Scientific Information Management in Collaborative Experimentation Environments

Kaletas, E.C.

Publication date
2004

Citation for published version (APA):

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
Chapter 6

The DNA Microarray Application Case

Among many others, microarray experiments can be named as one of the most important emerging and fast growing application domains within the life sciences domain. DNA microarrays allow genome-wide monitoring of changes in gene expression levels in response to some stimuli, for instance, the expression of a gene in response to certain drug. With the current technology and the state-of-the-art robotization applied in the production of microarrays, characteristics of thousands of genes can be studied in a single experiment.

This chapter describes how microarray experiments are supported within the VLAM-G experimentation environment, with a two-folded main focus on modelling the experiment and on all information management related aspects. In this direction, first the DNA microarray technology and microarray experiments are described in details, followed by an example scenario that is presented as motivation. Then components of the DNA microarray application and its development (as an implementation of the methodology described in Section 4.6) are described. Finally, how the application can be used by a biologist is illustrated, building further on the example scenario.

Please note here that the DNA microarray application is one of the two extensively studied application cases that provided the base for developing VLAM-G and VIMCO. Development of this application case was described in Section 1.4 as part of the research approach followed in this thesis work. However, this chapter presents the development of the DNA microarray application as a step-by-step application of the methodology described in Section 4.6, which in turn demonstrates the applicability of the methodology to a real-life application and constitutes a proof of concept for this methodology.
6.1 DNA Microarray Technology and Microarray Experiments

Microarray experiments allow genome-wide monitoring of changes in gene expression levels to answer the question ‘what genes are expressed in a particular cell type of an organism, at a particular time, under particular conditions?’ With the current technology, characteristics of thousands of genes can be studied in a single experiment towards identifying genes responding to certain stimuli (e.g. drugs, toxins), monitoring gene expression changes during disease progression, better understanding of mechanisms of gene regulation, assigning functions to novel genes, and identifying physiological pathways [1, 2, 3, 4].

Microarray technology exploits the preferential binding of complementary single-stranded nucleic acid sequences. A microarray is typically a glass slide, onto which DNA molecules are attached at fixed locations called spots. With the current technology of DNA microarrays, it is possible to spot up to 40,000 sequences on an array. For gene expression studies, each sequence ideally identifies one gene in the genome.

The steps involved in a microarray experiment can be grouped into five major tasks, as shown in Figure 6.1. Each of these tasks is briefly described below (hybridization, scanning, and image analysis are described separately for clarity):

**Clone preparation.** Preparing DNA clones that are uniquely representing the genes of the organism under study for spotting onto microarray slides.

**Microarray production.** Spotting and binding the prepared DNA clones onto tiny glass slides (microarrays). The size of a typical microarray is 3x8 cm, and it may contain up to 40,000 spots.

**cDNA-probe preparation.** Extraction of the mRNA pools representing all expressed genes in the tissues or cell types under investigation, synthesis of cDNA copies and labelling with (usually) either a red or green fluorescent dye. Each dye represents a different tissue or cell type. Note that in the remaining of this chapter, the terms cDNA-probe and mRNA probe will be used interchangeably.

| Experiment definition |
|------------------------|---------------------|-----------------|---------------------|
| 1 Clone preparation    | 2 Micro-array production | 4 Hybridization, Scanning & Image analysis | 3 cDNA-Probe preparation |
|                        |                     |                 |                     |
|                        |                     |                 | 5 Data analysis     |

Figure 6.1: Overview of the tasks involved in microarray experiments
Hybridization. Treating the microarrays with one or more labelled cDNA-probes. cDNAs will only bind to DNA clones that contain identical or highly similar nucleotide sequence.

Scanning. Reading the level of fluorescence for each dye and spot on the slide. This produces high-resolution images. In case of multiple cDNA-probes, the difference in fluorescence between the red and green dye is a measure for difference of gene expression levels in the tissues or cell types under investigation.

Image analysis. Analysis of the image to quantify the relative levels of gene expression for each gene represented on the microarray.

