CHAPTER 1

Introduction

1.1 BACKGROUND

In 1999, I was asked by my manager to lead an application development team to lay out a strategic plan for the next generation of chemical information systems for Merck Research Laboratories. Back then, Java technology was entering its fifth anniversary, and the J2EE 1.0 specification was just launched by Sun Microsystems. However, almost all chemical information systems used by chemical, pharmaceutical, agricultural, and biotech companies were developed using vendor proprietary technologies such as MDL ISIS, which is the de facto industry standard. Although many people recognized that the cost of licensing, developing, and maintaining these legacy systems was high, an alternative to those systems was unclear. I have to admit that there was probably no viable alternative at all back then.

Since its inception 30 years ago, object-oriented technology has been successfully applied in software development in many industries for many years. However, it is a new beast even now in the chemical informatics domain. Many chemistry software vendors have been slow in reacting to technology evolution. As a user or developer, not many technological choices are available. As an employer, it is difficult and costly to find and recruit developers who have experience in those vender proprietary development platforms. There is also a fear factor in many organizations; moving away from existing technologies to new ones, no matter how promising they may be, is risky. This risk is true even though many of the limitations of the existing technologies justify the changes: performance and flexibility are low, whereas development, maintenance, and licensing costs are high.

From the middle to late 1990s, the situation changed when major chemistry software vendors started migrating their chemical information databases from proprietary formats to Oracle-based relational databases. Another positive move was that these vendors also started releasing chemical structure

Developing Chemical Information Systems: An Object-Oriented Approach Using Enterprise Java, by Fan Li

data cartridges using the Oracle® Extensibility Framework. These products included Accelrys® Accord for Oracle, CambridgeSoft® Oracle Cartridge, Daylight® DayCart, Tripos® Auspyx for Oracle, and MDL® MDLDirect. These changes were caused at least in part by the competition among these vendors. These cartridges enable people to use direct SQL to query and update chemical databases, something that could only be done using vendor proprietary programming interfaces in the past. Software developers in the chemical informatics field now have the opportunity to use open, industry standards and more interesting technologies to do their work (like it or not, having fun is one of the biggest factors of software development productivity).

Having programmed in Java since its inception, I was a firm believer that Enterprise Java could be one alternative to vendor proprietary technologies. I proved to my managers that I was right when we finally released the first compound registration system using J2EE at Merck in 2003.

Chemical information systems are complex because they process chemical structures–a very special and complex sort of data. Indexing and querying chemical structure data require special techniques, and a handful of software vendors that have the domain expertise have come up with data storage and query solutions. The complexity also deterred many organizations from developing customized chemical information systems in-house. Instead, they hire outside consultants to implement these systems on their behalf. Many software developers in these consulting firms are not professional software devolopers by training but ended up becoming programmers for one reason or another. I remember during the technology boom in the 1990's, many "seasonal" programmers wanted to find IT jobs. Many of them did so simply because they were tired of what they were doing and believed IT jobs were easy and less stressful. People were under the impression that one could become a good programmer by just attending a two-week programming training course and learning how to write a "Hello World" program—a gross misperception. Software development projects are challenging and costly. They require special skills and disciplined practices, or they may fail badly.

The advantage for chemists in developing chemical information systems is obvious: they know the domain subject e.g., chemistry and what the systems are supposed to do very well. The disadvantage is that they do not necessarily know what it takes to develop enterprise strength software systems. There are certain people who know both very well, but it is not always the case. The consequence is that the systems developed can be hard to maintain and debug and are not as good in performance and scalability as you may expect. In many cases, only the person who wrote the code can understand and maintain it. I do not mean to offend anybody because this is purely due to a lack of training and experience and has nothing to do with talent. Neither am I suggesting that

being trained in software engineering automatically makes a person a good software developer. In fact, many chemists working in the pharmaceutical and chemical industries have advanced degrees and have trained themselves to be good software developers. I was a physicist by training initially myself and acquired a computer science degree later in my career. I learned low coupling and high cohesion principles in graduate school. They turned out to be the two most important principles in software development that have guided me since then. Software development is both an art and an engineering discipline, which in my mind requires formal training, years of practice, and continuous learning and exploration of new and better techniques.

Chemical informatics may mean different things to different people. I am not here to provide an authoritative definition. However, as it is the topic of this book, I will give a definition from the IT aspect. Chemical informatics is about capturing, storing, querying, analyzing, and visualizing chemical data electronically. Modern chemical information systems are challenged to facilitate industry's productivity growth by effectively handling a huge amount of data. Making sure these systems are robust and high-speed is crucial to the competitive advantage of any discovery research organization. Chemical information systems usually require the following tools.

