defined tables*, where the `coded values` are healthcare facility dependent, it is very easy to adapt or extend the definition of the table accordingly with those `coded values`.

4.3.1.1.3 Mapping Guidelines

As long as the test specification contains thousands of definitions, it is extremely important to be consistent in writing TTCN-3 test definitions and in maintaining a clear test suite and module structure. To keep the test specifications consistent, readable, reusable and well structured, a set of guidelines for writing and structuring the TTCN-3 specifications for HL7 based HISs was defined and applied. In the following, some of these guidelines are elaborated.

a) Guidelines for Naming the TTCN-3 Type Definitions

The adopted naming conventions and construction rules for the TTCN-3 type identifiers concern prefixing and postfixing rules and apply to all TTCN-3 types identifiers. To sum up, the used naming conventions are presented in Table 4.1.

b) Guidelines for Grouping the TTCN-3 Type Definitions

Having suggestive names for the TTCN-3 type definitions is not enough to define a clear, easy to understand, use, maintain or extend test specification. In section 4.2, the three possible levels of information for deriving guidelines for a test specification were discussed (Figure 4.1). It is equally important to keep and easily identify the dependency between the *physical level* and *architectural level* of a test specification. This means that the grouping of the TTCN-3 file modules into folders,

Table 4.1: TTCN-3 Naming Conventions

HL7 v2.x Elements	TTCN-3 Naming Convention for Corresponding Type Identifiers
Message Structure	Postfix the TTCN-3 type identifier with _Message_Type: <<HL7MessageTypeID>>_[<<HL7TriggerEventID>>] _Message_Type, e.g., ADT_A01_Message_Type (Listing 4.1). The <<HL7TriggerEventID>> is optional, for example in case of acknowledgement (ACK) HL7 *message structures*.
Segment structure	Postfix the TTCN-3 type identifier with _Segment_Type: «HL7SegmentID»_Type, e.g., MSH_Segment_Type (Listing 4.3).
Segment field within a *message structure*	Postfix the TTCN-3 **record field** identifier with _Segment: «HL7SegmentID»_Segment_[Nr.], e.g., the first **record field** of the TTCN-3 **record type** ADT_A01_Message_Type from Listing 4.1 is named MSH_Segment. The Nr. is optional and is used when an HL7 *segment* appears more than once within the same message structure.
Group of *segments*	Postfix the TTCN-3 type identifier with _Group_Type: «HL7SegmentID»_Group_Type, e.g., SFT_Group_Type (Listing 4.2).
Table	Postfix the TTCN-3 type identifier with _Table: <<HL7TableName>>_Table, e.g., Character_Sets_Table (Listing 4.7).
Group of *tables*	Postfix the TTCN-3 type identifier with _Group_Type: <<HL7TableName>>_Group_Type, e.g., Character_Set_Group_Type (Listing 4.6).

i.e., *physical layer*, has to be done in such a manner that the SUT interfaces, i.e., *architectural level*, can be intuitively identified. The definitions of the same SUT interface are grouped together. Figure 4.9 shows the structure of the folders and of the files of TTCN-3 type definitions referring to two versions of HL7 from 2.x family: 2.3.1 and 2.5.

Further naming conventions are used for the file names. The files carry suggestive names based on the types of definitions they contain, e.g., types of segments, types of tables. All files are postfixed by _Types and by the version of HL7, e.g., _v231 and _v25. This naming pattern bridges the information from the *language level* and *physical level* as described in Figure 4.1. Additionally, the files containing types which serve as libraries for HL7 *message structures* belonging to different *functional groups* such as ADT, ADR, are grouped into folders named libraries.

4.3.1.2 Representation of Actor Types in TTCN-3

Besides the identification of the data types used in an *integration profile*, at the first step of the HIS testing methodology process introduced in the previous chapter (section 3.3), the *actor types* are identified as well. They are used along the further steps of the process to create different *actors* participating in an *interaction scenario*. The final aim is to have these *actor types* and their instances represented in the target testing language.

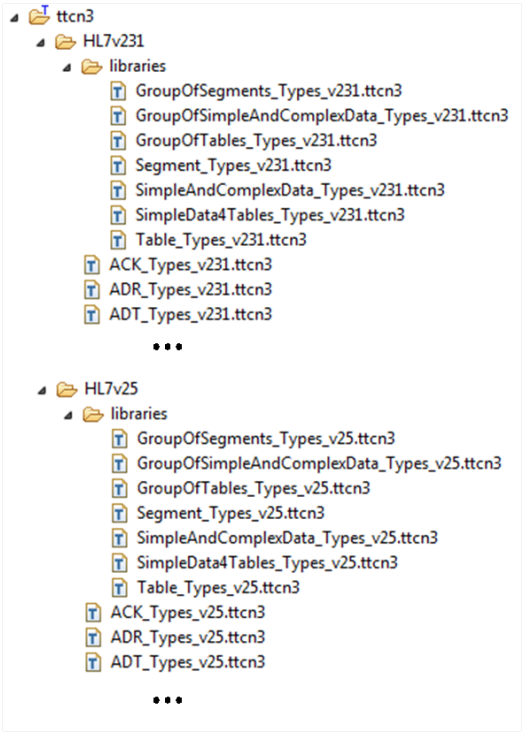


Figure 4.9: Structuring and Naming of TTCN-3 Type Definitions into Files and Folders

4.3.1.2.1 Annotations for Actor Types Derivation

The test configuration design impacts the overall test system by means of dynamic adaptation, flexibility, extendibility and even performance to changes to new test *interaction scenarios*. Figure 4.10 illustrates the generation of TTCN-3 elements (in the figure labelled as *TTCN-3 Design*) out of test scenario elements (in figure labelled as *Annotated Interaction Scenario Model*).

Figure 4.10 consists of two parts. The upper layer presents an example of an *interaction scenario* and how it can be modelled. The lower layer corresponds to the design for the corresponding TTCN-3 test code. In this particular example, a number of `n+m` *actors* are involved. Each actor, as, for example $\texttt{Actor}_j$, can communicate with other *actors* over different transport protocols by using one or more *message exchange patterns* which were presented in the previous chapter. Such a pattern is symbolised by a gray bidirectional arrow between $\texttt{Actor}_j$ and $\texttt{Actor}_{n+i}$.

For automation purposes this model has to be enhanced with elements of the tester mindset that can be technically realised by annotations. The annotations necessary to derive the TTCN-3 constructs to represent HIS *actor types* are next presented.

First of all, each *actor* must have an identifier, e.g., in Figure 4.10, $\texttt{Actor}_1 \ldots \texttt{Actor}_{n+m}$.

Furthermore, one has to distinguish between tested *actors* (with role SUT) and *actors* that the test system has to emulate (with role TS).

Additionally, each *actor* is an instance of an *actor type* from the HIS *integration profile*. In the *integration scenario* model, one can mark this aspect either as an additional annotation where the *actor type* is specified, e.g., the $Actor_j$ in has an *Actor Type* annotation, or by using modelling

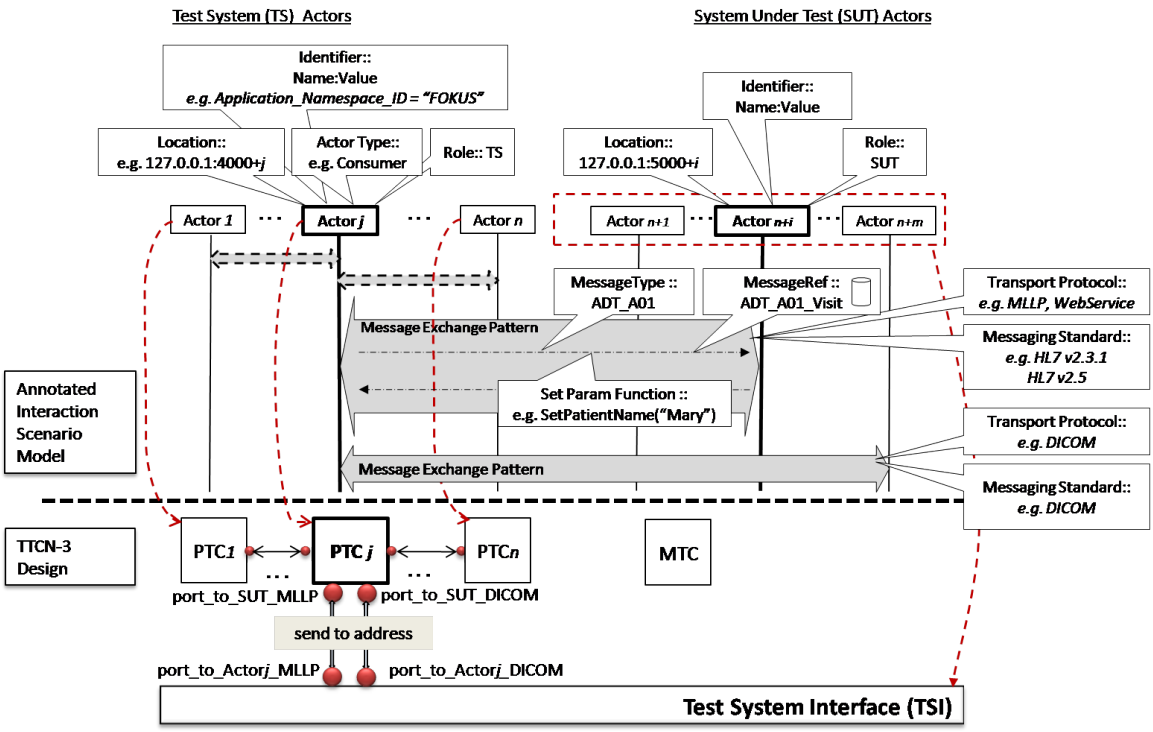


Figure 4.10: TTCN-3 Test Configuration Design and Derivation from an Interaction Scenario (I)

language proprietary elements e.g., in UML-SD the class attribute.

4.3.1.2.2 Derivation Algorithm and Test Design for Actor Types

TTCN-3 offers various mechanisms to express an HIS *actor*. The proposed functional architecture of a test system requires that the test system is able to simulate multiple *actors* at the same time. The TTCN-3 construct that enables this parallelism is the Parallel Test Component (PTC).

A TTCN-3 test case runs on top of an *MTC* that can create other Parallel Test Components (PTCs). The Test System Interface (TSI) is a special type of component and it represents an abstraction of the SUT at the TTCN-3 specification level.

The derivation algorithm used for representing the HIS *actor types* in TTCN-3 is presented below:

for each different transport protocol used between any TS *actor* and any SUT *actor*
within an *interaction scenario*
    generate a new TTCN-3 **port type**, e.g., Listing 4.10, lines 2, 6;

generate a **port type** used for internal communication between *actors* with TS role,
e.g., Listing 4.10, line 10;

for each *actor* with TS role within an *interaction scenario*
    identify its *actor type* as `actor_type_k`;
    if the `actor_type_k` not defined yet

generate a new TTCN-3 **component type** corresponding to
the `actor_type_k`, e.g., Listing 4.11, line 2;

for each `actor_type_k` within an *interaction scenario*
for each $Actor_j$ with TS role
generate on the TTCN-3 **component type**
corresponding to the `actor_type_k` a **port instance**
used for internal communication with $Actor_j$, e.g., Listing 4.11, lines 6, 7;

for each `actor_type_k` in all identified TS *actor types*
for each $Actor_j$ with role TS of type `actor_type_k`
for each `transport_protocol_i` used by $Actor_j$ to communicate with any SUT *actor*
if not yet added, add a new TTCN-3 **port instance** of TTCN-3 **port type** generated
for the `transport_protocol_i` to the TTCN-3 **component type**
corresponding to the actor_type_k, e.g., Listing 4.11, line 1;

define one empty-body **component type** as MTC e.g., Listing 4.11, line 11;

define one TTCN-3 **component type** as TSI, e.g., Listing 4.11, line 6;

for each $Actor_j$ with TS role
for each `transport_protocol_i` supported by the $Actor_j$ actor
add a new TTCN-3 **port instance** of TTCN-3 **port type** generated
for the `transport_protocol_i` to the TSI **component type**

Listing 4.10: TTCN-3 Test Configuration - Port Types

```
type port MLLPProtocol_Port_Type message {
        inout all
};

type port DICOMProtocol_Port_Type message {
        inout all
};

type port InternalCommunication_Port_Type message {
        inout all
};
```

Listing 4.11: TTCN-3 Test Configuration - Component Types

```
type component Consumer_Component_Type {
        port MLLPProtocol_Port_Type port_to_SUT_MLLP;
        port DICOMProtocol_Port_Type port_to_SUT_DICOM;

        port InternalCommunication_Port_Type port_to_Actor_1;
        port InternalCommunication_Port_Type port_to_Actor_2;
        // ...
        port InternalCommunication_Port_Type port_to_Actor_n;
}
```

```
                                                                    11
type component TSI_Component_Type {                                 12
        port MLLPProtocol_Port_Type port_to_Actor_j_MLLP;           13
        port DICOMProtocol_Port_Type port_to_Actor_j_DICOM;         14
}                                                                   15
                                                                    16
type component MTC_Component_Type { }                               17
```

4.3.1.2.3 Derivation Guidelines

Similar to the guidelines introduced along the TTCN-3 **data type** definitions, a set of guidelines is recommended as well when defining TTCN-3 **component types** and TTCN-3 **port types**. However, the guidelines in this case are far more simple since they refer to identifiers used for **component type** definitions, **port type definitions** and **port** instances definitions.

To sum up, the used naming conventions are:

- for any TTCN-3 **component type** that will be used to instantiate PTCs in the test configuration, the identifier shall use the pattern: `<<InteractionScenario_ActorTypeName>>_Component_Type`, e.g., Listing 4.11, line 2.
- for the TTCN-3 TSI **component type** the following identifier shall be used: `TSI_Component_Type`, e.g., Listing 4.11, line 12.
- for TTCN-3 MTC **component type** the following identifier shall be used: `MTC_Component_Type`, e.g., Listing 4.11, line 17.
- for any TTCN-3 **port type** used for communicating data between TS and SUT the following identifier shall be used: `«ProtocolName»Protocol_Port_Type`, e.g., Listing 4.10, lines 2, 6.
- for any TTCN-3 **port instance** used for communication with the SUT defined on a TTCN-3 **component type** whose instance is a PTC, the identifier shall respect the following pattern: `port_to_SUT_«protocolName»`, e.g., Listing 4.11, lines 3, 4.
- for any TTCN-3 **port instance** used for communication within TS defined on a TTCN-3 **component type** whose instance is a PTC, the identifier shall respect the following pattern: `port_to_TCj`, e.g., Listing 4.11, lines 6, 7.

4.3.2 TTCN-3 Test Data Layer

This layer covers the representation of the concrete messages exchanged within an HIS *interaction scenario* into the TTCN-3 testing language. The corresponding TTCN-3 construct is the **template**, a kind of instance of a TTCN-3 **type** definition. Besides expressing values for a particular message type definition, a **template** allows and enables additional functionalities such as definition of pattern values, wild-cards, parameters, etc., elements very useful when validating the received messages along the lifeline of a particular *actor*.

Furthermore, this layer includes the representation in TTCN-3 of the information designating the *actor's* location or semantic information that has to appear in conjunction with each message sent

or received by each *actor*. Since this information can be changed while keeping fixed the *actor's* roles (TS or SUT), hence not affecting the derived TTCN-3 test configuration, and the other annotations within an annotated *interaction scenario*, the suitable TTCN-3 construct to define and store *semantic and location parameters* is the TTCN-3 **module parameters**.

4.3.2.1 Annotations for Module Parameters and Templates

The setup configuration and location parameters for each *actor* have to be accessible at *interaction scenario* model level, independent of the modelling language chosen for its representation. This information can be annotated, for example, by using a convention such as `IP:port` for SUT *actors*, or only the `port` for TS *actors* (the IP will be obtained from the hosts where the test system itself runs). When using other communication protocols such as web services, obviously, the location parameters refer to URL values. This information is associated in TTCN-3 with *component instances* (i.e, PTCs), hence, it is used when deriving the TTCN-3 **test configuration**.

Another category of information refers to various values used within messages sent out from a particular *actor*. They can be annotated in the form of value pairs: `identifierName - value`, e.g., Application_Namespace_ID = "FOKUS". These value were designated as *semantic parameters* in the proposed test system architecture introduced in the previous chapter (Figure 3.25).

A further group of annotations regard the additional information that accompanies each *interaction* within an *interaction scenario*. First of all, the messaging standard and, where possible, its exact version, used in combination with a particular *interaction* shall be provided, e.g., HL7 v2.3.1 or HL7 v2.5 (in Figure 4.10). Secondly, the concrete identifier for the *message structure*, should be provided as well, e.g., ADT_A01. Additionally, an identifier for the message being used in conjunction with an *interaction* shall be supplied. This identifier points to an existing message from a message repository that corresponds to a TTCN-3 **template** identifier, e.g., in Figure 4.10, the identifier ADT_A01_Visit is the HL7 message that $\texttt{Actor}_j$ sends. However, that referred message may not be directly used as it is available in the repository. Hence, the pieces of information that may be changed within a message compared to the referred message have to be enabled as annotation also at *interaction* level. This can be realised by annotating on each *interaction* the identifier of a function with the corresponding parameters. This function identifier and signature shall be used during the derivation of TTCN-3 scripts to generate a corresponding TTCN-3 **function** whose role is to modify an existing TTCN-3 **template** by changing some of its fields and to produce a new **template** ready to be used for a particular *interaction*. Last but not least, also at *interaction* level, the transport protocol needs to be specified. This annotation element is used as well along the derivation of *actor types* into TTCN-3 elements.

4.3.2.2 Derivation Algorithm for Module Parameters

This is realised by generating a set of TTCN-3 **module parameters** which allow running the same test behaviour but with different locations or semantic values for each *actor*. The way it is represented into TTCN-3 is presented in the derivation algorithm bellow.

for each $Actor_j$ with TS role within an *interaction scenario*
    generate a **module parameter** corresponding to the location annotation of the $Actor_j$;
    e.g., `location_TS_Actor_j` from Listing 4.12;
    generate a **module parameter** corresponding to each identifier annotation of the $Actor_j$;

e.g., `Actor_j_Sending_Application_NamespaceID` from Listing 4.12;

for each *Actor*$_{n+i}$ with SUT role within an *interaction scenario*
generate a **module parameter** corresponding to the location annotation of the *Actor*$_{n+i}$;
e.g., `location_SUT_Actor_(n+i)` from Listing 4.12;
generate a TTCN-3 **const address** corresponding to the generated location
module parameter for the SUT *Actor*$_{n+i}$, e.g., `address_SUT_Actor_(n+i)`
from Listing 4.13;

The output of this algorithm is a set of module parameters used for localising the *actors* or for tuning messages issued or received by those *actors*. An example of such parameters is presented in Listing 4.12. Additionally, for localising SUT *actors*, for each SUT *actor*, a TTCN-3 **address** is generated as well, as shown in Listing 4.13. This **address** is used in **send to address** constructs and is assigned with the location parameters of SUT *actors*.

Listing 4.12: TTCN-3 Module Parameters

```
modulepar {

 charstring location_TS_Actor_j := "4000 + j";
 //...

 /**
  *  Sending Application and Facility Parameters
  */
 charstring Actor_j_Sending_Application_NamespaceID:="Application_FOKUS";
 charstring Actor_j_Sending_Facility_NamespaceID:="Facility_FOKUS";
 charstring Actor_j_Assigning_Authority_NamespaceID:="Authority_FOKUS";
 charstring Actor_j_OID:="1.3.6.1.4.1.21367.2010.2.1.419";


 charstring location_SUT_Actor_(n+i) := "127.0.0.1: 5000+(n+i)";
 //...

};
```

Listing 4.13: TTCN-3 Using Address Construct to Refer SUT Actors

```

type charstring address;

const address address_SUT_Actor_(n+i) := location_SUT_Actor_(n+i);
//...
```

4.3.2.3 Message Skeletons and Messages Ready to be Used

Message skeletons refer to a repository of HL7 v.2.x traces, which are obtained from real runs of interoperability scenarios or are manually defined. The main objective of these messages is to build on top of them the test data tuned with values of interest for particular fields. The reasoning behind this is that given the increased complexity of an HL7 v.2.x *message structure*, it is time consuming and effort demanding to fill in values for each test message, values which can remain

invariable when running different *interaction scenarios*. Hence, an optimisation of the whole process of defining and preparing test data is to load at a first step already existent HL7 messages. Tuning only the values of interest within a test message obtained from an existent trace, lead to a better optimisation of the whole process and to an increased focus over some key values.

Either in the form of not-refined test messages, i.e., *message skeletons*, or in the form of ready to be used test data, i.e., already tuned messages, their representation in TTCN-3 is achieved by employing the **template** construct.

From the point of view of an HIS *actor* involved in a *message exchange pattern*, with respect to the communication direction, there are two types of messages that the *actor* has to handle: the messages it sends and the messages it receives to, respectively from, a counterpart *actor*. Accordingly, in TTCN-3, for an *actor* with TS role, one represents those messages as **sending templates** and **receiving templates**. The first category contains **templates** fully defined, i.e., each leaf of the message which is not optional has to have a concrete value. To the second group belong TTCN-3 **templates** whose leaf fields can be assigned patterns, wildcards, etc., and are used to check the SUT responses. They enable an enhanced matching mechanism that does not restrict the test system to match concrete test oracle values only, but leaves place as well for a larger set of possibilities when reasoning an received message from the SUT.

Along the derivation algorithms presented in this section, the identifiers of the TTCN-3 **templates** generated out of HL7 v2.x *message skeletons* are referred to within the description of the test behaviour of an *actor*. Hence, at annotation step of an *interaction scenario*, specifically, when referencing a message identifier on a *interaction*, it is assumed that a set of identical identifiers for TTCN-3 *templates* are already generated, and made available at site using **import** statements.

4.3.2.4 Guidelines

The following guidelines apply to the construction of message identifiers:

- the identifier for the generated **module parameter** storing the location configuration of a TS or SUT *actor*, shall be prefixed by: `location_TS` respectively `location_SUT` and postfixed with the *actor*'s name e.g., `location_TS_Actor_j`, `location_SUT_Actor_(n+i)` in Listing 4.12, line 3 respectively line 15.

- given the complexity and the size of an HL7 **template** (even approaching 2000 TTCN-3 LOC), it is recommended to define one TTCN-3 **module** file for each **template** derived out of a **message skeleton**.

- all **templates** identifiers referenced within an annotated *interaction scenario* shall use the postfix `_Template`. Additionally, these identifiers have to encapsulate somehow the *message exchange pattern* where they are referenced and at which step in terms of chronological sequence number within the MEP they are used, e.g., `_MEP_j_Step_i_Template`. Finally, they have to contain the roles and names of the two *actors* (ordered by the sender-receiver direction) connected by the *interactions* referring these identifiers, e.g., `_TS_Actor_j_SUT_Actor_(n+i)_MEP_k_Step_y_Template`.

4.3.3 TTCN-3 Functions Layer

The functions layer from the proposed generic architecture for a HIS interoperability test system (Figure 3.25, section 3.4) can be easily represented into the TTCN-3 test technology, since this technology offers the language constructs and architectural design to support or to build a variate set of functions.

4.3.3.1 Template Generator

Since TTCN-3 **type** definitions resulting from mapping of HL7 v2.x *message structures* have a high degree of complexity, the **templates** based on these types are very complicated as well and almost impossible to be defined in a manual manner.

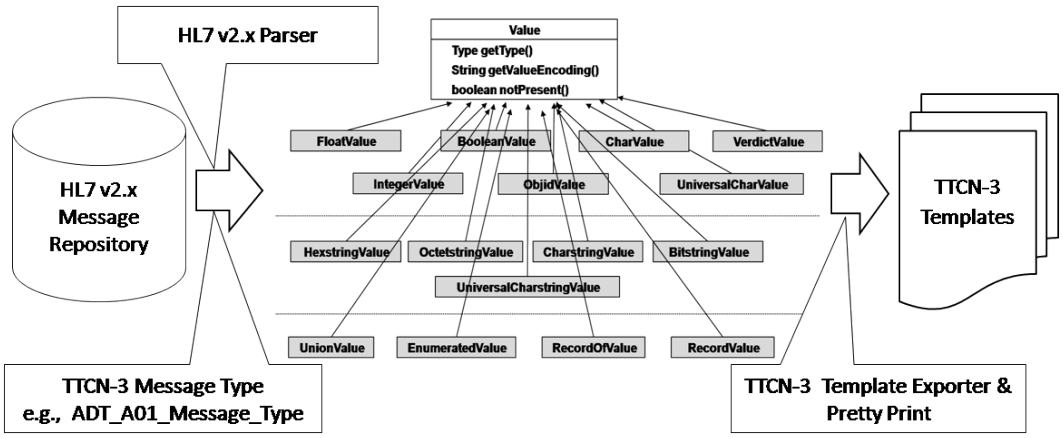


Figure 4.11: TTCN-3 Template Generator

In order to provide TTCN-3 templates based on a type system, one has the following possibilities:

- *manual template definition*: the templates are defined manually. A larger effort is demanded to achieve this. The templates are described directly in the TTCN-3 language, so that later changes are possible. However, due to the manual development a lot of inconsistencies may occur in the specification.
- *templates generated from a data pool*: a data pool which conforms to the messages of interest is provided. A transformer is needed to transform the data given in the data pool to TTCN-3 templates. The transformer is based on the data pool message structure and may produce TTCN-3 templates directly or may use an intermediate format. After the transformation, the test data is not only available in the data pool, but also in TTCN-3.
- *templates loaded via external functions*: also in this case, a data pool is required. The templates are not visible to the TTCN-3 test specification, but are loaded dynamically at the execution time into existent template variables. Afterwards, the loaded templates can be referenced like normal TTCN-3 templates. This mechanism is similar to the serialisation mechanism in the Java language. It has the advantage that the decoding function of the test system can be used instead of a transformer, but has the disadvantage that the template specification is not visible and no changes can be made in the template definition. Moreover, external function signatures for each template's type need to be defined.

In the framework presented in this thesis, the second approach was preferred. Consequently, the presented TTCN-3 framework was extended with a mechanism to automatically derive TTCN-3 **templates** out of HL7 v.2.x traces available as a *message skeletons* repository.

Figure 4.11 shows how this automated mechanism can be technically realised in TTCN-3. The proposed TTCN-3 *template generator* acts as a *message skeleton generator* function that accesses a repository of HL7 v2.x messages and translates them into their TTCN-3 correspondent, i.e., **templates**. However, from a technical view, this step is not as straightforward as it seems, since firstly it requires parsing an HL7 v2.x message and loading its content into a tree-like memory structure that, finally will be used to export and pretty print the resulted TTCN-3 **template**.

4.3.3.2 Template Tuning Functions

Template tuning functions are TTCN-3 **functions** used to change the content of test data, stimuli or oracle **templates** already generated with the *template generator*. As previously explained, the test data, i.e., TTCN-3 **templates**, is generated from real traces or messages. The generated messages partly contain valuable information for test purposes while other parts of the messages, such as controlling information or routing information, need to be adapted to the test environment needs by using so called tuning functions. The tuning functions need to be written separately by testers, since they cannot be generated.

The example provided in Listing 4.14 illustrates how the tuning functions are created and used. The `setSendingParam4Actor_j_ACK_Message_Type` gets as parameter a **template** of type `ACK_Message_Type` generated to TTCN-3 from a real HL7 ACK message and a second **template** of type `ADT_A01_Message_Type` generated as well.

The content of this message is partly changed within the function and it will be returned back to the calling behaviour. It is important to note that only some fields are changed. In the example, the `Message_Control_Id` and the `Time` **fields** are adapted to the testing needs, their values being generated from scratch by invoking other **external functions**. Other fields such as `Namespace_ID` are assigned a value taken from another **template** of type `ADT_A01_Message_Type` provided as input to the function as well. The most part of the content remains unchanged.

Listing 4.14: TTCN-3 Template Tuning Function Example

```
                                                                                1
function setSendingParam4Actor_j_ACK_Message_Type(                              2
   in HL7_ADT_Types_v231.ADT_A01_Message_Type the_ADT_A01_Template,              3
   in template HL7_ACK_Types_v231.ACK_Message_Type the_ACK_Template)             4
   return template PIX_Profile_Types_v231.ACK_Message_Type {                    5
                                                                                6
   the_ACK_Template.MSH_Segment.Sending_Application.Namespace_ID :=             7
      the_ADT_A01_Template.MSH_Segment.Receiving_Application.Namespace_ID;      8
   the_ACK_Template.MSH_Segment.Sending_Facility.Namespace_ID :=                9
      the_ADT_A01_Template.MSH_Segment.Receiving_Facility.Namespace_ID;         10
                                                                                11
   the_ACK_Template.MSH_Segment.Receiving_Application.Namespace_ID :=           12
      the_ADT_A01_Template.MSH_Segment.Sending_Application.Namespace_ID;        13
   the_ACK_Template.MSH_Segment.Receiving_Facility.Namespace_ID :=              14
      the_ADT_A01_Template.MSH_Segment.Sending_Facility.Namespace_ID;           15
                                                                                16
   the_ACK_Template.MSH_Segment.Message_Control_ID :=                           17
      getMessageId();                                                           18
   the_ACK_Template.MSH_Segment.DateTimeOfMessage.Time :=                       19
```

```
        getTimestamp();                                                    20
                                                                           21
    the_ACK_Template.MSA_Segment.Message_Control_ID :=                     22
        the_ADT_A01_Template.MSH_Segment.Message_Control_ID;               23
                                                                           24
    return the_ACK_Template;                                               25
}                                                                          26
```

Since the tuning functions need to be written manually, it is required that the tester is familiar with the message structures as it has to navigate using the dot notation in order to reach the **fields** to be modified.

4.3.3.3 Conformance Validation Functions

The main mechanism provided by the TTCN-3 test technology to detect conformance issues is the template matching mechanism, a mechanism embedded in any TTCN-3 test system implementation. It enables an easier and better verification of the SUT responses against the test oracles or expected reactions. **Templates** are closely coupled to TTCN-3 types, consequently, the closest the **type** definition is to the HIS messaging standard, the more powerful is the template matching mechanism. The TTCN-3 type system for HL7 v2.x introduced in this chapter copes with a high level checking of conformance compliance to the mentioned messaging standard. Furthermore, by additionally addressing the semantic mapping of HL7 *coded values* into the TTCN-3 type system, the conformance compliance validation during template matching increases in quality and assures a higher coverage of catching possible conformance issues. To summarise, the template matching mechanism enables the following types of validations:

- checking the *validity* of the HL7 v2.x messages and whether or not it is well formed.
- checking the *cardinality* which defines the minimum and maximum number of repetitions for a particular element.
- checking the *usage* whether an element must be present.
- checking the *length* of a particular element
- checking the *code sets* within messages whose compliance requires to use specific coding systems, but one may use alternative coding systems as supported by the type definition.

However, besides the complexity of its message structures, the HL7 messaging standard specifies a larger set of conditions concerning the dependability between the values of various fields. The meaning is that conditionality predicates must be associated with the referred elements (marked with “C”). They identify the conditions under which the elements must be or are allowed to be present. Each predicate is based on other values within the message and may be expressed as a mathematical expression or in text and may utilise operators such as equivalence, logical AND, logical OR and NOT. The conforming sending and receiving *actors* shall both evaluate such predicates.

The only possibility to verify such conditionality conditions for TTCN-3 **templates** is to manually implement a set of additional functions.

Figure 4.12 shows the definition of the `complex data type` Hierarchic Designator (HD) for HL7 v2.5.1. It consists of a sequence of 3 data elements. The second and third components, *Universal*

Seq.	Comp. No.	Index	Sec.	Table (this use)	Table (Comp.)	Opt.	Length	Data Type	Data Structure
1	139	Namespace ID		0300	0300	O	20	IS	IS
2	140	Universal ID				C	199	ST	ST
3	141	Universal ID Type		0301	0301	C	6	ID	ID

Figure 4.12: HL7 v2.5.1 Hierarchic Designator (HD) Complex Data Type

ID and *Universal ID Type* are `conditional` (marked with "C" in Opt. column). The meaning is: if the first component *Namespace ID* is present, the second and third components are optional. If the third component is present, then the second must also be present (although in this case the first is optional). The second and third components must either both be valued (both non-null), or both be not valued (both null).

Listing 4.15: Mapping Example - TTCN-3 HD

```
type record HD
{
   Assigning_Authority_Table Namespace_ID optional,
   ST Universal_ID optional,
   Universal_ID_Type_Table Universal_ID_Type optional
}
```

Listing 4.15 describes the corresponding TTCN-3 definition of the HL7 complex data type from Figure 4.12. The last two components of the HD marked as conditional, are defined in TTCN-3 **optional**. The first field in TTCN-3 is also optional since the HL7 counterpart is marked as conditional.

