

# A Model-Driven Case Study of Developing a Complex Domain-Specific System

Emanuel S. Grant\*

University of North Dakota, Department of Computer Science, Grand Forks, USA.

\* Corresponding author. Email: [grante@cs.und.edu](mailto:grante@cs.und.edu).

Manuscript submitted January 10, 2015; accepted April 8, 2015.

doi: 10.17706/jsw.10.5.639-649

---

**Abstract:** The increasing complexity of software projects makes it necessary to conduct software development by a well-defined methodology. The initial phases in the development process are crucial in the building of a correct and robust system. The system design has to be verified by the user to ensure that all requirements are met before coding commences. Models, with their supporting tools and prototypes provide a graphical representation of the overall system.

In this case study, the benefits of applying a model-driven approach in the development of a complex domain-specific project is assessed. The case study involves the interpretation of mass spectra of unknown compounds. Mass spectrum is widely used in the fields of chemistry, forensic sciences, biochemistry, and other related fields. Analysis of mass spectrum is a tedious and time-consuming process. While there are software tools available for analysis of mass spectrum, their complexity and high cost, limit their use to a few institutions.

**Key words:** Domain-specific model, software engineering, software specification, user requirement elicitation

---

## 1. Introduction

Software developers are often required to work in domains in which they may have little or no knowledge. Every requirement in the problem domain has to be studied in detail so that users' expectations of the system and the developers' understanding are the same. Many software development methodologies require users' involvement towards achieving the desired product.

Communication between users and developers is crucial to the success of any project. Inadequate communication may result in an undesirable product or require many changes to the product after deployment. Developers are required to interview users about existing systems, their drawbacks, and improvements expected from proposed systems. Developers must be able to communicate on the feasibility and limitation of proposed systems. The simple exchange of thoughts through interviewing is not enough to communicate between users and developers, when both have little knowledge of the others' domains. The communication gap between the user and developer can be bridged by pictorial representations of the thoughts of expression. Models fulfill this requirement of pictorial representation of ideas.

A model is an abstraction of a real world observation [1]. Models provide visual views of the system to be built. Modeling is useful in removing inconsistencies in understanding between developers and users.

Users will be able to visualize the expected final product with the help of prototypes, specifications, and flow diagrams. In this way, specifications can be reviewed and corrected at the requirements and designs level. In this project, modeling concepts are utilized in the development of a complex project, specifically the analysis of mass spectrum [2]. Mass spectrum is a plot of ion abundance (intensity) versus mass to charge ratio. In the graphical representation, the x-axis is the mass to charge ratio and the y-axis represents the intensity of the ions.

### 1.1. Research Objectives

This report presents a demonstration of the application of modeling techniques in the development of a complex domain-specific project. The modeling techniques used in this case study aim at eliciting a correct and complete set of specifications for the system before proceeding with further phases of the development process. The focus is on bridging the communication gap between users and the developers. With the use of a set of UML requirements models, users participate in the requirements elicitation and verification system development processes. Users are able to envisage the progress of the system development and provide input to correct or guide the developers' specification of the system requirements. The focus of the work presented in this report is on the complexity of the problem domain and the application of a model-driven approach to eliciting system requirements to alleviate the complexity in the system description.

### 1.2. Problem Description

Mass spectra are used by chemists, and scientists in related fields to aid in the determination of unknown compounds' molecular weight, formula, and structure. The mass spectrum, of an unknown compound, is obtained by ionizing the unknown compound with electrons, small gas bombardment, or laser light. The resulting ions usually have high energies and may further disintegrate into ions of smaller masses (daughter ions) as it passes through a magnetic field on its way to a detector. The detector counts the ions reaching it and determines the mass/charge ratio ( $m/z$ ), while neutral species are undetected (see Fig. 1) [2].

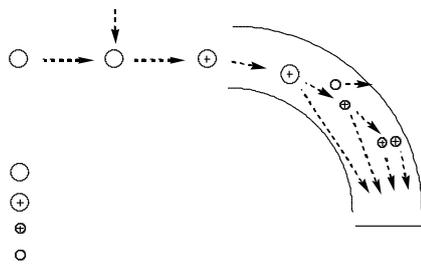


Fig. 1. Schematic of mass spectrometry.

The instrument plots the  $m/z$  ratio (determined by the detector) and the intensity of the peak, i.e. number of fragments corresponding to a particular  $m/z$  counted by the detector. This plot is the mass spectrum of the compound. The fragmentation pattern of the molecule depends on the structure of the compound. Analysis of the fragmentation pattern may reveal the structure of the compound.

