Analyzing Requirements for a Large Scale System for Cancer Research

Vito Perrone, Anthony Finkelstein
University College London
{v.perrone,a.finkelstein}@cs.ucl.ac.uk

Abstract

While most of the attention on building systems enabling semantic interoperability has been devoted to technical issues, human and organizational aspects are of equally if not higher importance. In this paper we focus on these aspects by recounting our experience and lessons learned working to the development of an innovative system in the cancer research domain.

1. Introduction

It is well recognized, in the cancer community, that the crucial step towards improving the research is to enable *easy access* and *interoperability* among the existing initiatives developing data repositories and services. In this light, in the UK, a specific initiative – namely the NCRI (National Cancer Research Institute) Informatics Initiative [2] – has been set out with the goal of increasing the impact of UK cancer research and improving prevention and treatment of cancer by effective use of informatics resources. A clear foundation to achieving this goal is to enable the development of an informatics platform that facilitates access to, integration and movement of, data generated from research funded by NCRI partner organisations. Given the complexity of the envisioned project, the platform development has been preceded by a project focusing on the requirements analysis. A multidisciplinary analysis group has been set up and a number of use cases – covering several cancer research sub-domains acquired by interviewing practitioners.

Alike most of the informatics literature in the biomedical which focuses on technical issues, in this paper we recount our experience in the requirements analysis and early architecture design for the NCRI Platform by focusing on human and organizational aspects which are, in our view, equally critical for the success of such a complex project. In particular, we introduce the approach we have used to collect and analyze a number of use cases and the key aspects of the adopted methods. Our approach and methods have shown to be effective in supporting the communication with busy domain experts and to unveil their "pains" and expectations in the system being developed.

2. The project context

Two major initiatives have been established in the US and the UK to enable large scale interoperability among the various initiatives operating in the cancer research domain. In the US, the NCI (National Cancer Institute) has funded the first large scale project, namely caBIG™ (Biomedical Informatics Grid) [1], aiming to build a global infrastructure to enable large scale integration of cancer research initiatives funded by the NCI. To enable the interoperability among these initiatives, caBIG provide a GRID infrastructure and a number of components including a large and very detailed ontology (namely the NCI-Thesaurus) and a metadata repository (caDSR). These components make up the semantic core of the network and are centrally maintained by the organization. Conversely, the system envisaged by the NCRI (National Cancer Research Institute) [2] in the UK aims to reuse as much as possible of the existing resources, including those provided by the NCI, to support the creation of an open and sustainable community.

The envisaged system (namely the NCRI Platform) aims to integrate and make accessible data sets and services produced by different research groups around the world working across the cancer research spectrum. They are subject to different access and use policies, have been recorded and often made accessible (typically via web sites) in non-standardized ways, that is, using different vocabularies, data structures, metadata standards and service interfaces. While there are not standards broadly adopted, a number of initiatives have been established to define common *ontologies* and *data models* which may have been adopted by some data repositories. Given the uncontrolled way this domain has developed, typical of an Interned based community, different organizations may have adopted different terminologies and data formats among those available in the domain. For instance, one of the most known ontologies, the GO ontology, has been adopted by hundreds of projects to annotate data about genes and the CDISC data format is broadly adopted for the exchange of clinical information.

3. The analysis approach: enhancing multidisciplinary communication

Since the very beginning of our project, active involvement of the community and the need of building a multidisciplinary team including requirements engineers, software developers and domain experts with a broad vision on the various sub-domains, have been considered two crucial success factors. In the initial phase, cancer related literature analysis and periodic team meetings have been intertwined to build up a common understanding of the platform's high level goals and to define a common language. Outcomes of the preliminary activities have been a context diagram, an initial set of stakeholders and a preliminary domain model. Subsequently, the main objectives of the analysis activities, shown in Figure 1, have been: (i) identifying the integration needs of researchers and understanding the role different resources described in the context diagram play (or could play) in the analyzed investigations; (ii) defining the high level architecture of the system to-be that will drive the following design and implementation activities. An introductory description and some considerations on the methods we have adopted to achieve these objectives are reported in the next sections.