Listing 4.16: TTCN-3 Conditionality Checking Function

```
function validationFunction_HD (
 in template SimpleAndComplexData_Types_v25.HD theHD)
 return boolean{

   if (ispresent (theHD.Universal_ID_Type) and
       not ispresent (theHD.Universal_ID_Type.Universal_ID)) {

          return false;
   }
   return true;
}
```

Listing 4.16 presents the TTCN-3 validation **function** for checking the conditionality textually specified above. This **function** has to be called in conjunction with other similar **functions** within a more generic validation function associated to a root HL7 *message structure*, e.g., `ADT_A01_Message_Type`. This way, one assures that each conditionality defined at any leaf level within the entire *message structure* is respected. Finally, the generic validation function has to be applied to each message sent or received by each *actor* with TS role, though, for sending templates, one can control and respect the imposed fields conditionality when defining the **templates**, and, in this case, the applicability of the generic validation function is redundant.

4.3.4 TTCN-3 Behaviour Layer

This layer contains the TTCN-3 realisation of the state machines corresponding to the *actors* from the *interaction scenario*.

The approach adopted in this thesis is rather different to the approaches previously presented in Fundamentals Chapter (Chapter 2). The main difference consists in the fact that in other approaches as in [Din09], [Sch03], a TTCN-3 **test component** emulates more than one *actor*. In this thesis, however, the concept of multi test *actors* is applied, i.e., each *actor* being simulated by only one TTCN-3 **test component**. The proposed design of the test configuration might be considered not efficient from a performance testing point of view, in case that a hundred *actors* have to be emulated by the test system. But this design is conceived in such a way that it suits to interoperability scenarios where the number of involved *actors* is to a lesser extent emphasised.

A TTCN-3 **test configuration** refers to creating and interconnecting **test components** (PTCs, MTC, TSI), i.e., instances of **component types**. This **test configuration** serves as an abstract setup for one **testcase** specification which corresponds to an annotated HIS *interaction scenario*. Consequently, different annotations applied to the same HIS *interaction scenario* lead to different TTCN-3 **test configurations** and parameterisable **testcases**.

The derivation algorithms presented next help to automatically generate compilable TTCN-3 test code in terms of **test configurations** and **testcase** definitions that suit a specific annotated HIS *interaction scenario*.

4.3.4.1 Annotations for Test Configurations and Testcase Specifications

Besides the annotations required to derivate TTCN-3 **type** definitions, the model of an *interaction scenario* has to provide additional information also in the form of annotations that are used to derive other TTCN-3 elements, e.g., *actor* behaviours. This will complete the TTCN-3 code to become a complete abstract test specification (ATS). However, the annotations introduced before in this section are employed as well along the derivation algorithms for designing *actor* behaviours presented next.

The designator of the *message exchange pattern* that contains a particular *interaction* shall be part of the annotation associated to an *interaction* as well. The access to the identified *message exchange pattern* is crucial when deriving the behaviour of an *actor*, since it contains the chronological sequence of interaction steps with another *actor*.

4.3.4.2 Derivation Algorithm and Test Design for Actor Behaviours

The generation of the *actor* behaviours assumes that the information annotated for each *actor* is mapped into TTCN-3.

The algorithm used for the derivation of TTCN-3 scripts for representing the behaviour of *actors* with TS role in TTCN-3 is presented below:

for each $Actor_j$ with TS role within an *interaction scenario*
    determine its *actor type* as «`ActorType`», e.g., in Figure 4.10 the type of
    $Actor_j$ is `Consumer`;
    search the identifier of the already generated TTCN-3 **component type** corresponding to

the «ActorType» as «ActorType»_Component_Type;

generate a TTCN-3 **function** with **runs on** clause refering to
the **component type** «ActorType»_Component_Type,
e.g., Listing 4.17 for $Actor_j$;
for each Message Exchange Pattern $\texttt{MEP}_k$ on the lifeline of the $Actor_j$
generate the TTCN-3 behaviour, i.e., state machine corresponding to the $\texttt{MEP}_k$,
e.g., in Listing 4.17;

The output of this algorithm is a TTCN-3 **function** with **runs on** clause. The core part of the algorithm is the generation of TTCN-3 state machines for each *message exchange pattern* (MEP) annotated in the interaction scenario. The state machines are constructed using **alt, send/receive, timer, altsteps** and other behavioural statements. For each MEP identified in the previous chapter a mapping strategy is applied. For example, the listing 4.17 corresponds to a <*Request - Immediate Response*> MEP.

Listing 4.17: TTCN-3 Actor's State Machine with one Message Exchange Pattern

```
function behavior_TS_Actor_j() runs on Consumer_Component_Type {

/**
 *     State Machine for Message Exchange Pattern (MEP1)
 *     <Request - Immediate Response>
 */
timer replyTimer_MEP1;

var template HL7_QBP_Types_v25.QBP_Q23_Message_Type reqTemplate_MEP1;
var template HL7_RSP_Types_v25.RSP_K23_Message_Type resTemplate_MEP1;

var HL7_RSP_Types_v25.RSP_K23_Message_Type resValue_MEP1;

//     apply tunning functions
reqTemplate_MEP1 := setSendingParam4TS_Actor_j_QBP_Q23_Message_Type(
        TS_Actor_j_to_SUT_Actor_nPlusi_MEP1_QBP_Q23_Template);

resTemplate_MEP1 := setReceivingParam4TS_Actor_j_RSP_K23_Message_Type(
        TS_Actor_j_to_SUT_Actor_nPlusi_MEP1_RSP_K23_Template,
        reqTemplate_MEP1);

port_to_SUT_MLLP.send(reqTemplate_MEP1) to address_SUT_Actor_nPlusi;

replyTimer_MEP1.start(20.0);
alt {
 [] port_to_SUT_MLLP.receive(resTemplate_MEP1) -> value resValue_MEP1
 {
      replyTimer_MEP1.stop;

      //     apply validation functions
      if (validationFunctions_MEP1(reqTemplate_MEP1, resValue_MEP1))
      {
        setverdict(pass);
      } else {
        setverdict(fail);
        mtc.stop;
      }
```

```
    }                                                                    39
    [] port_to_SUT_MLLP.receive {                                        40
                                                                         41
       replyTimer_MEP1.stop;                                             42
       setverdict(fail);                                                 43
       mtc.stop;                                                         44
    }                                                                    45
    [] replyTimer_MEP1.timeout {                                         46
       setverdict (fail);                                                47
       mtc.stop;                                                         48
    }                                                                    49
  }                                                                      50
}                                                                        51
```

The example instantiates firstly two template variables, one for the sending message and another one for the receiving message. Both messages are tuned by using *tuning functions* with information required by the test environment. These TTCN-3 functions, `setSendingParam4TS_Actor_j_QBP_Q23_Message_Type` and `setReceivingParam4TS_Actor_j_RSP_K23_Message_Type` have as parameter two existent sending, respectively receiving **templates**. These existent **templates** were previously generated by using the *template generator function* and were imported into the test behaviour **module**. They already contain a lot of information necessary for testing, but the message header information needs to be adapted to the test environment by using the *tuning functions*. Next, the sending message is sent to SUT over the `port_to_SUT_MLLP` port instance used for the communication over the MLLP transport protocol. The location of the SUT is explicitly emphasised by using the to `address_SUT_Actor_nPlusi` construct where `address_SUT_Actor_nPlusi` stores the location of the SUT $Actor_{n+i}$. The response of the SUT is evaluated with the **alt** block. The first alternative expects the correct response, which is stored into the `resValue_MEP1` variable. The response is evaluated twice. Firstly, it is matched against the `resTemplate_MEP1` **template** for structure and conformance validation. Secondly, a *checking and validation function* checks for different semantic correlations imposed by the messaging standard, e.g., the presence of fields depending on whether other fields are present. The failing behaviour of the SUT is cached by a timer **timeout** event or by receiving a not expected message.

4.3.4.3 Derivation Algorithm and Test Design for Test Configuration and Testcase

The way a TTCN-3 **testcase** corresponding to an annotated *interaction scenario* is designed is shown below in the following derivation algorithm:

for each different annotated *interaction scenario*
  generate a TTCN-3 **testcase**, e.g., in Listing 4.18 that **runs on** the empty-body `MTC_Component_Type` and has as **system** the `TSI_Component_Type`, both already defined in Listing 4.11;

  for each $Actor_j$ with TS role
    create a TTCN-3 **component instance**, $\texttt{v\_Actor}_j$, whose type is the TTCN-3 **component type** generated for the $Actor_j$'s *type*; the new component instance is assigned a name derived from $\texttt{Actor}_j$ name and postfixed with the value contained in the already generated **module parameter** `location_TS_Actor_j` for $\texttt{Actor}_j$, e.g., Listing 4.18;

for each `transport_protocol_k` used by the $Actor_j$ to communicate with any SUT *actor*
generate a TTCN-3 **map** statement that will map the port of the `v_Actor`$_j$ **component instance** to communicate with the SUT over the `transport_protocol_k`, `port_to_SUT_«transport_protocol_k»` with the **system** port defined on the `TSI_Component_Type` to communicate with $Actor_j$ over `transport_protocol_k`, e.g., Listing 4.18;

for each $Actor_i$ with TS role communicating with $Actor_j$
generate a TTCN-3 **connect** statement that will connect the **component instances** associated to the two *actors*, e.g., Listing 4.18;

A **testcase** behaviour sets up a test configuration by instantiating for each **actor** an TTCN-3 **component** instance. These components are mapped to the SUT through transport protocol specific ports. Similarly, the components are interconnected. The connections between components is derived automatically from the annotated *interaction scenarios*. An example of a test configuration is provided in Listing 4.18 where all steps in *creating*, *mapping* to SUT, *connecting* and *starting* with other components of $Actor_j$' component are presented. Each component is started with the function behaviour associated to that actor, e.g., `behavior_TS_Actor_j()`.

Listing 4.18: TTCN-3 Testcase's Configuration

```
testcase testcase_AnnotatedInteractionScenario() runs on
         MTC_Component_Type system TSI {

  // PTCs creating
  //...
  var Consumer_Component_Type v_TS_Actor_i:=
        Consumer_Component_Type.create("Actor_i" & "_" & location_TS_Actor_i);
  //...
  var Consumer_Component_Type v_TS_Actor_j:=
        Consumer_Component_Type.create("Actor_j" & "_" & location_TS_Actor_j);
  //...

  // PTCs mapping
  //...
  map(v_TS_Actor_j:port_to_SUT_MLLP, system:port_to_Actor_j_MLLP);
  map(v_TS_Actor_j:port_to_SUT_DICOM, system:port_to_Actor_j_DICOM);
  //...

  // PTCs connecting
  //...
  connect (v_TS_Actor_i:port_to_Actor_j, v_TS_Actor_j:port_to_Actor_i)
  //...

  // PTCs starting
  //...
  v_Actor_j.start(behavior_TS_Actor_j() );
  //...
  all component.done;
}
```

4.3.4.4 Guidelines

To increase the readability and maintainability of the generated test configurations and test behaviours, along the derivation algorithms a set of guidelines should be adhered to. Similar to previously presented guidelines, the following notation rules apply to the test components, test parameters and behavioural functions identifiers.

- for any TTCN-3 behaviour function used to describe the behaviour of an *actor*, the function's identifier shall use the following pattern: `behavior_TS_«Actor_j»`, e.g., Listing 4.17, line 2.
- each TTCN-3 **timer** associated with an identified *message exchange pattern*, has an identifier that shall follow the pattern: `replyTimer_«MEPi»`, e.g., Listing 4.17, line 8.
- the identifiers for TTCN-3 **template variables**, **type variables** used within a behaviour function and which correspond to a particular *message exchange pattern* shall be prefixed by `_«MEPi»`, e.g., Listing 4.17, lines 10, 11, 13.
- the identifier for each **component instance** created for each *actor* within a test configuration shall follow the pattern: `v_TS_«Actor_i»`, e.g., Listing 4.18, lines 6, 9.
- each **component** shall be created with an identifier obtained by concatenating the *actor*'s name with *actor*'s location value taken from the generated location **module parameter**: `«Actor_j»_location_TS_«Actor_j»`, e.g., Listing 4.18, lines 7, 10. This identifier plays a very important role in designing a dynamic test configuration in TTCN-3 since this identifier is the glue between the abstract test specification and the entity that correspond to a **component instance** in the communication layer.

4.3.5 TTCN-3 Means of Communication Layer

The means of communication layer between the test system and the SUT, as introduced in section 3.4 where the generic architecture of a test system for HIS interoperability testing is presented, are mapped in a TTCN-3 based test system into a *test Adapter* and a *test CoDec*. These two components implement the TTCN-3 Runtime Interface (TRI)[ETS07b] and TTCN-3 Control Interface (TCI)[ETS07c]. The task of making the TTCN-3 ATS executable is performed by TTCN-3 compilers.

Figure 4.13 is an extension of Figure 4.10 and additionally contains these two TTCN-3 elements of a test system. The framework developed in this work enables a transport protocol plug-in mechanism to support different communication protocols, i.e., multiple TTCN-3 *Adapters*, each protocol being supported by a separate plug-in. In order to exemplify the plug-in mechanism architecture, the MLLP [MLL09] protocol is used as an example. The suitable *CoDec* is triggered by the sending or receiving of messages by the activated *Adapter* depending on which transport protocol is in use.

For handling MLLP *interactions* between TS *actors* and the SUT *actors*, the *Adapter* uses a free Java implementation library for HL7 called HAPI [HAP09].

Figure 4.13 illustrates how the TTCN-3 **ports** (TTCN-3 Design) relate to the *Adapter's* components (Adapter Design). For each *actor* simulated by the test system, the *Adapter* defines a pair of

server and *client* entities. In case the actor communicates over more than one protocol, the a client/server pair is created for each protocol. The client/server pair is supported by the corresponding protocol adapter plug-in. The *client* becomes active whenever the test system stimulates the SUT and has the additional role of dispatching the message to the desired SUT *actor's* address. The *server* is created and started for each TTCN-3 PTC that has direct interaction with at least one SUT *actor*. This entity is responsible for handling connections from SUT *actors*. The main advantage of this design is that, whenever the test configuration changes, i.e., the number of PTCs, number of SUT *actors* and their interactions, the *Adapter* does not need to be changed.

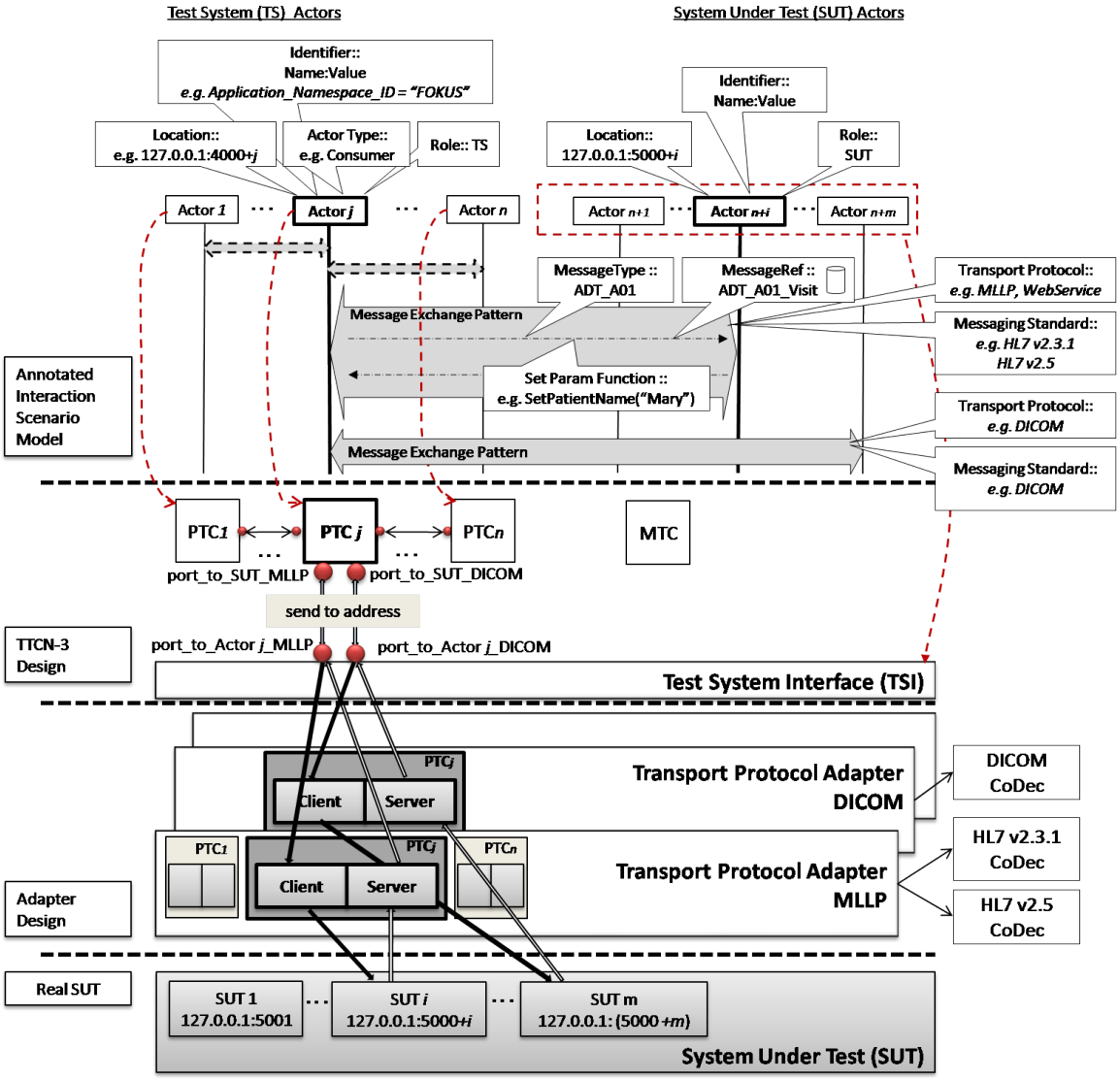


Figure 4.13: TTCN-3 Test Configuration Design and Derivation from an Interaction Scenario (II)

The role of the *CoDec* is to translate messages from TTCN-3 format into the format required by the SUT's interfaces, and vice versa. The implementation of the *CoDec* is also based on the HL7 library [HAP09] that acts as a parser for HL7 messages. A core library of functions that help to encode/decode HL7 v2.3.1 and v2.5 basic types was developed.

Chapter 5

Case Studies: Interoperability Tests for IHE PCD and ITI Domains

Ethical axioms are found and tested not very differently
from the axioms of science. Truth is what stands the test of experience.
– Albert Einstein

IHE [IHE97] standardised a set of profiles organised by domains such as cardiology, laboratory, radiology, patient care devices, etc. These integration profiles can be seen as an agreement between industry partners by means of eliminating the potential ambiguities that the HL7 standard allows for implementers. They offer developers a clear implementation path for communication standards supported by industry partners.

An IHE integration profile consists of *actor types*, which are abstract representations of the involved units, and a set of *transactions*, which define the communication between the *actors*, which in turn map to different *message exchange patterns* and consist of a set of *interactions*.

The proposed test design concepts proved their feasibility along two case studies. They have been applied to the implementation of the TSs for two profiles from the PCD [PCD06a] and ITI [ITI09a] domains.

5.1 IHE Patient Care Devices Domain

The first case study analysed in this thesis belongs to the domain of IHE Patient Care Devices (PCD) [PCD06a]. PCD area addresses the integration of medical devices in the healthcare enterprises, which could lead to significant improvements in patient safety and quality of care. PCD profiles standardise the communication scenarios and flows between medical devices directly connected to a monitored care unit, for instance ventilators, blood pressure sensors, infusion pumps, etc., and all other units from the medical environment like medical clinics, doctors, etc., interested and involved in receiving data from those medical devices.

PCD Technical Framework. The Patient Care Devices Technical Framework (PCD TF) identifies a subset of the functional components of the healthcare enterprise, called IHE actors, and specifies their interactions in terms of a set of coordinated, standards-based transactions. The PCD TF was initially published in 2006 as revision 1.1 of the trial implementation version. It

improved with each supplement document added to the original trial implementation version. The version we refer to within this thesis is the original published text, revision 1.1 of the trial implementation version from August 2006. The first volume (PCD TF-1) [PCD06b] provides a high-level view of IHE functionality and shows the integration profiles as functional units where the transactions and actors emphasise their capacity to address specific clinical needs. The second volume (PCD TF-2)[PCD06c] of the PCD TF provides detailed technical descriptions of each IHE transaction used in the PCD integration profiles. PCD TF revision 1.1 referred to in this thesis includes only one integration profile: Device Enterprise Communication (DEC). This integration profile is the target of the interoperability analysis within this section. However, the PCD TF was extended later on with further integration profiles such as Alarm Communication Management (ACM), Implantable Device Cardiac Observation (IDCO), Point-of-Care Infusion Verification (PIV), Rosetta Terminology Mapping (RTM). In this section we apply the concepts of automated interoperability testing to the PCD integration profile whose detailed description of the transactions is comprised mainly in volume 2 [PCD06c] of the PCD TF.

Relationship to Standards. As with other IHE TFs, the PCD TF identifies functional components of a distributed healthcare environment (referred to as IHE actors). They are regarded solely from the point of view of their interactions within the healthcare enterprise. IHE is an implementation framework, not a standard, hence, each IHE Technical Framework (IHE TF) defines a coordinated set of transactions based on different messaging standards. The DEC integration profile has as its underlying standard solely the HL7 messaging standard version 2.5 [HL703].

The Context of the Work. This work was carried out between December 2007 until November 2009 in the context of the PRO INNO research project Test Automation for the Next Generation of Medical Systems (TestNGMed) [TNG09], which was the German part of the bigger European research project Reliability Testing Of Medical Systems (ReTeMes) [ReT09]. The TestNGMed project consortium was built by the following German partners: 1) Technical University of Berlin, ETS chair [TUB10], 2) sepp.med GmbH. [SME10], and 3) Applied Biosignals GmbH. [ABI10]. The author of this thesis was affiliated along the project duration with the Technical University of Berlin. Besides the aforementioned German partners, two other European partners were part of the consortium of the European ReTeMes project: Politehnica University of Bucharest, Faculty of Automatic Control and Computers [UPB10] and the company Info World [IFW10].

The general target of the TestNGMed research project was to develop a test methodology and its instantiation test system based on TTCN-3 test technology for testing HL7/IHE based medical systems. In the end, the main result of the project was a test framework supporting the test methodology and providing prefabricated test configurations, test data and test procedures which enable an easy, flexible, reusable and scalable testing of HL7/IHE based medical systems.

The developed concepts and the test framework offer a basis for automated interoperability test solutions. This case study demonstrates the feasibility of the approach by applying the test methodology based on TTCN-3 test technology to test PCD DEC integration profile compliant implementations. This work captured a lot of interest from science and industry and is the starting point for further research projects. A first follow-up project of this work was the second case study treated in this thesis in the next section.

5.1.1 DEC Integration Profile

According to the PCD TF volume 1, revision 1.1 version [PCD06b] (PCD TF-1), one of the highest priority of the requirements addressed by PCD domain is the sharing of enterprise PCD data

including as goals the increase of the productivity by shortening the decision time, minimising transcription errors and increasing the contextual information regarding the data. PCD data includes periodic physiologic data (heart rate, invasive blood pressure, respiration rate, etc.) aperiodic physiologic data (patient weight, cardiac output, etc.), CLIA waived (or equivalent international waiver) point-of-care laboratory tests (i.e. home blood glucose, etc.) and may include contextual data such as the patient ID, caregiver identification, and patient care device configuration information.

The Device Enterprise Communication (DEC) integration profile, included in the PCD TF-1, targets the need for consistent communication of PCD data within the enterprise. Examples of medical enterprises that receives PCD data are Health Decision Support Systems (HDSSs), EHRs, etc. The DEC integration profile supports consistent communication by mapping the PCD data from proprietary legacy syntax and semantics into a single syntactic and semantic representation for communication to the enterprise. In general, IHE integration profiles do not operate independently. As mentioned in section 2.1, revision 1.1 PCD TF-1, in order to coordinate time across networked systems, there is a required dependency between the DEC profile and the Consistent Time (CT) integration profile from the IHE ITI domain [ITI09c]. Consequently, the PCD data is time stamped with a consistent enterprise time.

With respect to the security and privacy of the transfer of PCD data, in the DEC integration profile there are no security considerations treated, but only the assumption that the DEC profile is implemented in a single enterprise on a secure network.

5.1.1.1 Actors

IHE actors are information systems or components of information systems that produce, manage, or act on information associated with operational activities in the enterprise. Figure 5.1, as defined in [PCD06b] (Section 3.1), shows the actors directly involved in the DEC integration profile and the relevant transactions between them.

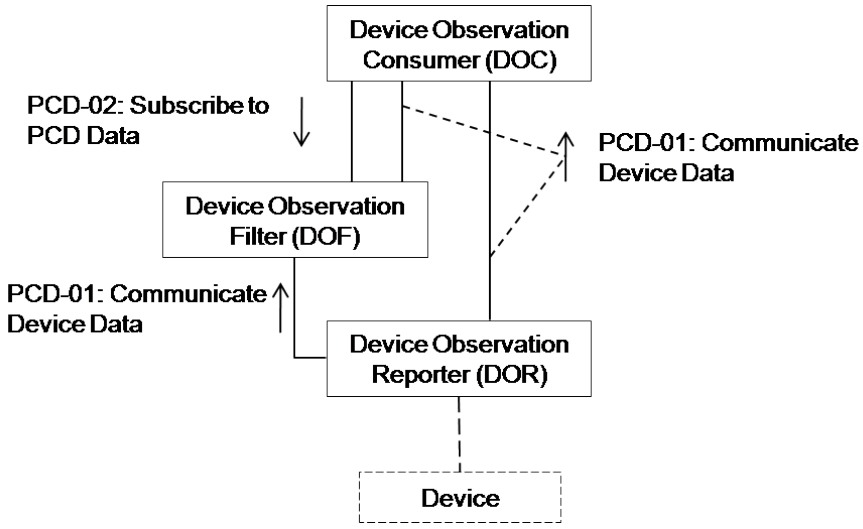


Figure 5.1: DEC Integration Profile: Actors and Transactions

Device Observation Reporter. The *Device Observation Reporter* (*DOR*)[1] receives data from

[1]The name of the *Device Observation Reporter* actor is used interchangeably with *DOR* or *Reporter* or *PCD Re-*

patient care devices (i.e., devices that collect PCD data) including those based on proprietary formats, and maps the received data to transactions providing consistent syntax and semantics. In other words, the *DOR* actor has the role of monitoring all medical devices from which data needs to be inserted into the system (e.g., ventilator, blood pressure sensor, etc.), and the role of converting that data into digital format, if necessary, acting as a data provider for the other actors of the PCD profile. The mechanism by which the *DOR* actor receives the PCD data is out of scope for the version of the investigated PCD TF.

Device Observation Filter. The *Device Observation Filter* (*DOF*)[2] actor is responsible for providing PCD data filtering services based on publish/subscribe predicates negotiated with client applications implementing the *Device Observation Consumer* actor. "Publish and subscribe" refers to the capability of one system, the "Publisher", to offer a data stream that can be sent to recipient systems upon a subscription. This actor is optional, which means that the implementers can support its functionality or not.

The general interaction model (Figure 5.1) corresponds to two possible interaction diagrams that illustrate the sequence of the possible interactions between the DEC actors. If an implementation does not support the *DOF* actor, then we have the situation presented in Figure 5.2. In this case the *DOR* actor sends PCD data directly to the *DOC* actor. The second figure (Figure 5.3) corresponds to the situation when a filter mechanism is used and implemented by the *DOF* actor.

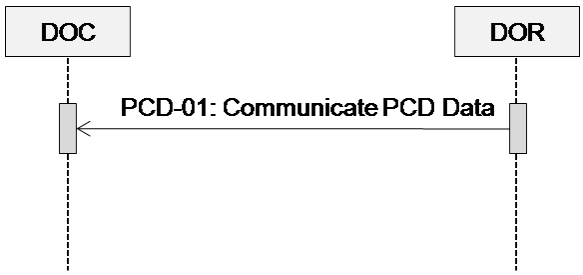


Figure 5.2: Communicate non Filtered PCD Data

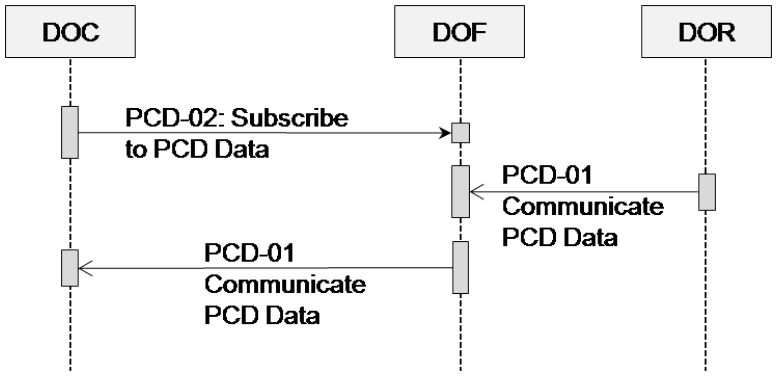


Figure 5.3: Communicate Filtered PCD Data

Device Observation Consumer. The *Device Observation Consumer* (*DOC*)[3] actor is responsible for receiving PCD data from the *Device Observation Reporter*, or from the *Device Observation*

porter.

[2]The name of the *Device Observation Filter* actor is used interchangeably with *DOF* or *Filter* or *PCD Filter*.

[3]The name of the *Device Observation Filter* actor is used interchangeably with *DOC* or *Consumer* or *PCD Consumer*.

Filter, or from both. The Device Observation Filter (DOF) is the entity responsible for handling a connection between a Device Observation Consumer (DOC) of the medical data (laboratory, medical clinic, etc.) and the Device Observation Reporter (DOR), in charge of providing PCD data. The DOF manages subscriptions and offers to the DOCs subscribers the possibility to receive a subset of the data stream, according to their needs and subscription predicates.

5.1.1.2 Transactions

Transactions are interactions between actors that transfer the required information through standards-based messages. In general, the transaction identifiers specified in IHE TFs documents are formed by postfixing the domain identifier with the transaction number. As underlying standard for the two transactions comprised in the DEC integration profile, version 2.5 of the HL7 messaging standard [HL703] was selected. In general, within IHE integration profiles, the selection of an HL7 version as underlying messaging standard for a certain transaction is motivated by the most predominantly adopted HL7 version among vendor implementations of actors involved in that transaction.

Table 5.1: DEC Actors and Transactions

Actors	**Transactions**	**Optionality**
Device Observation Consumer	PCD Communicate Device Data [PCD-01]	R
	PCD Subscribe to PCD Data [PCD-02]	O
Device Observation Filter	PCD Communicate Device Data [PCD-01]	R
	PCD Subscribe to PCD Data [PCD-02]	R
Device Observation Reporter	PCD Communicate Device Data [PCD-01]	R

Table 5.1 lists the transactions for each actor directly involved in the DEC integration profile. The third column indicates the optionality of the transactions (required transactions are labelled with "R", optional transactions are labelled with "O"). Each implementation claiming to support this integration profile must perform the required transactions labelled with "R" in the table. Volume 2 [PCD06c] of the PCD TF presents in detail these transactions.

Communicate Device Data [PCD-01] Transaction. Transaction [PCD-01] can be used by all actors. In the case of a non filtered DEC workflow (Figure 5.2), the [PCD-01] transaction is used to communicate PCD data from a *DOR* to a *DOC*. In the case of the communication of filtered PCD data (Figure 5.3) the *DOF* receives PCD data from the *DOR* and communicates a selected set of the messages based upon a subscription, which has been set up as a result of a [PCD-02] (Subscribe to PCD Data) transaction.

Every application which implements the *DOR* functionality receives data from one or more PCDs (devices close to the patient) using either standards based or proprietary protocols, which are outside the scope of the IHE PCD TF. The *DOR* also sends periodic reports (to *DOF* or *DOC*) at a minimum and maximum interval, which are configured at implementation. The *DOR* does not correlate or do any interpolation of data received from the PCD source. For both situations, *DOR* as a receiver or a sender, the associated event will trigger an unsolicited update message:

- R01 - ORU Subscription (Response).