Data analysis. Analyzing the numerical results to obtain information about (altered) gene expression patterns e.g. to identify genes with similar expression patterns through clustering and deduce relationships and pathways, to identify candidate disease genes, or to study the effect of drugs, chemicals or toxins.

6.2 A Motivating Microarray Experiment Scenario

Consider the scenario where a biologist at a pharmaceutical company is working on a new drug. Let us assume this drug is targeting a specific disease, for which the responsible gene is already known. The biologist wants to test whether the new drug is indeed preventing the activities of the target gene and under which conditions the drug performs more efficiently (e.g. concentration of the active materials). In order to investigate the activities of the target gene, s/he needs to carry out a microarray experiment. Figure 6.2 illustrates this example scenario.

To perform this complex study, the biologist first designs a microarray experiment. As shown in Figure 6.1, the steps in a microarray experiment are grouped into five major tasks: clone preparation; microarray production; cDNA-probe preparation; hybridization, scanning and image analysis; and finally analysis of the results. However, depending on the experiment setup and on the already available biomaterial in the lab, some of these steps (or even tasks) can be skipped. In our case, we assume that the pharmaceutical company does not prepare the clones itself rather acquires them from another biotechnology company, while all other required biomaterial will
be prepared in house. Thus, the experiment will consist of the steps involved in the remaining four tasks. Please notice here that the biologist should possess the necessary expertise about the procedure to follow (i.e. which steps are involved in each task) and the protocols to apply (i.e. how each step is performed). Due to its complexity, once the steps to follow and how to perform them have been defined, the experiment definition needs to be stored for future references.

During the actual experimentation phase, first the DNA clones obtained from the biotechnology company must be spotted on microarray slides using a spotter device. As mentioned earlier, current spotters are capable of spotting up to 40,000 clones on a single microarray slide (3 cm x 8 cm). The DNA clones uniquely represent the genes of the organism under study. At this stage, it is extremely important to keep track of which clone is spotted on which microarray slide and to which position on that slide, as well as the genes corresponding to the clones and the organism information. Meanwhile, samples from both untreated cells and cells treated with the drug should be prepared. Messenger RNA pools representing all expressed genes (active genes) in the cell types under investigation should be extracted from the samples, cDNA copies must be synthesized and labelled with a red or green fluorescent dye. Each dye will represent a different cell type, e.g. red dye for drug treated cells. Once both the microarrays and cDNA-probes are ready, they are hybridized, that is, microarrays are incubated with the two labelled cDNA-probes. cDNAs will only bind to DNA clones on the microarrays that contain identical or highly similar nucleotide sequences. The unbound probes will be washed away, leaving only the probes bound to some clones on the microarray. The hybridized microarrays are then scanned to read the level of fluorescence for each dye and each spot on the slide, producing high-resolution computer images. Analysis of the image will quantify the relative levels of gene expression for each gene represented on the microarray, resulting in around 300,000 useful data points for human. Please notice that it is crucial to log the detailed conditions and results at each step during the experiment, since without this information it is impossible to interpret the experimental results.

In order to see whether the developed drug indeed prevented the activities of the gene under study, the biologist has to analyze the experiment results. In our example, the biologist will cluster the expression results, in order to compare the target gene expression against the expression values of other known genes with similar activities. This comparison can also be extended to cover the results of the previous microarray experiments. Please notice that depending on the available information about the drug and target gene, several microarray experiments may be required, each using samples treated with the drug under certain specific conditions. In this case, information about all experiments has to be grouped to indicate the relation among them, and the analysis of the experiment results may include the usage of many software packages in combination, such as self-organizing maps for hierarchical clustering, or Pearson algorithm for statistical analysis. Only at this point, the biologist knows whether the target gene was expressed or suppressed after the drug treatment.
6.3 The DNA Microarray Application

The DNA microarray application is one of the application cases that has been developed within the VLAM-G. The main objective of the DNA microarray application is to manage the information related to microarray experiments performed at the Microarray Department of the Swammerdam Institute for Life Sciences (SILS) and the data/information generated by these experiments. This application has served as the main case study for many of the VLAM-G concepts and features.