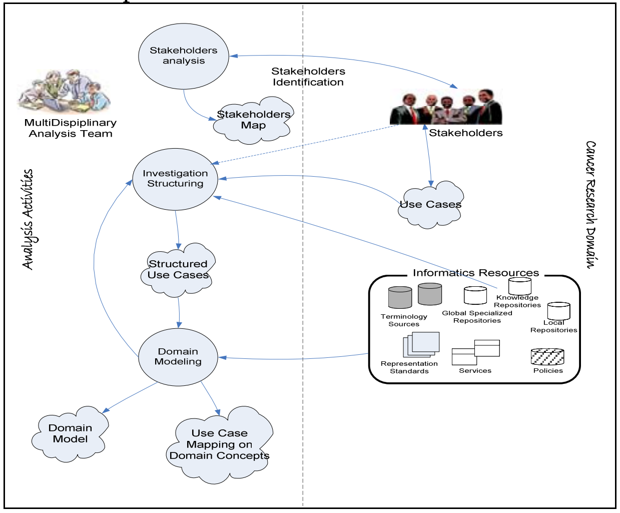
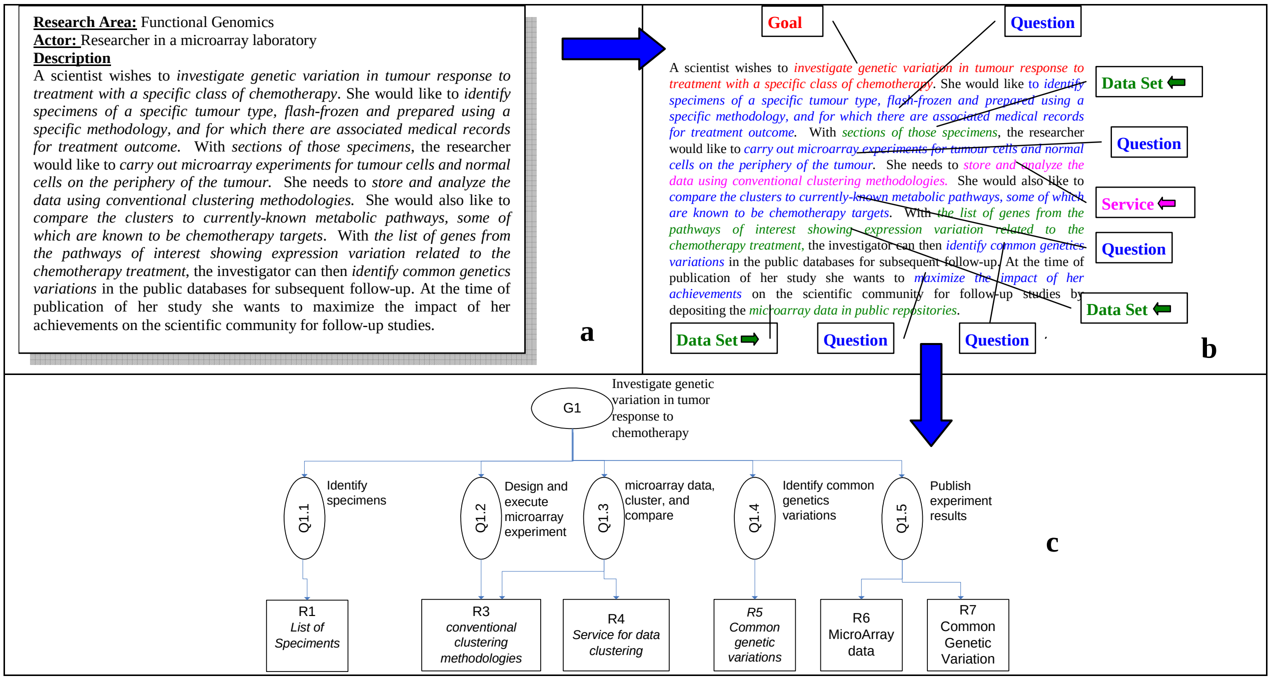


Figure 1: The use case analysis flow

4. Using an High Level Investigation Model in the Use Case Analysis

The collected use cases are examples of investigations researchers perform in their daily work and are akin to "user stories" in the Agile [4] approach rather than UML use cases which tend to describe the functionalities of the system to-be. Working side by side with domain experts to define such stories, we encouraged them to avoid thinking how the platform could satisfy their research needs and to instead carefully state their goals and what they would ask the system to achieve their goals. Such an approach permits uncovering of real user needs and avoids potential biases introduced by having in mind premature solutions.



Research Area: Functional Genomics
Actor: Researcher in a microarray laboratory
Description

A scientist wishes to *investigate genetic variation in tumour response to treatment with a specific class of chemotherapy*. She would like to *identify specimens of a specific tumour type, flash-frozen and prepared using a specific methodology, and for which there are associated medical records for treatment outcome.* With *sections of those specimens*, the researcher would like to *carry out microarray experiments for tumour cells and normal cells on the periphery of the tumour.* She needs to *store and analyze the data using conventional clustering methodologies.* She would also like to *compare the clusters to currently-known metabolic pathways, some of which are known to be chemotherapy targets*. With *the list of genes from the pathways of interest showing expression variation related to the chemotherapy treatment*, the investigator can then *identify common genetics variations* in the public databases for subsequent follow-up. At the time of publication of her study she wants to maximize the impact of her achievements on the scientific community for follow-up studies.

