



Universiteit
Leiden
The Netherlands

Image Analysis and Platform Development for Automated Phenotyping in Cytomics

Yan, K.

Citation

Yan, K. (2013, November 27). *Image Analysis and Platform Development for Automated Phenotyping in Cytomics*. Retrieved from <https://hdl.handle.net/1887/22550>

Version: Corrected Publisher's Version

License: [Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden](#)

Downloaded from: <https://hdl.handle.net/1887/22550>

Note: To cite this publication please use the final published version (if applicable).

Cover Page



Universiteit Leiden



The handle <http://hdl.handle.net/1887/22550> holds various files of this Leiden University dissertation.

Author: Yan, Kuan

Title: Image analysis and platform development for automated phenotyping in cytomics

Issue Date: 2013-11-27

Chapter 6

Reasoning over Data in HT/HC Management

This chapter is based on the following publications

Yan, K., Larios, E. LeDevedec S.E., van de Water, B., and Verbeek, F.J.(2011), “Automation in Cytomics: Systematic Solution for Image Analysis and Management in High Throughput Sequences.”, Proceedings IEEE Conf. Engineering and Technology (CET 2011), Vol 7. Shanghai 2011, 195-198

Chapter Summary

In the Chapter 4 and Chapter 5, we have demonstrated two case studies on the HT/HC live cell screen. These two case studies are producing an amount of data on the scale of terabytes, which makes the organization and storage of these image data nontrivial. Additionally, phenotypical measurements of these HT/HC data must be integrated with other omics resources such as Entrez DB, Ensembl DB, BLAST engine to allow the construction of genotype-to-phenotype models. To that end, an HT/HC database management system (HT/HC database system) is required to provide a platform for both data management and integration.

Initially, the HDF5 format, a self-contained data storage file format has been employed in our study as a hierarchical data storage solution. The HDF5-based storage has proven to be an efficient data storage and transportation solution. However, it lacks a build-in searching and querying mechanisms allowing examine data in a comprehensive manner. To overcome the limitation self-contained data storage, we proposed a prototype HT/HC database system. This database system is not only designed for the management of HT/HC image data but also facilitates the integration with omics via a flexible programming API. Together with tools for data mining and semantic analysis tools, the HT/HC database system will eventually provide a multi-layer view of an experiment, which cannot be easily accomplished by self-contained file format such as HDF5.

Several major challenges are encountered during the design of this HT/HC database system. Compared to other designs of concurrent database applications, the design of HT/HC database system faces the following challenges: (1) megabyte-scale data must be locked per transaction, (2) complex image analysis may lead to longer execution duration per transaction, and (3) a multithread-safe programming API is required when accessing data.

In this chapter, a prototype design of HT/HC data management system is proposed to meet with these challenges. The chapter first introduces the principle design of the architecture of the HT/HC data management system. Subsequently, the architecture is further divided into a multilayer model which is comparable to the layered foundation of the OSI model. In the following sections, each layer is explained.

6.1. Principle Design of HT/HC Management System

Initially, we focus on designing a seamless automated framework to complete the data processing for HT/HC screen studies [145][146]. Since the purpose of this system is straightforward, here we employ a task-driven user-centered design principle taking the empirical workflow of the HT/HC screen study (cf. Figure 1-1) as starting point. A HT/HC analysis first starts with the design of the experiment setting based on biological interests. The experiment design is further stored as plate designs. Subsequently, the specimens are visualized using microscopy. The captured images are uploaded to a network-attached storage (NAS) and quantified using image analysis. Eventually, the phenotypical measurements are further probed using machine learning and statistical analysis to provide a comprehensive data representation. From the workflow (cf. Figure 1-1), we defined a simplified functional division between each task during the HT/HC analysis. Moreover, the essential data type passed between each task is also defined. From the definitions, the general design of HT/HC data management system is illustrated as Figure 6-1.

The general design follows the layered architecture similar to the OSI model [147]; as we descend each layer is closer to the raw data. During the design of our HT/HC data management system, we first start from the analysis workflow of an empirical HT/HC experiment (cf. Figure 1-1). Compared to a software design following a requirement-driven approach, our approach is very applicable as there are no clear mutual requests definitions underlying different HT/HC screen experiments. The design of HT/HC screen experiment is often semi-systematic and there are very little similarities in the organization of experiment materials and perturbations employed in the experiment design (cf. Ch. 4 and Ch. 5). Moreover, the existing similarities are mostly related to the raw data produced during HT/HC screen experiment instead of biological materials. Therefore, the analysis workflow of HT/HC empirical experiments is a good starting point to study the limited amount of similarities among HT/HC screen experiments. This workflow illustrates the major steps in a HT/HC screen experiment.