6.3.1 The Expressive System

The system that is developed to support the experiment and information management requirements of DNA microarray application is called *Expressive*. The main component of the Expressive System is a database to store all relevant data/information about a microarray experiment. In addition, the necessary user interfaces for the manipulation and querying of the database contents, and tools for the analysis of experiment results have been developed as part of the VLAM-G environment. The following subsection describes the instantiation of the methodology given in Section 4.6 for the development of the Expressive System and its integration in VLAM-G.

Before describing the Expressive System in details, it is worth to note that the Expressive System distinguishes the manual (i.e. laboratory-related) and automated (i.e. execution- and analysis-related) parts of experiments. That means, the laboratory-related part of the experimental information is stored and maintained by the laboratory information management system (LIMS) installed at the SILS Microarray Department. Namely, information about the clone preparation, microarray production, probe preparation, hybridization, scanning tasks is stored and maintained by the LIMS. However, in order to analyze and interpret the experiment results/draw conclusions, some minimum set of information about the experiment must be available. This minimum set must include sufficient information to provide the correct context for the experiment and for the analysis of its results. Therefore, for microarray experiments, a copy of some of the laboratory-related experimental information is also stored and maintained by the Expressive System (namely information about microarray production, probe preparation, hybridization, and scanning tasks). On the other hand, information related to the analysis procedure as well as the analysis results (i.e. image analysis and data analysis) are only stored and maintained by the Expressive System. Figure 6.3 shows the relation between the Expressive System and the LIMS with respect to the contents of their databases.

6.3.2 Developing and Integrating the Expressive System

As also mentioned earlier, the DNA microarray application is one of the two extensively studied application cases that provided the base for developing VLAM-G and VIMCO, and its actual development was described in Section 1.4 as part of the research approach followed in this thesis work. However, this subsection uses the Expressive System and presents its complete development and integration in VLAM-G.
as an “example” of how a new application case shall be developed and integrated in VLAM-G. The purpose of this subsection is thus two-folds: First to describe the development of the Expressive System and its integration in VLAM-G; and second to illustrate the applicability of the methodology presented in Section 4.6 to a real-life application. As such, here the methodology presented in Section 4.6 is applied to the DNA microarray application, and this section presents and describes the involved steps. As a result, this subsection also demonstrates the applicability of the methodology to a real-life application and constitutes a proof of concept for the methodology.

Integration of the New Domain in VLAM-G

Domain integration considers developing the gene expression database, integrating and registering this database into the VLAM-G, and defining and registering the users. The steps taken to integrate the microarray domain into the VLAM-G are described below.

Step 1 – Study domain experiments. The study of microarray technology and experiments involved weeks-long discussions with the experts at the SILS Microarray Department, which formed the base knowledge on the domain. Further research in the literature complemented this base knowledge with the related work. This study resulted in the step-wise description of a typical microarray experiment.

Step 2 – Generate process-data flows (PDFs). The results of Step 1 are used to define the process-data flows (PDFs) for microarray experiments. The PDF for the clone preparation task is omitted here due to its complexity (it involves 51 steps). Figures 6.4, 6.5 and 6.6 respectively show the PDFs for microarray production, cDNA-probe preparation, and hybridization, scanning and image analysis tasks of microarray experiments. All these figures use the same notation as Figure 6.4.

Figure 6.3: Contents of the Expressive System and LIMS databases
6.3. The DNA Microarray Application

Figure 6.4: Process-data flow for the microarray production task of microarray experiments

Figure 6.5: Process-data flow for the cDNA-probe preparation task of microarray experiments

**Step 3 – Model PDFs using EEDM.** The data elements, physical entities, and activities identified in the PDFs are mapped to EEDM constructs. Namely, data elements and physical entities are modelled as subtypes of the DataStep class (see Figure 6.7) and activities are modelled as subtypes of the ActivityStep class (see Figure 6.8).