There are different techniques available for interpreting the structure of the unknown compounds. A majority of these techniques require a relatively large quantity of the sample compound. In many cases, the compound is available in small quantities, so techniques such as NMR (Nuclear Magnetic Resonance Spectrometry [3], a technique to determine molecular structure) are not possible. Only mass spectrum can be performed for the structure determination when the quantity is found in traces.

The interpretation of mass spectrum of a compound is not trivial. Though mass spectrum is widely used, the user often may not have expertise in the interpretation of mass spectrometry. Because of this problem,

the mass spectrum is often underutilized and enormous amount of structural information is either lost or never realized.

There are a few computer systems (Probability Based Matching (PBM) and Self-Training Interpretive and Retrieval System (STIRS) [4]) available for determining the closest match for a compound by searching a database of mass spectrums of known compounds. The new compound is matched with the existing spectra, and if any spectrum shows similar pattern, it is reported with the matching percentage – There may be multiple matches. Thus, the system provides a list of chemical formulas with a specific match probability. The problem with this method is that even a 95% match may not confirm the structure of a component. Moreover, if the search does not find any match in the database, it fails to provide any structural information.

It would be beneficial to have a software system that complements the existing search approach in determining the structural features of the compound and its fragments. The greatest difficulty in developing such a system is the elicitation of the system requirements from the domain experts. This difficulty exists because of the high level of complexity, and specialization of the problem domain, i.e. unknown chemical compound mass spectrum analysis. This is a field, which relies on the expertise of highly skilled and experience domain experts.

### **1.3. Scope of the Work**

The initial system was designed for identifying compounds of low molecular weights, so that the number of possibilities of molecular formulas and fragments will be small, leading to easier interpretation. Once this has been successfully accomplished, the system may be evolved to handle compounds with larger molecular weights. Additionally, the work was limited to the identification of organic compounds. Because of the complexity of the problem, the work was further limited to those compounds for which molecular ions exist. The system is interactive – The system will be able to check the validity of the input data but only to a certain limit; the user will have full control to query the state of the analysis at any point during the processing of a mass spectrum.

The system will be able to provide possible molecular formula or formulae. Non-isotopic peaks will be used in determining the low mass ion series, which identifies the functional groups of the compound. By use of isotopic peaks and a fragment ion database, the system will be able to predict the possible composition and structure of the fragmented ions. The possible neutral losses directly from the molecular ion will be determined by taking the difference between the molecular ion and the fragment ions. The identification of the neutral loss will be performed by using the database of the neutral losses. In providing all this information, the proposed system would help scientists in analyzing the unknown organic compound.

### **1.4. Expected Results**

A model-driven software development approach is used in an effort to capture the requirements for a system to provide inference about the structure of an unknown organic chemical compound whose mass spectrum is provided. This work produced unambiguous models of the required system, from which source code may be generated using any programming language [5]. The implemented system will be able to provide important features of the unknown compound. Some of these features are:

- Report inconsistencies in the input data (e.g. the user will be alerted if the overall elemental analysis is not 100%),
- The molecular mass of the sample compound,
- The molecular formula of the compound,
- The logical neutral losses from the molecular mass and its chemical formula,
- The fragment ions with their chemical formulas and weight, and

- Functional groups present in the compound (e.g. alcohols, amines, carboxylic acids, etc.) and their match percent with the low mass ion series.

The system may not be able to provide the complete structure of the unknown compound, but it will provide valuable assistance to the user in determining the possible structure of the unknown compound.

## **1.5. Related Works**

Two computer-based systems that are used for the interpretation of unknown mass spectra were reviewed. The two systems are Probability Based Matching (PBM) [4][6] and Self-Training Interpretive and Retrieval System (STIRS) [4][7].

PBM is based on the retrieval system approach, where the unknown compound spectrum is compared with reference spectrum from a spectrum database. PBM uses two unique features; data weighting and reverse search for determining the best match. The data weighting is based on the probability of the abundance of mass. It takes into account that high mass fragments may break into small mass fragments, and thus the abundance of higher mass is less. The reverse search technique searches for the peaks of reference spectrum in the unknown spectrum. Depending on the weighting and the reverse search match probability is assigned to the reference spectrum. In PBM search, the user is required to provide all the peaks from the mass spectrum for the unknown compound to find the right match and reduce the errors [4]. The program gives the best fits with their ranking. This is a time consuming process as the search and comparison is performed on each mass spectrum in the database. In addition, if no match is found, the system fails to provide any information about the compound.