1.2 CHEMICAL STRUCTURE ENCODING SCHEMA

One of the most widely used chemical structure-encoding schemas in the pharmaceutical industry is the MDL® Connection Table (CT) File Format. Both Molfile and SD File are based on MDL® CT File Format to represent chemical structures. A Molfile represents a single chemical structure. An SD File contains one to many records, each of which has a chemical structure and other data that are associated with the structure. MDL Connection Table File Format also supports RG File to describe a single Rgroup query, rxnfile, which contains structural information of a single reaction, RD File, which has one to many records, each of which has a reaction and data associated with the reaction, and lastly, MDL's newly developed XML representation of the above—XD File. The CT File Format definition can be downloaded from the MDL website: http://www.mdl.com/downloads/public/ctfile/ctfile.jsp.

Other structure-encoding schemas are developed by software vendors and academia such as Daylight® Smiles, CambridgeSoft® ChemDraw Exchange (CDX), and Chemical Markup Language (CML), and they all have advantages and disadvantages. The MDL CT File Format is the only one that is supported by almost all chemical informatics software vendors.

Figure 1.1 is the structure of aspirin.

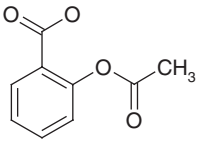

Figure 1.1 Structure of the aspirin molecule.

The Molfile representation of the above structure is as follows.

```
-ISIS-  07240513032D

 13  13  0  0  0  0  0  0  0  0  0999  V2000
-1.1556   -0.1291    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
-1.1568   -0.9565    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
-0.4419   -1.3694    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
 0.2745   -0.9560    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
 0.2716   -0.1255    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
-0.4437    0.2836    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
-0.4462    1.1086    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
-1.1667    1.5250    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
 0.9846    0.2897    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
 1.7006   -0.1201    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
 2.4135    0.2951    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
 1.7037   -0.9451    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
 0.2677    1.5221    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
 1  2  2  0  0  0  0
 6  7  1  0  0  0  0
 3  4  2  0  0  0  0
 7  8  2  0  0  0  0
 5  9  1  0  0  0  0
 4  5  1  0  0  0  0
 9  10  1  0  0  0  0
 2  3  1  0  0  0  0
 10  11  1  0  0  0  0
 5  6  2  0  0  0  0
 10  12  2  0  0  0  0
 6  1  1  0  0  0  0
 7  13  1  0  0  0  0
M  END
```

The Smiles representation of the same structure is far simpler: C(=O)(O)c1ccccc1OC(=O)C.

1.3 CHEMICAL STRUCTURE RENDERING AND EDITING TOOLS

MDL® ISISDraw and CambridgeSoft® ChemDraw are probably the most widely used structure editing tools. Both companies have a Web browser

plug-in version of these structure editing tools—MDL® ChimePro Plug-in and CambridgeSoft® ChemDraw Plug-in. MDL ChimePro also includes a JavaBean component, which can be used either as applets or in Java Swing based client applications.

Other products on the market include Daylight® Depict Toolkit, Accelrys® Discovery Studio ViewerPro, and Chem Axon® Marvin Bean.

1.4 CHEMICAL INFORMATION DATABASES

Data storage and querying are the most fundamental requirements of all informatics systems. Thanks to the Oracle® Extensibility Framework (a.k.a. Oracle Data Cartridge Technology), chemical structure data can be stored and queried using direct SQL and special query operators, such as substructure search, flexmatch search, similarity search, and formula search. Also, some indexing techniques make these otherwise slow searches fast. Detailed discussions about these databases and cartridges are beyond the scope of this book. Please refer to the vendor's website and product documentation for more information.

1.5 CHEMISTRY INTELLIGENCE SYSTEMS

These tools perform structure validations, making sure molecule structures follow certain conventions that are defined by an organization, property calculations such as molecular weight, molecular formula, pK_a, and so on, and salt handling. Many chemistry software vendors provide chemistry intelligence software. Some vendors may encapsulate chemical intelligence components in their data cartridge products. Some may bundle it with their structure editing tools. Some may offer it as independent products. MDL, for example, used to have it as part of its ISIS product suite. Now it has a product called Cheshire that is independent of ISIS and can be integrated with both Microsoft and Java platforms.

Since each organization has unique business rules, it is highly desirable that the chemistry intelligence software is flexible to allow customized implementations of chemistry rules handling. MDL Cheshire does a pretty good job from that perspective.

The above tools provide fundamental building blocks of chemical information systems. With these tools in place, you can pretty much develop customized solutions that meet your specific technical and business needs.