Note that the names of some of the steps may change at this stage. This is not only because the representations are becoming more formal, but also because more generic terminology is being used with more formal representations. For example, the term target is used in the EEDM-based constructs instead of the term mRNA-Probe
Figure 6.6: Process-data flow for the hybridization, scanning and image analysis task of microarray experiments

used in the PDFs.

**Step 4 – Identify and model common elements.** The elements that are common to different experiments are identified. Some of these elements are already identified during the definition of the PDFs (i.e. generic descriptive elements). Furthermore, the relationships between these elements and the experiment steps are identified and modelled.

Figure 6.9 shows the data elements and physical entities in a microarray experiment together with the elements common to all experiments, such as Organism and Gene. This figure also includes the ArrayTemplate class, which is defined to efficiently represent certain array designs that are frequently used in experiments. Note that the data type to hold analysis results is also defined at this stage, namely Ch-Measurement (to keep the measurement value for each channel/mRNA probe). Furthermore, the relationships among the data steps are identified and modelled here, such as the relationships between MicroArray and ArrayMeasurement and between ArrayMeasurement and Target.

Similarly, other elements related to the activities are also identified and modelled at this stage (see Figure 6.10). For microarray experiments, the only activity-related element that is common to all experiments is the laboratory protocol used during the activity. In order to model the utilized protocol, a class called Process is defined as a subtype of the EEDM ActivityStep, which is related to the Protocol class. Then
all activities that were initially modelled as subtypes of the EEDM ActivityStep are modified to extend the Process class. The instruments and software programs used during these activities are also modelled at this stage. For this purpose, two classes called DNALabHWTool and DNALabSWTool are defined, which respectively extend the HwTool and SwTool types of the EEDM. Then a relationship is defined between each activity that uses a hardware or software tool and one of these two classes. The result is the core database schema for the domain of gene expression studies.

**Step 5 – Generate the schema in ODL.** Matisse ODBMS is used for the implementation of the microarray gene expression database. Matisse requires the database schema to be defined in the Object Definition Language (ODL). Hence, data structures in the core database schema, as defined in the previous step, are mapped to ODL. Other data models required by the VLAM-G and VIMCO are also added to the ODL, such as the PFT data model, study-PFT links, origin-copy class, and the class for attribute properties. All these classes together comprise the Expressive DB
Figure 6.8: Mapping activity steps in the PDFs to EEDM constructs

schema. Besides the extensive study of the microarray experiments as explained in the previous steps, existing and emerging standards have also been considered during the design and development of the Expressive DB schema, such as MIAME (Minimum Information About a Microarray Experiment) [169, 170] and Dublin Core [81].

Due to its large size, the schema diagram of the Expressive DB is divided into five parts as shown in Figure 6.11. Class diagrams for the types in each part of the Expressive DB schema were given in the previous chapters. Therefore, the diagrams for these parts are not repeated here again. Please refer to Figure 4.14 for the EEDM (Part A), Figure 6.9 for the specific data types and common elements (Part B), Figure 6.10 for the specific activity types and common tools (Part C), and Figure 5.8 for the PFT Data Model (Part D). Other required data types in Part E include the study-PFT links (Figure 5.9), attribute properties (Figure 5.26), and origin-copies (as described in Subsection 4.4.2).

Although most of the constructs in the Expressive DB schema were described earlier, a brief description of the Expressive DB schema is presented below to provide an overall view.

The modelling approach of the Expressive DB reflects the flow-of-processes nature of microarray experiments. In Expressive DB, microarray experiments are modelled as a recursive flow of experiment steps, where steps can be either activity steps (activities), or data steps (data elements and physical entities). Although in theory they
can occur in any order, the steps in an experiment actually follow a specific order consistent with the experiment logic. An experiment step can in turn consist of other experiment steps. Such a recursive definition of experiment steps in Expressive DB
Figure 6.10: Activity steps, common tools (generic descriptive elements), and the relationships among them in microarray experiments

allows the modelling of any complex microarray experiment as well as the storage of any experimental information.
The `Project` type in the Expressive DB schema is used to group logically related experiments; however, it is also possible to represent a complete microarray experiment as a project, and the sub-parts of this experiment (e.g. microarray production, hybridization) as the experiments in that project. This allows logical groupings at different levels.

