PLANNING OF AUTOMATED CELL CULTIVATION SYSTEM

Sándor Kopácsi – János Nacsa

1. SUMMARY

The early results of a research and development project, called CytoFab [1], funded by the European Union, is published in this paper. The idea of CytoFab is to develop an automated, self-learning modular manufacturing platform for flexible and patient-specific cell production. Our first tasks in this project were the specification of the requirements, or rather the requirement engineering, as well as, modelling the system using standard tools, like UML, that will serve as a basis of the planning and implementation.

2. INTRODUCTION

Old age of human beings is often associated with several diseases and reduction of particular organic functions. This is why, in parallel to the increasing life expectancy, the demand for modern therapies and implants raises. One way to fulfil this demand is the application of personalised medicine, with which patients can be cured individually according to their personal responses to the treatment. Creating and testing personalized medicines requires the patient-specific reproduction of human cells. Another possible application of cell cultivation is the reproduction of organs that can be applied for chronic diseases and acute injuries. This can be done by donating healthy cells from humans to produce new tissues in large quantities for transplantation. Today, the main challenge within these approaches is the patient-specific production of human cells in sufficient volumes.

3. STATE OF THE ART

The current situation in biotechnological science shows a high amount of accepted applications in the field of personalized medicine due to the high benefits of this technology. However, producing the required amounts of cells for the personalized medicine in lab scale is not sufficient to distribute this well-established technology to a broad population, which could highly benefit from these technologies due to the individuality of therapy.

Currently, there is no automated production system to produce cell material for personalized medicine available. There are some automated solutions for cell production (e.g. BioCel Automation System [2] or ELISA STARlet [3]), but these do not satisfy the needs and regulations to prevent cross-contamination between different cell cultures, which is necessary to keep them pure and to avoid reactions of the cells to foreign material.

4. CYTOFAB CONCEPT

The main concept of the CytoFab project is to transfer patient-specific cell production from manual, laboratory scaled production to an industrial scale which enables the production of specific cell material in required quantity in an affordable way.

Figure 1. illustrates the overall CytoFab approach, including a modular production line which consists of plug-ins adapted to modules by standardized interfaces and implementing the functionalities necessary to execute cell production process steps (e.g. liquid handling, heating, image processing, measurement of cell stress factors and common environment conditions, etc.) So it will be possible to configure the platform's capabilities and capacity by exchanging or adding plug-ins or modules. The production modules are combined with a control station for user interaction with the platform and a hybrid workplace for manually performed process steps (e.g. for processes necessary before or after cell production such as cell preparation), integrated by man machine interaction. An intelligent Manufacturing Executing System (MES) software, configured via the control station accomplishes the production management. Moreover, the software is capable of optimizing the processes before and during the execution as it applies a knowledge-based model of cell behaviour and advanced process control functionalities.