From the HT/HC analysis workflow (cf. Figure 1-1), two element types are defined. (1) The **procedure element** type defines a number of essential steps in a HT/HC experiment. (2) The **data element** type defines all sorts of basic raw data produced and transported in between procedures. Each procedure element receives an input data element and performs a collection of operations to produce an output data element. To that end, each procedure element is considered to be a **functional module** [148][149]. Furthermore, here we further generalized the HT/HC analysis workflow into a top-down **layered architecture** (cf. Figure 6-1). Each layer represents a collection of functional modules in the HT/HC analysis workflow.

From the highest to the lowest layer (cf. Figure 6-1), these layers include (1) **end-user GUI layer**, (2) **WS-API layer**, (3) **web service host layer**, (4) **database layer**, and (5) **computation layer**. The top layer, namely the **end-user GUI layer**, is an encapsulated user interface that allows the end-user to access all functional modules with a general view. From this layer, the end-users are given the opportunity to perform a collection of customized pipelines of HT/HC screen analysis without having to understand the architecture of the functional modules such as image analysis, data analysis, and data management. The seamless integration of all lower layers functional modules is accomplished via the implementation of the **Web Service based**

Application Programming Interface (WS-API) layer. A **Web Service (WS)**, by World Wide Web Consortium (W3C) definition [150], is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically Web Services Description Language (WSDL)). Other systems interact with the Web service in a manner prescribed by its description using Simple Object Access Protocol (SOAP) messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

The **WS-API layer** is in fact a pseudo layer in which the functional module is wrapped in a web service shell. Such design takes advantage of the compatibility of WS-API and allows module developed in different programming languages to be able to communicate with each without implementing complex cross-language programming. Moreover, with the web service shell, updating in each module does not necessarily require additional changes in the dependent module. To physically host the WS-API layer, the **web service host layer** is introduced. These web service server hosts the foundations of programming API and software architecture. The **image database layer** is the major management layer providing an organized storage and distribution of image data, auxiliary image results, phenotypical measurements and supplementary documentation. The lowest layer, namely the **computation layer**, is the foundation layer which provides computational power for all higher layers.

Following the top-down design of each layer, this chapter is organized as follows. In the next sections, we will focus on end-user GUI layer, WS-API layer, and the database layer since the computation layer is beyond the scope of management and the web service host layer employs only commercial and open-source software. Finally, we will address the database layer.

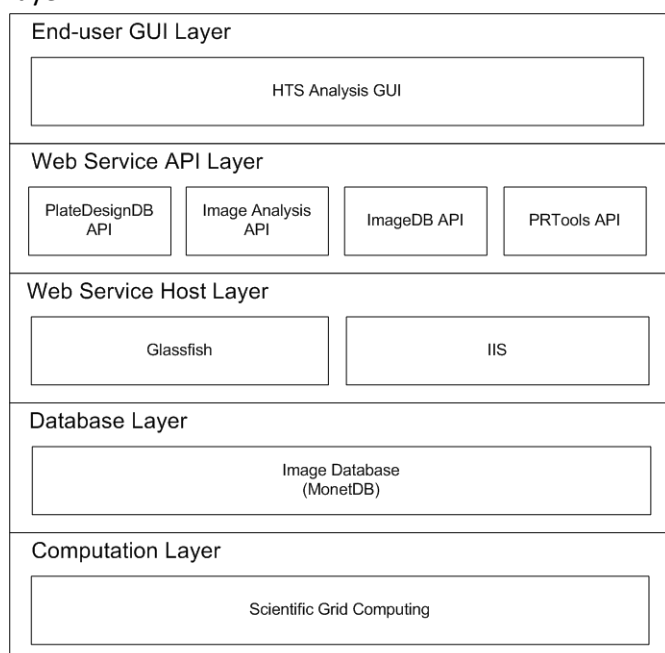


Figure 6-1 the structure of HTS analysis platform

6.2. Implementation of Layers

This section illustrates the fundamental design of the system architecture. The section follows the top-down design flow of the layered model depicted in Figure 6-1. In this section, the top three layers will be described. We will first start with the highest layer.