From the message structure point of view, the [PCD-01] transaction is conducted by the HL7 ORU (unsolicited transmission of an observation message) message type. The message structure is derived from the ORU_R01 HL7 v2.5 message structure, as shown in Table 5.3. Detailed descriptions of segments are provided in volume 2 [PCD06b] of the PCD TF. Also, along each segment structure, IHE introduces a set of restrictions to the original structure standardised by HL7 version 2.5.

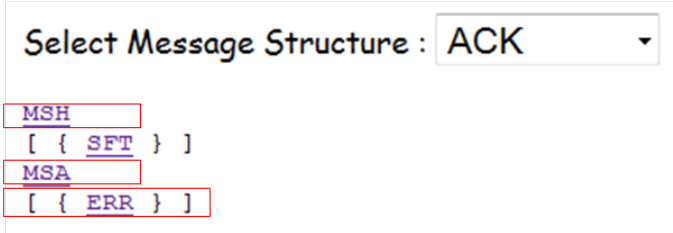


Figure 5.4: HL7 v2.5 ACK Message Structure

Discussion. The ORU_R01 message structure, as described in HL7 messaging standard version 2.5 [HL703] is more complex than the structure defined by IHE in the PCD TF. Attributes such as optionality ([ ] brackets), repetition ({ } brackets) or substructures (e.g., PATIENT, VISIT, ORDER_OBSERVATION, etc.) are preserved and clearly specified. To ease the interoperability between systems, IHE reduced the complexity of this message structure. At this place a simplification is introduced by simply removing some segments that are optional in the HL7 standard. In Table 5.3, the segments marked with "X" on the "Usage" column shall not appear in [PCD-01] messages, e.g., the optional sequence of STF segments ("[{STF}]") marked with "X" was removed from the [PCD-01] transaction even though it could appear in HL7 messages compliant with the original HL7 ORU_R01 message structure.

Table 5.2: ACK - Acknowledgement Messages

ACK	**[PCD-01] ACK Message**
MSH	Message Header
MSA	Message Acknowledgement
[ERR]	Error

Upon receipt of an ORU message, *DOC* and *DOF* validate it and shall respond with an accept acknowledgement message (HL7 ACK message), i.e., acknowledgements using the so-called *HL7 Original Mode*. The ACK message is also part of the *Communicate Device Data* [PCD-01] transaction. The returned ACK message to the initiator of the [PCD-01] transaction (*DOR* or *DOF*) is based on HL7 standard version 2.5 standard and has the structure as presented in Table 5.2. It differs slightly from the original ACK message structure specified by the HL7 standard by removing the optional sequence of STF segments (see Figure 5.4).

Subscribe to PCD Data [PCD-02] Transaction. The transaction [PCD-02] is used by the *DOC* to subscribe for PCD data from a *DOF*. It appears only in use cases corresponding to filtered communication of the PCD data (Figure 5.3). Upon receipt of a subscription request from a *DOC*, the *DOF* sets up filtering such that only those [PCD-01] messages, which satisfy the filter predicates are communicated to the *DOC*. When no explicit predicates are available regarding starting and stopping, the *DOF* will start as soon as the configuration of the predicate filters is finished and will

Table 5.3: [PCD-01] and ORU_R01 (HL7 v2.5) Message Structure

Segment	Meaning	Usage	Card.	HL7 chapter
MSH	Message Header	R	[1..1]	2
[{SFT}]	Software Segment	X	[0..0]	2
{	--- PATIENT_RESULT begin			
[	--- PATIENT begin			
PID	Patient Identification	R	[1..1]	3
[PD1]	Additional Demographics	X	[0..0]	3
..[{NTE}]	Notes and Comments	X	[0 0]	2
..[{NK1}]	Next of Kin/Associated Parties	X	[0..0]	3
[	--- VISIT begin			
PV1	Patient Visit	O	[0..1]	3
[PV2]	Patient Visit – Additional Info	X	[0..0]	3
]	--- VISIT end			
]	--- PATIENT end			
{	---ORDER_OBSERVATION begin			
[ORC]	Order Common	X	[0..0]	4
OBR	Observation Request	R	[1..1]	7
[{NTE}]	Notes and Comments	O	[0..1]	2
[{	--- TIMING_QTY begin			
TQ1	Timing/Quantity	O	[0..1]	4
[{TQ2}]	Timing/Quantity Order Sequence	X		4
{]	--- TIMING_QTY end			
[CTD]	Contact Data	X	[0..0]	11
[{	--- OBSERVATION begin			
OBX	Observation Result	R	[1..1]	7
[{NTE}]	Notes and comments			2
}]	--- OBSERVATION end			
[{FT1}]	Financial Transaction	X	[0..0]	6
[{CTI}]	Clinical Trial Identification	X	[0..0]	7
[{	--- SPECIMEN begin			
SPM	Specimen	X	[0..0]	7
[{OBX}]	Observation related to Specimen	X	[0..0]	7
}]	--- SPECIMEN end			
}	--- ORDER_OBSERVATION end			
}	--- PATIENT_RESULT end			
[DSC]	Continuation Pointer	X	[0..0]	2

continue until an explicit stop transaction is received. Each *DOF* is capable of supporting one or more subscriptions from a *DOC*.

Table 5.4: [PCD-02] Messages

QSB	[PCD-02]: QSB^Z02^QSB_Q16 Message	Usage
MSH	Message Header	R
QPD	Query Parameter Definition	R
RCP	Response Control Parameter	R
[DSC]	Continuation Pointer	CE

According to IHE PCD TF (TF-2, Section 3.2.6.1), the event triggering a subscription message on the *DOC* side is Z02. From the message structure point of view, the [PCD-02] transaction is conducted by the HL7 QSB (create subscription) message type. The message structure is derived from the QSB_Q16 HL7 v2.5 message structure. However, the HL7 standard does not associate any trigger event Z02 to the QSB message type, the only possible triggers associated with this message type being Z83 (which stands for ORU Subscription) and Q16 (which stands for QSB create subscription).

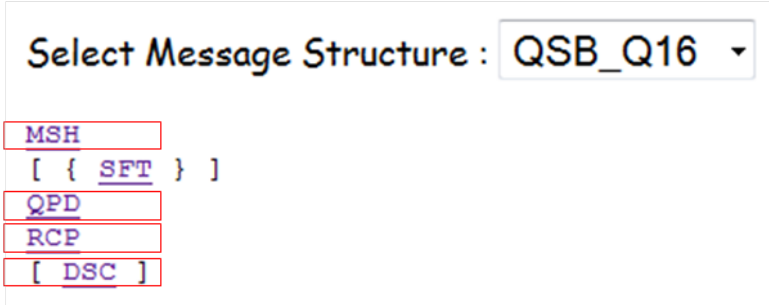


Figure 5.5: HL7 v2.5 QSB_Q16 Message Structure

Discussion. The QSB_Q16 message structure, as described in HL7 messaging standard version 2.5 [HL703] is different to the structure defined by IHE and presented in Table 5.4. Figure 5.5 shows the whole sequence of segments in the message structure as specified in HL7 standard [HL703]. The segments marked with a box are also kept in the PCD TF. The optional sequence of STF segments ("[{STF}]") was removed from the [PCD-02] transaction (Table 5.4) even though it could appear in HL7 messages compliant with the original HL7 QSB_Q16 message structure.

The *Subscribe to PCD Data* transaction is characterised by an *immediate query mode*, i.e., at the receipt of the subscription message no acknowledgement message (ACK) will be sent. Instead, the immediate sending of the *Communicate PCD Data* messages fulfilling the subscription constraints occurs as a response to the [PCD-02] initiator.

5.1.2 Test System

DEC integration profile is the first case study where the generic TTCN-3 test system introduced in the previous chapter was customized and applied. This case study is the main outcome of the TestNGMed research project, which had as its main research goal to investigate interoperability

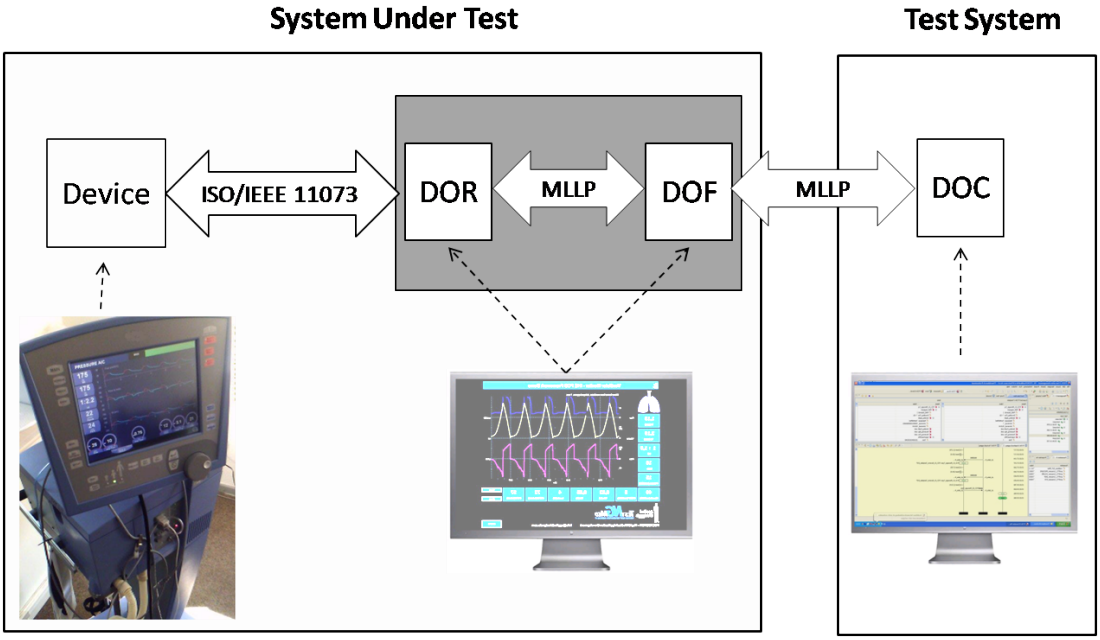


Figure 5.6: TTCN-3 Test Bed Setup for DEC

issues in HISs and how current testing technologies can cope with the challenges of revealing such issues.

The test system supports different *interaction scenarios* derived from the specification of the DEC. These *interaction scenarios* were provided by sepp.med GmbH., one of the TestNGMed project's partners. These scenarios are generated with the .getmore which implements the TTCN-3 generation algorithms proposed in this thesis.

As TTCN-3 IDE and execution environment, the TTworkbench tool commercialised by Testing Technologies has been used [TTE09].

5.1.2.1 Test Bed Description

As proof of concept, within the project, a test bed was created. The test bed consists of one instance of each actor type described in the DEC integration profile. The test bed is presented in Figure 5.6. In this particular set-up, each DEC *actor type* is instantiated only once; therefore, the experimented *interaction scenarios* are based on this configuration.

As shown in the figure, the SUT consists of one DOR and one DOF *actor*. Both actors are implemented within the Polybench tool, provided by another project partner, namely Applied Biosignals GmbH. Polybench is a software system, designed for analysing and processing signals from physiological sources, i.e., PCD devices, and for developing measurement protocols. Among the sources that can be supported by the DOR interfaces within Polybench, the following categories of devices can be connected: ventilator, heart and lung monitor, cardiac output monitor, infusion pump, cerebral function monitor. For the specific experiments conducted along the case study, an AVEA patient lung simulator produced by Core Fusion [CaF10] was used.

The test system plays the role of the DOC *actor* and interacts directly with the DOF. In the sup-

ported interaction scenarios, the PCD subscription mechanism is tested. For different subscription requests, the DOF actor forwards to DOC the PCD data received from the AVEA ventilator via DOR. The group of AVEA ventilator, DOR and DOF is considered an SUT as a whole.

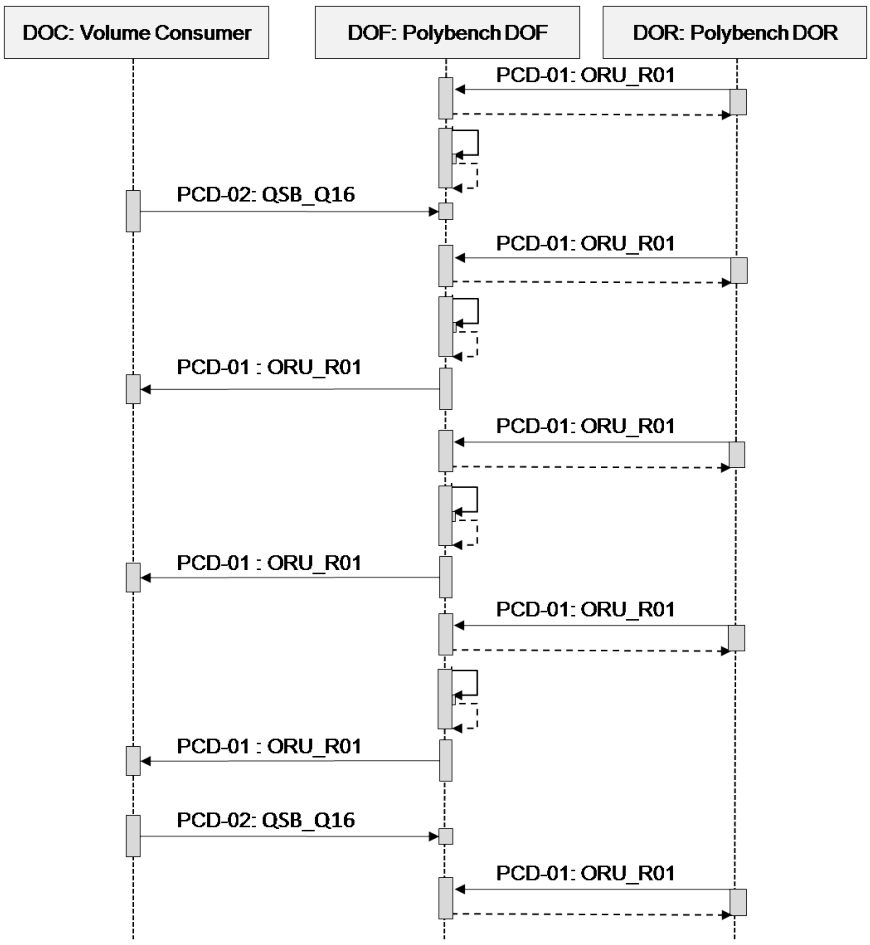


Figure 5.7: Example of a DEC Interaction Scenario

5.1.2.2 Results and Evaluation

5.1.2.2.1 DEC SUBSCRIBE_UNSUBSCRIBE Interaction Scenario

A set of *interaction scenarios* were ran using the test configuration described above. An example is shown in Figure 5.7. In this particular interoperability test, the goal is to test whether, for a certain subscription made by the DOC, the DOF *actor* is capable of forwarding the PCD data that it receives from the DOR *actor* to that *Consumer*.

The scenario steps are:

- *first step*: the *Consumer* sends a QSB_Q16 HL7 *message structure* for subscribing to the DOF for receiving different breath parameters like: airway pressure (Paw), mean airway pressure (Pmean), Peak pressure (Ppeak) expiratory tidal volume (Vte), Inspiratory time

(Ti), fraction of inspired oxygen (FiO2), the ratio of the duration of inspiration to the duration of expiration (I:E Ratio), etc.

- *second, third and fourth step*: the *Filter* sets up the filtering mechanism according to the subscription predicate indicated in the *Consumer's* subscription. After the subscription, the DOF starts checking whether [PCD-01] data (continuously) received from the DOR matches the requirements read from the [PCD-02] message. If a [PCD-01] fits into those constraints, the DOF sends to the DOC that [PCD-01] data.

- *fifth step*: the *Consumer* waits for three (this is configurable) [PCD-01] messages and afterwards it unsubscribes from the DOF. The unsubscription is similar to a subscription and is realized with the same [PCD-02] transaction; the difference is a coded value "D" for the "Action Code" field in the QPD *segment*.

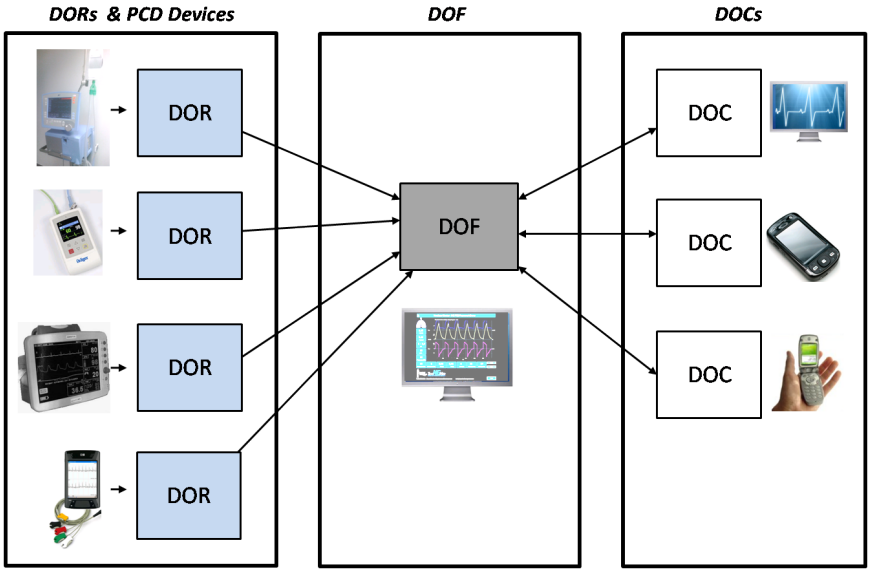


Figure 5.8: DEC Complex Setup

From the perspective of the *message exchange patterns* matching the communication between the TTCN-3 DOC *actor* and the SUT DOF *actor*, this scenario corresponds to a *<Request - Deferred Responses> message exchange pattern*. The unsubscription corresponds to *<Information - No Response> message exchange pattern*. The SUT internal communication between its two *actors*, DOF *actor* and DOR *actor* corresponds to *<Information - Immediate ACK Response> message exchange pattern*.

These *interactions* can be combined in more complex sequences. An example of a more complex set-up which involves multiple DORs and multiple DOCs connected to the same DOF is illustrated in Figure 5.8. Further *interaction scenarios* include the resubscription with new predicates, multiple subscriptions of the same DOC, and even parallel subscriptions from more than one DOCs.

A prerequisite of this test is that the DOF has already stored the ID of that particular *Consumer* in its internal list of registered *Consumers*.

5.1.2.2.2 TestNGMed Project Results

The main result of this case study is the prototypical realisation of the HL7 v2.5/TTCN-3 test framework and its application in a real test set-up. The designed *interaction scenarios*, including SUBSCRIBE_UNSUBSCRIBE, conceptually prove the feasibility of the proposed generic interoperability test architecture and the suitability of the TTCN-3 test design.

The SUBSCRIBE_UNSUBSCRIBE scenario may uncover flaws in the *Filter* implementation with respect to *Consumers*' subscriptions. Based on the interaction diagram presented in Figure 5.7, some of the interoperability test objectives can be identified:

- DOC subscription: validate a subscription transaction with valid/invalid *Consumer* ID.
- DOC receives [PCD-01] data: validate that after the subscription, the *Consumer* receives data from the *Filter* according to the subscription predicate.
- DOC unsubscription: validate an unsubscription with valid/invalid *Consumer* ID.
- DOR data sending frequency: validate that the *Reporter* sends particular [PCD-01] data to the *Filter* with the correct frequency.

Along the TestNGMed project, some interoperability and conformance issues, with respect to [PCD-01] and [PCD-02] have been revealed by the TTCN-3 test system which helped the DOF's functionality (which was also developed during the project) to improve and comply with the IHE DEC technical framework specification.

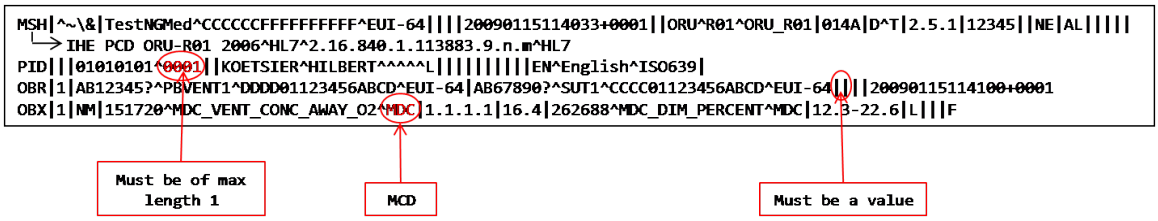


Figure 5.9: Example of a [PCD-01] Message

An example of a [PCD-01] message that is not compliant with the IHE PCD TF is shown in Figure 5.9. This message was received by the *TTCN-3 Consumer* from the *Filter* in one of the incipient phases of the DOF development.

Based on this trace, different conformance issues were uncovered by the *TTCN-3* test system on the DOF side. Here are some examples:

- *not respected coded values*: the third component of the third field of the OBX *segment* (OBX-3) must have a value taken from the list of values available in the HL7 v2.5 Table 0396. Instead, the value "MCD" is used. A correct value close to the "MCD" is "MDC", and the similarity of these two values may indicate a possible typing fault that appeared while implementing the DOF. Thanks to the capability of the *TTCN-3* type system of capturing semantic information by restricting some values to a limited set of values, i.e., coded values, this conformance issue was revealed.

- *not respected optionality*: the fourth *field* of the OBR *segment* (OBR-4) is a required *field*. Instead, the [PCD-01] message trace does not contain any value in that position. The TTCN-3 *Consumer* detected this problem thanks to the mechanism of expecting an SUT reaction using a type hypothesis. This type encapsulates the optionality characteristic to each element in the message.

- *not respected length restriction*: the second *component* (Check Digit CX-2) of the third *field* of the PID *segment* (PID-3) must be of type ST with maximum length 1. Instead, the present value is "0001" which exceeds this length.

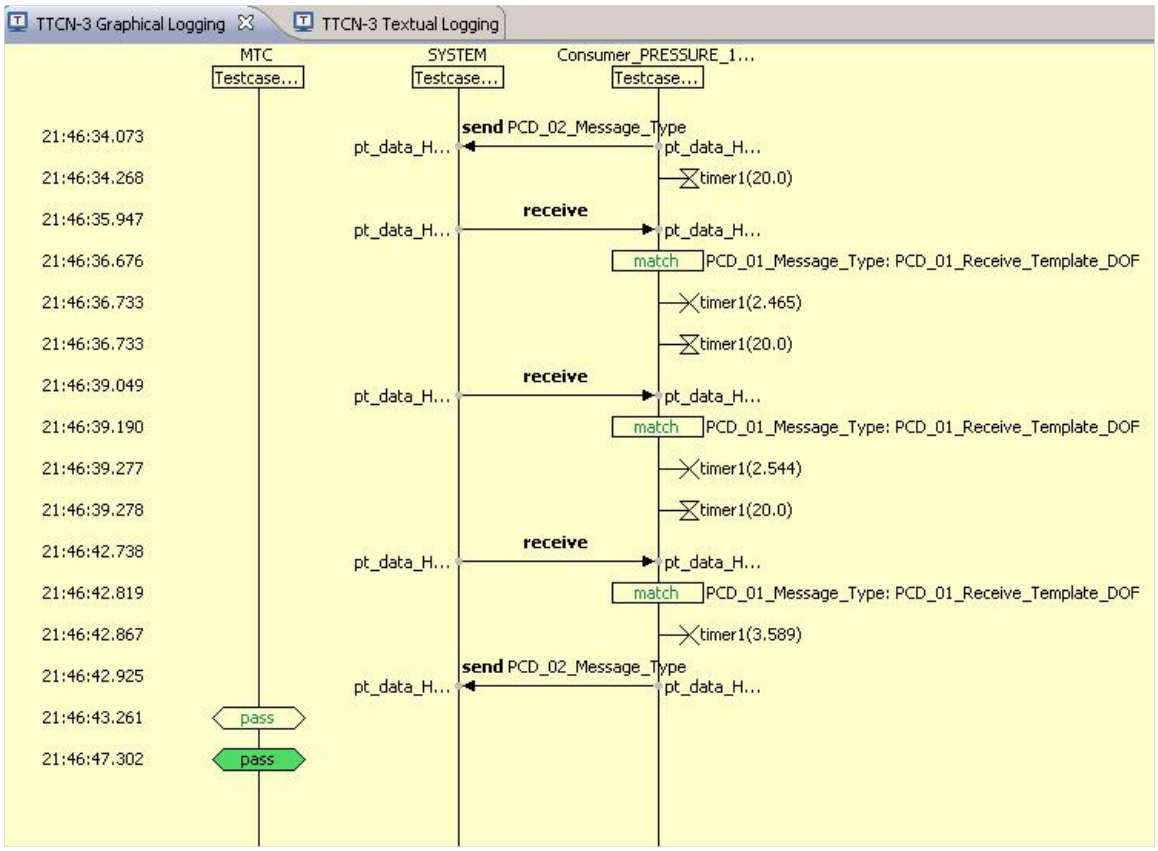


Figure 5.10: DEC SUBSCRIBE_UNSUBSCRIBE Interaction Scenario - Run Example: Success

With this configuration, TTCN-3 test system emulates only the *Consumer*, it is impossible to establish whether the DOR's implementation contains the same flaws when producing [PCD-01] data as the DOF when forwarding these data to the *Consumer*. To be able to investigate the DOR's functionality, the TS should simulate the DOF role. These issues discovered in the SUT functionality were recognised as conformance issues and fixed accordingly. An example of a smooth run is shown in Figure 5.10 which contains the graphical logging produced by the TTworkbench tool. The SYSTEM component displays the activity of the SUT. The right lifeline shows the behaviour of the TS, i.e., the DOC. On this lifeline, the three **match** statements correspond to the three [PCD-01] messages sent by DOF to DOC.

5.2 IHE IT Infrastructure Domain

The second case study belongs to the domain of IHE ITI [ITI09a]. This area supplies the infrastructure for sharing healthcare information. The IT Infrastructure Technical Framework (ITI TF) addresses specific implementations of established standards to achieve integration goals that encourage appropriate sharing of medical information to support better patient care.

ITI Technical Framework. The ITI TF improved with each published version. The version we refer to is revision 6.0 from August 2009 for the final text. The ITI TF identifies a subset of the functional components of the healthcare enterprise, called IHE actors, and specifies their interactions in terms of a set of coordinated, standards-based transactions. The first volume (ITI TF-1)[ITI09b] provides a high-level view of IHE functionality and presents the integration profiles as functional units where the transactions and actors emphasise their capacity to address specific IT Infrastructure requirements. Volumes 2a, 2b and 2x of the ITI TF provide detailed technical descriptions of each IHE transaction used in the ITI integration profiles. Volume 3 comprises specifications used by multiple transactions. The referred ITI TF version includes integration profiles developed in the previous revision versions, e.g., Retrieve Information for Display (RID), Enterprise User Authentication (EUA), Patient Identifier Cross-referencing (PIX), as well as new profiles such as Patient Demographics Query (PDQ), Audit Trail and Node Authentication (ATNA), etc. This section exemplifies the concepts of automated interoperability testing for the PIX integration profile whose detailed description is comprised mainly in volume 2a [ITI09c] of the ITI TF.

Relationship to Standards. The ITI TF distinguishes functional components of a distributed healthcare environment (referred to as IHE actors), solely from the point of view of their interactions in the healthcare enterprise. The current version, addressed in this thesis, defines a coordinated set of transactions based on ASTM International, DICOM, HL7 SDO, Internet Engineering Task Force (IETF), ISO, Organisation for the Advancement of Structured Information Standards (OASIS) and World Wide Web Consortium (W3C) standards. With the increase of the covered areas by the IHE initiative, transactions based on other standards may be included as required. The PIX profile includes as underlying standard solely HL7 messaging standard version 2.5 [HL703] and version 2.3.1 [HL799].

The Context of the Work. This work has been carried out in the context of a joint project between the Fraunhofer FOKUS Institute, MOTION department, [MOT10] and ETSI, Centre for Testing and Interoperability (CTI) [CTI10]. The main target of the project was to demonstrate the feasibility of the approach and to apply the test methodology based on TTCN-3 test technology to test PIX integration profile implementations at the Connectathon 2010 [Con10] plug-in event. The IHE Connectathon is the largest healthcare IT industry interoperability testing event, where yearly over one hundred IT companies jointly test their products for interoperability. Additionally, IHE Connectathon organizers provide certificates for the IT healthcare solutions. The main target of the participation at the 10th annual IHE European Connectathon (April 12-16, 2010 at the Cité Mondiale in Bordeaux, France) was to reveal that an automated interoperability test framework is possible and could be adopted on a large scale. The TTCN-3 technology used to build such a test framework not only satisfied the requirements for automation for interoperability testing, but also captured the interest of many vendors and raised discussions on possible integration with the test management tool, named Gazelle [Gaz10] used at Connectathon event. An additional advantage of the adopted TTCN-3 test technology is the simultaneously check of the interoperability and conformance of healthcare informational systems compliant with IHE profiles.

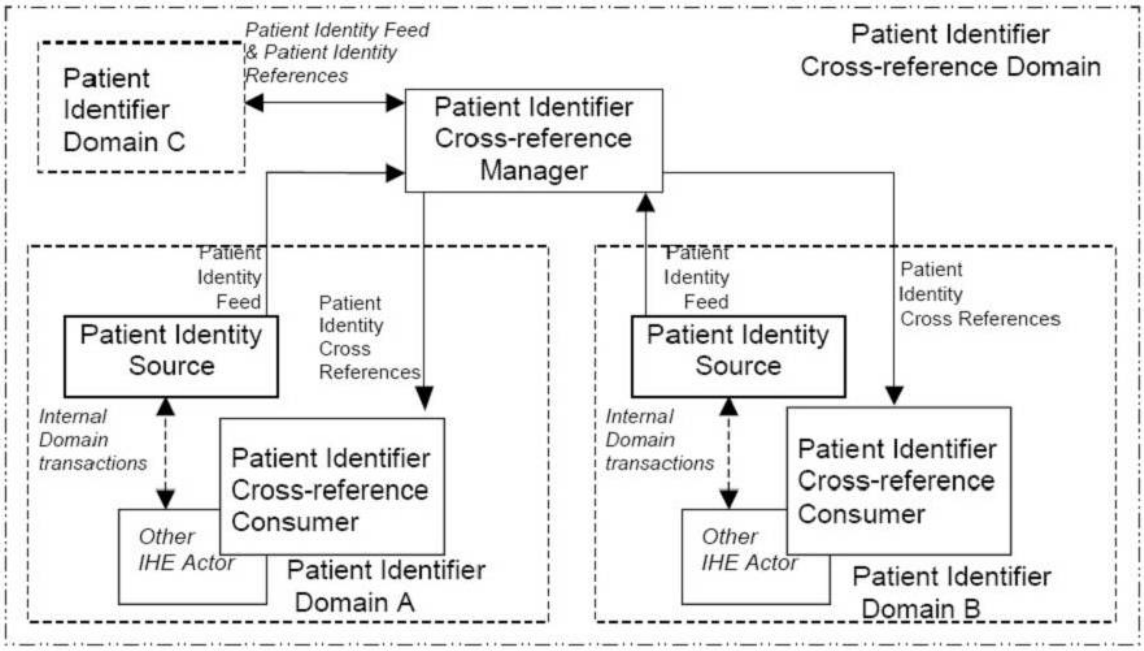


Figure 5.11: Process Flow with Patient Identifier Cross-referencing

5.2.1 PIX Integration Profile

Part of the ITI domain, the IHE PIX integration profile, standing for *Patient Identifier Cross-referencing Integration Profile*, targets healthcare enterprises of a broad range of sizes such as hospital, clinic, physician office, etc. The PIX integration profile defines how to resolve a patient identifier from one local domain to other connected systems. In other words, it enables the cross-referencing of patient identifiers from multiple patient identifier domains by:

- transmitting patient identity information from an *Identity Source* to the *Patient Identifier Cross-reference Manager*.
- providing the ability to access the list(s) of cross-referenced patient identifiers either via a *query / response* or via an *update notification*.

This integration profile does not define any specific enterprise policies or cross-referencing algorithms, it simply defines the behaviour for each single actor by specifying the above transactions among specific actors. As a consequence, this integration profile provides the necessary interoperability while maintaining the flexibility for each enterprise regarding the algorithm or cross-referencing policy.

The diagram represented in Figure 5.11, as presented in the ITI TF [ITI09b] (Section 5) shows the intended goal of this profile. As illustrated in the diagram, there are two types of *Identifier Domains*: *Patient Identifier Domain* and *Patient Identifier Cross-reference Domain*.