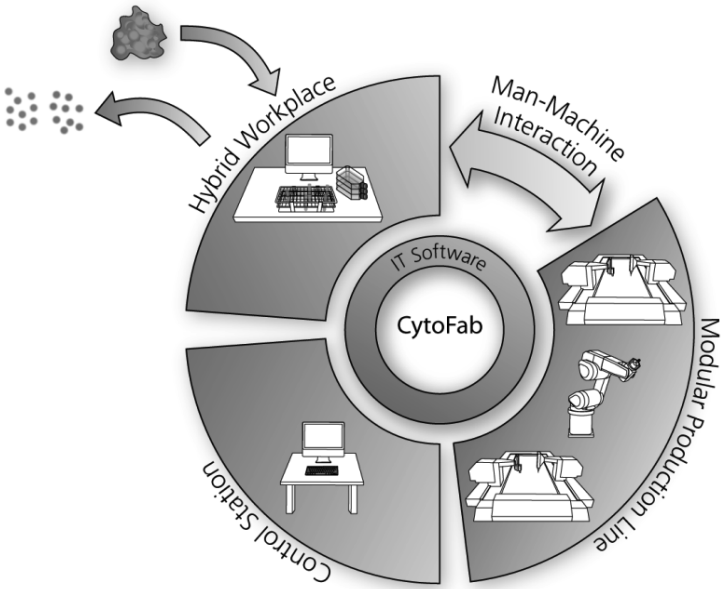

Figure 1.
The Overall CytoFab concept

5. REQUIREMENT ENGINEERING

The primary measure of success of a software/hardware system – like Cytofab – is the degree that meets the purpose for which it was intended. Broadly speaking, Requirements Engineering (RE) is the process of discovering that purpose, by identifying stakeholders and their needs, and documenting these in a form that is amenable to analysis, communication, and subsequent implementation. There are a number of inherent difficulties in this process. Stakeholders (including paying customers, users and developers) may be numerous and distributed. Their goals may vary and conflict, depending on their perspectives of the environment in which they work and the tasks they wish to accomplish. Their goals may not be explicit or may be difficult to articulate, and, inevitably, satisfaction of these goals may be constrained by a variety of factors outside their control.

Requirement Engineering is a multi-disciplinary activity, deploying a variety of techniques and tools at different stages of development and for different kinds of application domains. Methods provide a systematic approach to combining different techniques and notations, and method engineering plays an important role in designing the RE process to be deployed for a particular problem or domain. Methods provide heuristics and guidelines for the requirements engineer to deploy the appropriate notation or modelling technique at different stages of the process. A variety of approaches have been suggested to manage and integrate different RE activities and products. To enable effective management of an integrated RE process, automated tool support is essential. Requirements management tools, such as DOORS, Requisite Pro, Cradle, and others, provide capabilities for documenting requirements, managing their change, and integrating them in different ways depending on project needs [4].

In this project we have selected Enterprise Architect (EA) UML tool for modelling the requirements of the system. EA provides full life cycle modelling for business and IT systems, software and systems engineering, as well as real-time and embedded development. With built-in requirements management capabilities, Enterprise Architect helps in tracing high-level specifications to analysis, design, implementation, test and maintenance models using UML and other open standards [5].

The main steps of requirement specification in UML using Enterprise Architect by Sparx Systems will be discussed in following chapters. See more details in [6].

6. CREATING TEMPLATES FOR THE NON TECHNICAL PARTNERS

Although we have decided, that Enterprise Architect will be used in the requirement engineering [7], we have defined our own templates to help and formulize the work with the those partner who have no experience with software engineering tools, but their knowledge in the field of cell cultivation is essential.

One of the templates was the *Requirement Specification Template,* which based on the attributes of requirement element of EA, but some new elements (open issues, ideas, future, and source) have been added to it.

Another template was the *Use-Case Template*, which is also based on the attributes of use case elements of EA, but with bigger modifications. Hopefully it helps to describe the activities (flows), the conditions and the timing aspects of the biological processes. It forces the experts to provide information that is often skipped, but rather important for the control experts (failed post conditions, timing details, exception handling, etc.).

Using these templates the experts can 1) supervise the requirements and use cases created by software engineers, 2) easily provide new requirements, 3) formulize their know-how about the details of the processes.

7. REQUIREMENT MODEL IN ENTERPRISE ARCHITECT

As the CytoFab system is going to be rather complex, a lot of requirements could be collected in the very beginning of the project. More requirements will be added later on, and the existing ones might be restructured and evaluated.

To solve the complexity somehow, we have decided to build a strongly structured requirement graph. Many requirements are linked to other ones. The links are mainly “dependency”, “realization”, and “aggregation”. One of the difficulties of requirement engineering is that the type of such link is very dependent on the designer’s taste. The structure of the requirement graph and many concrete links can be defined in many other ways.