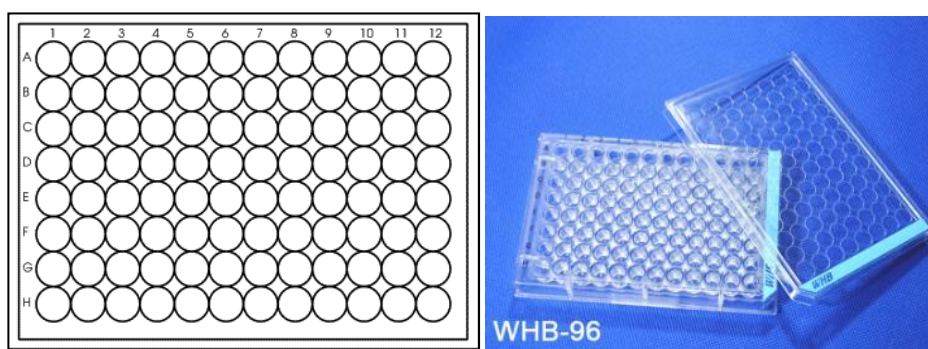
6.2.1. End-user GUI Layer

The end-user GUI layer is the highest layer providing a data input/output and a visualization mechanism. It is designed to be both starting point and end point of a HT/HC analysis. This layer is originally designed as a solution for metadata management in HT/HC such as the bookkeeping of experiment protocols. As introduced in Figure 1-1, HT/HC screen study starts with the design of the experiment; during which the researcher decides which materials and what conditions will be included. Such design is often referred as **plate design**.

The **plate design** begins with the researcher first chooses one cell culture plate as the design model. A cell culture plate (cf. Figure 6-2a) is a plastic or glass plate containing a number of small wells each resembling an individual Petri disk that can hold live specimens. The actual dimensionality, meaning width and length, of cell culture plate is fixed. The numbers of rows and columns of the wells within the culture plate can be in 4x6, 6x8, 8x12 or even higher. A higher number of well allows more experimental perturbations (conditions) to be considered in one experiment. However, more wells also mean longer imaging time during HT/HC screen since, for microscopy, there always is a minimum duration to the imaging of each well. Depending on the complexity of experiment, the researcher often chooses a culture plate with sufficient wells to hold most experimental conditions meanwhile it should guarantee the shortest imaging duration as possible. To visualize the design procedure, here we introduce two empirical spreadsheet-based plate designs of HT/HC screen.

Figure 6-2b is the experimental design of the growth factor regulation experiment described in Ch. 4. In this study, the researcher wants to extract phenotypical variability of cancer cells under the influence of different growth factors. To achieve this purpose, the researcher first selected four growth factors and two cell types from one cancer cell line. Each cell type will be treated with a single or combination of growth factors. Therefore, the researcher divides one 8x12 culture plate into left and right section. The designated growth factors are mapped into the row and column of and stored in a spreadsheet.

Figure 6-2c is the experiment design of a dynamic study EGF transportation pattern [29]. In this study, the researcher wants to quantify the transportation model of EGF in cells under different treatments. Again, two cell types are chosen for the study. Instead of employing live cell imaging, the researcher decided to fix cells at each time point including 10 min, 20 min, 40 min, 80min, and 160 min (cf. Figure 6-2c [column]). Similarly, the whole culture plate is divided into left and right section to house two cell types and each column represent the fixing time of cells. The row represents different treatment the cells received.



(a) a 8x12 cell culture plate

DMSO	HGF	EGF+HGF	FGF+TGFbeta	DMSO	HGF	EGF+HGF	FGF+TGFbeta
DMSO	HGF	EGF+HGF	FGF+TGFbeta	DMSO	HGF	EGF+HGF	FGF+TGFbeta
EGF	TGFbeta	EGF+TGFbeta	HGF+TGFbeta	EGF	TGFbeta	EGF+TGFbeta	HGF+TGFbeta
EGF	TGFbeta	EGF+TGFbeta	HGF+TGFbeta	EGF	TGFbeta	EGF+TGFbeta	HGF+TGFbeta
FGF	EGF+FGF	FGF+HGF	all	FGF	EGF+FGF	FGF+HGF	all
FGF	EGF+FGF	FGF+HGF	all	FGF	EGF+FGF	FGF+HGF	all
pGFP				GB1			

(b) Spreadsheet-based data management: growth factor regulation

	pGFP					GB1				
	10 min	20 min	40 min	80 min	160 min	10 min	20 min	40 min	80 min	160 min
ctrl										
mock										
+EGF										

(c) Spreadsheet-based data management: EGF translocation experiment

Figure 6-2 plate design in HT/HC screen

When conducting a wet-lab experiment, this plate design will be followed. During the image analysis and data analysis, the same spreadsheet will also be used as bookkeeping information for data comparison. However, for large scale experiment such as siRNA functional screens, hundreds siRNA targets (conditions) will be manually mapped into the plate designs and each experiment requires 20~40 different plate designs to hold all targets. It is difficult to produce an error-free scenario when performing the design. Moreover, an error in the plate design will lead to a false-conclusion during image and data analysis since the original expectation of output may no longer fulfill due to the mismatch between bookkeeping information and the practical experiment being conducted. Therefore, to minimize or eliminate the manual process of the important bookkeeping mechanism, namely the plate design, here we design a seamless GUI for the plate design in HT/HC screen studies.