A *Patient Identifier Domain* represents a single system or a set of interconnected systems that all share the same identification scheme, i.e., an identifier and an assignment process to a patient, and the same issuing authority for patient identifiers. Additionally, it is characterised by the following properties:

- a set of policies that describe how identities will be defined and managed according to the specific requirements of the domain.
- an administration authority for administering identity related policies within the domain.
- a single system, known as a *Patient Identity Source*, that assigns a unique identifier to each instance of a patient-related object as well as maintaining a collection of identity traits.
- ideally, only one identifier is uniquely associated with a single patient within a given *Patient Identifier Domain*, but a single *Patient Identity Source* actor may assign multiple identifiers to the same patient and communicate this fact to the *Patient Identifier Cross-reference Manager* actor.
- an identifier of the *Patient Identifier Domain*, known as *Assigning Authority*, that is unique within a *Patient Identifier Cross-reference Domain*.
- other systems in the *Patient Identifier Domain* rely upon the identifiers assigned by the *Patient Identity Source* system of the domain to which they belong.

A *Patient Identifier Cross-reference Domain* consists of a set of *Patient Identifier Domains* known and managed by an entity called the *Patient Identifier Cross-reference Manager* actor. The *Patient Identifier Cross-reference Manager* actor is responsible for creating, maintaining and providing lists of identifiers that are aliases of one another across different *Patient Identifier Domains*. The *Patient Identifier Cross-reference Manager* actor is not responsible for improving the quality of identification information provided to it by the *Identity Source Actors*.

5.2.1.1 Actors

Figure 5.12 as defined in [ITI09b] (Section 5.1) shows the actors directly involved in the PIX integration profile and the relevant transactions between them. The diagram presents only the actors described in the PIX profile, other actors that may be indirectly involved due to their participation in other related profiles are not shown.

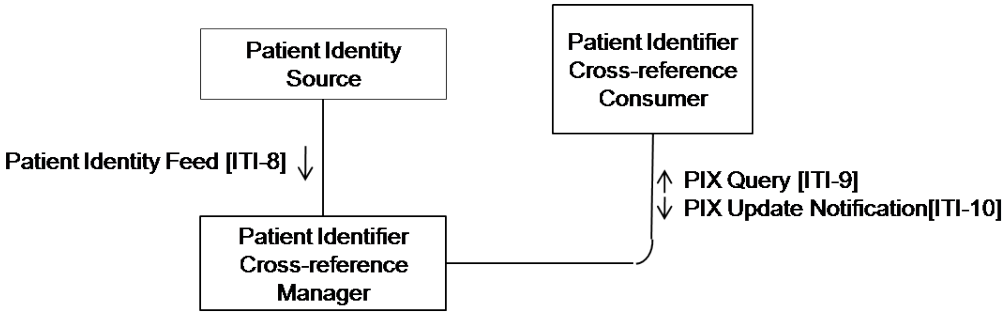


Figure 5.12: Patient Identifier Cross-referencing Actor Diagram

Patient Identity Source. The *Patient Identity Source* (*Source*) (PIX Source)[4] is a single system in any subunit of a healthcare facility, e.g., laboratory, Intensive Care Unit (ICU) , etc., which can generate an internal patient ID for a specific purpose. These subunits of the healthcare facility map

[4]The name of the *Patient Identity Source* actor will be used interchangeably with *Source* or *PIX Source*.

to the concept of the *Patient Identifier Domain* introduced above and the healthcare facility plays the role of the *Patient Identifier Cross-reference Domain*. For a given *Patient Identifier Domain* there shall be one and only one *Patient Identity Source* actor, but a given *PIX Source* actor may serve more than one *Patient Identifier Domain*. All objects associated with a single patient such as reports, laboratory orders, etc., have to be linked with the same internal patient ID within a domain. The *PIX Sources* belonging to different *Patient Identifier Domains*, which are part of the same *Patient Identifier Cross-reference Domain* have their own notion of internal patient identity. Additionally, this internal ID needs to be mapped to the medical record number (MRN) of the patient, which is generated by the main ADT system in that healthcare facility and is used as the patient identity.

Patient Identifier Cross-reference Manager. The internal ID generated by a *PIX Source* is transferred to the *Patient Identifier Cross-reference Manager* (*Manager*)[5] by using the *Patient Identity Feed*[6] transaction identified by IHE as of being of type *ITI-8*. Once the *Manager* receives the *Feed* transactions, it performs its internal logic to determine which, if any, patient identifiers can be "linked together" as being the same patient based on the corroborating information included in the *Feed* transactions it has received. The cross-referencing process (algorithm, human decisions, etc.) is performed within the *Patient Identifier Cross-reference Manager* and is outside the scope of IHE.

Patient Identifier Cross-reference Consumer. Within a *Patient Identifier Domain* it may be necessary to get information associated with a patient (that the domain knows by its internal patient ID) from another *Patient Identifier Domain*. For example, a clinician from an ICU wants to review the glucose level of a patient which is included in a laboratory report stored in the main laboratory system (LIS) of the hospital. To receive this, a request for information using the internal patient ID including the identifier of the *Patient Identifier Domain* (known as *Assigning Authority*) is sent by the querying *Patient Identifier Domain* to the desired *Patient Identifier Domain*. Upon receipt of the request, the questioned *Patient Identifier Domain* determines that the request is for a patient outside of its own domain. It then requests a list of patient ID aliases corresponding to the patient ID from the *PIX Manager*. Such a system within a *Patient Identifier Domain* sending requests to the *PIX Manager* for resolving patient IDs is called the *Patient Identifier Cross-reference Consumer* (*Consumer*)[7]. Having resolved or linked the Patient ID, the *Manager* returns a list of Patient ID aliases to the *Consumer*. This list helps in retrieving the information for the desired patient within the requested *Patient Identifier Domain*. Finally, this information is returned to the querying *Patient Identifier Domain*.

An example of these three types of actors and interactions between them is represented in Figure 5.13 and detailed in the ITI TF [ITI09b] (page 42).

Another feature of the *PIX Consumer* is the capability to receive *Update Notifications* from the *PIX Manager*. To receive such notifications, the *PIX Consumer* can be configured to be informed about the patient identifier aliases or other changes. This notification is done in order to allow systems that are aware of multiple identifier domains to maintain synchronisation with patient identifier changes that occur in any of the identifier domains that they are aware of.

[5] The name of the *Patient Identifier Cross-reference Manager* actor will be used interchangeably with *Manager* or *PIX Manager*.

[6] The name of the *Patient Identity Feed* transaction will be used interchangeably with *Feed* or *PIX Feed* transaction.

[7] The name of the *Patient Identifier Cross-reference Consumer* actor will be used interchangeably with *Consumer* or *PIX Consumer*.

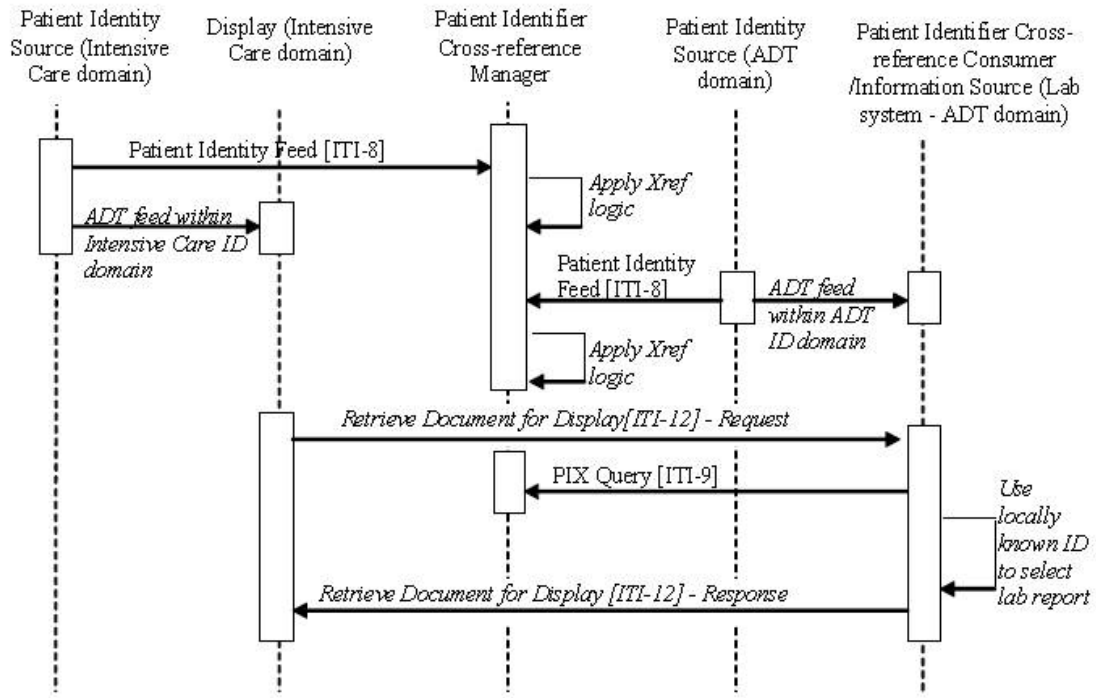


Figure 5.13: Multiple ID Domains in a Single Facility Process Flow in PIX Profile

5.2.1.2 Transactions

Table 5.5 lists the transactions for each actor directly involved in the PIX profile. Each implementation claiming to support this integration profile must perform the required transactions labelled "R" in the table. Transactions labelled "O" are optional. The volume 2a [ITI09c] of the ITI TF presents in detail these transactions.

Table 5.5: PIX Actors and Transactions

Actors	Transactions	Optionality
Patient Identity Source	Patient Identity Feed [ITI-8]	R
Patient Identifier Cross-reference Consumer	PIX Query [ITI-9]	R
	PIX Update Notification [ITI-10]	0
Patient Identifier Cross-reference Manager	Patient Identity Feed [ITI-8]	R
	PIX Query [ITI-9]	R
	PIX Update Notification [ITI-10]	R

Patient Identity Feed [ITI-8] Transaction. Transaction [ITI-8], as indicated in Figure 5.13 (on top, left side), is used by the *Patient Identity Source* and *Patient Identifier Cross-reference Manager* actors. It serves to communicate patient information, including corroborating demographic data after the patient's identity is established, modified or merged or after the key corroborating demographic data has been modified. As underlying standard, the version 2.3.1 of HL7 messaging standards [HL799] was selected. One reason for choosing this version was the broader potential

of *Patient Identity Source* actors capable of participating in other integration profiles associated with this transaction. Secondly, it allows existing ADT actors from within the IHE radiology field, where the version 2.3.1 of HL7 is predominant among implementers, to participate as *PIX Source* actors. As a general remark, the selection of the HL7 version for a certain transaction within an integration profile, is mainly determined by the most predominantly adopted HL7 version among vendor implementations of the actors involved in that transaction.

The *Patient Identity Source* actor uses this transaction to provide a notification to the *Patient Identifier Cross-reference Manager* for any event related to the patient ID: creation, updates, merges, etc. The *PIX Manager* shall only perform cross-referencing logic on messages received from *PIX Source* actors.

Table 5.6: ADT Patient Administration Messages

ADT	**A01, A04, A05, A08 Patient Administration Message**	**Chapter in HL7 2.3.1**
MSH	Message Header	2
EVN	Event Type	3
PID	Patient Identification	3
PV1	Patient Visit	3

With respect to the patient identity management, the following events from a *PIX Source* will trigger one of the *Admit*, *Register* or *Update* messages:

- A01 - Admission of an in-patient into a facility.
- A04 - Registration of an outpatient for a visit of the facility.
- A05 - Pre-admission of an in-patient (i.e., registration of patient information ahead of actual admission).

Changes to patient demographics (e.g., changes in patient name, patient address, etc.) shall trigger the following Admit/Register or Update message:

- A08 - Update Patient Information.

From the message structure point of view, the [ITI-8] transaction is conducted by the HL7 ADT message type. The *PIX Source* actor shall generate the message whenever a patient is admitted, pre-admitted, or registered, or when some piece of patient demographic data changes. Pre-admission of in-patients shall use the A05 trigger event. The message structure is derived from the ADT_A01 HL7 message structure and the required segments are listed in Table 5.6. Detailed descriptions of these segments are provided in volume 2a [ITI09c] of the ITI TF. Along each segment structure, IHE introduces a set of restrictions to the original structure standardised by HL7 version 2.3.1.

Discussion. The ADT_A01 message structure, as described in HL7 messaging standard version 2.3.1 [HL799], is more complex than the structure defined by IHE and presented in Table 5.6. Figure 5.14 shows the whole sequence of segments in the message structure as specified in the HL7 standard [HL799]. Attributes such as optionality ([ ] brackets), repetition ({ } brackets) or substructures (e.g., PROCEDURE, INSURANCE) are clearly specified. To ease the interoperability between systems, IHE reduced the complexity of this message structure by preserving only the outlined mandatory segments: MSH, EVN, PID, PV1.

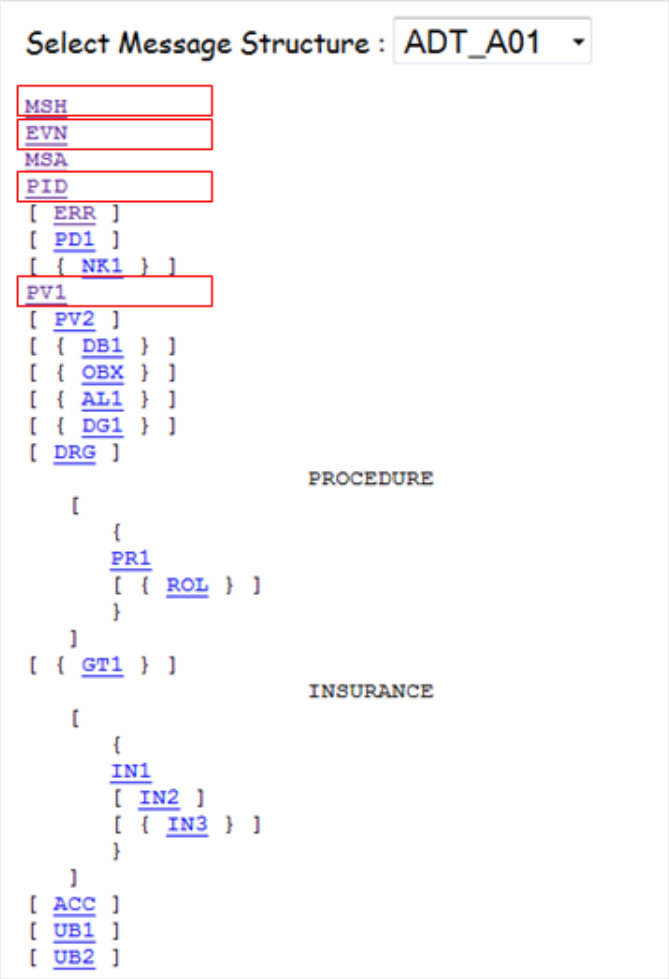


Figure 5.14: HL7 v2.3.1 ADT_A01 Message Structure

When two patients' records identify the same patient in a *Patient Identifier Domain* and are merged, the *Source* shall trigger the following message:

- A40 - Merge Patient - Internal ID.

The message triggered with A40 event is not based on ADT_A01 message structure as those messages triggered with A01, A04, A04 and A08, but on the ADT_A39 message structure standardised by HL7 version 2.3.1. It deals with patient identity *Merge*, different to the other messages triggered by the *PIX Source* which deal with *Admission, Pre-admission, Registration* or *Update* a patient data. The structure of the *Merge* messages adopted by IHE is shown in Table 5.7.

Even though IHE includes these two types of message structures in the same [ITI-8] transaction, one can actually differentiate two types of *Feed* transactions: a) *Admission, Pre-admission, Registration, Update* transactions, whose underlying structure is ADT_A01 message structure (Figure 5.14) and specified in Table 5.6 and b) *Merge* transactions based on ADT_A39 message structure (Figure 5.15) and described in Table 5.7.

In all cases when the *PIX Source* is triggering an ADT message, an acknowledge message (HL7

Table 5.7: ADT A40 Patient Identity Merge Message

ADT	A40 Patient Identity Merge M Message	Chapter in HL7 2.3.1
MSH	Message Header	2
EVN	Event Type	3
PID	Patient Identification	3
MRG	Merge Information	3
[PV1]	Patient Visit	3

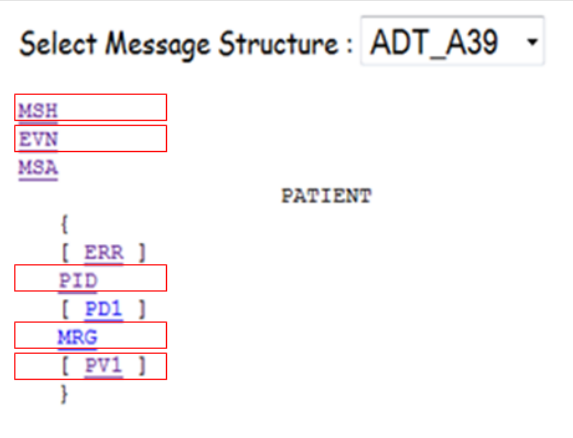


Figure 5.15: HL7 v2.3.1 ADT_A39 Message Structure

ACK message) shall be sent by the receiver of ADT message to its sender. The ACK message is also part of the *Feed* [ITI-8] transaction. Irrespective of the triggering event that conducted to the sending of the ADT message within *PIX Feed* transaction, the ACK message returned to the *PIX Feed* transaction initiator has the same message structure. In this case, IHE does not specify any changes to the general ACK message structure in terms of sequence of segments and it remains as it was defined by the HL7 standard version 2.3.1 (Figure 5.16).

PIX Query [ITI-9] Transaction. Transaction [ITI-9], as indicated in Figure 5.13, is used by the *Patient Identifier Cross-reference Consumer* and *Patient Identifier Cross-reference Manager* actors. It serves to request a list of patient identifiers that correspond to a patient identifier known by the *PIX Consumer*. The query is sent by the *PIX Consumer* to the *PIX Manager*. The *PIX Manager* processes the request and returns a response in the form of a list of corresponding patient identi-

Figure 5.16: HL7 v2.3.1 ACK Message Structure

fiers. In case no other patient identifiers are found, the response contains an empty list. Different than *PIX Feed* transaction, the *PIX Query* transaction selects as underlying messaging standard, version 2.5 of the HL7 messaging standards [HL703]. As indicated in the volume 2a [ITI09c] of the ITI TF (page 55), the reason for selecting this version was that it was considered the most stable version that contained the functionality required by transactions [ITI-9] and [ITI-10].

Table 5.8: QBP Query By Parameter Messages

QBP	Query By Parameter	Chapter in HL7 2.5
MSH	Message Header	2
QPD	Query Parameter Definition	5
RCP	Response Control Parameter	5

As we presented at the beginning of this section, a patient can have different identifiers in different *Patient Identifier Cross-reference Domains*. In general, a *PIX Consumer* actor knows at least a patient identifier associated with one of the *Patient Identifier Cross-reference Domains*. In order to obtain information about the same patient, but within other *Patient Identifier Cross-reference Domains*, the *PIX Consumer* needs firstly to find the patient identifier from the targeting *Patient Identifier Cross-reference Domain*. To achieve this, the *PIX Consumer* will trigger a request message based on the following HL7 trigger event:

- Q23 - Get Corresponding Identifiers.

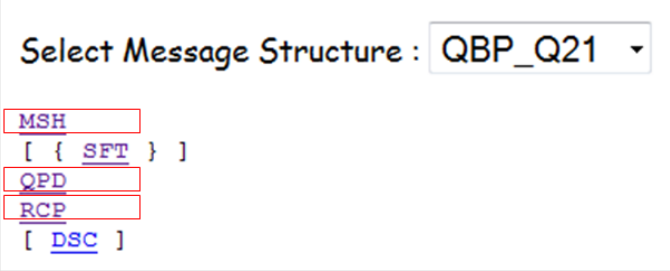


Figure 5.17: HL7 V2.5 QBP_Q21 Message Structure

From the message structure point of view, the [ITI-9] transaction is conducted by the HL7 QBP message type and triggers Q23. The message structure is derived from the QBP_Q21 HL7 message structure and the segments required by the the ITI TF are listed in Table 5.8. Detailed descriptions of these segments are provided in volume 2a [ITI09c] of the ITI TF. Similar to the [ITI-8] used in the *PIX Feed* transaction, along each segment structure, IHE introduces a set of restrictions to the original structure standardised by HL7 version 2.5.

Discussion. The QBP_Q21 message structure, as described in HL7 messaging standard version 2.5 [HL703] is only a bit more complex than the structure defined by IHE and presented in Table 5.8. Figure 5.17 shows the whole sequence of segments in the message structure as specified in the HL7 standard [HL703]. To ease the interoperability between systems, IHE reduced the complexity of this message structure by preserving only the mandatory segments: MSH, QPD, RCP.

Table 5.9: RSP Segment Pattern Response Messages

RSP	Segment Pattern Response	Chapter in HL7 2.5
MSH	Message Header	2
MSA	Message Acknowledgement	2
[ERR]	Error segment	2
QAK	Query Acknowledgement	5
QPD	Query Parameter Definition	5
[PID]	Patient Identification	3

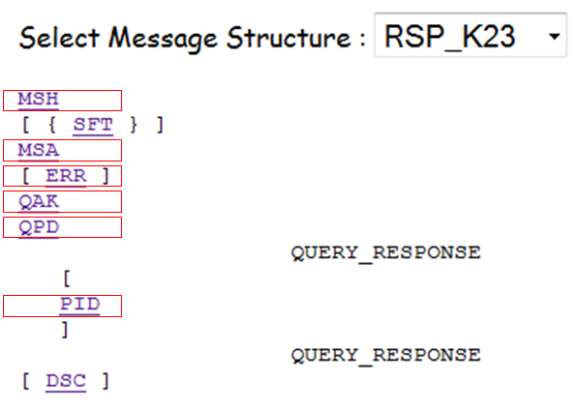


Figure 5.18: HL7 V2.5 RSP_K23 Message Structure

Also part of the [ITI-9] transaction, similar to the ACK message from the [ITI-8] transaction, is the query response message which returns the corresponding patient identifiers. This response is issued by the *Patient Identifier Cross-reference Manager* actor irrespective whether an identifier, a list or no patient identifier was found. The trigger event is:

- K23 - Corresponding patient identifiers.

The message structure of the response is conducted by the HL7 RSP message type and by trigger K23. The message structure is derived from the RSP_K23 HL7 message structure and the segments required by the ITI TF are listed in Table 5.9. These segments are described in more detail in volume 2a [ITI09c] of the ITI TF.

Discussion. The RSP_K23 message structure, as described in HL7 messaging standard version 2.5 [HL703] is only a bit more complex than the structure defined by IHE and presented in Table 5.9. Figure 5.18 shows the whole sequence of segments in the message structure as specified in HL7 standard [HL703]. To simplify the interoperability between systems, IHE reduced the complexity of this message structure by preserving only the following segments: MSH, MSA, ERR, QAK, QPD, PID.

PIX Update Notification [ITI-10] Transaction. Transaction [ITI-10] (Figure 5.13) shall be triggered by the *Patient Identifier Cross-reference Manager* in order to notify the *Patient Identifier Cross-reference Consumer* whenever a change occurs in any of the patient identifiers belonging to *Patient Identifier Domains* of interest to the *Consumer*. The *Patient Identifier Domains* of interest to a *PIX Consumer* are configured and maintained by the *PIX Manager* actor. The *PIX Manager*

may have to issue several notification messages to a *PIX Consumer* to reflect all the changes on the resulting sets of cross-reference patient identifiers belonging to *Patient Identifier Domains* of interest.

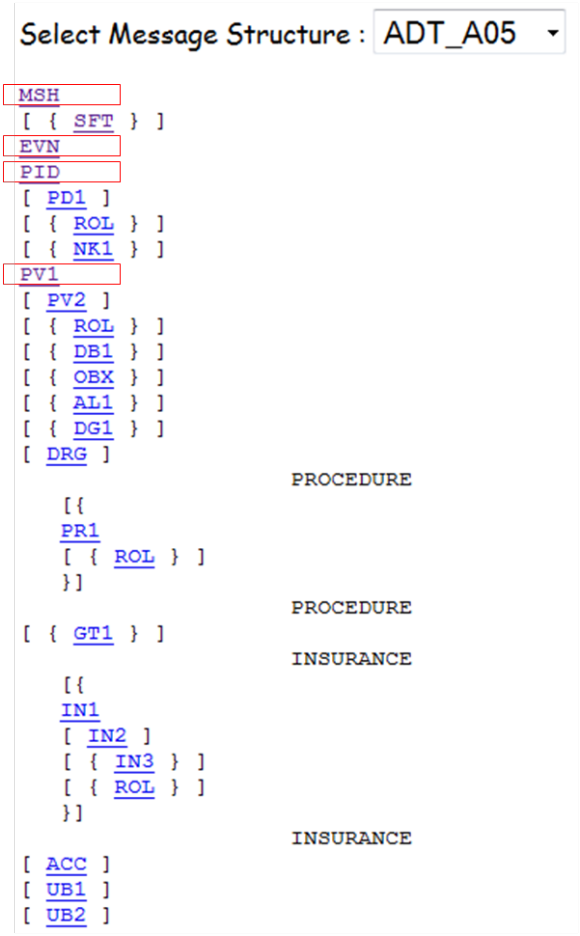


Figure 5.19: HL7 V2.5 ADT_A05 Message Structure

Similar to the *PIX Query* ([ITI-9]) transaction, the *PIX Update Notification* transaction selects as the underlying messaging standard, version 2.5 of HL7 messaging standards [HL703]. Details about this transaction are presented in volume 2a [ITI09c] of the ITI TF (page 67).

The HL7 trigger event used to send update notifications:

- A31 - Update Person Information

The message structure of an update notification message is conducted by the HL7 ADT message type and the trigger A31. The message structure is derived from the ADT_A05 HL7 message structure and the segments chosen in the ITI TF are listed in Table 5.10. These segments are described in more detail in volume 2a [ITI09c] of the ITI TF.

Discussion. The ADT_A05 message structure, as described in HL7 messaging standard version 2.5 [HL703] is more complex than the structure defined by IHE and presented in Table 5.10. Figure 5.19 shows the whole sequence of segments in the message structure as specified

Table 5.10: ADT Patient Administration Update Notification Messages

ADT	Patient Administration Message	Chapter in HL7 2.5
MSH	Message Header	2
EVN	Event Type	3
PID	Patient Identification	3
PV1	Patient Visit	3

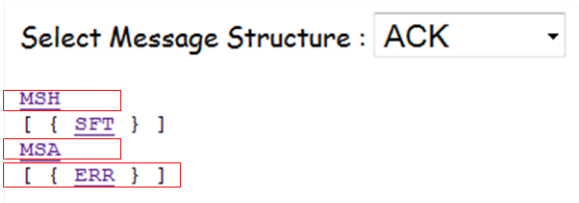


Figure 5.20: HL7 V2.5 ACK Message Structure

in HL7 standard [HL703]. To simplify the interoperability between systems, IHE reduced the complexity of this message structure by preserving only the following segments: MSH, EVN, PID, PV1.

Table 5.11: Common ACK Messages

ACK	Meaning	Usage	Chapter in HL7 2.5
MSH	Message Header	R	2
MSA	Message Acknowledgement	R	2
ERR	Error segment	C	2

In response to an update notification message, the *PIX Consumer* shall send back to *PIX Manager* an acknowledge message (HL7 ACK message). The ACK message is also part of the *Update Notification* [ITI-10] transaction. The volume 2x [ITI09d] of the ITI TF (Section C.2.3, Acknowledgement Modes), gives details about the definition of the ACK message which is based on HL7 messaging standard version 2.5. The ACK message returned to the *PIX Update Notification* transaction initiator has a message structure as presented in Table 5.11. IHE slightly changed the general ACK message structure as defined by the HL7 standard version 2.5 (Figure 5.20).

5.2.2 Test System

The realisation of the TTCN-3 test system for supporting different *interaction scenarios* from the PIX *integration profile* is based on the instantiation of the generic test architecture for HISs presented in the previous chapter.

The specification for the supported PIX *interaction scenarios* was defined by IHE Connectathon 2010 [Con10] organisers and distributed via Gazelle test management system [Gaz10] and made available online before the plug-in event.

The design principles of the *TTCN-3* test specification introduced in this thesis were entirely applied for running the PIX *interaction scenarios*. A tedious task was the adaptation of the HL7 v2.x *TTCN-3* type system to IHE PIX *integration profile* constraints specified in the ITI TF. The work was made more difficult by the IHE specification document itself, which, without a formalism behind, many times lead to misunderstandings and, for clarification reasons, required additional interaction with different authors of the specification. These IHE constraints also impacted the generic HL7v2.x CoDec implementation to be harmoniously coupled with the modified TTCN-3 type system.

An important problem appeared during the implementation of the test system itself due to the fact that, before the participation to the Connectathon, no real SUT systems were available. This implied that the test system itself could not have been validated against real SUT *actors* within various *interaction scenarios*. The solution adopted was to implement also in TTCN-3 the behaviour of the SUT *actors* involved in the interaction scenarios proposed for the Connectathon plug-in event. This way, both sides, the TS and the TS-SUT simulators could have been monitored and the correctness of test flows validated.

Similar to the previous case study, as TTCN-3 IDE and execution environment, the TTworkbench tool commercialised by Testing Technologies has been used [TTE09].

5.2.2.1 Test Bed Description

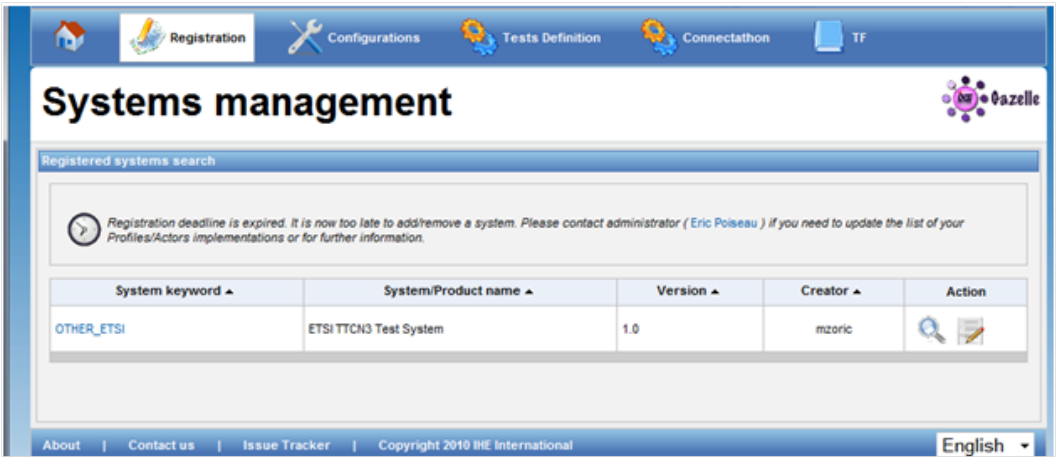


Figure 5.21: TTCN-3 Test System Registration into Gazelle

The test management system used during Connectathon 2010 [Con10] is Gazelle [Gaz10], a web based application. This system was used to store all test execution traces and manage the test

sessions along the event. A test session represents an instance of a test description (equivalent to an *interaction scenario*) between two or more interoperability *actors* and it contains the final test execution trace, which confirms that the involved systems are interoperable.

6675	Peer To Peer Test	PIX_FEED	2010-04-15 13:25:28.937					
				OTHER_ETSI	J5	PIX	PAT_IDENTITY_SRC	NONE
				EHR_Tiani - Cisco	E9	PIX	PAT_IDENTITY_X_REF_MGR	NONE
2 Test participants								
4573	Peer To Peer Test	PIX_FEED	2010-04-13 10:23:22.767					
				OTHER_ETSI	J5	PIX	PAT_IDENTITY_SRC	NONE
				PIX_X_REF_MGR_Nextgate	J4	PIX	PAT_IDENTITY_X_REF_MGR	NONE
2 Test participants								
7191	Peer To Peer Test	PIX_CLIENT_Upd_Notif_Option	2010-04-16 11:18:41.547					
				OTHER_ETSI	J5	PIX	PAT_IDENTITY_CONSUMER	PIX_UPDATE_NOTIFICATION
				OTHER_GE_ICW	E3	PIX	PAT_IDENTITY_X_REF_MGR	NONE
				OTHER_ETSI	J5	PIX	PAT_IDENTITY_SRC	NONE
3 Test participants								
6457	Peer To Peer Test	PIX_CLIENT	2010-04-14 17:55:04.906					
				OTHER_GE_ICW	E3	PIX	PAT_IDENTITY_X_REF_MGR	NONE
				OTHER_ETSI	J5	PIX	PAT_IDENTITY_CONSUMER	NONE
2 Test participants								
7169	Peer To Peer Test	PIX_CLIENT_Upd_Notif_Option	2010-04-16 10:59:53.157					
				OTHER_ETSI	J5	PIX	PAT_IDENTITY_CONSUMER	PIX_UPDATE_NOTIFICATION
				EHR_ITH-ICOSERVE_SIEMENS_1	J11	PIX	PAT_IDENTITY_X_REF_MGR	NONE
				OTHER_ETSI	J5	PIX	PAT_IDENTITY_SRC	NONE
3 Test participants								

Figure 5.22: Connectathon 2010 - Example of Test Sessions with TTCN-3 Test System

The TTCN-3 test system was registered a few weeks before the Connectathon event such that any partner could see the available *actors* that the test system can emulate. As TTCN-3 IDE and test execution environment, the TTworkbench tool commercialised by Testing Technologies is used [TTE09]. The screen capture in Figure 5.21 shows the TTCN-3 Test System with the "ETSI TTCN-3 Test System" identifier registered into Gazelle.