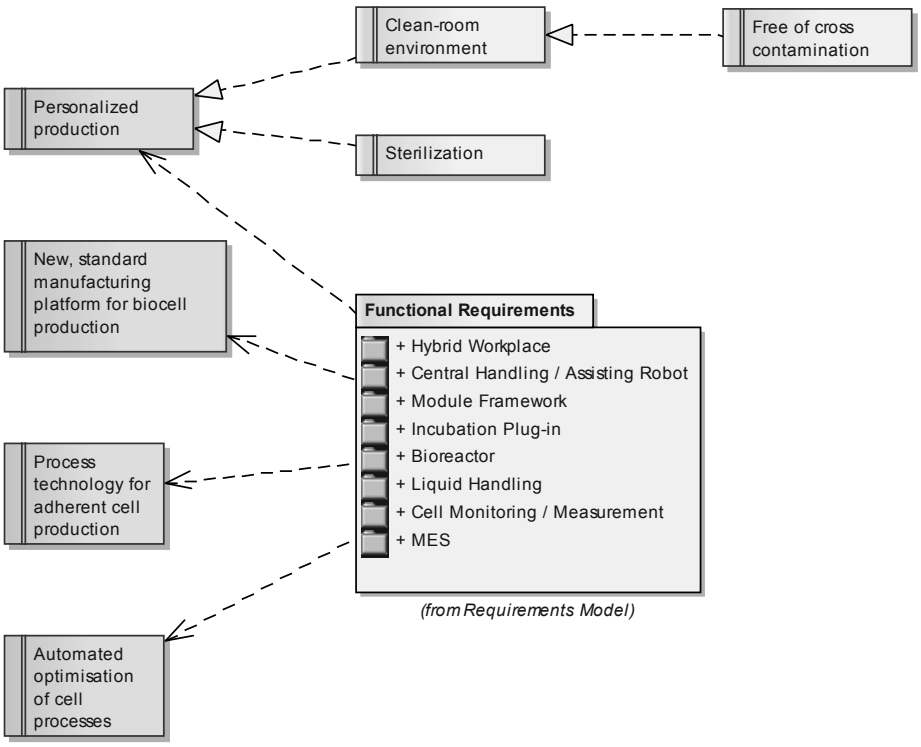

Figure 2.
Features and Requirements

The requirements were grouped also in three categories: features, functional and non-functional requirements. A requirement tends to be more granular, and is usually written with the implementation in mind. Those who consume requirements tend to be engineers or technical audience. Requirements are collected in two groups: functional and non-functional. In our model a feature is a “higher-level” objective than a requirement – and is usually more focused on business needs rather than implementation. We have distinguished features that collect different functional requirements (see Figure 2.), and features related to non-functional requirements.

8. USE CASE MODEL IN ENTERPRISE ARCHITECT

In the main Use Case model two human actors were defined originated from the CytoFab user stakeholder.

1) The everyday worker of the system is the laboratorian, who support and control the system.
2) The Lab Manager makes the advanced work with the special features of the system.

As an option a special use case was defined to allow the health inspector stakeholder as an actor to check the system on-line.

As physical objects the following key elements have been defined:

- Bioreactor is a place where a given cell culture can grow. The system transports the bioreactors between different reactor slots.
- Robot is the device that moves the reactors automatically from one place to the other one (defined in Central Handling/Assisting Robot packages).
- Module Framework is the main part of the system that contains the different plug-ins. It includes the Module Framework Handling Unit.
- Incubator plug-in is the environment where the bioreactors with the cell cultures are stored.
- Liquid Handling Unit supplies the bioreactors with certain liquid medium, includes the medium tank, pump, tubing; also able to remove the waste.
- Medium storage component to keep the different liquids before adding them to the bioreactors.
- Optical monitoring plug-in to monitor the cell cultures.

9. CONCLUSIONS

The requirement gathering will continue during the next phase of the project, where the following tasks have been defined:

- The requirement team will add the last collected information to the model and fix some bugs.
- The use cases will be defined more detailed to add activity diagrams etc. to the most of them.
- The links between use cases and requirements will be established. We want to link most of the requirements to use case elements to ensure the system will be able to fulfil the given requirement.
- A big set of requirements will be selected and tests will be defined to help after the development to check how the system fulfils the given requirement.
- A continuously update will be done during the further phases of the CytoFab development.

10. LIST OF REFERENCES

[1] CytoFab – Self learning modular manufacturing platform for flexible, patient-specific cell production, Description of Work, EU project proposal, 260100, 2010
[2] Access Agilent eNewsletter, August 2010, www.chem.agilent.com/en-US/Newsletters/accessagilent/2010/aug/pages/ngs.aspx
[3] www.hamiltonrobotics.com/fileadmin/user_upload/products/ELISA_Starlet/MR-0702-01_ELISA-STARlet.pdf
[4] Pamela Zave: Classification of Research Efforts in Requirements Engineering, ACM Computing Surveys Vol. 29, No 4, 1997
[5] http://www.sparxsystems.com/products/ea/index.html
[6] CytoFab Deliverable D 1.1 – Overall Requirements specification
[7] Requirement Management with Enterprise Architect; By Sparx Systems, 2008, version 1.2 http://www.sparxsystems.com/downloads/whitepapers/Requirements_Management_in_Enterprise_Architect.pdf