STIRS is based on the interpretive approach, wherein the system tries to interpret in the same way as a human would [4]. It is a self-learning process, which combines pattern matching with artificial intelligence to provide logical substructure of the compound [4], [7]. STIRS uses the fragmentation process of the compound [2] in its search for the reference spectra. It divides the mass spectra into classes according to its significant structural features. Then it tries to match the unknown spectra with these classes of spectra to find a perfect match. The match will indicate the similarity between the unknown spectra and the class of spectra, which possess certain functionality. The matches are found more effective in the cases when the substructure has distinct features [2]. While this is an effective method, its usage is limited to only few institutions due to the high cost of the software.

## **2. Research Motivation**

Mass spectrum has been used in various fields of sciences for analyzing unknown compounds. The first mass spectrometer instrument was built in 1918 [8]. As the commercial instruments are getting cheaper, more available, and easier to operate, the use of mass spectrum in many academic areas, research institutes, and industrial areas is increasing exponentially [7]. The interpretation of mass spectrum requires a skilled person in the field of mass spectrometry. Mass spectrum interpretation is similar to solving a puzzle, which may take a very long time even for a skilled interpreter. In compounds where the molecular masses are high, the possibility of assignment of fragment ions increases. In these cases, determination of the structure of the compound becomes a tedious task. Efforts to automate the interpretation have been taken, but they have limitations.

A number of chemists (domain experts) expressed the need for a software system that would aid in conducting mass spectrum analysis. Preliminary discussions between the experts and software developers highlighted the communication difficulties that would persist through the project. It was immediately determined that the greatest difficulty with this project would be the developers gaining an understanding of the problem domain, and conveying this understanding to the experts for corrective activities. Based on the developers experience, on prior projects, with model-driven development it was decided that the

Unified Modeling Language (UML) [9] notation would be used in developing 1) the models for communication with the experts, and 2) the design of the system.

The benefits of model-driven development are easily identifiable. Models (e.g. UML class diagrams) serve as high-level representations of a software system. These high-level models allow software engineers to focus on the structure, performance, functionality, etc. of the system. Models also serve as blueprints for development, documentation, and reducing time-to-market [10].

## **2.1. Background**

The primary influence for this work comes from the OMG's UML notation. UML is a graphical modeling notation that supports model-driven software development, as the system models are the fundamental entities that direct the software development process. A major benefit of the UML notation is in the use of the model for communication between developers and customers and for documenting the design decisions of the system development process.

## **2.2. UML**

The UML Specification describes the UML as "... a language used for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system" [9]. UML provides different models to be used in modeling different views of the system. One specific model is not enough to provide all information about the system. Therefore, it is important to find the combination of selected models to describe the pertinent features and functions of the system. There are two main types of diagrams; structural diagrams and behavior diagrams. The structural diagrams provide the static view of the system. Structural diagrams are of four types:

- 1) Class diagram: is composed of classes, interfaces and the relationships between them.
- 2) Object diagram presents the objects and the relationships between them. Object diagrams provide a snapshot of instances of the objects of a class diagram.
- 3) Component diagram models the physical aspect of an object-oriented system. It focuses on the visualization of the system's components (e.g. executables, tables, libraries, etc.) and their interaction to form an executable system.
- 4) Deployment diagram also models the physical aspect of an object-oriented system. It mainly deals with the static view of the management of hardware devices during the execution of the software.

Behavior diagrams provide the dynamic view of the system. This dynamic view is in contrast to the static view, as provided by the UML class diagram. They visualize and construct the dynamic aspects of the system. There are five types of behavior diagrams:

- 1) Use case Diagram consists of use cases, actors, and the relationships between them.
- 2) Activity Diagram is similar to a flowchart in which the flow of control is depicted between activities.
- 3) Sequence diagram is an interaction diagram (diagrams which illustrate objects, their relationship, and the messages dispatched between them) where emphasis is given to the ordering of messages in chronological order.
- 4) Collaboration diagram is also an interaction diagram where emphasis is given to the structural organization of the objects, which interact through messages.
- 5) Statechart diagram presents a state machine view of the entities of the system. It helps in the modeling of reactive objects (reactive objects respond to external stimuli).