The plate design GUI (cf. Figure 6-4) starts with an review of the design of a plate[145]. From a number of empirical studies, we extract the basic elements employed in most HT/HC screen studies. We introduce the concept of **plate**, **well**, **condition**, and **group** (cf. Figure 6-3). The design concept is to emulate the natural plate design procedure that researchers have used. To that point, the design of the plate design GUI follows a top-down flow instead of a requirement-driven. The **plate** is a representation of plate layout. Each plate contains a number of wells. There is fixed a one-to-many relationships between plate and wells. Instead of giving a fixed row and column number, we allow users to choose the number-id of row and column and fit a plate layout with the number of wells. The **well** represents a physical well in a culture plate. Each well contains a number of **conditions** which represents the actual treatment, treatment duration, cell type or any experiment perturbations that may be employed in the HT/HC screen. There is a many-to-many relationship between conditions and

wells since a well can have a combination of perturbations (conditions). The **group** is a higher level concept of wells following some reasoning. For example, Figure 6-2c contains two groups which represent different cell types.

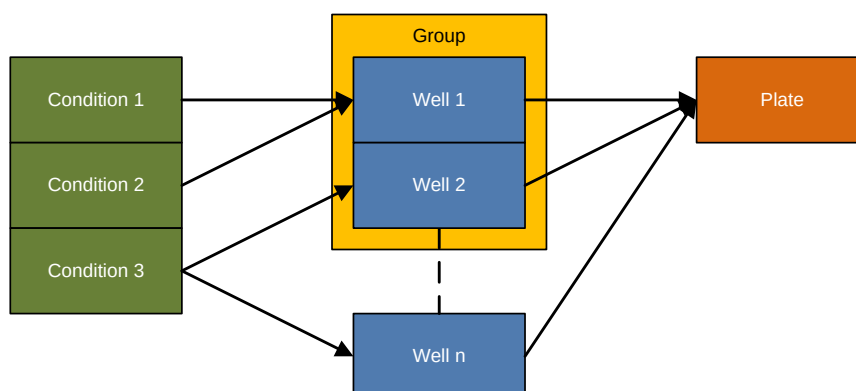


Figure 6-3 Components in plate design

With the GUI design (cf. Figure 6-4), we allow the end-user to first create an empty plate of chosen size. Then the end-user imports the master sheet of conditions which contains all experiment perturbations; meanwhile conditions can be added to the wells. Image analysis of logical groupings can be then performed. The major improvement of this GUI design over spreadsheet-based design is the introduction of what-you-see-is-what-you-get (**WYSIWYG**) principle and **drag-&-drop** design.

The **WYSIWYG** design principle provides the foundation for GUI of plate design. Instead of using a table to emulate a plate layout, here we employ a row-column well map layout which is an exact map of the physical plate. It provides a straightforward overview of the whole plate without losing the possibility of zooming into a single-well.

The **drag-&-drop** design allows a faster mapping from the master list of conditions to desirable well in the plate. By allowing multiple selections, the same condition can also be assigned into multiple wells with a single dragging; similarly, one well can be assigned with multiple conditions with a single dragging. Moreover, with colored labels for conditions, wells contain different conditions can be easily distinguished by color (cf. Figure 6-4). When displaying different conditions together, colors can be combined or switched off to improve fast recognition. In the screenshots (cf. cf. Figure 6-2c), the similar operation using spreadsheet-based solution would require end-users to constantly switch between spreadsheets. Moreover, wells containing combined conditions cannot be easily visualized together.

In Figure 6-6, the sequence diagram of the use of a plate design GUI is displayed. The user first finishes plate designs using the plate layout design interface (cf. Figure 6-4) and based on the design user will conduct wet-lab experiment. After collecting all images associated with the plate design, the user can upload all images and map them to the database using the ImageDB API. Subsequently, the user will initialize analysis of the screen and then the GUI will invoke image analysis API and pattern recognition API to produce intermediate results of the image and data analysis (cf. Figure 6-5a & b). By using different data analysis approaches, the user

may verify the initial hypothesis or compare phenotypical characteristics under different treatments.