All executed tests have been registered within Gazelle and they were evaluated by the monitors (i.e. IHE representatives for validating the interoperability criteria). The screen capture in Figure 5.22 presents some of the tests executed with the TTCN-3 test system which were registered into Gazelle. These tests have been described above and their results are discussed in the following subsection. As a general remark, most of the test sessions revealed at least one issue in the tested systems which obviously had to be investigated. In all situations, the interoperability issues were fixed accordingly and tests passed correctly all test steps.

In the executed interoperability *interaction scenarios*, the TTCN-3 test system emulated the *PIX Source* and *PIX Client actors* and it has been executed against several implementations of different *PIX Manager actors*. According to Gazelle requirements, at least three correct runs per test description against three different counterpart *actors* should be available in order to pass a test. The status of the test executions with the TTCN-3 test system indicates that this system passed all types of tests in which it has been involved. The certification status is reflected in Figure 5.23 and Figure 5.24.

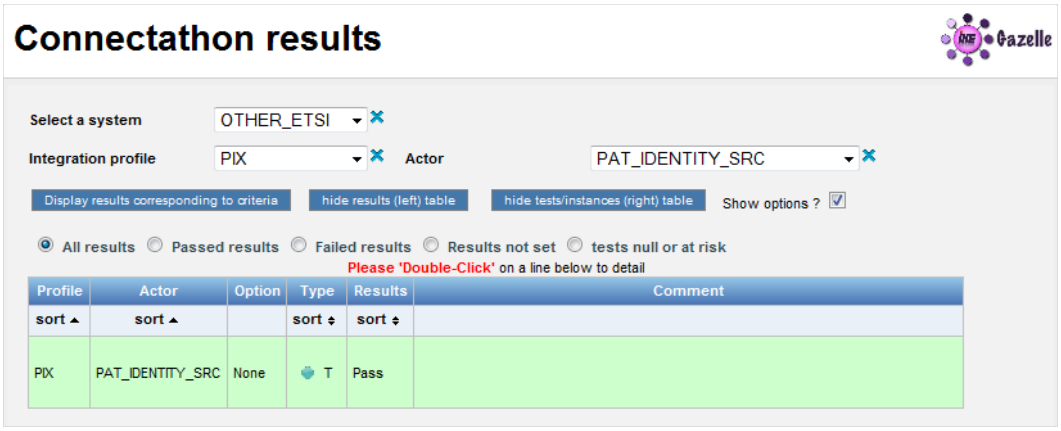


Figure 5.23: Connectathon 2010 Results for the PIX TTCN-3 Test System (I)

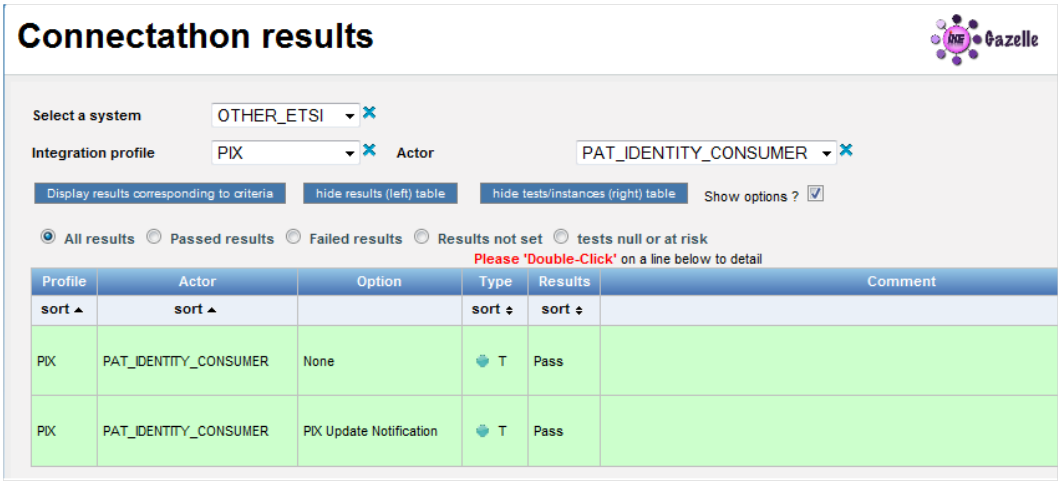


Figure 5.24: Connectathon 2010 Results for the PIX TTCN-3 Test System (II)

5.2.2.2 Results and Evaluation

In the following, examples of PIX *interaction scenarios* used at Connectathon are presented. As a general rule, all Connectathon *interaction scenarios* contained no more than one instance of each PIX *actor type*. Furthermore, for each case, beside the flow of *interactions*, the *actors* emulated by the TTCN-3 test system are clearly specified.

5.2.2.2.1 PIX_Seed_Mgr - Preamble for all Interaction Scenarios

In order to be able to perform the required *interaction scenarios* at Connectathon, a preamble step must be executed on the *PIX Manager actor's* side. The purpose of this test is to place a set of common patients in each *PIX Manager* available, to provide a common basis for testing. *PIX Manager* systems run only one instance of this test and the list of the patients is available in advance before the Connectathon event starts, e.g., published on a wiki page. Each of these patient records

was registered with a *global* patient ID that has the master *Assigning Authority* established for Connectathon. These demographics include several reference patients. One is the "well-known" FARNSWORTH ^ STEVE, which is used frequently in the test scenarios. The list also includes one woman patient for each *Source*, who has the last name matching the Gazelle system keyword of the *Source* and who has the first name Mary, e.g., patient "EHRxyzmedical ^ Mary", which will be used to create a special test case.

Additionally, each *PIX Source* is required to have its own *Assigning Authority* assigned in order to identify that *Source* system. That *Assigning Authority* will consist of a text label and an OID. Different IHE *integration profiles* treat the HL7 *Assigning Authority* (PID 3.4) differently. The *Assigning Authority* has three subcomponents: *Namespace ID* (text string), *Universal ID* (OID) and *Universal ID type*. The Connectathon requirements for the *Assigning Authority* impose: a) the usage of the string “ISO” when an OID is used and b) having populated either the *Namespace ID*, or both *Universal ID* and *Universal ID type*, or all three subcomponents.

Later *interaction scenarios* will use the same demographics of patients, but with *local* patient IDs and the Assigning Authorities given to each *PIX Identity Source* at the Connectathon.

On the TTCN-3 test system side, the aforementioned requirements related to patient demographics have to be regarded from the perspective of each *actor* involved. For example, when emulating the *PIX Source* in TTCN-3, an *Assigning Authority* must be chosen, while, when emulating the *PIX Manager*, the patient’s records that each *Manager* has to seed must be translated in TTCN-3 **templates** and made available in different test configurations.

5.2.2.2.2 PIX_FEED Interaction Scenario

The form chosen by the IHE Connectathon organisers to describe the PIX_FEED *interaction scenario* is presented in Figure 5.25. This *interaction scenario* is characterised by the participation of two *actors* from the PIX *integration profile*.

Description. The purpose of the test is to feed the *PIX Manager* with *local* patient identifiers from a *PIX Source*. The *PIX Manager* will cross-reference the fed patient information with information that is already available in its database. The test implies that the *PIX Source* will send its local patient identifier (patient ID) and *Assigning Authority* for three patients.

The *interaction scenario* involves the following steps on the *PIX Source* side:

- *first step*: admit, pre-admit or register a first patient, FARNSWORTH ^ STEVE.
- *second step*: admit, pre-admit or register a second patient, SINGLETON ^ MARION (this is a patient’s maiden name)
- *third step*: admit, pre-admit or register a third woman patient who has the last name matching the name of the system’s keyword name of the *PIX Source* and its first name of MARY, e.g., patient EHRxyzmedical ^ MARY.
- *fourth step*: update the third patient’s first name to a new name, i.e., MARION; after update, the third patient’s name is EHRxyzmedical ^ MARION
- *fifth step*: merge the second and the third patients, i.e., the SINGLETON ^ MARION and EHRxyzmedical ^ MARION

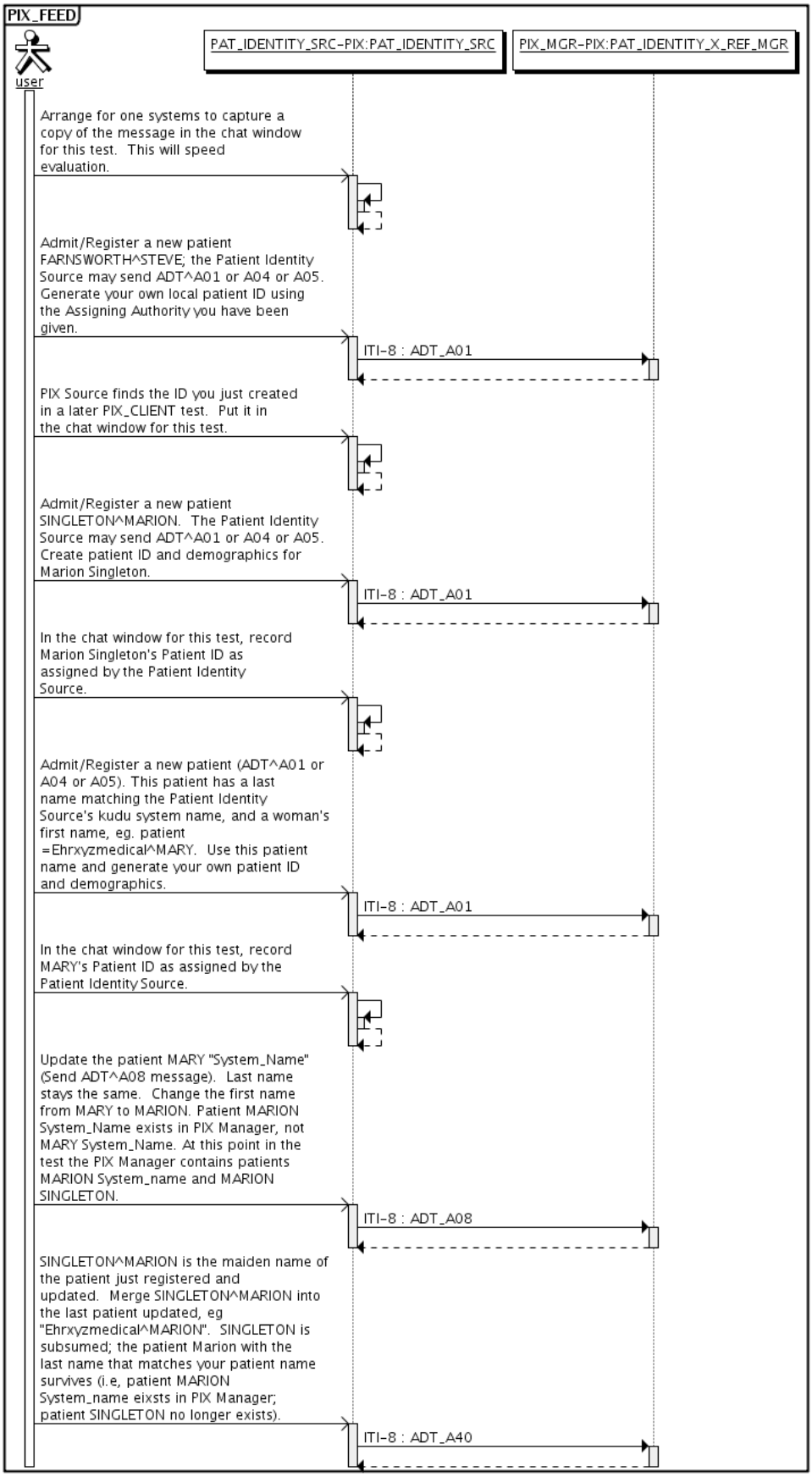


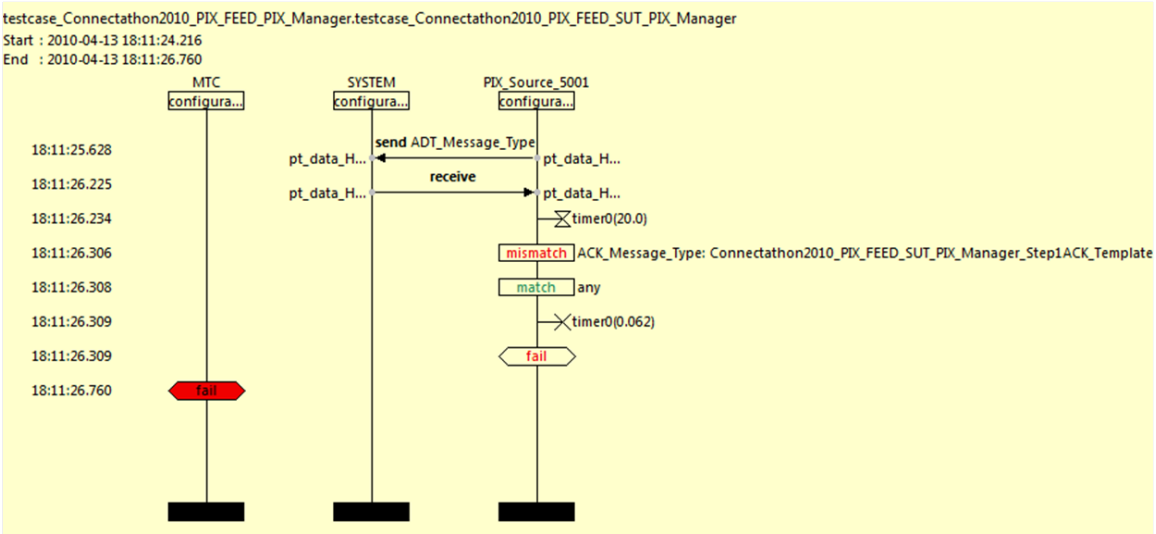
Figure 5.25: PIX_FEED Interaction Scenario

The scope of the test is to verify if different types of [ITI-8] transaction are supported by the *PIX Manager*: 1) admit, pre-admit, register a patient (using A01, A04, A05 trigger events); 2) update a patient (using the A08 trigger event); 3) merge patient identities (with the A40 trigger event).

In this *interaction scenario*, the TTCN-3 test system played the role of the *PIX Source*. This scenario implies the usage of only one version of the HL7 (v2.3.1) messaging standard. The transport protocol required to transmit feed messages is MLLP. Hence, on the TTCN-3 test system side, these requirements must be fulfilled.

Evaluation. The *PIX Source actor* must be able to perform all types of [ITI-8] transactions: admit, register (A01, A04, A05), update (A08) of a patient and patient identity merge (A40). If one of these capabilities is missing, the Connectathon test does not pass. The Connectathon validation process imposes that one of the two *actors* involved in the PIX_FEED *interaction scenario* will have to show the transmitted HL7 messages. In all tests executed with the TTCN-3 test system, the transmitted HL7 messages have been recorded on the test system side.

After executing the first step, the patient FARNSWORTH ^ STEVE is inserted into the *PIX Manager* with the *local Assigning Authority* of the *PIX Source* system in this test, i.e., the TTCN-3 test system. The *PIX Manager* must cross-reference this patient to the existent patient FARNSWORTH ^ STEVE which has a *global* ID previously fed by PIX_Seed_Mgr preamble test. This could be checked by using a *PIX Consumer* to query for the *Patient Identity Source's local* ID, and retrieve the *global* ID for the patient FARNSWORTH.




```
MSH|^~\&|OTHER_ETSI|ETSI|ETAH|ETAH|20100413061526||ADT^A04^ADT_A01|ETSI-5216076|P|2.3.1
EVN||20090224104145-0600
PID|||101^^^ETOT&1.3.6.1.4.1.21367.2010.2.1.419&ISO||FARNSWORTH^STEVE||19781208|M
PV1||O
```

```
MSH|^~\&|ETAH|ETAH|OTHER_ETSI|ETSI|20100413181536||ACK^A04|127117533649618|2.5
MSA|AA|ETSI-5216076
```

Figure 5.26: PIX_FEED Interaction Scenario Run Example: ACK with Different Version

Similarly, the patient SINGLETON ^ MARION was registered in the database of the *PIX Manager* during the second step of the test. For the patient EHRxyzmedical ^ MARY, the first name was updated from MARY to MARION after the *fourth step* of the test, hence, no change in patient ID

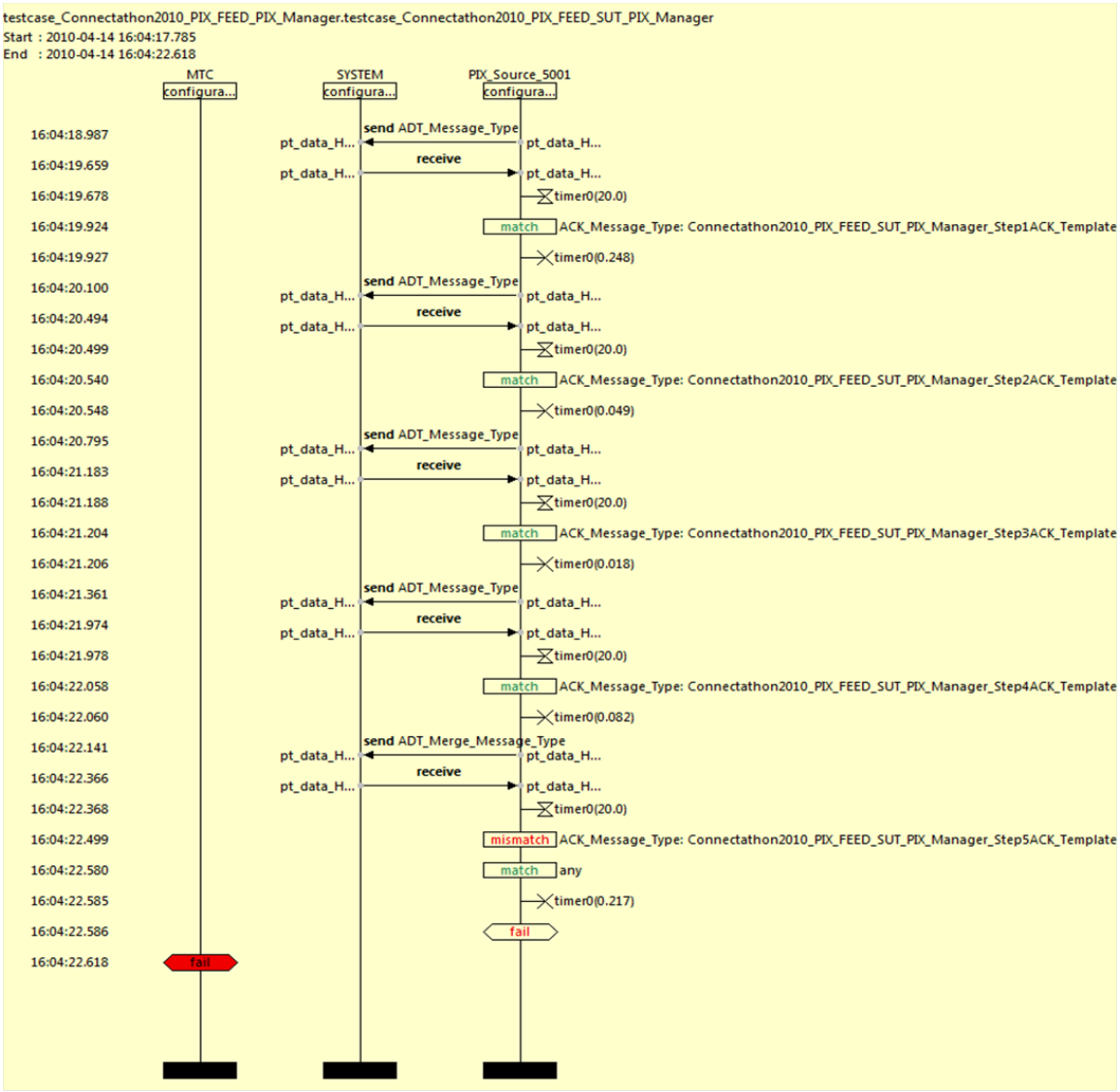
occurred. After performing the *fifth step* of this *interaction scenario*, the patient EHRxyzmedical ^ MARION (which was updated in the *fourth step* from EHRxyzmedical ^ MARY to EHRxyzmedical ^ MARION), must have been merged with the patient inserted in the second step, i.e., SINGLETON ^ MARION. With respect to the recorded IDs during the test's steps, in the end, patient ID for EHRxyzmedical ^ MARION survives while patient ID for SINGLETON ^ MARION has been subsumed.

During the Connectathon week, this *interaction scenario* was performed with the TS having the role of the *PIX Source*. From the perspective of the *message exchange patterns* matching the communication between the TTCN-3 *Source actor* and the SUT *Manager*, the PIX_FEED scenario steps correspond to a sequence of *<Information - Immediate Response> message exchange patterns*. Basically each step presented above matches this MEP and each step can be recognised as a dialogue between an *Informer* (the TTCN-3 PIX *Source*) and a *Tracker* (the SUT PIX *Manager*).

The test was performed against 5 different *PIX Manager* implementations and revealed 8 problems on the SUT side. In the following, some examples of test executions are presented.

Figure 5.26 shows the graphical logging produced after executing the PIX_FEED test. The test execution ended with the verdict **fail** because, the ACK message received by the *PIX Source* (TS) immediately after sending the first feed, (ADT_A01) *message structure*, has a different HL7 version to the expected one. This problem was revealed by the TTCN-3 *PIX Source* only, even though similar erroneous ACK messages were sent as answers by the same *PIX Manager* to other counterpart *PIX Sources* and the verdict for their executed scenario was pass. The TTCN-3 *PIX Source* cached this problem thanks to a very restrictive TTCN-3 type system.

A different case of a failed test is presented in Figure 5.27. Instead of sending an ACK message with application accept (AA) as a reaction to the merge feed sent by the TS *PIX Source*, the SUT *PIX Manager* reported an internal application error. Different to the case shown before, where the received ACK did not match a field restriction (HL7 version field in MSH *segment*), here the received ACK did not match the expected message structure. On the contrary, the received ACK contains an unexpected *segment* ERR (third *segment*) in its structure. This case not only helped uncover an issue raised by the merge transaction on the *Manager* side, but it also helped to improve the CoDec of test system itself. Prior to Connectathon, no ACK message with error has been tested, hence the decoding of such message structures could not have been validated in advance. This situation again shows the importance of plug-in events such as the Connectathon event for improving the interoperability of systems on both sides.




```
MSH|^~\&|OTHER_GE_ICW|GEICW|OTHER_ETSI|ETSI|20100414160431||ACK^A40|5072|P^T|2.3.1
MSA|AR|ETSI-6359026
ERR|^^^207&Application internal error&HL70357&&HTTP operation failed invoking
http://localhost:8080/pxs/exporter with statusCode: 500
```

Figure 5.27: PIX_FEED Interaction Scenario - Run Example: Application Internal Error

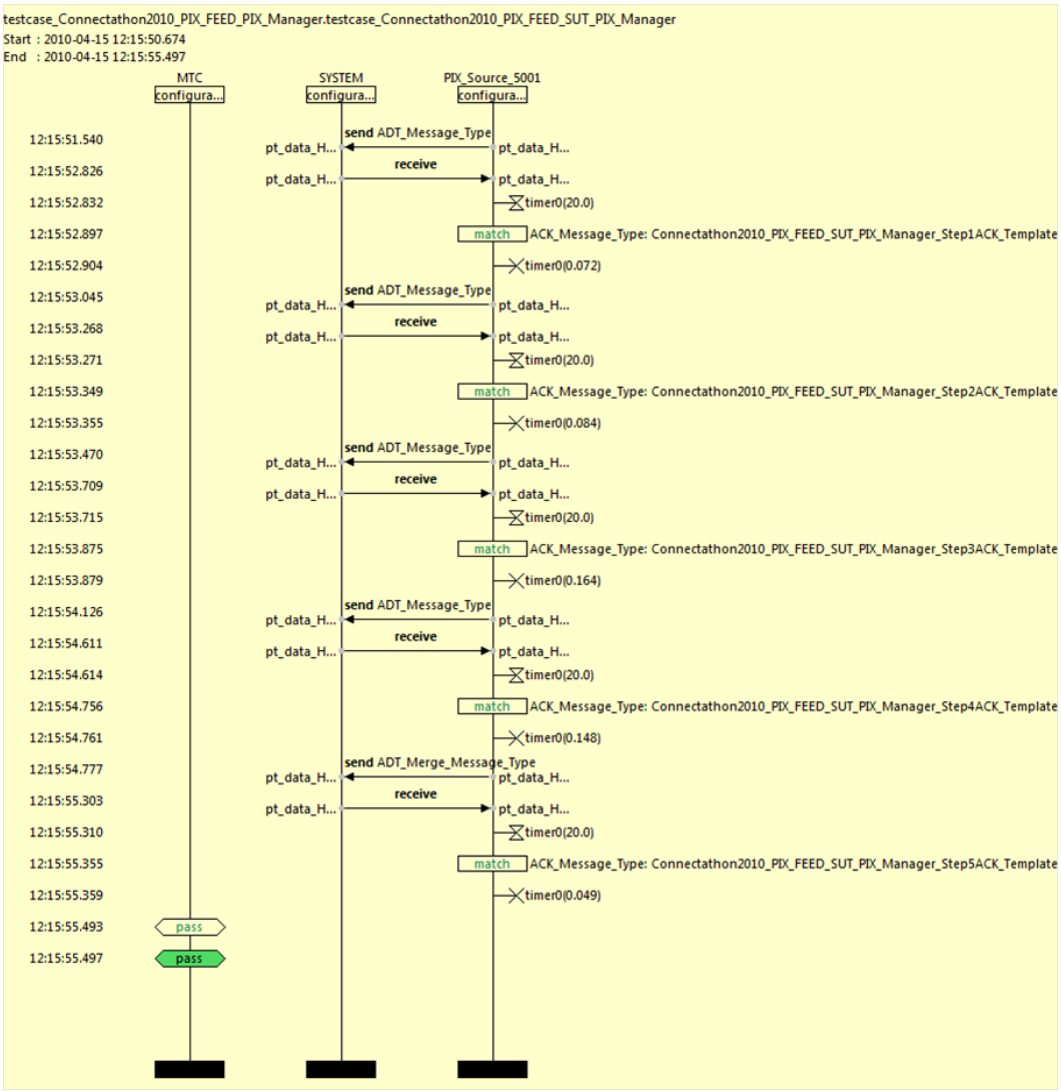


Figure 5.28: PIX_FEED Interaction Scenario - Run Example: Pass

Figure 5.28 contains the log from a smooth test execution where all 5 steps imposed by the scenario were performed according to the specification. The TTCN-3 *PIX Source* validated each received ACK answer and the test finished with the verdict **pass**.

5.2.2.2.3 PIX_CLIENT Interaction Scenario

Similar to the PIX_FEED, the PIX_CLIENT *interaction scenario* is described by the IHE Connectathon organisers as a sequence of steps as shown in Figure 5.29. It again involves two interacting *actors* of *PIX Consumer* and *PIX Manager* types.

Description. A *PIX Consumer actor integration profile* is designed to use both a query mechanism (transaction [ITI-9]) and an optional push mechanism for notifications (transaction [ITI-10]). This is a test for the query mechanism.

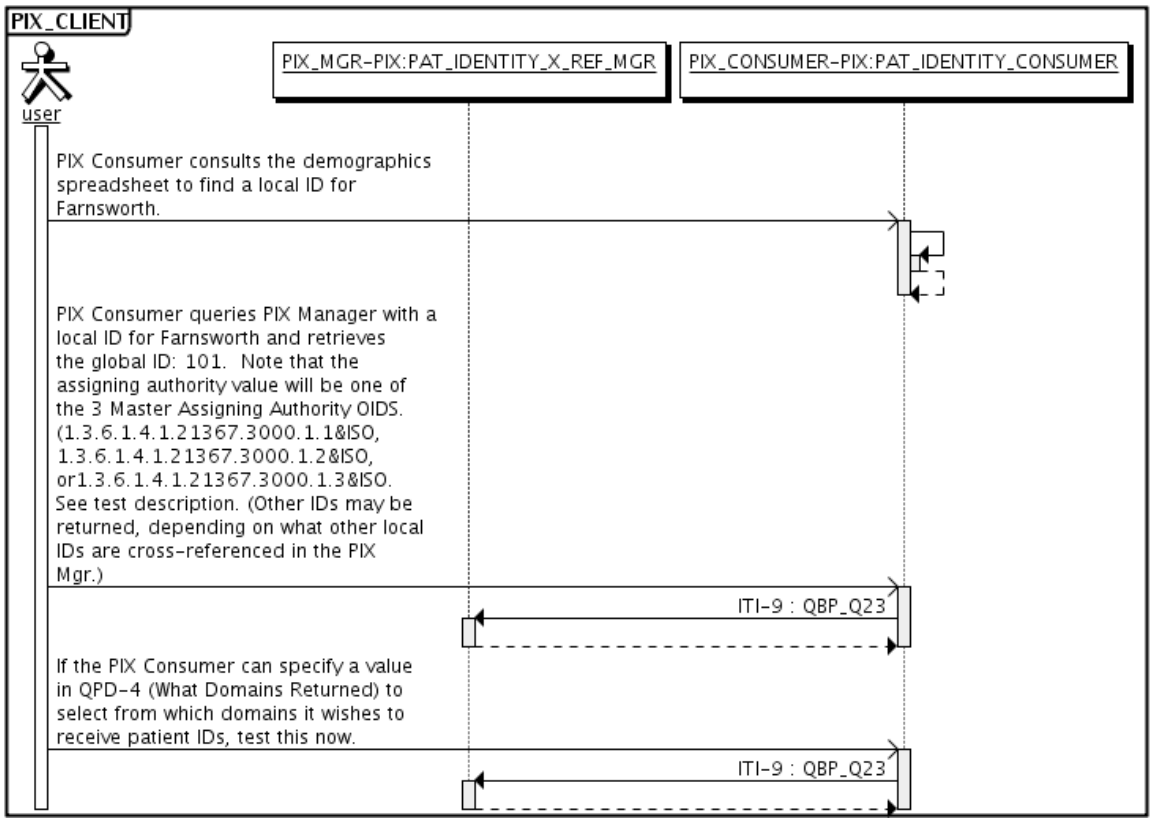


Figure 5.29: PIX_CLIENT Interaction Scenario

The prerequisite for this scenario is to have completed the above presented scenarios: the preamble scenario PIX_Seed_Mgr and the PIX_FEED *interaction scenario*. This test ensures that the *PIX Manager* contains many patient IDs for the patient FARNSWORTH ^ STEVE who was registered: 1) with the patient ID 101 (defined in Connectathon demographics tables) with a *global* affinity domain, i.e., *Assigning Authority*, registered after running the PIX_Seed_Mgr preamble scenario and 2) with different patient IDs and with *local Assigning Authorities* identifying the different *Sources* that have run the PIX_FEED scenario with that particular *Manager*.

The *interaction scenario* involves the following steps on the *PIX Consumer* side:

- *first step*: *PIX Consumer* queries the *PIX Manager* with a *local* ID for FARNSWORTH ^ STEVE and retrieves records with the *global* ID 101 and the *Assigning Authority* value will be one of the three master *Assigning Authority* OIDs. (1.3.6.1.4.1.21367.3000.1.1&ISO, 1.3.6.1.4.1.21367.3000.1.2&ISO, or 1.3.6.1.4.1.21367.3000.1.3&ISO). Other IDs may be returned, depending on what other local IDs are cross-referenced in the *PIX Manager*.
- *second step*: if the *PIX Consumer* can specify a value in the fourth *field* of the QPD *segment* (QPD-4: "what domains returned") in order to select from which domains it wishes to receive patient IDs, it queries the *PIX Manager* with such a constraint.