`Experiment` can represent a complete microarray experiment or a sub-part of it. `Experiments` can be ordered (hasNextExperiment/hasPrevExperiment links), for instance to model a time-series microarray experiment that look at the changes in the gene expressions with respect to time. Every experiment has one owner, usually the `User` who inserted this experiment definition. There can also be comments written about this experiment, either by the owner or by other `Persons`.

Experiments contain a set of steps, where every step can be either a `DataStep` or an `ActivityStep`. Steps in an experiment are also ordered in a similar way to experiments using hasNextStep/hasPrevStep relationships. A step can be an aggregate step, that is, it can consist of other steps (hasSubStep/hasSuperStep links). The `Step`, `ActivityStep`, and `DataStep` types, however, are not directly used in the Expressive DB; there are no direct instances of these types. Information about the steps of an experiment is represented as instances of the microarray specific types that are extending the base step types. For example, instances of `MicroArray` type are used to capture the information about the microarrays spotted and used in an experiment while `CellSample` objects represent the untreated samples used in the experiment. As such, all information specific to microarray experiment steps are represented by
the types that are extending the base Step type. Every step has a User performing the step. For each experiment step, some comments can be given by the performer or by other people. In case that a certain piece of experimental information is not explicitly modelled (for instance, when an institute decides to save the sequence of mRNA probes), such information is represented by Property objects as name-value pairs.

Instruments and software packages used during a microarray experiment are stored as instances of the Hardware and Software types.

There are some other types in the Expressive DB schema, whose instances are common to all microarray experiments and do not directly represent an experiment step, but are used in some steps (i.e. generic descriptive elements). Examples of these types are Clone and Gene.

The large bulk data generated during a microarray experiment is not stored in the Expressive DB. Instead, the data is saved in the VLAM-G archive (large-capacity and high-performance disk storage) as flat files, and pointers to the file locations are stored in the database; eliminating the performance decrease of the database that would otherwise occur when transferring large data sets. The array image, for instance, is stored in the archive as a tiff file, while a pointer to the file is kept in the corresponding ArrayImage object in the database. Results of the array image analysis, however, are stored in the Expressive DB. One ChMeasurement object is created for the intensity of each mRNA probe.

Next to the experimental steps and results information, the microarray experiment templates (PFTs) are also stored in the Expressive DB. In order to maintain the link between the PFTs and the studies created using these PFTs, the StudyPFT and StudyElmPFTElm types are defined; and to track the shared/reused objects, Origin-Copy type is defined. Finally, the AttProperty type contains information about how each of the attributes in the Expressive DB schema must be handled by the user interfaces. It specifies whether the attribute, for example, is a date, a pointer to a file, or link to another object. During start-up, the user interface reads the properties for all attributes and applies the appropriate handling mechanism when the scientist is working on an attribute.

**Step 6 — Create the database.** An empty Matisse database is created for Expressive DB, and the ODL file developed in the previous step was loaded to generate the database schema.

**Step 7 — Register the database.** The Expressive DB is registered to VIMCO by creating an instance of the DB class in the VIMCO DB. This object provides the information about the Expressive DB, such as the name of the Matisse database, the host running the database, the port number for the database, and the name of the driver to be used to access this database. Since all drivers implement the same interface, specifying which driver must be used is sufficient for VIMCO to access the database. Therefore, the database in fact can be implemented by using different DBMSs. For instance, relational Oracle can also be used to implement the Expressive DB if the driver for Oracle is available.
Step 8 – Define users. The Expressive users have been defined and registered in the three databases of VIMCO DB, RTS DB, and Expressive DB using the VIMCO user management functionality.