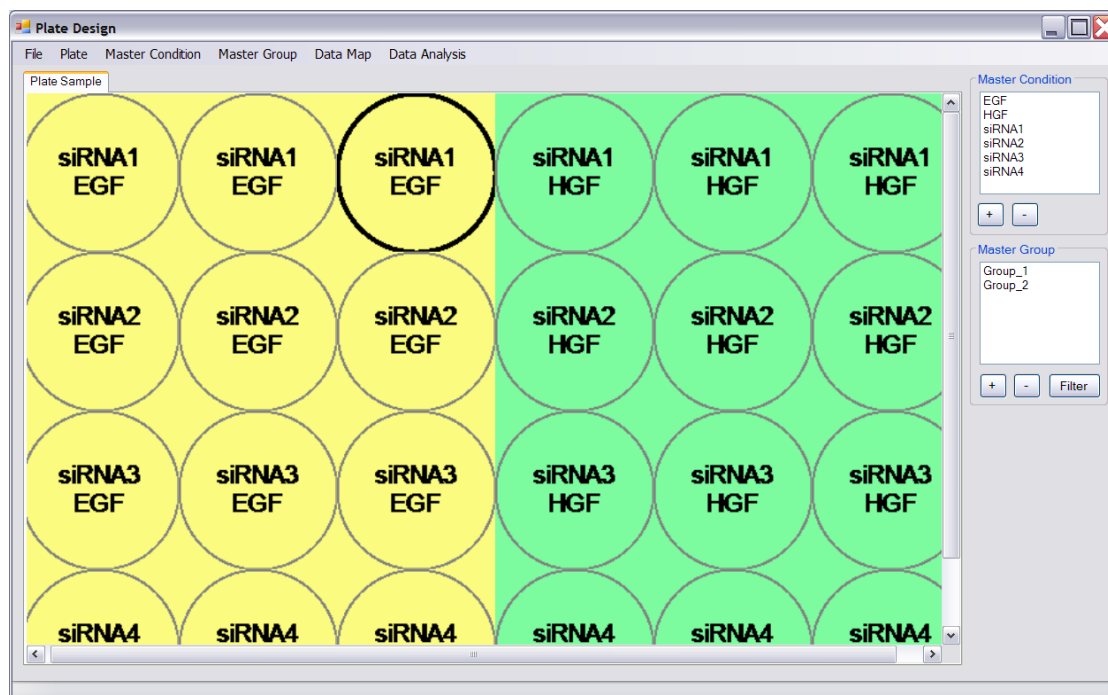
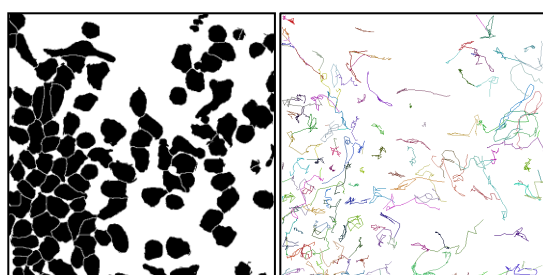
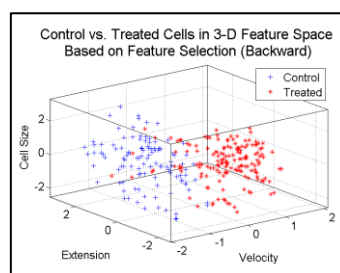