A scientist wishes to *investigate genetic variation in tumour response to treatment with a specific class of chemotherapy*. She would like to *identify specimens of a specific tumour type, flash-frozen and prepared using a specific methodology, and for which there are associated medical records for treatment outcome*. With *sections of those specimens*, the researcher would like to *carry out microarray experiments for tumour cells and normal cells on the periphery of the tumour*. She needs to *store and analyze the data using conventional clustering methodologies.* She would also like to *compare the clusters to currently-known metabolic pathways, some of which are known to be chemotherapy targets*. With *the list of genes from the pathways of interest showing expression variation related to the chemotherapy treatment*, the investigator can then *identify common genetics variations* in the public databases for subsequent follow-up. At the time of publication of her study she wants to *maximize the impact of her achievements* on the scientific community for follow-up studies by depositing the *microarray data in public repositories*.

a
Goal
Question
Data Set
Question
Service
Question
Data Set
Data Set
Question
Question
b
G1
Investigate genetic variation in tumor response to chemotherapy
Q1.1
Identify specimens
Q1.2
Design and execute microarray experiment
Q1.3
microarray data, cluster, and compare
Q1.4
Identify common genetics variations
Q1.5
Publish experiment results
c
R1 List of Speciments
R3 conventional clustering methodologies
R4 Service for data clustering
R5 Common genetic variations
R6 MicroArray data
R7 Common Genetic Variation

Figure 2: An example from the use case collection and progressive analysis

An example of use case is shown in Figure 2a as collected from the field. It is unstructured and hard to understand for non-specialists like the computer scientists in charge of the analysis activities. An important step in the analysis is to structure the use cases so that the investigation's goals, flow and the informatics resources needed by researchers can be clearly identified (Investigation Structuring activity in Figure 1). A key aspect of our analysis has been the definition of a high level model to describe research investigations. In defining such a model we have addressed three key questions: what are the objectives of this investigation? If the NCRI Platform were available, how would the researcher use it? What are the existing resources that can be used to address the user needs and how would the system use them? In defining the model we had to compromise between two conflicting goals: we needed to work together with the domain experts to break down the use cases and we needed a systematic approach to structure them in a way suitable for our analysis. In other words, the model was required to be easy to understand by people with no information modelling background but systematic enough to enable systematic analysis. The model we have defined, namely GQR (Goal Question Results) [8], is grounded by goal oriented requirements engineering principles [5] and is inspired by the Goal/Question/Metric method [6] used to plan the measurement of success of a software system from the user point of view. Lack of space prevents us to describe the model and all its elements so we can only introduce the key ones, that is, goal, question and result. A ***goal*** represents the investigation's objective. Goals can be more or less specific like "the role of diet in cancer" or "investigate whether a disease responds to a drug". A goal entails one or more ***questions*** that must be answered to achieve its fulfilment. Questions are answered by way of *data sets* or *services* which may need to be integrated with one another to produce the actual ***results*** the researcher was looking for. Data sets and services can be considered either inputs or outputs of an investigation.

Using the GQR method to analyze the use cases, the experts and the analysis team work together to progressively structure the use case according to the concepts contained in the GQR model. Its simplicity (easy to explain to the interviewees) and the limited number of concepts allow a first quick structuring to be carried out together with the domain expert on the initial description (figure 2b), preferably on paper. In our experience, it is important to have a rapid way to structure the use case and to enable further analysis without having to organize several further meetings. Once *goals* and *questions* have been identified, a number of elicitation questions, gradually more specific, are asked to the experts. Examples of these questions include: “What is your desiderata about this question”; “What are the current typical obstacles”; “Examples of repositories, records, etc.”, and so forth. The GQR method has shown up to be effective in supporting the conversation between the analysts and the domain experts in this phase. The information collected in the interview is then organized in a more structured way as shown in figure 2c. Once the desired *results* have been identified, these have to be described in terms of common domain concepts so that all the use cases can be compared (Domain Modelling in Figure 1). This required the definition of a domain model as described in the next section. Important is to notice that describing the results by way of domain concepts may require several iterations. The domain experts who provided the use case may not be always available in all the iterations thus the domain experts in the multidisciplinary team play a key role in this activity.