At this moment, the Expressive DB is registered as a new domain database to the VLAM-G. However, in order to store and manage gene expression data, PFTs for different parts (tasks) in a microarray experiment must be defined.

Integration of the New Application in VLAM-G

Application integration for the DNA microarray application considers the design and development of the PFTs for different parts of microarray experiments, development of software modules for data analysis, and defining user roles and access rights.

Step 1 – Define user roles and access rights. The two user roles defined for Expressive System are the Expressive User and Expressive Expert. Expressive User is the default user role for all microarray users, while Expressive Expert is the role for domain experts.

Access rights for these two roles are defined in terms of restrictions. Figures 6.12 and 6.13 show the restrictions for regular Expressive User and Expressive Expert users respectively.

<table>
<thead>
<tr>
<th>Role</th>
<th>Data Type</th>
<th>Data Source</th>
<th>Rest. Ops</th>
<th>Restriction</th>
</tr>
</thead>
<tbody>
<tr>
<td>PFT</td>
<td>Expressive</td>
<td>MACS</td>
<td>W</td>
<td></td>
</tr>
<tr>
<td>Study</td>
<td>Expressive</td>
<td>MACS</td>
<td>RW</td>
<td></td>
</tr>
<tr>
<td>Module definition</td>
<td>RTS</td>
<td>W</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Topology</td>
<td>RTS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Session</td>
<td>VIMCO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data source</td>
<td>VIMCO</td>
<td></td>
<td>W</td>
<td></td>
</tr>
<tr>
<td>User information</td>
<td>VIMCO</td>
<td>W</td>
<td></td>
<td>name = &quot;$User.name&quot;</td>
</tr>
<tr>
<td>Access rights (roles and restrictions)</td>
<td>VIMCO (for VIMCO data types)</td>
<td>RW</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>VIMCO (for MACS data types)</td>
<td>RW</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>VIMCO (for RTS data types)</td>
<td>RW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td>VIMCO</td>
<td>RW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Study-PFT links</td>
<td>Expressive</td>
<td>MACS</td>
<td>RW</td>
<td></td>
</tr>
<tr>
<td>Orgin-Copies</td>
<td>MACS</td>
<td>RW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Info data types (Session information, DB)</td>
<td>Expressive</td>
<td>MACS</td>
<td>RW</td>
<td></td>
</tr>
<tr>
<td>Write objects</td>
<td>ANY</td>
<td>RW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Get reuse objects</td>
<td>Expressive</td>
<td>MACS</td>
<td>RW</td>
<td></td>
</tr>
<tr>
<td>Get user type</td>
<td>MACS</td>
<td>RW</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 6.12: Restrictions defined for the regular Expressive User
Step 2 – Define PFTs. Seven PFTs have been designed for different parts of microarray experiments. These PFTs correspond to microarray production, sample preparation, RNA isolation, target preparation, hybridization, array image analysis, and data analysis. PFT for microarray production is given in Figure 6.14. It defines the microarray production task of microarray experiments, and contains two elements, namely ArraySpotting and MicroArray. As mentioned in Chapter 4, the first two elements of a PFT always correspond to Project and Experiment. The PFT also includes descriptive elements, such as for the comments and properties of Array Spotting and MicroArray, for the hardware/software tools and protocols used during the Array Spotting.

When designing the PFTs, the whole microarray experiment is divided into parts, each of which generates an output that is used by the next part as input. For example, the experiment part corresponding to microarray production PFT generates microarrays, and the experiment part corresponding to sample preparation PFT generates samples, which are used as input by the experiment part corresponding to the hybridization PFT.

Once the PFTs have been designed, they are defined and stored in the Expressive DB using the PFT Editor (see Figure 5.1).