Figure 6-4 plate design GUI



(a) intermediate results of image analysis



(b) data analysis result

Figure 6-5 front end of the data management system

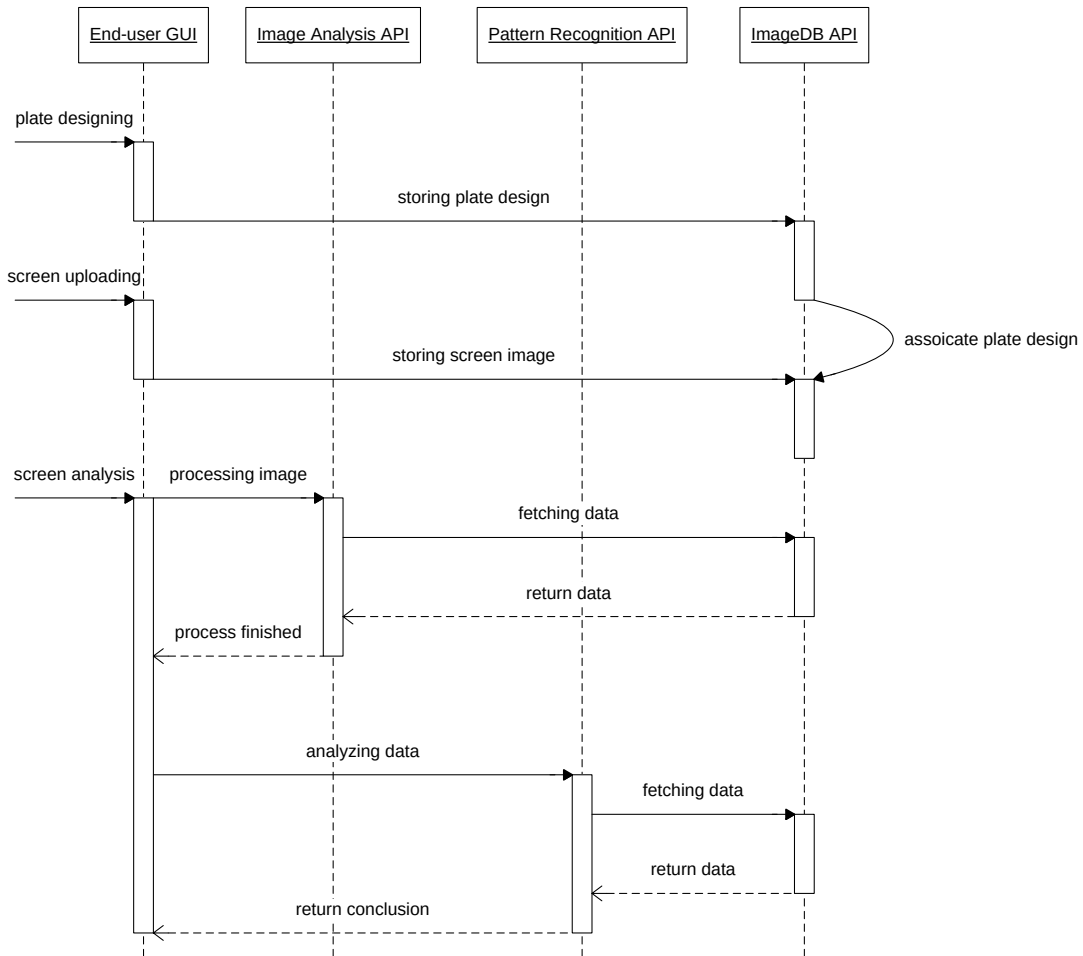


Figure 6-6 sequence activity diagram of using end-user GUI

6.2.2. WS-API Layer

In order allow different functional modules to communicate with each other, a universal wrapping application programming interface (API) is required. Although it is possible to hardcode cross-module communication, such implementation may limit the extensibility of software and violates the encapsulation design. Therefore, we adapted the principle of **Web Service based Application Programming Interface (WS-API)** as the solution. The W3C organization defines a "Web service" (WS) as "a software system designed to support interoperable machine-to-machine interaction over a network" (cf. Figure 6-7) [150].

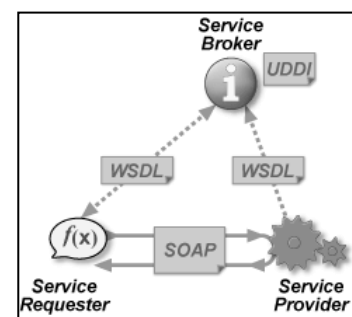


Figure 6-7 web service architecture

To take advantage of the flexibility of the Web Service, the WS-API of each function module is wrapped in a Web Service shell by following modular programming and published by web server supporting Web Service, such as Java Glassfish and Microsoft IIS. We will first use the ImageDB API as a sample to illustrate the abstract design of WS-API communication (cf. Figure

6-8). Furthermore, we will briefly introduce the implementation of other functional modules such as plate design API, image analysis API, and pattern recognition API.

ImageDB API

The ImageDB API is a WS-API that grants database accessibility to other function modules. The design of ImageDB WS-API is illustrated in Figure 6-8. Such design provides several advantages:

- 1) It is possible to access data set without understanding the complex internal relationships
- 2) Internal structure or physical location of data are not visible to the end-users
- 3) Internal modifications do not require updates at client state
- 4) The modular design significantly improves extensibility and flexibility of cross-module communications.

With the ImageDB API, other modules may exchange data without direct communication with each other.