5. Domain Modelling

The domain model has played three crucial roles in our project: (1) has acted as a bridge between the problem analysis and the solution design so it is used to ensure that the analysis that went into it applies to the final product, the software system; 2) has been the *backbone* of the language used by all the team members, including analysts, domain experts and developers; (3) has been the teams’ agreed-upon way to structure the domain knowledge so that when new knowledge surfaces the domain model can be used to interpret it or to identify uncovered aspects. Defining the domain model is an iterative and incremental activity where rigorous analysis of meeting minutes, use cases, documentations, etc. by means of engineering techniques is intertwined with discussions with domain experts to reach a common vision.

We had to face two main challenges which we believe are common to similar projects aiming to develop integrative systems in a complex and open environment. First, the model should easily accommodate change which is likely to occur in an open and evolving environment such as the cancer research. This requires the model to be extensible and flexible. Fine-grained and very specialized models are hardly extensible and it is often very difficult (if not impossible) to reach an agreement in heterogeneous teams involving domain experts with different specializations. In this light, a requirement on our domain model has been to be sufficiently generic and to accommodate different points of view, while identifying the key entities and relationships. Second, several different points of view [7] need to be considered in the analysis. As far as the perspective changes, different typologies of entities and relationships may be required to describe the domain. For instance, in the use case in Figure 2 words such as “specimens”, “experiments” refer to the process of carrying out an investigation, whereas “gene”, “pathway” to the cancer biology. An important issue we have identified in this project is that, in order to clearly identify all the needed information, a multi-perspective analysis is needed. This entails a multi-dimensional domain model to be defined so that each dimension can evolve separately. In our project, the domain model has been organized across three dimensions: the ***cancer research*** including all the concepts involved in the investigation/experiment execution such as *samples, patient data, protocols, publishable data*, etc.; the ***cancer biology*** including concepts like *tumour, drug, gene, pathway, etc.;* the ***system integrator*** which models the environment where the platform will operate, the different types of available resources and their relationships such as *bioinformatics repository, ontology, registry, data format, service, etc.*. As a general consideration, the first dimension is specific of the system we intend to develop (to support research investigations), the second one is specific of the reference domain and the last one represents the informatics point of view.

In our analysis, the domain model entities have been used to describe the results required to answer the researcher's questions as identified by the GQR analysis. As an example of analysis, Q1.4 (use case in figure 2) can be answered by the result R5 which represents the integration of data coming from different repositories. From the *biological* (CB) perspective the result can be described by saying that "the platform should query repositories known to contain information about *CB_Gene, CB_Pathways, CB_ExpressionVariation and CB_Agent* (where a class of chemotherapy is considered an agent)". From the *system integrator (SI)* perspective, these repositories are modelled as *SI_KnowledgeBases* (e.g. "REACTOME" and "PharmGKB") and *SI_ScientificLibrary* (e.g. PubMed). The information is *semantically annotated* by metadata elements (*SI_MetaDataElements*) whose domain is defined within *SI_TerminologySource* such as the GO Ontology (in this case).

7. Using a system metaphor to describe the early architecture

In the early stage of a new system development it is crucial to remove ambiguities about the envisaged system. Clients, analysts and developers need to come up with a shared vision about the system to-be which enables to generalize the system highlighting the main components and actors of a possible architecture. To this end, the Agile approach has introduced the concept of *system metaphor*. The simpler and more effective definition for this concept is "*a story that everyone - customers, programmers, and managers - can tell about how the system works*" [4]. In our project the system metaphor has served three main purposes. It has been used to introduce the project to the domain experts who had heard about the NCRI initiative but did not know much about the system. Being the system highly innovative, it was impossible to refer to other similar systems hence the use of an analogy has been determinant to communicate the essential elements of the envisaged system. The metaphor has also been used to describe the architecture of other similar systems so that these could be easily compared with the high level architecture of our system. Finally, it has provided the analysis and development teams with a guide used throughout the development process and against which all the design and implementation solutions have been checked.