The transaction [ITI-9] involves a *query* initiated by the *Patient Identifier Cross-reference Consumer actor* for a list of patient identifiers that correspond to a patient identifier known by the

Consumer. The request is received by the *PIX Manager* which firstly processes it and afterwards returns a *response* in the form of a list of corresponding patient identifiers, if any.

Evaluation. For this *interaction scenario*, the patient ID used by the *Consumer* in its query is the one associated to the patient FARNSWORTH ^ STEVE previously created in test PIX_FEED. The patient FARNSWORTH ^ STEVE should have a record in the *PIX Manager* with the *global* ID 101 entered during the test PIX_Seed_Mgr. The same patient should have other similar records entered by *Sources* from the test PIX_FEED, but with different patient IDs. As a *PIX Client*, one needs to find one or more *local* IDs for the patient FARNSWORTH using the *Assigning Authority* value assigned to each *PIX Source* at the Connectathon. One can then use one of the *local* IDs to form a PIX *query* to get the *global* ID (and/or other IDs) for that patient.

The evaluation process at Connectathon demands that a so-called monitor (human) will observe the *query* and the *response* for each *query* and looks for evidence that a *query* was made with a *local* ID and returned the *global* ID (101) for the patient FARNSWORTH ^ STEVE. Other *local* IDs may also be returned, depending on which IDs for FARNSWORTH ^ STEVE have been cross-referenced by the *PIX Manager*.

This evaluation task has been implemented in a *TTCN-3* **testcase**. This time the TTCN-3 test system emulates the *PIX Consumer actor*. The sequence of *interactions* within this *interaction scenario* corresponds to a sequence of two *<Request - Immediate Response> message exchange patterns* introduced in Chapter 3. For each MEP, by sending a *query*, the *PIX Consumer* acts as a *Placer* while the respondent *actor*, the *Manager*, has the role of the *Fulfiller*. The TTCN-3 test behaviour for the *Consumer* is derived out of these MEPs.

During the plug-in event week, this *interaction scenario* was executed against 3 different *Manager* systems and the TTCN-3 test system revealed several problems regarding the compliance of the *Manager* to the [ITI-9] transaction.

Figure 5.30 shows an example of a test that finished with the verdict fail. The revealed problem was rather an HL7 conformance issue because it regarded the misplacement of the coded value "S" on the position of the 8^{th} instead of the 7^{th} component of the last present *field* of the PID *segment* in the *query response*. Even though the query processing functionality of the tested *Manager* proved to be according to the specification, i.e., the list of patient IDs was encapsulated in the *query response*, the TTCN-3 claimed an error in the message from SUT. Moreover, contradictory discussions arose from the fact that the numbering of *components* within that *field* should start from 1 and not from 0, a fact that could have explained the wrong positioning of the "S" value. After a carefully inspection of the HL7 messaging standard and of the IHE PIX specification document, the conclusion was that the HL7 v2.5 TTCN-3 type system was correctly specified. This test run proved once again that the detailed conformance validation at Connectathon is still an issue. This fact encourages the owner of different *actor's* implementations to rely on their successful certification at previous 10 participations at Connectathon, even though such defects had been present in their systems but they could not have been revealed. Additionally, the case revealed by this particular run als questioned the validity of the previous runs of the same scenario where the same *PIX Manager actor* was involved, since it indirectly uncovers a poor validation of the *query responses* on the counterpart *PIX Consumers*.

Furthermore, after fixing the *query response* message structure on the *PIX Manager* side to be compliant with the specification, the next runs against the same system revealed another interesting type of problem. Figure 5.31 presents the trace of a *query response* which does not comply with the type specification regarding one mandatory field. As the Figure shows, this field is not included in the set of required fields of one of the PID *components* (the missing is contained in

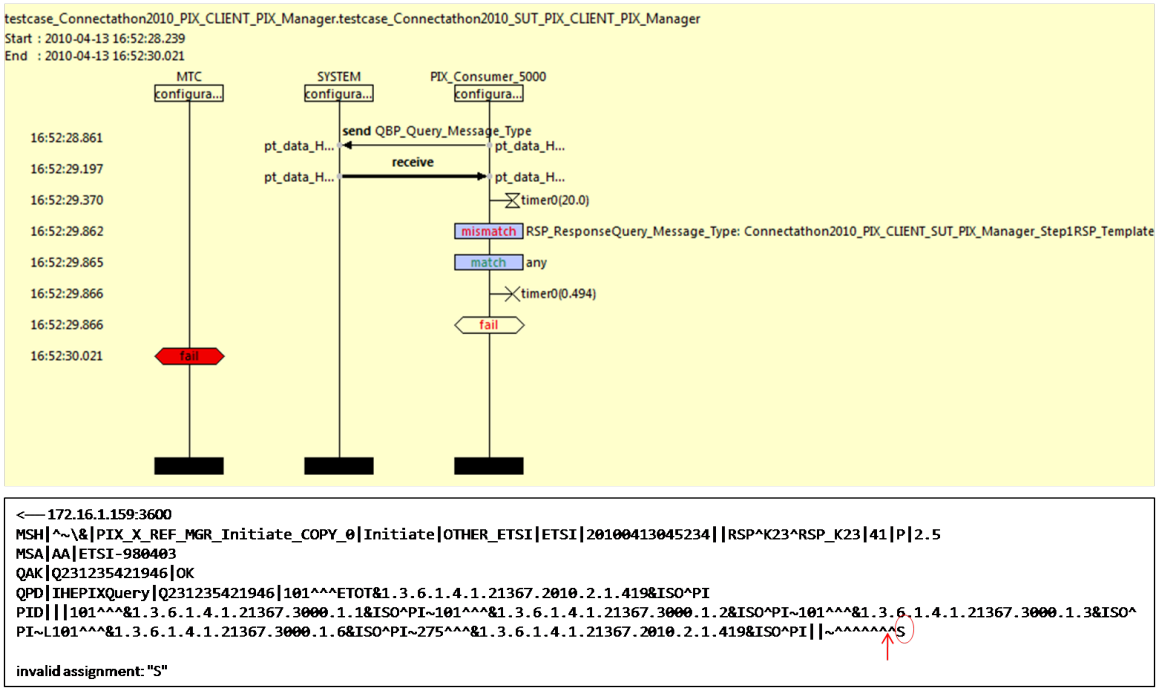


Figure 5.30: PIX_CLIENT Interaction Scenario - Run Example: "S" Problem in Query Response

the attached box). This issue demonstrates a strong optionality checking capability on the *TTCN-3* test system side thanks to a very powerful type system.

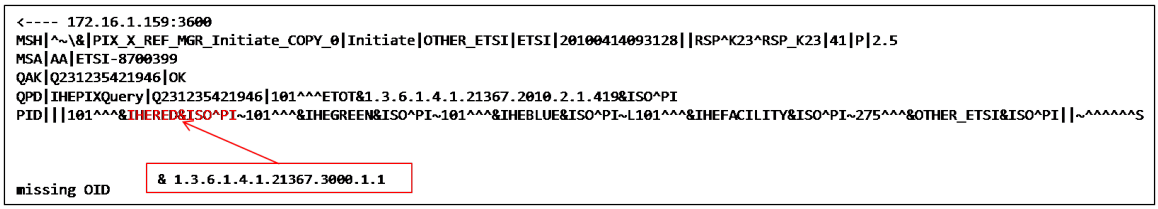


Figure 5.31: PIX_CLIENT Interaction Scenario - Run Example: "Missing OID" Problem in Query Response

Another type of issue that questioned the functionality of the counterpart *PIX Manager* regarding the implementation of the query mechanism, is reflected in the graphical logging from Figure 5.32 and outlined in Figure 5.34. This run proves that the *PIX Manager* was incorrect in processing a *query* since the list of patient identifiers, which was supposed to be returned was empty. This issue has been cached on the *TTCN-3* test system side thanks to the template matching mechanism: the expected value (left side of the data view in Figure 5.34) was set to be **any** value, i.e., "?" symbol while the received value from the SUT was empty, i.e., **omit** (on the right side of the same Figure).

A successful test is presented in Figure 5.33. It contains the log from a smooth test execution where the 2 steps imposed by the scenario were performed according to the specification. The TTCN-3 *PIX Consumer* validated each received *query response* answer and the test finished with the verdict **pass**.

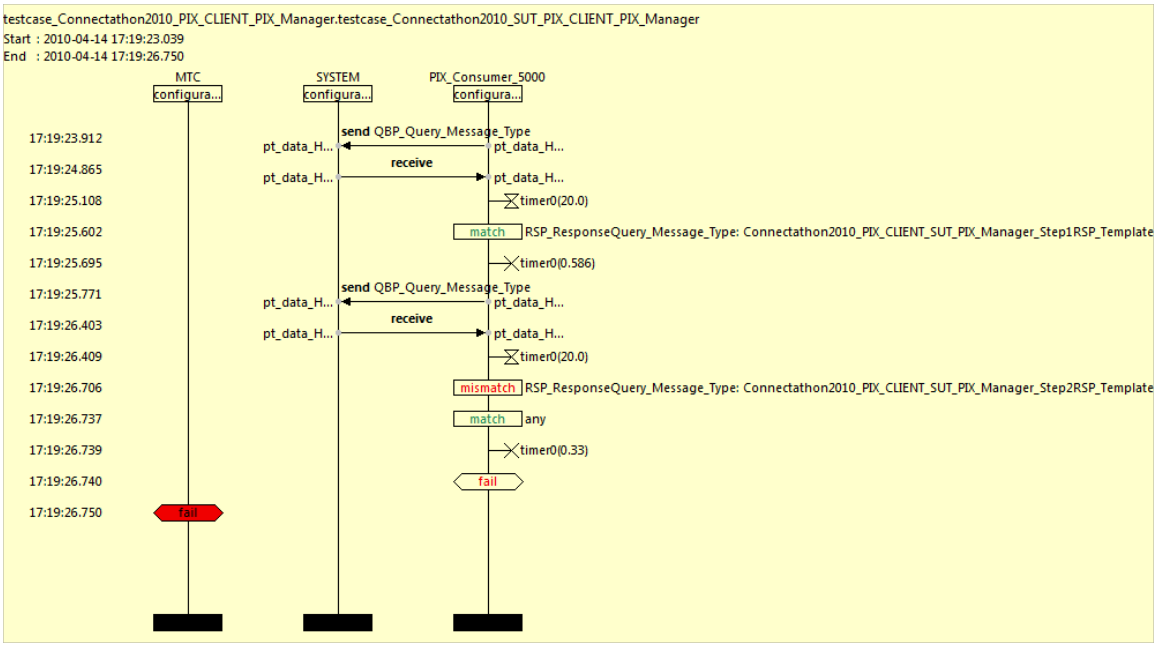


Figure 5.32: PIX_CLIENT Interaction Scenario - Run Example: Returned Empty List of Patient Identifiers (2)

5.2.2.2.4 PIX_CLIENT_Upd_Notif_Option Interaction Scenario

While the query mechanism is mandatory for each *PIX Consumer* implementation and it is tested with the PIX_CLIENT scenario, the support for update notifications is optional for Consumer implementations. However, those *PIX Consumers* that implement this feature and all *PIX Managers* are required to run at least one *PIX_CLIENT_Upd_Notif_Option* scenario during the Connectathon.

Furthermore, different to the previous examples, in order to run this test, instances of all three types of PIX *actors* are needed. The scenario consists of a sequence of steps as presented in Figure 5.35. Both the "Geneva Tool", i.e., the *PIX Source* made available by the Connectathon organisers and a vendor's *Patient Identity Source actor* are needed to run this scenario. The *PIX Source* is marked as optional in this test because the focus is not to validate its functionality but rather to use it to help realise the whole flow and the test itself.

Description.The scenario is conceived to test the notifications mechanism (transaction [ITI-10]). This test ensures that the *PIX Manager* notifies the *PIX Consumer* using [ITI-10] transaction, whenever a change occurred in a set of cross-reference patient identifiers belonging to the patient identifiers in the "domains of interest" of the *PIX Consumer*.

As a prerequisite for this test, the *PIX Manager* shall configure the "domain of interest" of the counterpart *PIX Consumer actor*. For this test, the "domain of interest" shall indicate two values: 1) the value of the *local Assigning Authority* for the *PIX Source actor* involved in the scenario and 2) the value of the *global* (master) *Assigning Authority* for the Connectathon affinity domain, i.e., of the *Assigning Authority* of the "Geneva Tool" *PIX Source*).

The steps to be performed within this *interaction scenario* are presented bellow. The last three

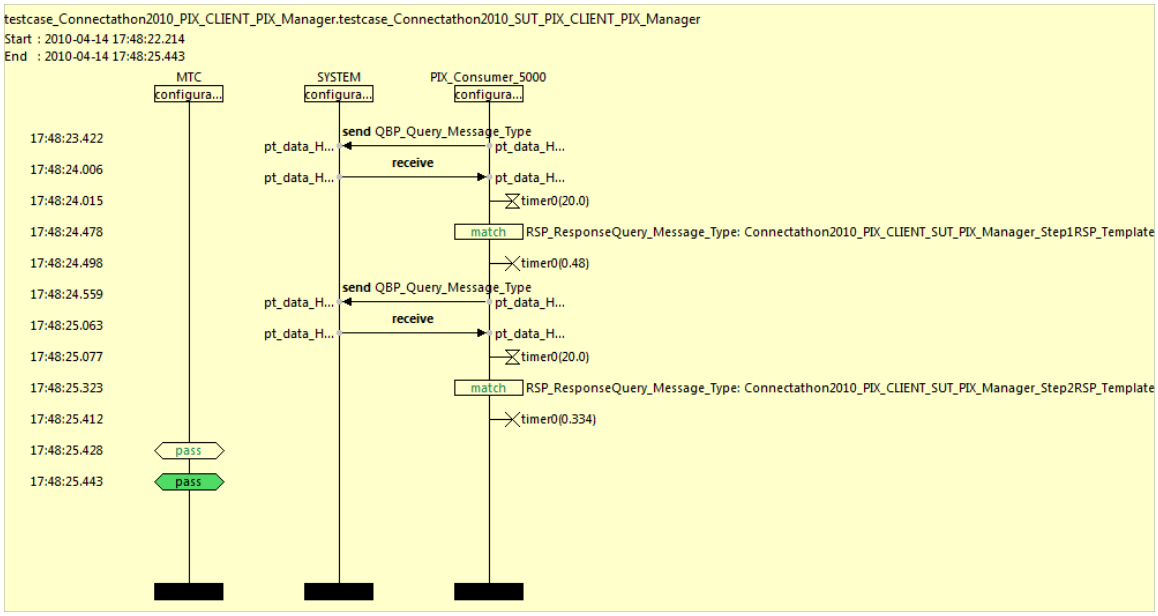


Figure 5.33: PIX_CLIENT Interaction Scenario - Run Example: Pass

steps are optional.

- *first step*: "Geneva Tool" *PIX Source* sends a feed for the admission of an in-patient into a facility: ANDERSˆ MARIAN, 1944.04.04, female, with the address "444 Main St. Seattle, WA". This patient is admitted with the *global* (master) *Assigning Authority* value for the Connectathon affinity domain.

- *second step*: an *update notification* must be sent from the *PIX Manager* to the *Consumer* and must contain the *global Assigning Authority*.

- *third step*: the same patient, i.e., ANDERSˆ MARIAN, 1944.04.04, female with address 444 Main Street, Seattle, WA, is admitted at a different *local* facility. This patient is admitted with the *local Assigning Authority* value of the second *PIX Source actor* involved in this *interaction scenario*.

- *fourth step*: [PIX Manager internal]: the *PIX Manager* cross-references the patient ANDERSˆ MARIAN.

- *fifth step*: an update notification is sent from the *Manager* to the *Consumer*. It contains IDs from both the *local Assigning Authority* of the *Patient Identity Source* and the *master Assigning Authority*.

- *sixth step* [optional]: The "Geneva Tool" does not support this functionality. Although they follow the scenario in the Technical Framework, not all PIX Managers will unlink the patients because of the address change.

- *seventh step* [optional]: On the Local Patient Identity source, change Marian Ander's address to 111 New Street, Portland, Oregon.

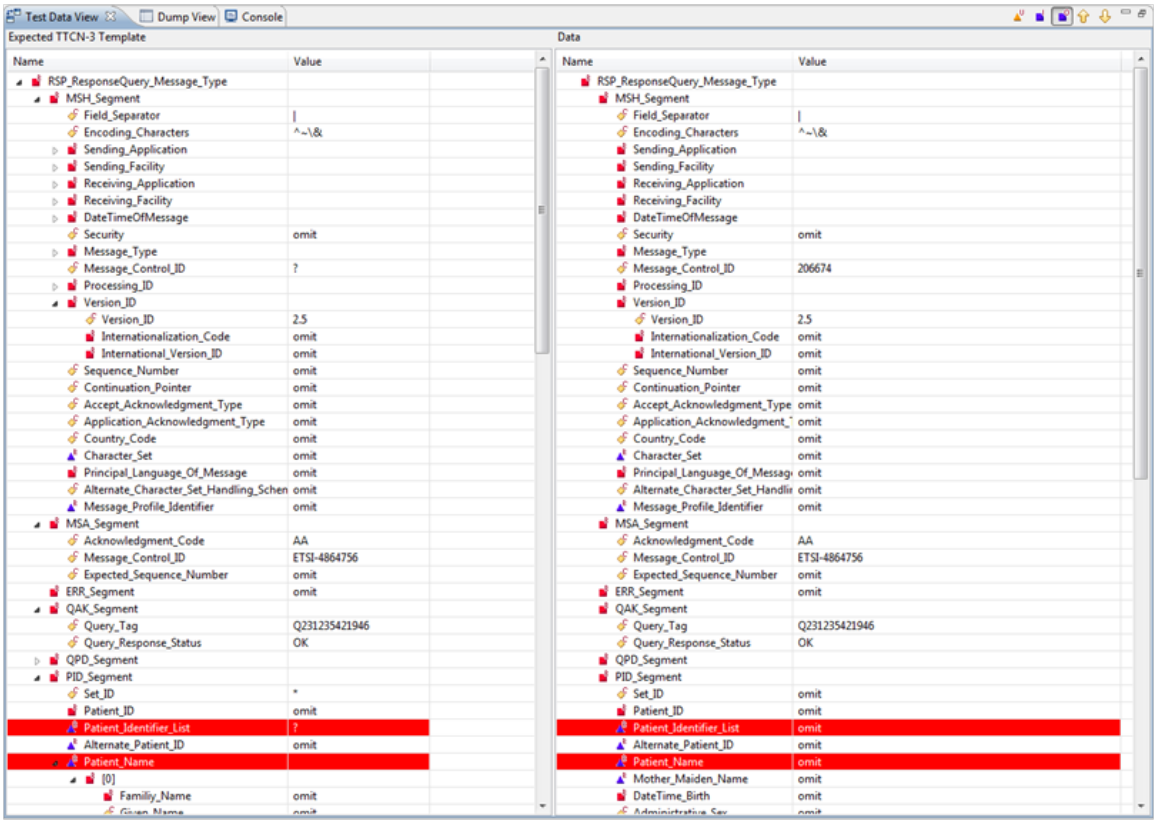


Figure 5.34: PIX_CLIENT Interaction Scenario - Run Example: Returned Empty List of Patient Identifiers (1)

- *eighth step* [optional]: The PIX Mgr determines that this patient from the local domain is no longer the same patient as that in the master affinity domain. It sends an update notification to the Consumer with Marian Anders' patient ID from the local Assigning Authority.

- *ninth step* [optional]: PIX Manager sends a second update notification with Marian Anders' patient ID from the Master Assigning Authority.

Evaluation. The validation process at Connectathon implies that a monitor should observe evidence that the *PIX Consumer* is receiving messages from the *PIX Manager* when patients are admitted or updated in their "domains of interest" configured for that *Consumer* on the *PIX Manager*. This can be done by examining the stored traces captured by the participants in this test. To achieve this, firstly a patient has to be registered or updated either by a *Source* with *local Assigning Authority* or by the "Geneva Tool" *PIX Source* with *global Assigning Authority*. As a result of this feed, the notifications shall come in real-time from the *Manager*.

This *interaction scenario* offered the possibility to make evidence of an important capability of the *TTCN-3* test system design: its capability to simultaneously emulate multiple *actors*. The situation at Connectathon was that, given the optionality of this update notification feature on *Consumers* side, not many systems supported this mechanism in their implementation. On the other side, each PIX Manager had to run at least once this scenario. Hence, the few available *Consumers* offering update notifications were somehow over demanded. Furthermore, as described above in the test

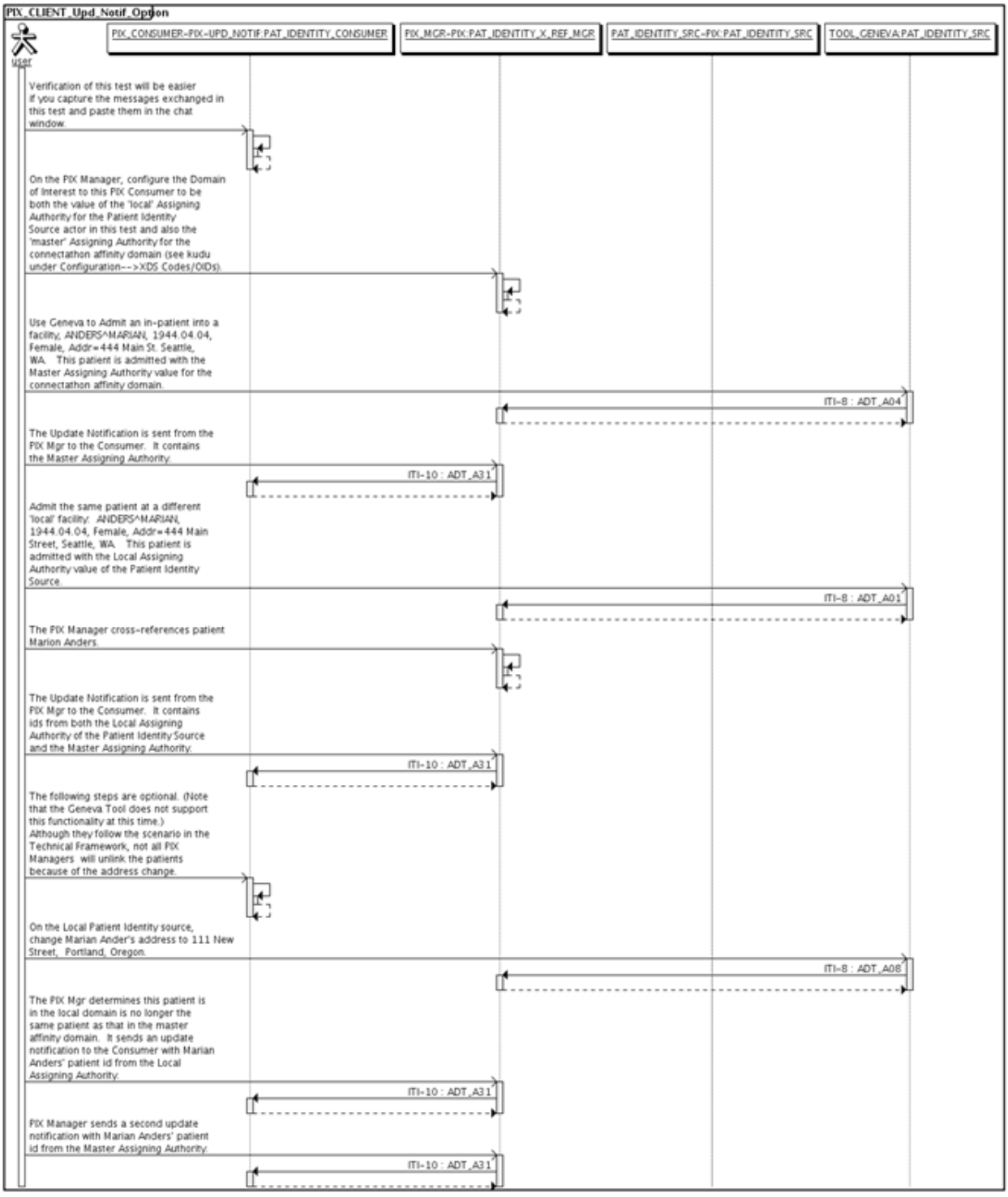


Figure 5.35: PIX_CLIENT_Upd_Notif_Option Interaction Scenario

steps, this flow involves the participation, besides the *PIX Manager* and *PIX Consumer*, of two additional *Sources*. The problem that arose here was again the over-stressing of the only available "Geneva Tool" *PIX Source* with *global Assigning Authority*. By offering all counterpart *actors* for a *PIX Manager* demanded by this scenario, including a replace for the "Geneva Tool" *PIX Source*, the *TTCN-3* test system was indirectly set on the list of preferences of *Manager* actors regarding the validation of the update notification mechanism. Within this interaction scenario, the *TTCN-3* test system emulated two *PIX Sources* and one *Consumer*.

With respect to the communication pattern between different *actors*, the *<Information - Immediate Response> message exchange pattern* is applied between any two *actors* involved in the scenario.

During the Connectathon week, the *TTCN-3* test system could perform this *interaction scenario* in an automated manner and, in the end, it captured and reported inconsistencies and issues along a number of runs performed against four *PIX Managers*.

The most common interoperability issue discovered by running this scenario is revealed in Figure 5.36 and Figure 5.37. The problem clearly marked with "fail" on the lifeline of the *Consumer* PTC indicates that the *PIX Manager* does not send the update notification messages that the interoperability scenario requires. In the first Figure, the *PIX Manager* does not send any of the demanded update notifications. In the second Figure, the *PIX Manager* is able to react with an update notification message only to the second feed, which still indicates an interoperability problem. Both situations have been corrected on the SUT side, i.e., *PIX Manager*. In the end, the performed sequence of interactions conforms to the *interaction scenario*. This result is presented in Figure 5.38 which contains a clear execution of the PIX_CLIENT_Upd_Notif_Option *interaction scenario*.

5.2.2.2.5 Connectathon 2010 Results

The main results from this case study were obtained during the Connectathon 2010 event where the test system helped revealing many interoperability issues in the tested SUTs. An important aspect worth mentioning, is the industrial character of the event; different than the DEC case study where the results were purely for research purposes, the PIX test system proved that the methodology and its technical realisation can also cope with industrial needs. Many examples of such issues have been described above, along with the presentation of the interoperability test scenarios. Next, it is perhaps also interesting to look at these types of errors and provide some statistical information.

During the four and a half days at Connectathon, 18 interoperability test flows were executed. Each test required at least 4-5 runs to fix the problems detected in the SUTs. The minimum number of runs was 1 (with a system which proved to be error free along the event) and the maximum 6 (with systems which proved to be rather at prototypical level).

Overall, about 90 runs were performed to fix about 15 problems in the other systems and about 2 problems in the TS itself. Each run required about 15 minutes to configure the TS with required parameters. Each issue required about one hour of discussions, investigations, debugging and problem identification. Most of the times, the IHE specifications were investigated to clarify and prove that the requirements have been correctly implemented on the TS side. In a few situations, the partners required one day to fix the problem.

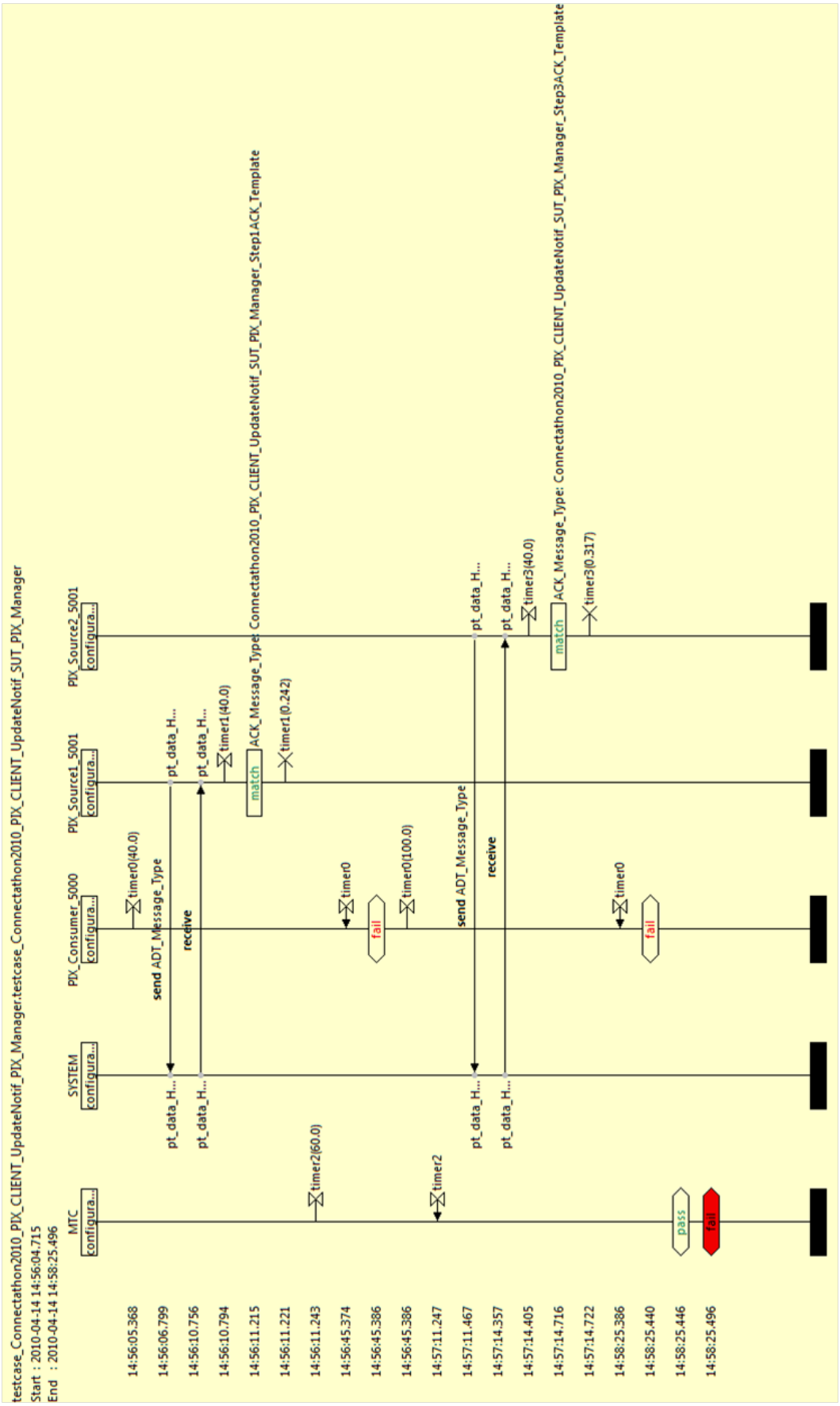


Figure 5.36: PIX_CLIENT_Upd_Notif_Option Interaction Scenario - Run Example: No Update Notifications