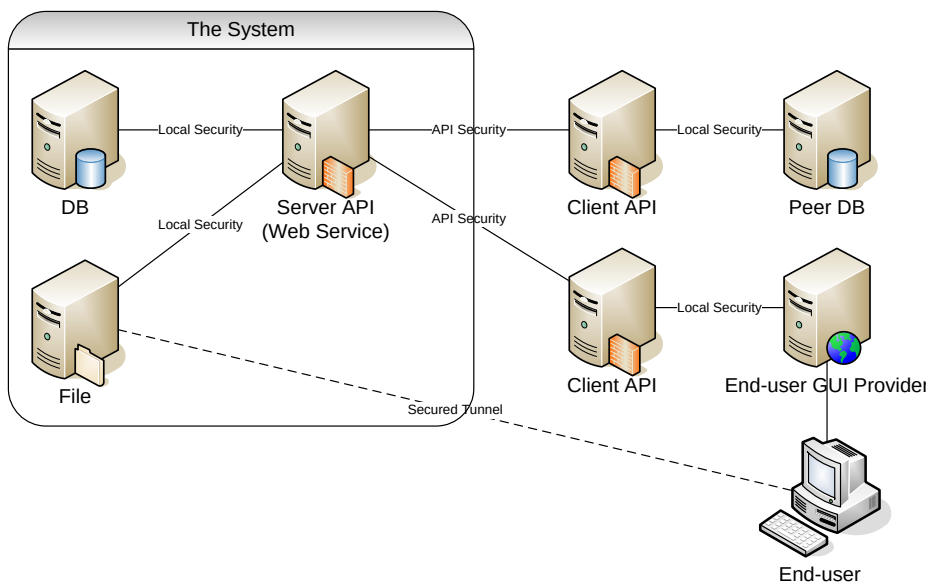


Figure 6-8 abstract design of WS-API communication

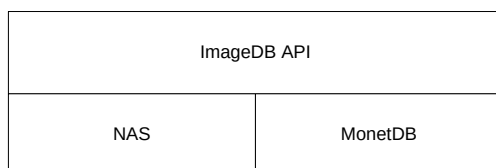


Figure 6-9 underline structure of the ImageDB API

In the design of ImageDB API, we have chosen for the MonetDB DBMS [151][152]. MonetDB is a leading open source database system that has been dedicated to the management of large datasets. Compared to other DBMSs, it is well-known for its performance in processing analytical queries on large scale data sets. Instead of storing data in a row-based memory block, the MonetDB stores data in a column-based memory block (cf. Figure 6-10). Thus, when processing analytical queries, only the required columns are loaded. Such design is in particular important when data contain extensive amounts of primary keys or if not all column data will

be involved in the analytical queries. According to our study [145][146], such storage structure provides an increased access speed since the conditional filtering can be executed much faster.

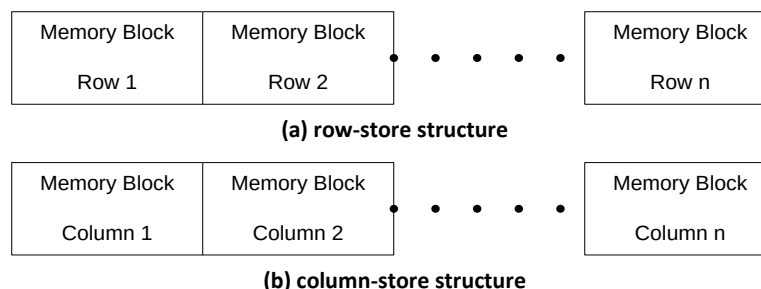


Figure 6-10 row-store structure vs. column-store structure

Plate Design Module

The plate layout design provides a graphical user interface allowing end-users to rapidly deploy, modify, and search through plate designs, to which auxiliary data such as experimental protocols, images, analysis result and supplementary literature is attached. In addition the plate design provides fast cross-reference mechanism in comparing data from various origins. This module is also used as the front end for the visualization of results such as using heatmaps, cell detection, or motion trajectories. The underlying structure of the plate design module is illustrated in Figure 6-11.

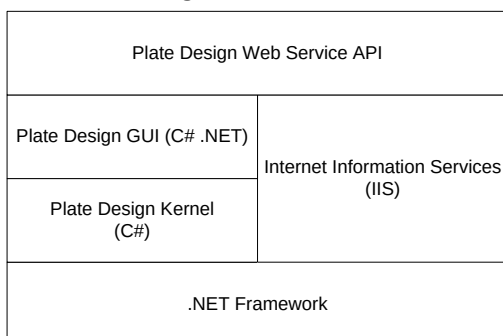


Figure 6-11 underline structure of the plate design API

The plate design module is constructed in the JAVA language. Since most of the end-users are more familiar with Windows-based applications such as Excel, the Windows-based GUI using JAVA may further improve user satisfaction. Compared to other programming languages, JAVA is a platform independent programming language which allows the GUI to be directly run on computers with a Java Virtual Machine (JVM) installed without the need to recompile the code.