<table>
<thead>
<tr>
<th>Role</th>
<th>Data Type</th>
<th>Data Source</th>
<th>Rest. Ops</th>
<th>Restriction</th>
</tr>
</thead>
<tbody>
<tr>
<td>PFT</td>
<td>Expressive</td>
<td>MACS</td>
<td>RW</td>
<td></td>
</tr>
<tr>
<td>Study</td>
<td>Expressive</td>
<td>MACS</td>
<td>W</td>
<td></td>
</tr>
<tr>
<td>Module definition</td>
<td>RTS</td>
<td></td>
<td>RW</td>
<td></td>
</tr>
<tr>
<td>Topology</td>
<td>RTS</td>
<td></td>
<td>W</td>
<td></td>
</tr>
<tr>
<td>Session</td>
<td>VIMCO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data source</td>
<td>VIMCO</td>
<td></td>
<td>W</td>
<td></td>
</tr>
<tr>
<td>User information</td>
<td>VIMCO</td>
<td></td>
<td>W</td>
<td></td>
</tr>
<tr>
<td>Access rights (roles and restrictions)</td>
<td>VIMCO (for VIMCO data types)</td>
<td>W</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>VIMCO (for Expressive data types)</td>
<td>W</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>VIMCO (for MACS data types)</td>
<td>W</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>VIMCO (for RTS data types)</td>
<td>W</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td>VIMCO</td>
<td></td>
<td>RW</td>
<td></td>
</tr>
<tr>
<td>Study-PFT links</td>
<td>Expressive</td>
<td>MACS</td>
<td>RW</td>
<td></td>
</tr>
<tr>
<td>Origin-Copies</td>
<td>Expressive</td>
<td>MACS</td>
<td>RW</td>
<td></td>
</tr>
<tr>
<td>Info data types (Session information, DB)</td>
<td>Expressive</td>
<td>MACS</td>
<td>RW</td>
<td></td>
</tr>
<tr>
<td>Write objects</td>
<td>ANY</td>
<td></td>
<td>RW</td>
<td></td>
</tr>
<tr>
<td>Get reuse objects</td>
<td>Expressive</td>
<td>MACS</td>
<td>RW</td>
<td></td>
</tr>
<tr>
<td>Get user type</td>
<td>MACS</td>
<td></td>
<td>RW</td>
<td></td>
</tr>
<tr>
<td>Authenticate user</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 6.13: Restrictions defined for the Expressive Expert
Figure 6.14: PFT for Microarray Production task of microarray experiments
Step 3 – Develop software modules. As part of the Expressive System, some software tools are being ported to the VLAM-G, so that they can be used in topologies and executed on high-performance computers for the analysis of microarray data. The analysis tools include among others image analysis tools (i.e. data extraction); visualization tools, necessary to visualize the extracted data in order to make decisions about what analysis steps should be followed; data normalization tools, necessary to correct systematic errors (e.g. differences in labelling between Cy3 and Cy5) or to correct print tip effects; and clustering tools, implementing a number of clustering algorithms, such as unsupervised clustering (hierarchical or non-hierarchical) and supervised clustering.

In addition to these microarray specific modules, Expressive System naturally benefits from the generic facilities provided by the VLAM-G, such as visualization and file management modules.

6.3.3 Data Loading

As mentioned earlier in this chapter, the laboratory-related information about microarray experiments is stored and maintained by the LIMS at the laboratory, while the minimum information about the microarray experiment is stored and maintained by the Expressive System. Since experiments are performed at the laboratory, the data and information are first inserted into the LIMS database. In order to extract the minimum information from this database and insert it to Expressive DB, an export/import tool has been developed. This tool reads the information from the LIMS database, re-formats the information as studies so that it complies with the PFTs defined for microarray experiments, creates the necessary objects for studies, creates the links between the studies and PFTs, and inserts the studies and all related information into the Expressive DB. Biologists can then view and manipulate this information, and analyze it within the VLAM-G.

6.3.4 Representing Different Types of Experiments

Figure 6.15 shows how a complex microarray experiment can be represented in the Expressive DB. The figure depicts a microarray experiment (represented as a project, printed in bold) consisting of three tasks (represented as experiments, printed in bold): Microarray production (Experiment-1), mRNA-probe preparation (Experiment-2), and hybridization (Experiment-3). Every task contains a number of data or activity steps. Some of the steps can be used in many experiments to represent a physical entity or data element generated in one experiment as output which is used in another experiment as input (e.g. the MicroArray generated in Experiment-1 is used in Experiment-3 during hybridization, depicted as a rectangle).