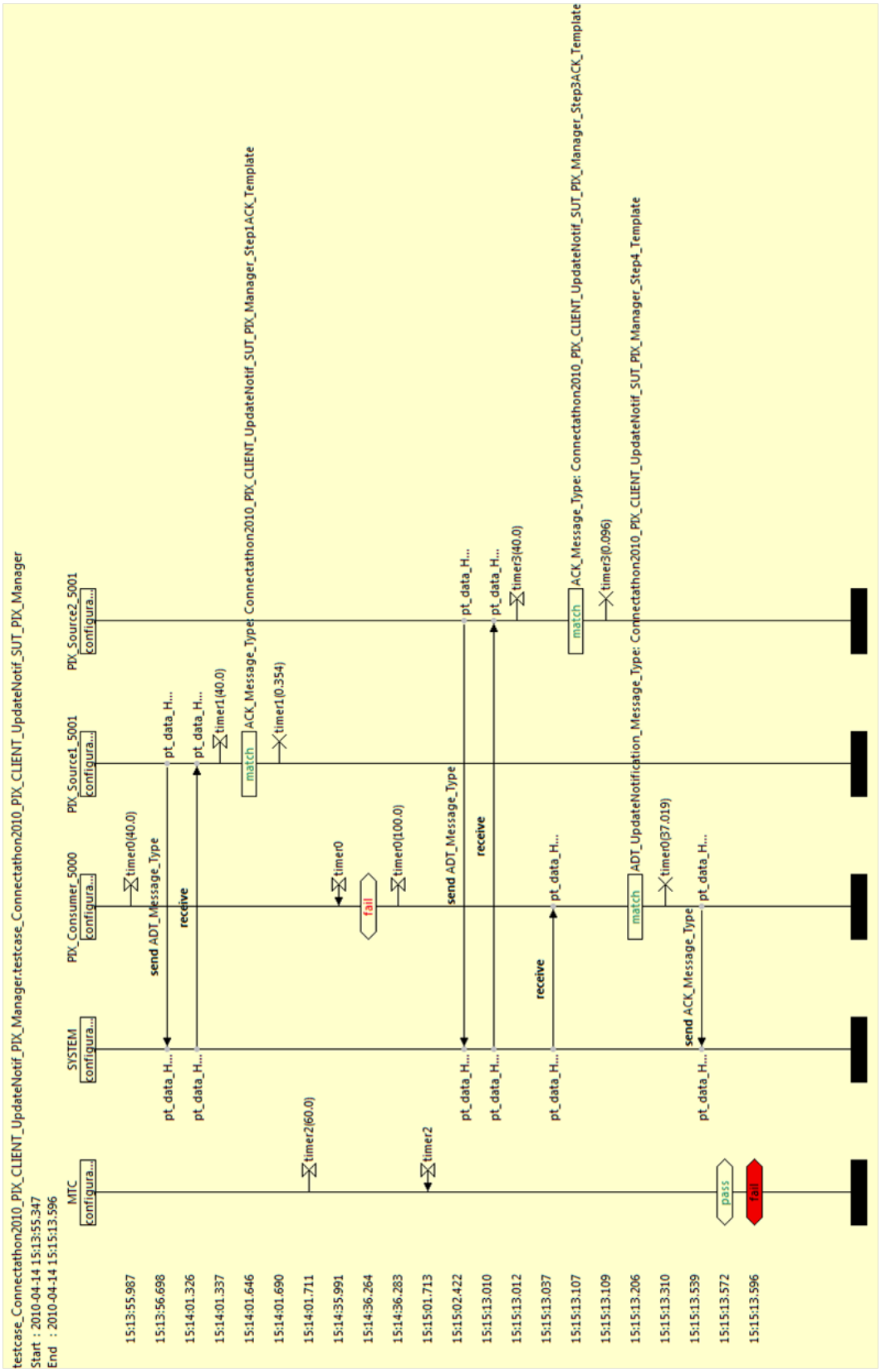


Figure 5.37: PIX_CLIENT_Upd_Notif_Option Interaction Scenario - Run Example: No Update Notification for Registration

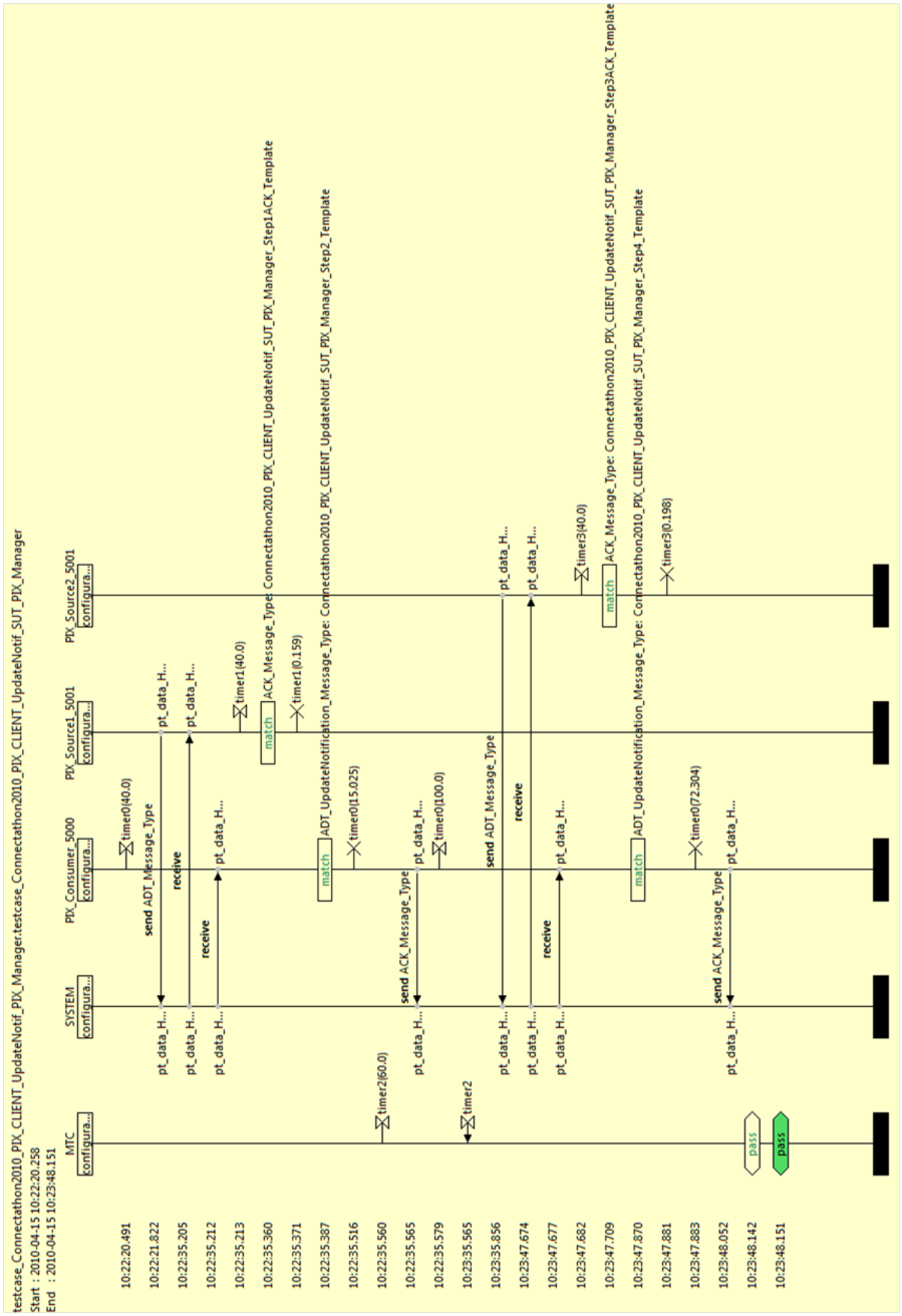


Figure 5.38: PIX_CLIENT_Upd_Notif_Option Interaction Scenario - Run Example: Success

The types of interoperability issues can be classified into five categories:

- *flow issues*: these issues are detected by timer's timeout events occurring when certain messages are not received by the test system. For example, this type of issue occurred in the *PIX_CLIENT_Upd_Notif_Option interaction scenario* when the update notifications were not transmitted. This case is presented in Figure 5.36 and Figure 5.37.
- *conditionality issues*: these issues occur when fields in the *query response* should contain values derived from the *query* but they do not. For example, such an issue occurs when the MSA *segment* in the *query response* does not contain the message control ID used in the MSH *segment* of the initial *query* that triggers the *response*.
- *message type conformance issues*: these issues occur when the type of received message does not conform with HL7 or IHE specifications. This type of issue was encountered in the situation presented in Figure 5.26 where the structure of the ACK message is of a different HL7 version than expected.
- *message content issues*: these issues occur when the content of the response message does not contain the expected values or is missing. Figure 5.31 indicates such an error where the OID is missing in the PID segment.
- *IHE constraints issues*: these issues occur when the IHE constraints were not respected. For example, IHE imposes that the value "S" is set on the 7^{th} component of the PID *segment* within a *query response* message. An example of this error is presented in Figure 5.30 where the "S" value is set wrongly at the 8^{th} position and consequently, the seventh component was left empty.

Chapter 6

Conclusions and Outlook

Mehr als die Vergangenheit interessiert mich die Zukunft, denn in ihr gedenke ich zu leben.
– Albert Einstein

The delivery of eHealth services is a critical and complex application area and non-interoperability of HISs might lead to serious damage or even death of patients. An example is the misinterpretation of the measurement units expressed in coded values withing an ePrescribing system when exchanging messages between the physician order entry system and the receiving system available to the pharmacy. This can lead to an incorrect dosage being applied, which, in the end, puts the patient's life in danger. In the pursuit of achieving and assessing interoperability of HISs, the interoperability testing must take place long before products are sold on the market and even before the direct interaction with the additional healthcare applications, which complete the operational environment needed to perform an interaction scenario.

This thesis elaborates a novel methodology for interoperability testing applied to healthcare information systems. The methodology introduces an interoperability test design process and provides the necessary concepts to enable the design of efficient and extensible interoperability test systems. The particular challenges to be considered while testing for interoperability of healthcare information systems, in general, or during specially organised interoperability test events are firstly identified. Taking into consideration these challenges, the methodology proposes the idea of testing an SUT actor or a group of SUT actors by emulating on the TS side the rest of the interacting actors. A number of identified message exchange patterns between different application roles are used to discover the possible interaction patterns, which serve as a basis for automated derivation of the test behaviour. To the author's knowledge, this approach has never been used before for this purpose.

The relevant components for a TS, in order to cope with the particularities of healthcare systems and to enable an enhanced automation capability, are identified in a generic test system architecture, which is the core concept used to build test systems based on the proposed methodology. Furthermore, the thesis provides design guidelines for implementing an interoperability testing platform based on the TTCN-3 test technology. An important component of the realisation of TTCN-3 test system bases on the semantic mapping of HL7 v2.x message structures to a *TTCN-3* type system that preserves the ontology. Additionally, a set of derivation algorithms for providing TTCN-3 test behaviours and configurations are presented. Moreover, the design of the TS-SUT communication layer allows for flexibility and generality. The communication layer shall recognise the dynamic changes of the test configuration and automatically adapt to them. It

also automatically detects which protocols should be used for communication (as required by the IHE profile) and through a plug-in based mechanisms activates the adequate encoding/decoding scheme of data.

Along two case studies, which serve as the basis for experimental work, the feasibility and efficiency of the proposed methodology and test design concepts are assessed. The results were also evaluated during the Connectathon event where the PIX test suite was used to test real implementations of PIX IHE integration profile.

The target of this work was to realise a generic framework for interoperability testing with conformance checking capability of HIS systems, rather than a single purpose implementation, making it easier to extend it for new SUTs with various types of TS-SUT interactions underlying different communication means.

The main outcome of the applicability of this methodology is that it lowers the costs for running interoperability tests and allows for even more thorough testing by extending the set of scenarios, which are usually available at interoperability test events. This happens thanks to the possibility to emulate the interacting parts, and thus, covering the required operational environments for particular applications. This method, which can be applied on site, represents a pre-interoperability checking method to assess interoperability of the SUT in isolation, not an alternative to traditional method of participating to an interoperability plug-in event, where the tested systems interwork directly one against each other.

Moreover, the evaluation of the test system itself can be achieved long before the SUT is available. Similar to emulating TS actors, the SUT actors can be emulated as well. This way, the TS behaviour can be tested against the emulated SUT behaviour, thus, the interoperability issues can be revealed in advance.

The enhancement of interoperability testing with conformance assessment not only that it helps to discover interoperability issues but also helps to remediate these issues on the SUT side by precise information about the problem's coordinates. This combination of interoperability with conformance testing allows the validation of the whole interoperability stack, covering also the syntactic and semantic levels besides the business and technical levels, which is not provided by other methods or tools.

The realisation of the TTCN-3 test system for HL7/IHE PCD and PIX profile and the successful participation at the Connectathon Europe 2010 event not only broadened the area of applicability of the TTCN-3 test technology to further domains such as healthcare, directly in an industrial environment, but it also opened further possibilities to continue and sustain the work.

The approach presented in this thesis investigates the interoperability testing by proposing a generic and modular test framework. Targeting the efficiency, re-usability and flexibility of the testing system[ZVS+07], the test system design is conceived in such a way that its components are modular, pluggable and dynamically adaptable to scenario changes. This way the framework enables the integration within the same framework of further extensions meant to support new healthcare messaging standards and diverse means of communication layers. In this respect, a first step towards the framework's extendibility is the provision with support for additional messaging schemes introduced in Chapter 2. For example, HL7 version 3 [HL705] shall be supported. More and more vendors and key players in developing healthcare informational systems adopt for their applications version 3 of HL7 and comply with the IHE integration profiles for HL7 version 3. Investing and supporting the efforts in developing a TTCN-3 test framework sustaining and implementing these requirements will considerably increase the area of the applicability of the proposed

methodology within healthcare domain and lift the potential for further evaluations.

Furthermore, new IHE integration profiles can be supported. This will allow the access to a larger spectrum of applications complying with HL7, DICOM/IHE, HITSP which, in the end will also lead to a more mature test framework and simultaneously increase the trustworthiness and confidence in testing healthcare informational systems with TTCN-3 within the TTCN-3 community, etc. Additionally, the popularity of the only one standardised test technology will increase considerably, especially by attending other Connectathon plug-in events outside Europe such as Connectathon US.

With respect to the applicability of the introduced methodology to other application sectors, a good candidate are the genome information systems, which, similarly to HISs, are *data-intensive* systems and present likewise characteristics and challenges as HISs. Furthermore, the identified messages exchange patterns can be used, extended and applied to eGovernment contexts.

A long-term applicability of the proposed approach in the direction of its integration with other test management frameworks used at interoperability plug-in events or employed by certification bodies can be considered. For example, the conformance validation of the exchanged messages between various peers during the Connectathon event in 2010 was done manually by human monitors. Given the complexity of the messages, this task is obviously subject to many errors. For example, during Connectathon 2010 the developed IHE PIX test system was able to detect many inconsistencies, which could not have been detected at previous Connectathon events. However, integration of such syntactic and semantic validators demands great deal of effort because it requires knowledge of both platforms and a combination of different underlying technologies. In this respect, a first step has been already undertaken, when, in the context of the currently running European Healthcare Interoperability Testing and Conformance Harmonisation (HITCH) project, an evaluation of the developed test framework for HL7 v2.x / IHE PIX profile was conducted [HIT11] (Deliverable 2.1 on Tools Selection, Table 4). Compared to the other six selected test frameworks, the TTCN-3 test system provides support for all the investigated aspects: message and content validation, workflow, test cases and test management.

In a domain in continuous expansion and subject to constant and frequent changes in standards as eHealth, the interoperability testing will be, without doubt, a debated topic for the next years. Furthermore, especially in this new era of the cloud computing explosion, the actor based approach thinking developed within thesis addresses the need for applications virtualization in order to conduct interoperability tests in a simulated and complete operational environment.

Glossary

ePrescribing (eRX)
: ePrescribing (or electronic prescribing) refers to the transmission of prescription information from the prescriber's computer to a pharmacy computer.

HL7 SDO
: Health Level Seven (HL7) is one of several American National Standards Institute (ANSI) accredited Standards Developing Organisations (SDOs) operating in the healthcare arena. HL7 is an international community of healthcare experts, promoting standards within and among healthcare organisations and is organised in the form of a global organisation (Health Level Seven, Inc.) and country-specific affiliate organisations. Most SDOs acting in the healthcare arena produce standards for particular healthcare domains such as pharmachy, medical devices, etc. The domain of HL7 is clinical and administrative data. HL7 was adopted by International Organization for Standardization (ISO) and their first mutually published standard was ISO/HL7 21731:2006 Health informatics - HL7 version 3 - Reference Information Model - Release 1.

IHE TF
: Technical Frameworks issued by the Integrating the Healthcare Enterprise (IHE) are detailed documents which specify the Integration Profiles and the associated actors (systems) and transactions.

ITI TF
: IT Infrastructure Technical Framework issued by the Integrating the Healthcare Enterprise (IHE) are detailed documents, which identify a subset of the functional components of the healthcare enterprise, called IHE actors, and specify their interactions in terms of a set of coordinated, standards-based transactions.

MBT
: Model-Based Testing is the automatic generation of efficient tests using models of system requirements and specified functionality.

PCD TF
: Patient Care Devices Technical Framework issued by the Integrating the Healthcare Enterprise (IHE) are detailed documents, which identify a subset of the functional components of the healthcare enterprise, called IHE actors, and specifies their interactions in terms of a set of coordinated, standards-based transactions.

system
: System indicates mainly the system to be tested. Since this thesis focuses on Healthcare Information Systems (HISs) or its subcomponents, e.g., Electronic Health Records (EHRs) systems, the term system refers mainly to these types of systems.

Telehealth
: Telehealth represents the delivery of health-related services and information using telecommunications technologies. The delivery could be very simple such as a discussion about a patient case over the phone between two health professionals, or sophisticated, e.g., using videoconferencing between providers at facilities located in different countries, or even more complex such as robotic technology. It encompasses preventive, promotive and curative aspects.

Telemedicine
: Telemedicine refers to the transfer of medical information over the phone, Internet or through the network with the scope of consulting and sometimes remote medical examinations.

Acronyms

ADT	Admission Discharge and Transfer
ANSI	American National Standards Institute
ASTM International	ASTM International, originally known as the American Society for Testing and Materials (ASTM), is one of the largest voluntary standards development organisations in the world
ATCB	Authorised Testing and Certification Body first
ATNA	Audit Trail and Node Authentication
ATS	Abstract Test Specification
CCHIT	Certification Commission for Healthcare Information Technology
CCR	Continuity of Care Record
CDA	Clinical Document Architecture first
CDS	Clinical Decision Support
CEN	European Committee for Standardisation
CPOE	Computerised Physician Order Entry
CPR	Computer-based Patient Record (also *Electronic Patient Record (EPR)* or *Electronic Health Record (EHR)* or *Electronic Medical Record (EMR)*)
CT	Consistent Time
CTMF	Conformance Testing Methodology and Framework first
DEC	Device Enterprise Communication
DICOM	Digital Imaging and Communications in Medicine
DOC	Device Observation Consumer
DOF	Device Observation Filter
DOR	Device Observation Reporter
ebBP	Business Process Specification Schema
ebXML	eBusiness eXtensible Markup Language

EC	European Commission first
EDI	Electronic Data Interchange
EHC	Electronic Health Card
EHR	Electronic Health Record (also *Electronic Patient Record (EPR)* or *Computerised Patient Record (CPR)* or *Electronic Medical Record (EMR)*)
EIF	European Interoperability Framework
EMF	Eclipse Modelling Framework
EMR	Electronic Medical Record (also *Electronic Patient Record (EPR)* or *Computerised Patient Record (CPR)* or *Electronic Health Record (EHR)*)
ePHR	electronic Personal Health Record
EPM	Electronic Practice Management
EPS	Electronic Prescription System
ER7	HL7 encoding rules: *vertical bar* or *pipe notation* syntax)
ETSI	European Telecommunications Standards Institute
GAIT	Generic Approach to Interoperability Testing Methodology first
HDSS	Health Decision Support System
HIMSS	Healthcare Information and Management Systems Society
HIPAA	The Health Insurance Portability and Accountability Act
HIS	Healthcare Information System (also used interchangeably with Hospital Information System)
HIT	Health Information Technology
HITECH	Health Information Technology for Economic and Clinical Health first
HITSP	Healthcare Information Technology Standards Panel
HL7	Messaging standards published by HL7 SDOs or HL7/ISO - define how information is packaged and communicated from one party to another within the healthcare domain, e.g., versions HL7 v2.x and v3.0.

ICD	International Statistical Classification of Diseases and Related Health Problems first
ICT	Information and Communications Technology
ICU	Intensive Care Unit
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force - develops and promotes Internet standards
IHE	Integrating the Healthcare Enterprise
ISO	International Organisation for Standardisation
ISSS	Information Society Standardisation System
ISTQB	International Software Testing Qualifications Board
ITI	IT Infrastructure
LIS	Laboratory Information System
LOINC	Logical Observation Identifiers Names and Codes
MDT	Model-Driven Testing
MIF	Model Interchange Format
MLLP	Minimal Lower Layer Protocol
MMS	Massachusetts Medical Society
MTC	Main Test Component
NAHIT	National Alliance for Health Information Technology, a US organisation
NCPDP	National Council for Prescription Drug Programs
NEMA	National Electrical Manufacturers Association
NHS	National Health Service
NIH	National Institutes of Health
OASIS	Organisation for the Advancement of Structured Information Standards
OSI	Open Systems Interconnection (OSI)
PACS	Picture Archival and Communication System
PAS	Patient Administration System
PCD	Patient Care Devices
PIX	Patient Identifier Cross-Referencing
PTC	Parallel Test Component
RIM	Reference Information Model first

RIS	Radiology Information System
SDL	Specification and Description Language
SDO	Standards Developing Organisation
SNOMED CT	Systematised Nomenclature of Medicine - Clinical Terms first
SOAP	Simple Object Access Protocol
SUT	System Under Test
TMA	Telemedicine Alliance: cooperation between four international organisations: EC, WHO, ITU, ESA first
TS	Test System
TSI	Test System Interface
TTCN-3	Testing and Test Control Notation, version 3
UML	Unified Modeling Language
UN/EDIFACT	United Nations Electronic Data Interchange for Administration, Commerce and Transport
W3C	World Wide Web Consortium - main international standards organisation for the World Wide Web (abbreviated WWW or W3)
WHO	World Health Organisation
XML	Extensible Markup Language

Bibliography

[ABI10] Applied Biosignals GmbH. www.appliedbiosignals.com, 2010. Accessed 15 Jan. 2011.

[aCRSSH90] J. Gadre amd C. Rohrer, C. Summers, S. Symington, and D. Hogrefe. A COS Study of OIS Interoperability. In *Computer Standards & Interfaces*, pages 217–237, 1990.

[Ama04] M. K. Amatayakul. *Electronic Health Records: A Practical Guide for Professionals and Organizations*. American Health Information Management Association (AHIMA), 233 North Michigan Avenue, Suite 2150 Chicago, Illinois 60601-5800, 2nd edition, 2004. ISBN 1-58426-133-1.

[AMA+09] S. Aftab, Y. Mehmood, F. Ahmad, Y. Javed, M. Hussain, and M. Afzal. Mapping integrating the healthcare environment (IHE) to business process execution language for people (BPEL 4PPL). In *Multitopic Conference, 2009. INMIC 2009. IEEE 13th International*, pages 1–6, Dec. 2009. ISBN 978-1-4244-4872-2.

[AO08] P. Ammann and J. Offutt. *Introduction to Software Testing*. Cambridge University Press, New York, NY, USA, 2008. ISBN: 978-0-521-88038-1.

[ASM10] Microsoft Corporation. Abstract State Machine Language (AsML). http://research.microsoft.com/en-us/projects/asml/, 2010. Accessed 15 Jan. 2011.

[AST] ASTM International. ASTM E2369 - 05e1 Standard Specification for Continuity of Care Record (CCR). www.astm.org/Standards/E2369.htm. Accessed 15 Jan. 2011.

[BCKZ01] C. Besse, A. Cavalli, M. Kim, and F. Zaïdi. Two methods for interoperability tests generation: An application to the TCP/IP Protocol. www-lor.int-evry.fr/articles/main_input.ps, 2001.

[BDG+07] P. Baker, Z.R. Dai, J. Grabowski, Ø. Haugen, I. Schieferdecker, and C. Williams. *Model-Driven Testing: Using the UML Testing Profile*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[Ben09] T. Benson. *Principles of Health Interoperability HL7 and SNOMED (Health Informatics)*. Springer, 1 edition, December 2009.

[Ber08] Ensuring Interoperability with Automated Interoperability Testing. www.etsi.org/WebSite/document/aboutETSI/CTI/Ensuring%20Interoperability%20with%20Automated%20Interoperability%20Testing_rev6a.pdf, 2008. Accessed 15 Jan. 2011.

[BG94] B. Baumgarten and A. Giessler. *OSI Conformance Testing Methodology and TTCN*. Elsevier Science Inc., New York, NY, USA, 1994. ISBN 0-444-89712-7.

[BH10] P. Brooks and A. Health. Standards and Interoperability in Healthcare Information Systems: Current Status, Problems, and Research Issues. In *Proceedings of the Midwest Association for Information (MWAIS) 2010*, MWAIS. AIS Electronic Library (AISeL), 2010. http://aisel.aisnet.org/mwais2010/18/.

[Bin99] R. Binder. *Testing Object Oriented Systems. Models, Patterns and Tools.: Models, Patterns and Tools*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999. ISBN 0-201-80938-9.

[Bri03] P. Brittenham. Understanding the WS-I Test Tools – Test how the WS-I Basic Profile 1.0 defines interoperability. www.ibm.com/developerworks/webservices/library/ws-wsitest/, November 18 2003. Accessed 15 Jan. 2011.

[BUT+09] B. Bánfai, B. Ulrich, Z. Török, R. Natarajan, and T. Ireland. Implementing an HL7 Version 3 Modeling Tool from an Ecore Model. In K.-P. Adlassnig, B. Blobel, J. Mantas, and I. Masic, editors, *MIE*, volume 150 of *Studies in Health Technology and Informatics*, pages 157–161. IOS Press, 2009.

[CaF10] Care Fusion GmbH. AVEA Comprehensive Ventilator. www.viasyshealthcare.com/prod_serv/prodDetail.aspx?config=ps_prodDtl&prodID=284, 2010. Accessed 15 Jan. 2011.

[CAL10] CALLIOPE Project: CALL for InterOPErability, ICT Policy Support Programme (ICT PSP). www.calliope-network.eu, 2008–2010. Accessed 15 Jan. 2011.

[CCH04] The Certification Commission for Health Information Technology (CCHIT). www.cchit.org, 2004. Accessed 15 Jan. 2011.

[CEN00] European Committee for Standardization (CEN). CEN ENV 13608-1:2000, Health informatics - Security for healthcare communication - Concepts and terminology, 2000.

[CEN09] CEN Global eBusiness Interoperability Test-bed Methodologies (GITB). www.ebusiness-testbed.eu/publicaties/4752, 2008-2009. Accessed 15 Jan. 2011.

[CMZ04] A. Cavalli, A. Mederreg, and F. Zaïdi. Application of a Formal Testing Methodology to Wireless Telephony Networks. *International Journal of the Brazilian Computer Society (JCBS)*, 10(2):56–68, November 2004. ISSN 0104-6500.

[Com04] European Communities. European Interoperability Framework Version 1.0. http://ec.europa.eu/idabc/servlets/Docd552.pdf?id=19529, 2004. ISBN 92-894-8389-X. Accessed 15 Jan. 2011.

[Com08] European Communities. Draft document as basis for EIF 2.0. http://ec.europa.eu/idabc/servlets/Docb0db.pdf?id=31597, 2008. Accessed 15 Jan. 2011.

[Con10] IHE-Europe Connectathon. www.ihe-europe.net/content/connectathon_2010.htm, Bordeaux, France, 2010. Accessed 15 Jan. 2011.

[CTI10] European Telecommunications Standards Institute (ETSI), Centre for Testing and Interoperability (CTI). www.etsi.org/WebSite/AboutETSI/HowWeWork/Testingandinteroperability.aspx, 2010. Accessed 15 Jan. 2011.

[Dai06] Z. R. Dai. *An Approach to Model-Driven Testing - Functional and Real-Time Testing with UML 2.0, U2TP and TTCN-3*. PhD thesis, Technical University of Berlin, 2006. ISBN: 3-8167-7237-4 Fraunhofer Verlag IRB.

[DC02] J. Dahle and D.W. Callahan. Methodology for the development of an electronic medical record. In *System Theory, 2002. Proceedings of the Thirty-Fourth Southeastern Symposium on*, pages 406–411, 2002. ISSN 0094-2898.

[dCJB+10] E. C. A. de Carvalho, M. K. Jayanti, A. P. Batilana, A. M. O. Kozan, M. J. Rodrigues, J. Shah, M. R. Loures, S. Patil, P. Payne, and R. Pietrobon. Standardizing Clinical Trials Workflow Representation in UML for International Site Comparison. *PLoS ONE*, 5(11):e13893, 11 2010.

[DGM+08] R. Dingwell, A. Gregorowicz, R. McCready, H. Sleeper, T. Taylor, A. Trivedi, and D. Wilson. LAIKA Software Design Document. http://laika.sourceforge.net/docs/Laika_Design_Document.pdf, 2008. Accessed 15 Jan. 2011.

[DIC85] Medical Imaging & Technology Alliance, division of National Electrical Manufacturers Association (NEMA). Digital Imaging and Communications in Medicine (DICOM). Known as ISO Standard 12052 or NEMA Standard PS3. http://dicom.nema.org/, First DICOM standard published in 1985. Accessed 15 Jan. 2011.

[Din09] G. Din. *A Performance Test Design Method and its Implementation Patterns for Multi-Services Systems*. PhD thesis, Technical University of Berlin, 2009. ISBN: 978-3-8167-7973-5 Fraunhofer Verlag IRB.

[DK03] S. Dibuz and P. KrÃl'mer. Framework and Model for Automated Interoperability Test and Its Application to ROHC. In Dieter Hogrefe and Anthony Wiles, editors, *Testing of Communicating Systems*, volume 2644 of *Lecture Notes in Computer Science*, pages 607–607. Springer Berlin / Heidelberg, 2003. ISBN 3-540-40123-7.

[DK06] S. Dibuz and P. Kremer. An easy way to test interoperability and conformance. In *Testbeds and Research Infrastructures for the Development of Networks and Communities, 2006. TRIDENTCOM 2006. 2nd International Conference on*, page 408. IEEE, 2006. ISBN 1-4244-0106-2.

[DKG+06] A. Dogac, Y. Kabak, O. Gulderen, T. Namli, A. Okcan, O. Kilic, Y. Gurcan, U. Orhan, and G. Laleci. ebBP Profile for Integrating Healthcare Enterprise (IHE). www.srdc.metu.edu.tr/webpage/publications/2006/ebBPprofile-forIHE-1.0-draft1-July19-2006.pdf, 2006. Submitted to OASIS ebXML Business Process Technical Committee.

[Dog10] A. Dogac. Testing Conformance and Interoperability of eHealth Applications. *Methods of information in medicine*, 49(3):389–399, 2010. ISSN 0026-1270.

[DRU10] Drummond Group Inc. www.drummondgroup.com/html-v2/EHR, 2010. Accessed 15 Jan. 2011.

[DV05] A. Desmoulin and C. Viho. Formalizing Interoperability Testing: Quiescence Management and Test Generation. In Farn Wang, editor, *Formal Techniques for Networked and Distributed Systems - FORTE 2005*, volume 3731 of *Lecture Notes in Computer Science*, pages 533–537. Springer Berlin / Heidelberg, 2005. ISBN 3-540-29189-X.

[DV11] I. Schieferdecker D. Vega. *Automated Interoperability Testing of Healthcare Information Systems*. Elsevier Inc., 2011.

[DVS08] G. Din, D. Vega, and I. Schieferdecker. Automated Maintainability of TTCN-3 Test Suites based on Guideline Checking. In *SEUS '08: Proceedings of the 6th IFIP WG 10.2 international workshop on Software Technologies for Embedded and Ubiquitous Systems*, pages 417–430, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 978-3-540-87784-4.

[EAR+05] M. Eichelberg, T. Aden, J. Riesmeier, A. Dogac, and G.B. Laleci. A survey and analysis of Electronic Healthcare Record standards. *ACM Comput. Surv.*, 37(4):277–315, 2005. ISSN 0360-0300.

[EAR+06] M. Eichelberg, T. Aden, J. Riesmeier, A. Dogac, and G.B. Laleci. Electronic Health Record Standards - A Brief Overview. In *ITI 4th International Conference on Information & Communications Technology (ICICT' 2006)*. IEEEXplore Digital Library, 2006. Digital Object Identifier: 10.1109/ITICT.2006.358222.

[ebX07] Organization for the Advancement of Structured Information Standards (OASIS). ebXML Business Process Specification. http://docs.oasis-open.org/ebxml-bp/2.0.4/OS/spec/, 2007.

[ELT08] Elvior LLC. TestCast Generator. www.smartesting.com/index.php/cms/en/explore/products, 2008. Accessed 15 Jan. 2011.

[EPS11] epSOS Project: Smart Open Services for European Patients, partially funded under the ICT Policy Support Programme (ICT PSP). http://www.epsos.eu, 2008–2011. Accessed 15 Jan. 2011.

[ERA09] eHealth ERA: Towards the Establishment of a European eHealth Research Area, Study funded by the European Commission, DG Information Society and Media, ICT for Health Unit. www.ehealth-era.org, 2009. Accessed 15 Jan. 2011.

[ETI10] Etasoft Inc. Beginners Guide to EDI X12 (including HIPAA). www.xtranslator.com/prod/beginguidex12.pdf, 2010. Accessed 15 Jan. 2011.