7.1. A system metaphor for the NCRI's platform

From the initial discussion with the NCRI stakeholders, only a few key requirements were identified. NCRI wanted to build a system to support an *open environment* where the needed data resources could be easily found and effectively used by researchers. The system *should not create another centralized ontology* to annotate the existing resources but should use existing ontologies hence exploiting the relationship between existing ontologies and data resources. The system should provide the needed functionalities to *assure quality information and services*. From these basic requirements, the analysis of similar systems and the initial dialogue with the community, we have identified the system metaphor, that is, an ***open knowledge marketplace*.** The notion of a marketplace captures the central importance of *providing a system of mutual incentives* to bring both requesters and resource providers to the platform. The system will give providers a means of offering access to their resources and users a means of exploiting them while assuring *certainty*, *authority* and *consistency***.** This will require bringing together information, services, data standards, data models, metadata elements, business models, business logic and business policies. The key elements of such a system will be Products, Providers and Requesters. ***Products*** are bioinformatics data and services. ***Providers*** are organizations which manage and/or curate either *data repositories* (local databases, specialized global repositories, knowledge bases, etc.) or *services registries*. ***Requesters*** are researchers or applications which need to access the data stored in repositories or to execute services in order to fulfil their *research* *goals*. In this light, the system should provide the needed infrastructure and mediation services to run the marketplace.

The metaphor as shown on the left of Figure 3 is the one we defined after a few initial meetings with the NCRI stakeholders and the community representatives, while on the right side is the metaphor in a later stage when the key components were already identified. Even in its early definition, the metaphor conveys the key traits of the system to-be we needed to validate and reach an agreement upon from both technical and organizational points of view. For instance, it clearly communicates that the system will need a management/configuration division to be set up in the NCRI; that both final users (researchers) and other applications will use the services provided by the system, and so forth. Depending on the target audience, different parts of the metaphor can be used to drive the discussion. Finally, we have found it important to define a metaphor that can be represented visually because this will be much more effective in supporting the discussions with stakeholders.

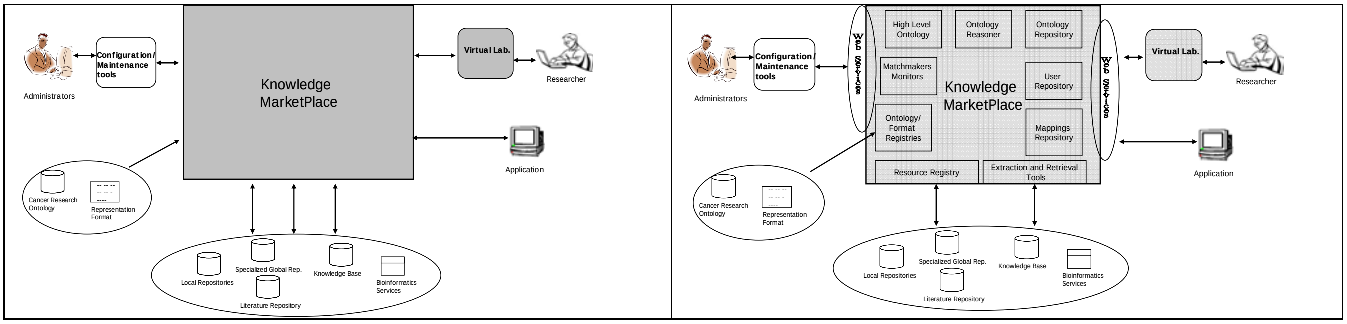


Figure 3: NCRI Platform's metaphor in an early stage (left) and later stage (right)

Conclusions

In this paper we talk about our experience in the analysis and early architecture design of the NCRI Platform system. However, we believe that the paper contains a number of more general points which can be of interest for project managers, analysts and developers working to the development of biomedical systems aiming to integrate knowledge in complex and open environments. The methods we have briefly introduced in this paper have shown to be effective to support the communication between software engineers and domain experts and to elicit the role of the various resources belonging to the knowledge network established in a domain.

Acknowledgements: This project has been partially funded by the grant Generative Software Development provided by British Telecom. We are also grateful to the NCRI Informatics Initiative group for their support during the elicitation and analysis activities.

References

[1] P. Covitz et al., caCORE: A common infrastructure for cancer informatics. Bioinformatics 19(18), 2003.

[2] NCRI Informatics Initiative www.cancerinformatics.org.uk

[3] A. Brazma, et al.: ArrayExpress—a public repository for microarray gene expression data at the EBI. Nucleic Acids Res., 31, 68–71, 2003

[4] Kent Beck: Extreme Programming Explained. Addison-Wesley Longman Publishing Co., Inc, 1999

[5] A. Dardenne, A. van Lamsweerde, S. Fickas: Goal-Dircted Requirements Acquisition. Science of Computer Programming, 20 (1993)

[6] R. Solingen, E. Berghout: The Goal/ Question/ Metric Method McGraw-Hill, 1999

[7] A. Finkelstein, et al.: Viewpoints: a framework for integrating multiple perspectives in system development" Int. Journal of Software Engineering and Knowledge Engineering, vol. 2, 1992

[8] Project's web site: http://www.cs.ucl.ac.uk/CancerInformatics/