Image Analysis Module

The customized image processing and analysis is applied to obtain phenotypical measurements for each of the different treatments. An open source image processing kernel, i.e. ImageJ, is extended with packages providing customized and robust image segmentation and object tracking algorithms dedicated to various types of cytomics (cf. Figure 6-5b). The current package covers solutions to cell migration, cellular matrix dynamics and structure dynamics analysis (cf. Chapter 4 & Chapter 5). It has been practiced in HTS experiments for toxic compound screening of cancer metastasis [10][15], wound-and-recovery of kidney cells [70] and cell matrix adhesion complex signaling[29], etc. This module (cf. Figure 6-12) is designed as

WS-API. As the image analysis computation requires large image volumes to be processed, GRID computing is used to obtain results in reasonable time.

The current WS-API serves as a wrapping class around the image analysis package kernel and transforms the functional module into a black box that can be access by other functional module.

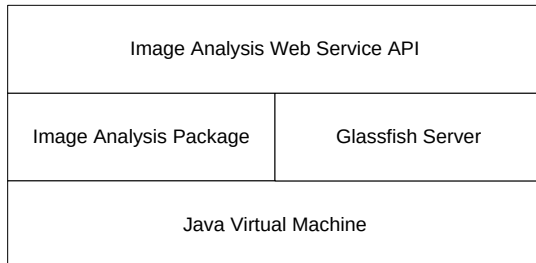


Figure 6-12 underline structure of the image analysis API

Data Analysis Module

To maximize the availability of data analysis tools, we have included the PRTools package [153] as the machine-learning toolset kernel for the data analysis module. Furthermore, we develop a customized pattern recognition toolset for data analysis with spatio-temporal data. The data analysis module (cf. Figure 6-13) is implemented as WS-API using .NET output of MATLAB deployment tools. Such architecture allows a rapid adaptation to complex mathematical algorithms. Furthermore, the flexibility of GUI-based data mining procedures can be operated by the end-user with a minimum of knowledge on machine learning.

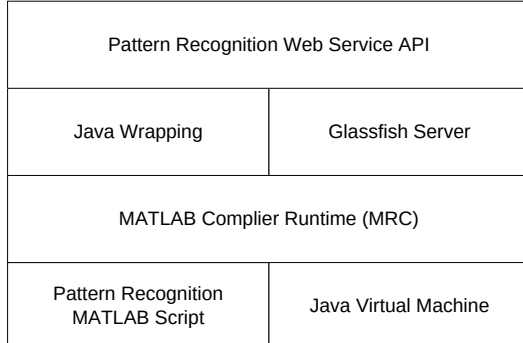


Figure 6-13 underline structure of the pattern recognition API

Since we implemented centralized and platform-independent software architecture, end-users can update algorithms in real-time without compatibility or package dependency.

6.3. Conclusion and Discussion

The beta test shows that a workload previously taking one month can now be accomplished within a week using the HT/HC database system. The major workload is now mostly on experiment preparation and image acquisition since these two procedures are labor-intensive. Normally, the experiment preparation takes one day to accomplish while image acquisition will take another day. However, the image and data analysis of this image set takes a couple of hours. Complex experiment design requires more time to prepare and often prone to errors. In contrast, image analysis and data analysis can easily adapt the increase of in the amount of data by simply adding more computational power.

In terms of software design, the HT/HC DMS follows a modular design and all modules are implemented in the form of web services, therefore, updating the system is virtually instantaneous. Moreover, this framework is very flexible as it allows connecting other web services. Consequently, a fast response to new progress in image and data analysis algorithms can be realized. Additionally, the seamless design of HT/HC database system has significantly decreases manual errors in the data transportation, image analysis and data analysis. Comparing with solutions such as CellProfiler[82] or ImagePro, our solution provides a unique approach for HT/HC image analysis. It allows end-users to perform high-profile HT/HC analysis with a minimum level of prior experience on image analysis and machine learning. Moreover, the modular design allows faster connecting with other web services such as BLAST [154]. Consequently, a faster response to progress in image and data analysis can be realized. Further integration with online bio-ontology databases and open gene-banks is considered so as to allow integration of the data from other sources. Therefore, the platform can eventually evolve into an interdisciplinary platform for cytomics.

Acknowledgement

The authors would like to thank Wouter Zomervrucht, Alice Bodanzky, Tijl Kindt, Arthur Stuivenberg, and Leah Winkel for their meticulous work on developing the prototype of the HT/HC data management system.