[ETS] European Telecommunications Standards Institute (ETSI). PlugtestsTM Interop Events. www.etsi.com/WebSite/OurServices/plugtests/home.aspx.

[ETS03] European Telecommunications Standards Institute (ETSI) TS 102 237-1 (2003): Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) Release 4; Interoperability test methods and approaches; Part 1: Generic approach to interoperability testing, 2003. DTS/TIPHON-06025-1R4.

[ETS07a] European Telecommunications Standards Institute (ETSI). European Standard (ES) 201 873-1 V3.2.1 (2007-02): The Testing and Test Control Notation version 3; Part 1: TTCN-3 Core Language, 2007.

[ETS07b] European Telecommunications Standards Institute (ETSI). European Standard (ES) 201 873-5 V3.2.1 (2007-02): The Testing and Test Control Notation version 3; Part 5: TTCN-3 Runtime Interface (TRI), 2007.

[ETS07c] European Telecommunications Standards Institute (ETSI). European Standard (ES) 201 873-6 V3.2.1 (2007-02): The Testing and Test Control Notation version 3; Part 6: TTCN-3 Control Interface (TCI), 2007.

[ETS08] European Telecommunications Standards Institute (ETSI) TS 102 517: Methods for Testing and Specification (MTS); Internet Protocol Testing (IPT): IPv6 Core Protocol; Interoperability Test Suite (ITS), 2008.

[ETS09] European Telecommunications Standards Institute (ETSI) TS 186 011-2: Technical Committee for IMS Network Testing (INT);IMS NNI Interworking Test Specifications; Part 2: Test descriptions for IMS NNI Interworking, 2009.

[ETS10a] European Telecommunications Standards Institute (ETSI). TS 102 766 v1.1.1 (2009-10): ICT GRID Interoperability Testing Framework and survey of existing ICT Grid interoperability solutions. Technical Report DTR/GRID-0002, 2009-2010.

[ETS10b] European Telecommunications Standards Institute (ETSI). TS 102 786 v1.1.1 (2009-10): ICT GRID Interoperability Testing Framework. Technical Report DTR/GRID-0002, 2009-2010.

[ETS10c] European Telecommunications Standards Institute (ETSI). www.etsi.org, 2010.

[ETS10d] European Telecommunications Standards Institute (ETSI). European Standard (ES) 202 237: Methods for Testing and Specification (MTS); Internet Protocol Testing (IPT); Generic Approach to Interoperability Testing, version 1.2.1, 2010.

[Eur03] EUROREC Institute (EuroRec). www.eurorec.org, 2003. Accessed 15 Jan. 2011.

[FAC] United Nations Economic Commission for Europe (UNECE). United Nations Electronic Data Interchange for Administration, Commerce and Transport (UN/EDIFACT). www.unece.org/trade/untdid/down_index.htm. Accessed 15 Jan. 2011.

[Fer04] J. Fernando. Factors that have contributed to a lack of integration in health information system security. *The Journal of Information Technology in Healthcare*, 2(5):313–328, 2004. ISSN 1479-649X.

[Fow99] Martin Fowler. *Refactoring: Improving the Design of Existing Code*. Addison-Wesley, Boston, MA, USA, 1999. ISBN 0-201-48567-2.

[FSB09] M. A. Friedman, A. Schueth, and D. S. Bell. Interoperable Electronic Prescribing In The United States: A Progress Report. *Health Affairs - The Policy Journal of the Health Sphere*, 28(2):393–403, November 2009. Accessed 15 Jan. 2011.

[Fun06] The Commonwealth Fund. New International Survey of Primary Care Physicians: Most U.S. Doctors Unable to Provide Patients Access to After-Hours Care; Half Lack Access to Drug Safety Alert Systems. www.commonwealthfund.org/usr_doc/Press_Release3.pdf?section=4056, November 2006. Accessed 15 Jan. 2011.

[Gaz10] Gazelle Project: IHE Gazelle Test Management Tool. http://gazelle.ihe.net/, 2010. Accessed 15 Jan. 2011.

[GCS09] Y. Yuri Glickman, A. Ciemniewska, and I. Stergiou. White paper on technical interoperability software testing. www.qualipso.org/technical_interop_champion, 2009. Accessed 15 Jan. 2011.

[Gli10] Quality Platform for Open Source Software (QualiPSo). Systematic Technical Interoperability Testing. http://fossa2010.inrialpes.fr/files/2010/10/QualiPSo-Interoperability-fOSSa2010.pdf, November 8, 2010. Accessed 15 Jan. 2011.

[GS84] R.A. Greenes and E.H. Shortliffe. Medical informatics: an emerging academic discipline and institutional priority. *The structure of medical informatics. Medical Informatics*, (9):175–80, 1984.

[GS90] R.A. Greenes and E.H. Shortliffe. Medical informatics: an emerging academic discipline and institutional priority. *JAMA, The Journal of the American Medical Association*, 263(4):1114–1120, 1990. ISSN 0098-7484.

[HAP09] HAPI Project: HL7 Application Programming Interface. hl7api.sourceforge.net, 2009. Accessed 15 Jan. 2011.

[HHA96] A. Hasman, R. Haux, and A. Albert. A systematic view on medical informatics. *Computer Methods and Programs in Biomedicine*, 51(3):131–139, 1996. ISSN 0169-2607.

[Hin07] A. Hinchley. *Understanding Version 3 - A Primer on the HL7 Version 3 Interoperability Standard - Normative Edition*. Alexander Mönch Publishing, 4th (completely revised) edition, 2007.

[HIT11] HITCH Project: Healthcare Interoperability Testing and Conformance Harmonisation, partially funded by the European Commission under the 7th Framework Programme. www.hitch-project.eu, 2010–2011. Accessed 15 Jan. 2011.

[HL787a] Health Level Seven Standards Developing Organizations (HL7 SDOs). www.hl7.org, 1987. Accessed 15 Jan. 2011.

[HL787b] Health Level Seven Standards Developing Organizations (HL7 SDOs). Health Level Seven (HL7) Messaging Standards, Version 2.x, 3.0. www.hl7.org/implement/standards/ansiapproved.cfm, First HL7 Messaging Standard published in 1987. Accessed 15 Jan. 2011.

[HL787c] Health Level Seven Standards Developing Organizations (HL7 SDOs). Health Level Seven (HL7) Messaging Standards, Version 2.x. www.hl7.org/implement/standards/v2messages.cfm, First HL7 Version 2.x Standard published in 1987. Accessed 15 Jan. 2011.

[HL799] ANSI/HL7 V2.3.1-1999. Health Level Seven Standard Version 2.3.1 - An Application Protocol for Electronic Data Exchange in Healthcare Environments. http://www.hl7.org/implement/standards/ansiapproved.cfm?ref=nav, Release 14.04.1999. Accessed 15 Jan. 2011.

[HL703] ISO/HL7 27931:2009. Data Exchange Standards – Health Level Seven Version 2.5 – An application protocol for electronic data exchange in healthcare environments. www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=44428, Release 30.08.2003. Accessed 15 Jan. 2011.

[HL705] Health Level Seven Standards Developing Organizations (HL7 SD0s). Health Level Seven (HL7) Messaging Standards Standard, Version 3. http://www.hl7.org/v3ballot/html/index.htm, First HL7 Version 3 Standard published in 2005. Accessed 15 Jan. 2011.

[HL706] ISO/HL7 21731:2006. Health informatics – HL7 version 3 – Reference information model (RIM) – Release 1. www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=40399, 2006. Accessed 15 Jan. 2011.

[Hog90] D. Hogrefe. Conformance testing based on formal methods. In J. Quemada and A. Fernandez, editors, *FORTE 90 Formal Description Techniques*, 1990.

[IFW10] Info World Company - Connecting Healthcare. www.infoworld.ro/, 2010. Accessed 15 Jan. 2011.

[IHE97] Integrating the Healthcare Enterprise (IHE). www.ihe.net, 1997. Accessed 15 Jan. 2011.

[IIC11] OASIS ebXML Implementation Interoperability and Conformance (IIC) TC. www.oasis-open.org/committees/tc_home.php?wg_abbrev=ebxml-iic, 2011. Accessed 15 Jan. 2011.

[Inf10] InfoGard Laboratories. www.infogard.com/ig.nsf/html/EHR, 2010. Accessed 15 Jan. 2011.

[ISO92] ISO/IEC 9646-1 (1992) Information Technology - Open Systems Interconnection - Conformance Testing Methodology and Framework - Part 1: General concepts, 1992.

[ISO04] ISO/IEC Standard No. 9126: Software engineering – Product quality; Parts 1–4. International Organization for Standardization (ISO) / International Electrotechnical Commission (IEC), Geneva, Switzerland, 2001-2004.

[ISO05] ISO/TR 20514:2005, Health informatics – Electronic health record – Definition, scope and context, 2005.

[IST10] International Software Testing Qualifications Board (ISTQB) - Standard glossary of terms used in Software Testing, version 2.1, April 2010. www.astqb.org/documents/ISTQB_Glossary_of_Testing_Terms_2.1.pdf, 2010. Accessed 15 Jan. 2011.

[IST11] International Software Testing Qualifications Board (ISTQB). http://istqb.org/display/ISTQB/Home, 2011. Incorporated society at 1990 Brussels. Accessed 15 Jan. 2011.

[ITI09a] Integrating the Healthcare Enterprise (IHE). IT Infrastructure Technical Framework (ITI TF). www.ihe.net/Technical_Framework/index.cfm#IT, 2009. Accessed 15 Jan. 2011.

[ITI09b] Integrating the Healthcare Enterprise (IHE). IT Infrastructure Technical Framework, Volume 1 (ITI TF-1), Integration Profiles, Revision 6.0, Final Text, August 10, 2009. www.ihe.net/Technical_Framework/upload/IHE_ITI_TF_6-0_Vol1_FT_2009-08-10-2.pdf, 2009. Accessed 15 Jan. 2011.

[ITI09c] Integrating the Healthcare Enterprise (IHE). IT Infrastructure Technical Framework, Volume 2a (ITI TF-2a), Transactions Part A, Revision 6.0, Final Text, August 10, 2009. www.ihe.net/Technical_Framework/upload/IHE_ITI_TF_6-0_Vol2a_FT_2009-08-10-2.pdf, 2009. Accessed 15 Jan. 2011.

[ITI09d] Integrating the Healthcare Enterprise (IHE). IT Infrastructure Technical Framework, Volume 2x (ITI TF-2x), Appendices, Revision 6.0, Final Text, August 10, 2009. http://www.ihe.net/Technical_Framework/upload/IHE_ITI_TF_6-0_Vol2x_FT_2009-08-10.pdf, 2009. Accessed 15 Jan. 2011.

[IV07] I.K. Indrajit and B.S. Verma. DICOM, HL7 and IHE: A basic primer on Healthcare Standards for Radiologists. *Indian Journal of Radiology and Imaging*, 17(2):66–68, 2007. ISSN Print: 0971-3026, Online: 1998-3808. Accessed 15 Jan. 2011.

[Jos05] T. Joseph, editor. *E-Health Care Information Systems: An Introduction for Students and Professionals*. Jossey-Bass Inc., Publishers, 1st edition, 2005. ISBN 0-7879-6618-5.

[KAP09] C. Kuziemsky, N. Archer, and L. Peyton. Towards E-Health Interoperability: Challenges, Perspectives and Solutions. *Journal of Emerging Technologies in Web Intelligence*, 1(2), 2009.

[KC09] H. Kubicek and R. Cimander. Three dimensions of organizational interoperability - Insights from recent studies for improving interoperability frame-works. *European Journal of ePractice*, 6(6):3, 2009.

[KFN99] C. Kaner, J. Falk, and H. Q. Nguyen. *Testing Computer Software, Second Edition*. John Wiley & Sons, Inc., New York, NY, USA, 1999. ISBN 0-47135-846-0.

[KK97] S. Kang and M. Kim. Interoperability Test Suite Derivation for Symmetric Communication Protocols. In *FORTE X / PSTV XVII '97: Proceedings of the IFIP TC6 WG6.1 Joint International Conference on Formal Description Techniques for Distributed Systems and Communication Protocols (FORTE X) and Protocol Specification, Testing and Verification (PSTV XVII)*, pages 57–72, London, UK, 1997. Chapman & Hall, Ltd. ISBN 0-412-82060-9.

[KSK00] S. Kang, J. Shin, and M. Kim. Interoperability test suite derivation for communication protocols. *Computer Networks*, 32(3):347–364, 2000.

[L. 08] L. Gebase and R. Snelick and M. Skall. Conformance Testing and Interoperability: A Case Study in Healthcare Data Exchange. In *Proceedings of the 2008 International Conference on Software Engineering Research & Practice, SERP 2008, July 14-17, 2008, Las Vegas Nevada, USA, 2 Volumes*, pages 143–151. CSREA Press, 2008. ISBN 1-60132-088-4.

[LAI10] LAIKA Project: open source electronic health record (EHR) testing framework. www.laika.sourceforge.net, 2010. Accessed 15 Jan. 2011.

[MA05] P. Mäki-Asiala. Reuse of TTCN-3 Code. Master's thesis, VTT Technical Research Centre of Finland, Helsinki, 2005.

[MAI09] Quality Platform for Open Source Software (QualiPSo). Methodology using Model for Automated Interoperability Testing (MAIT). http://forge.morfeo-project.org/docman/view.php/83/483/4.%20MAIT.pdf, 2009. Accessed 15 Jan. 2011.

[McC08] R. McCready. LAIKA: An Open Source Electronic Health Record Testing Framework . www.mitre.org/news/events/exchange08/3813.pdf, 2008. Accessed 15 Jan. 2011.

[Men10] Object Mentor. JUnit.org Resources for Test Driven Development. www.junit.org, 2010. Accessed 15 Jan. 2011.

[MGM07] K. Merouane, C. Grepet, and S. Maag. A Methodology for Interoperability Testing of a MANET Routing Protocol. *Wireless and Mobile Communications*, 0:5, 2007. ISBN: 0-7695-2796-5. Accessed 15 Jan. 2011.

[MIJ99] Sun Microsystems, Inc, and Javasoft. *Java Look & Feel Design Guidelines*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

[MLL09] Health Level Seven. Transport Specification: Minimal Lower Layer Protocol (MLLP), Release 2. www.hl7.org/v3ballot/html/infrastructure/transport/transport-mllp.htm, Last Published on 12.06.2009. Accessed 15 Jan. 2011.

[MOT10] FOKUS Fraunhofer Institute for Open Communication Systems, Competence Center MOTION, Modeling and Testing for System and Service Solutions. www.fokus.fraunhofer.de/en/motion/index.html, 2010. Accessed 15 Jan. 2011.

[MT08] J. A. Mykkänen and M. P. Tuomainen. An evaluation and selection framework for interoperability standards. *Inf. Softw. Technol.*, 50(3):176–197, 2008. ISSN 0950-5849.

[NAD09] T. Namli, G. Aluc, and A. Dogac. An interoperability test framework for HL7-based systems. *Trans. Info. Tech. Biomed.*, 13(3):389–399, May 2009. ISSN 1089-7771.

[Neu04] H. Neukirchen. *Languages, Tools and Patterns for the Specification of Distributed Real-Time Tests*. PhD thesis, Dissertation, Universität Göttingen, November 2004 (electronically published on http://webdoc.sub.gwdg.de/diss/2004/neukirchen/index.html and archived on http://deposit.ddb.de/cgi-bin/dokserv?idn=974026611 . Persistent Identifier: urn:nbn:de:gbv:7-webdoc-300-2), Nov 2004.

[NIH06] National Institutes of Health (NIH) National Center for Research Resources. MITRE Center for Enterprise Modernization. Electronic Health Records Overview. www.ncrr.nih.gov/publications/informatics/EHR.pdf, April 2006. Accessed 15 Jan. 2011.

[NIS03] National Institute of Standards and Technology (NIST)- Manufacturing Systems Integration Division. OAG/NIST B2B Testbed Overview - Meeting with ETeB

Delegation. www.mel.nist.gov/msid/b2btestbed/b2bTestbed_11_24.pdf, november 2003. Accessed 15 Jan. 2011.

[NR10] R. Noumeir and B. Renaud. IHE cross-enterprise document sharing for imaging: interoperability testing software. *Source Code for Biology and Medicine*, 5(1):9, 2010.

[NZ07] H. Neukirchen and B. Zeiß. Automation of Refactoring and Refactoring Suggestions for TTCN-3 Test Suites – The TRex TTCN-3 Refactoring and Metrics Tool (Extended Abstract). In *Proceedings of the* 1^{st} *Workshop on Refactoring Tools held in conjunction with* 21^{st} *European Conference on Object-Oriented Programming (ECOOP 2007), July 31st 2007, Berlin. (Editors: Danny Dig, Michael Cebulla). Technical Report No 2007-8, Technische Universität Berlin, Fakultät IV - Elektrotechnik und Informatik, ISSN 1436-9915*, July 2007.

[oD09] HIMSS Board of Directors. Interoperability Definition and Background. www.himss.org/content/files/interoperability_definition_background_060905.pdf, 2009. Accessed 15 Jan. 2011.

[ORLS06] J. C. Okika, A. P. Ravn, Z. Liu, and L. Siddalingaiah. Developing a TTCN-3 test harness for legacy software. In *AST '06: Proceedings of the 2006 international workshop on Automation of software test*, pages 104–110. ACM Press, 2006. Shanghai, China. ISBN 1-59593-408-1.

[PCD06a] Integrating the Healthcare Enterprise (IHE). Patient Care Devices Technical Framework (PCD TF). www.ihe.net/Technical_Framework/index.cfm#pcd, 2006. Accessed 15 Jan. 2011.

[PCD06b] Integrating the Healthcare Enterprise (IHE). Patient Care Devices Technical Framework (PCD TF), Volume 1, Integration Profiles, Revision 1.1, Trial Implementation Version, Year 1: 2005-2006. www.ihe.net/Technical_Framework/upload/IHE_PCD_TF_rev1.pdf, 15 Aug. 2006. Accessed 15 Jan. 2011.

[PCD06c] Integrating the Healthcare Enterprise (IHE). Patient Care Devices Technical Framework (PCD TF), Volume 2, Transaction, Revision 1.1, Trial Implementation Version, Year 1: 2005-2006. www.ihe.net/Technical_Framework/upload/IHE_PCD_TF_rev1-2.pdf, 15 Aug. 2006. Accessed 15 Jan. 2011.

[Pia08] O.S. Pianykh. *Digital Imaging and Communications in Medicine (DICOM): A Practical Introduction and Survival Guide*. Springer-Verlag Berlin Heidelberg, Tiergartenstr. 17, D-69121 Heidelberg, Germany, 1st edition, 2008. ISBN 978-3-540-74570-9.

[PPP08] G. Pentafronimos, S. Papastergiou, and N. Polemi. Interoperability testing for e-government web services. In *Proceedings of the 2nd international conference on Theory and practice of electronic governance*, ICEGOV '08, pages 316–321, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-386-0.

[PRR+09] M. Pascale, M. Roselli, U. Rugani, C. Bartolini, A. Bertolino, F. Lonetti, E. Marchetti, and A. Polini. Automated testing of healthcare document transformations in the PICASSO interoperability platform. In *31st International Conference on Software Engineering, ICSE 2009, May 16-24, 2009, Vancouver, Canada,*

Companion Volume, pages 163–171. IEEE Communications Society, IEEEXplore, 2009.

[QPS09] Quality Platform for Open Source Software (QualiPSo). ETSI Generic Approach to Interoperability Testing methodology (GAIT). http://forge.morfeo-project.org/docman/view.php/83/481/2.%20GAIT.pdf, 2009. Accessed 15 Jan. 2011.

[QTR10] Conformiq Inc. Conformiq Tool Suite ™, Eclipse based workbench for automated test design. www.conformiq.com/products.php, 2010. Accessed 15 Jan. 2011.

[ReT09] ReTeMes Project: Reliability Testing Of Medical Systems, EUREKA European Project (E!4053 RETEMES). www.eurekanetwork.org/project/-/id/4053, 2007-2009. Accessed 15 Jan. 2011.

[RGS10] T. Rings, J. Grabowski, and S. Schulz. On the Standardization of a Testing Framework for Application Deployment on Grid and Cloud Infrastructures. In *Proceedings of the 2nd International Conference on Advances in System Testing and Validation Lifecycle (VALID 2010), August 22-27 2010, Nice, France*, pages 99–107. IEEE, August 2010. ISBN 9-781-4244-7784-5.

[RID07a] RIDE Project: A Roadmap for Interoperability of eHealth Systems in Support of COM 356 with Special Emphasis on Semantic Interoperability. www.srdc.metu.edu.tr/webpage/projects/ride/, 2006-2007. Accessed 15 Jan. 2011.

[RID07b] RIDE Project: A Roadmap for Interoperability of eHealth Systems. www.srdc.metu.edu.tr/webpage/projects/ride/, 2007. Accessed 15 Jan. 2011.

[RIN09] Towards systematic interoperability testing of distributed systems. Ph.D. Symposium at the 2009 International Conference on Software Testing, Verification, and Validation (ICST 2009)., 2009. Denver, Colorado, USA.

[RJHSD10] C. Rinner, S. Janzek-Hawlat, S. Sibinovic, and G. Duftschmid. Semantic validation of standard-based electronic health record documents with W3C XML Schema. *Methods of information in medicine*, 49(3):271–280, 2010. ISSN 0026-1270.

[RNG08] T. Rings, H. Neukirchen, and J. Grabowski. Testing Grid Application Workflows Using TTCN-3. *Software Testing, Verification, and Validation, 2008 International Conference on Software Testing, Verification, and Validation (ICST 2008). Lillehammer , Norway, April 9-11 2008*, 0:210–219, 2008.

[Sar04] S. N. Sarbadhikari. Basic medical science education must include medical informatics. *Indian J Physiol Pharmacol*, 48(4):395–408, 2004. ISSN 0019-5499.

[SC06] E.H. Shortliffe and J.J. Cimino, editors. *Biomedical Informatics: Computer Applications in Health Care and Biomedicine, 3rd Edition*. Springer Science - Business Media LLC, 233 Spring Street, New York, 10013, USA, June 2006. ISBN 0-38728-986-0.

[SCA11] National Council for Prescription Drug Programs (NCPDP). ePrescribing Industry Information. www.ncpdp.org/eprescribing.aspx, 2011. Accessed 15 Jan. 2011.

[Sch03] M. Schmitt. *Automatic Test Generation Based on Formal Specifications - Practical Procedures for Eficient State Space Exploration and Improved Representation of Test Cases*. PhD thesis, Georg-August-Universität Göttingen, 2003.

[Sch10] I. Schieferdecker. Test Automation with TTCN-3 - State of the Art and a Future Perspective. In *Testing Software and Systems - 22nd IFIP WG 6.1 International Conference, ICTSS 2010, Natal, Brazil, November 8-10, 2010. Proceedings*, Lecture Notes in Computer Science, pages 1–14. Springer, 2010. ISBN 978-3-642-16572-6.

[SCR11] National Council for Prescription Drug Programs (NCPDP). NCPDP Electronic Prescribing Standards. www.ncpdp.org/pdf/NCPDP.eprescribing.101.ppt, 2011. Accessed 15 Jan. 2011.

[SEM08] SemanticHEALTH Project: Sharing knowledge in e-Health Information Systems, funded by the 6th R&D Framework Programme (FP6). www.semantichealth.org, 2006–2008. Accessed 15 Jan. 2011.

[SEP09] sepp.med GmbH. .getmore (GEnerating Tests from MOdels to Reduce Effort), UML based Test Generator. www.seppmed.de/getmore.65.0.html, 2009. Accessed 15 Jan. 2011.

[Ser07] Geotechnical Software Services. Unit Testing Guidelines, Version 4. www.geosoft.no/development/unittesting.html, April 2007. Accessed 15 Jan. 2011.

[Sho84] E.H. Shortliffe. The science of biomedical computing. *Medical Informatics*, (9):185–93, 1984.

[SL09] R.D. Sriram and B. Lide. The role of standards in healthcare automation. In *Proceedings of the fifth annual IEEE international conference on Automation science and engineering*, CASE'09, pages 79–82, Piscataway, NJ, USA, 2009. IEEE Press. ISBN 978-1-4244-4578-3.

[SME10] sepp.med GmbH. www.seppmed.de, 2010. Accessed 15 Jan. 2011.

[SOD+09] C. Schoen, R. Osborn, M. M. Doty, D. Squires, J. Peugh, and S. Applebaum. A Survey of Primary Care Physicians in 11 Countries, 2009: Perspectives on Care, Costs, and Experiences. *Health Affairs (Project HOPE)*, 28(6):1171–1183, November 2009. http://dx.doi.org/10.1377/hlthaff.28.6.w1171. Accessed 15 Jan. 2011.

[SOH+06] C. Schoen, R. Osborn, Phuong T. Huynh, M. Doty, J. Peugh, and K. Zapert. On The Front Lines Of Care: Primary Care Doctors' Office Systems, Experiences, And Views In Seven Countries. *Health Affairs - The Policy Journal of the Health Sphere*, 25(6):w555–w571, November 2006. eISSN 1544-5208. http://content.healthaffairs.org/content/25/6/w555.full. Accessed 15 Jan. 2011.

[SPE10] Microsoft Corporation. Spec Explorer™. http://research.microsoft.com/en-us/projects/specexplorer/, 2010. Accessed 15 Jan. 2011.

[Spr06] R. Spronk. The Role Agent: a new middleware concept. Version 1.3 (Final). www.ringholm.de/docs/00400_en.htm, November 15 2006. Accessed 15 Jan. 2011.

[STD08] Smartesting. Test Designer ™. www.smartesting.com/index.php/cms/en/explore/products, 2008. Accessed 15 Jan. 2011.

[Ste04] H. Stevens. HL7 Version 3 - From Design to Implementation, Part 1 - Design. www.cihi.ca/cihiweb/en/downloads/v3Advanced_AM_e.pdf, April 22-23 2004. HL7 Canada Meeting and Tutorials, Calgary, Alberta. Accessed 15 Jan. 2011.

[Str00] B. Stroustrup. *The C++ Programming Language*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 3rd edition, 2000.

[Sul07] Frost & Sullivan. E-Prescription in Europe: First Things Fast. www.frost.com/prod/servlet/market-insight-top.pag?docid=102824378, 30 Jul 2007. Accessed 15 Jan. 2011.

[Sul09] Frost & Sullivan. E-Healthcare in Western Europe A Huge Market Opportunity for Wireless Technologies, October 2009.

[TNG09] TestNGMed Project: Testing of Next Generation Medical System, 2007-2009, PRO INNO Germany. www.testngmed.org, 2007-2009. Accessed 15 Jan. 2011.

[TPI10] TIPI - Testing Internet Protocols Interoperability. www.irisa.fr/tipi/wiki/doku.php/home, 2010. Accessed 15 Jan. 2011.

[TTE09] Testing Technologies IST GmbH. TTworkbench Professional, an Eclipse based TTCN-3 IDE, Version 1.1.9. www.testingtech.com/products/ttworkbench.php, 2009. Accessed 15 Jan. 2011.

[TUB10] Technische Universität Berlin, ETS - Entwurf und Testen kommunikationsbasierter Systeme. www.ets.tu-berlin.de, 2010. Accessed 15 Jan. 2011.

[TW98] J. Grabowski T. Walter, I. Schieferdecker. Test Architectures for Distributed Systems - State of the Art and Beyond. In A.Petrenko, editor, *International Federation for Information Processing -IFIP-, Technical Committee 6, Communication Systems: Testing of communicating systems : Proceedings of the IFIP TC6 11th International Workshop on Testing of Communicating Systems (IWTCS'98) August 31 - September 2, 1998, Tomsk, Russia*, pages 149–174, 1998. ISBN 0-412-84430-3.

[UL06] M. Utting and B. Legeard. *Practical Model-Based Testing: A Tools Approach*. Morgan Kaufmann Publishers Inc., San Francisco, USA, 2006. ISBN 0-1237-2501-1.

[UPB10] Politehnica University of Bucharest, Faculty of Automatic Control and Computers. www.upb.ro/index.php/en/faculties-and-departments/faculties/automatic-control-and-computers, 2010. Accessed 15 Jan. 2011.

[VBT01] C. Viho, S. Barbin, and L. Tanguy. Towards a Formal Framework for Interoperability Testing. In *Proceedings of the IFIP TC6/WG6.1 - 21st International Conference on Formal Techniques for Networked and Distributed Systems*, FORTE '01, pages

53–68, Deventer, The Netherlands, The Netherlands, 2001. Kluwer, B.V. ISBN 0-7923-7470-3.

[VDS10] D. Vega, G. Din, and I. Schieferdecker. Application of TTCN-3 Test Language to Testing Information Systems in eHealth Domain. In *MCIT 2010: Proceedings of the International Conference on Multimedia Computing and Information Technology, University of Sharjah, U.A.E.*, pages 123–130. IEEE Communications Society, IEEEXplore, 2010. ISBN 0-909925-88-7.

[Veg10] D. E. Vega. Towards an Automated and Dynamically Adaptable Test System for Testing Healthcare Information Systems. In *Proceedings of the 2010 Third International Conference on Software Testing, Verification and Validation*, ICST, pages 331–334, Washington, DC, USA, 2010. IEEE Computer Society. ISBN 978-0-7695-3990-4.

[VSD08] D. Vega, I. Schieferdecker, and G. Din. A TTCN-3 based Test Automation Framework for HL7-based Applications and Components. In *CONQUEST 2008: Proceedings of the Conference on Quality Engineering in Software Technology*, 2008.

[VSD10] D. Vega, I. Schieferdecker, and G. Din. Design of a Test Framework for Automated Interoperability Testing of Healthcare Information Systems. In *eTELEMED 2010: Proceedings of the Second International Conference on eHealth, Telemedicine, and Social Medicine, St. Maarten, Netherlands Antilles*, pages 123–130. IEEE Computer Society, 2010. ISBN 0-909925-88-7.

[VW06] H. Veer and A. Wiles. European Telecommunications Standards Institute (ETSI): Achieving Technical Interoperability - the ETSI Approach. http://ec.europa.eu/idabc/servlets/Docd552.pdf?id=19529, 2006. ETSI White Paper No. 3. Accessed 15 Jan. 2011.

[WDT+05] C. Willcock, T. Deiß, S. Tobies, S. Schulz, S. Keil, and F. Engler. *An Introduction to TTCN-3* . John Wiley & Sons, 1 edition, April 2005.

[Wil03] A. Wiles. Relevance of Conformance Testing for Interoperability Testing. Copyright ETSI. www.acats-forum.org/pdf/anthony_wiles.pdf, 2003. ACATS ATS-CONF, Stuttgart 20-21 January, 2003.

[WLG09] K. A. Wager, F. W. Lee, and J. P. Glaser. *Health Care Information Systems: A Practical Approach for Health Care Management*. Jossey-Bass, 2nd edition, June 2009.

[Woo07] J. Woo. *Agile Test Methodology for B2C/B2B Interoperability*. PhD thesis, Department of Industrial and Management Engineering Pohang University of Science & Technology, 2007.

[WSI04] Web Service Interoperability Organization (WS-I). www.ws-i.org, 2004. Accessed 15 Jan. 2011.

[WST03] P. Brittenham. Understanding the WS-I Test Tools – Test how the WS-I Basic Profile 1.0 defines interoperability. www.ibm.com/developerworks/webservices/library/ws-wsitest/, 2003. Accessed 15 Jan. 2011.

[Zei10] B. Zeiß. *Quality Assurance of Test Specifications for Reactive Systems*. PhD thesis, Dissertation, Universität Göttingen, Juni 2010 2010.

[ZNG+06] B. Zeiß, H. Neukirchen, J. Grabowski, D. Evans, and P. Baker. Refactoring and Metrics for TTCN-3 Test Suites. In *System Analysis and Modeling: Language Profiles. 5^{th} International Workshop, SAM 2006, Kaiserslautern, Germany, May 31 - June 2, 2006, Revised Selected Papers. Gotzhein, Reinhard, Reed, Rick (Eds.)*, Lecture Notes in Computer Science, Vol. 4320, Computer Communication Networks and Telecommunications., pages 148–165. Springer, December 2006. ISBN 978-3-540-68371-1.

[ZVS+07] B. Zeiß, D. Vega, I. Schieferdecker, H. Neukirchen, and J. Grabowski. Applying the ISO 9126 Quality Model to Test Specifications – Exemplified for TTCN-3 Test Specifications. In *Software Engineering 2007 (SE 2007). Lecture Notes in Informatics (LNI) 105. Copyright Gesellschaft für Informatik*, pages 231–242. Köllen Verlag, Bonn, March 2007.