Figure 6.16 depicts a case where some of the experiments in a project follow a specific order (e.g. time-series experiments). In this figure, there are two groups of experiments, namely Experiments 1 - 3 and Experiments 4 - 5. Experiments have a specific order within each group. However, although they are related, there is no order among the experiments in different groups.
6.4 Using the Expressive System

In order to illustrate the functionality offered by VLAM-G to microarray studies, this section further builds on the scenario presented in Section 6.2 and describes the scenario to show how the VLAM-G can help the biologist in her/his microarray experiment.

**Designing the experiment.** In the example scenario, a number of DNA microarray PFTs are presented to the biologist, which contain the steps involved in a typical microarray experiment. S/he will go through these templates step by step, and enter the required information for each step into the fill-in forms displayed. Every form corresponds to a specific step in the experiment, and captures the information required for that step. For instance, the molarity of the solvent used in hybridization, resolution parameter used for the image analysis software, or comments of the biologist on the quality of the clones obtained from the biotechnology company.

**Performing the experiment.** The analysis of the scanned microarray images requires a certain amount of computational power and storage. In this step, the biologist selects the most suitable analysis module among the available biology related software modules, and attaches it to the two modules; one reading the images from the archive and the other storing the analysis results in the database. When defined, the experiment topology (consisting of these three modules) will be first saved and

![Diagram of experiment topology](image)

Figure 6.15: Example project with different microarray experiments in Expressive DB

![Diagram of different ordering of experiments](image)

Figure 6.16: Representing different ordering of experiments within a project in Expressive DB
then executed by the VLAM-G RTS. The VLAM-G RTS executes the modules in the topology on the Grid in a distributed manner using the Globus toolkit. This allows the user to specify the computational, networking, and storage requirements for the execution of her/his modules and acquire the required resources from a distributed pool of available resources. Furthermore, an intelligent assistant agent will provide scientists some statistical information, such as information about similar tools, most-used software modules, run-time statistics about a module, etc.

Analysis of the experiment results. In order to extract the knowledge and gain better insights into the problem under study, the data sets obtained after the experiment need to be analyzed. The analysis can involve among others utilization of high computational power, high-speed networks, storage facilities, and visualization tools. In the microarray example, the biologist will be able to choose the software modules to use in her/his topology to analyze her/his expression result set. The analysis procedure will first read the current experiment result set from the expression database, and perform a query on the database to retrieve the results of all experiments performed in the past for the same drug and for the same target gene. Then the current result set will be compared to the others to find any common expression behavior. Finally the comparison results will be displayed. Assume that there are two analysis tools available for such a comparison, one performing the comparison in a sequential way on the same machine, the other distributing the result sets on a cluster and compare in parallel. The biologist will select the second module, connect the database module as input generator to this analysis module (which reads the data sets from the database), and connect a visualization module (which displays the comparison results) and another database module (which stores the results in the database) as output consumers to the analysis module. Alternatively, s/he can modify the parameters of the analysis module, display the results, and repeat the analysis until a better result is obtained, and store the results in the database.

6.5 Conclusions

This chapter described a complex application case from the life sciences domain, namely the DNA Microarray Application. Due to its complexity, the study of the DNA Microarray Application was used and contributed to many of the information management related developments in the VLAM-G. In addition to providing general information about microarray experiments, this chapter described how this application case supports biologists as part of the VLAM-G virtual laboratory environment. Furthermore, this chapter represented the development of this application case in a stepwise manner, following the methodology described in Section 4.6.

Development of the DNA Microarray Application as part of the VLAM-G does not only illustrate the applicability of the methodology of Section 4.6 to a real life application, but has also served as the verification/validation ground for the information management framework defined in Chapter 4 and its implementation described in Chapter 5.