The work presented in this report consists of UML class diagrams (analysis and detailed design), use cases and use case diagram, activities diagrams, and sequence diagrams. The UML state chart was don't developed in this project, at the time of the completion of the first iteration, but will be considered in future work on the next iteration.

### 2.3. Mass Spectrum Terminology

The complexity of the domain of work is highlighted in the following description of a subset of the domain-specific terminology. Mass spectrum is a plot of ion abundance (intensity) versus ion mass to charge ratio. In the graphical representation, the x-axis is the mass to charge ratio and the y-axis represents the intensity of the ions.

High-Resolution Mass Spectrometry (HRMS) allows an accurate mass measurement for a molecular formula. As the actual masses of elements are not integer values, the exact molecular mass is usually calculated with an accuracy of up to four decimal places. This allows scientists to distinguish between molecular formulas with the same integral molecular formula and the same integral molecular masses, but different exact masses.

The abundance of most of the abundant isotope (A) is set to 100%. The abundance of other isotopes (for example A+1, A+2, etc.) is found by normalizing their abundance with respect to the A's abundance. This is the relative intensity for isotopes of elements. Some relative intensity of isotopes of elements are listed in Table 1.

Table 1. Natural Isotopic Abundance of Elements

Element	A		A + 1		A + 2	
	Mass	Relative abundance	Mass	Relative Abundance	Mass	Relative Abundance
Hydrogen	1	100	2	0.016	NA	NA
Carbon	12	100	13	1.08	NA	NA

### 3. Methodology

Since interpreting an unknown mass spectrum is not a trivial task, a systematic model-driven approach was applied. First software requirement engineering was applied in documenting the users' requirements. Interviews with the domain experts often uncovered problems in interpreting the data. The experts' input was considered at each step to verify the understanding of the problem. The problem domain was defined; prototypes were developed to check for consistency between the problem defined by the experts and the problem understood by the software developer. Necessary corrections were made.

Next UML tools were used in modeling the system. Use case diagram described the services necessary for the system. The actions of each use case were described in detail and they were used in defining the activity diagrams. The activity diagram gave a clear picture of the flow of information and data in the system. The detailed activity diagrams helped identifying the domain objects, which were used in defining the class diagrams.

#### 3.1. System Requirement Specifications

The specifications and requirements are mainly based on the study from [2]. There are nine steps in the interpretation of mass spectrum in the system. A brief description of three of these steps is provided below:

- 1) Data Collection - The first step is collecting all data available for the sample compound. The data may include high-resolution mass spectrum (HRMS), molecular formula, elemental analysis, and mass/charge ratios with their relative intensities.
- 2) MI Identification - The next step is the identification of the molecular ion (MI). This is a very crucial step in analyzing the mass spectrum of the compound. Without the molecular ion, it will be difficult to determine the elemental composition or the compound's molecular formula.
- 3) MF Identification - After identifying the MI, the molecular formula (MF) is calculated. In cases where HRMS is available, the molecular formula can be found using the table of HRMS formula for various combinations of carbon, hydrogen, nitrogen, and oxygen.

- 4) Hydrogen Deficiency Index - After the molecular formulas are found, the index of hydrogen deficiency for the molecular formulas are calculated. This index is the sum of the number of rings, the number of double bonds, and twice the number of triple bonds.
- 5) User Assistance for MF Verification and A Peaks - The user is shown the calculated MI and molecular formulas for verification and correction.
- 6) LMIS Matches - After the user approval on some molecular formulas, each MF is analyzed separately for LMIS, fragments, and neutral losses.
- 7) Fragment and Neutral Loss Composition - Next, the fragment ions are analyzed for the determination of structure of the compound. The fragment ion can help in identifying the substructure of the compound.
- 8) Assign Weight to Fragment Formulas - After determining the fragment formulas, the fragment formulas are assigned weight. The previous process of finding fragment formulas and the neutral loss can lead to many fragment formulas for one fragment mass.
- 9) Report Results - For reporting information, care has to be taken to eliminate invalid choices wherever possible. There will be a molecular ion and related molecular formulas. With each MF, there will be a list of fragment formulas, neutral loss, and functional group information.

#### 4. System Models

System models were developed from the requirement specifications. The open-source StarUML tool [11] was used to develop the system models. It is a stand-alone program and is easy to use with minimal training.

##### 4.1. Use Case Diagram for the System

To illustrate the functional requirements of the system use case diagram was developed, Fig. 2.

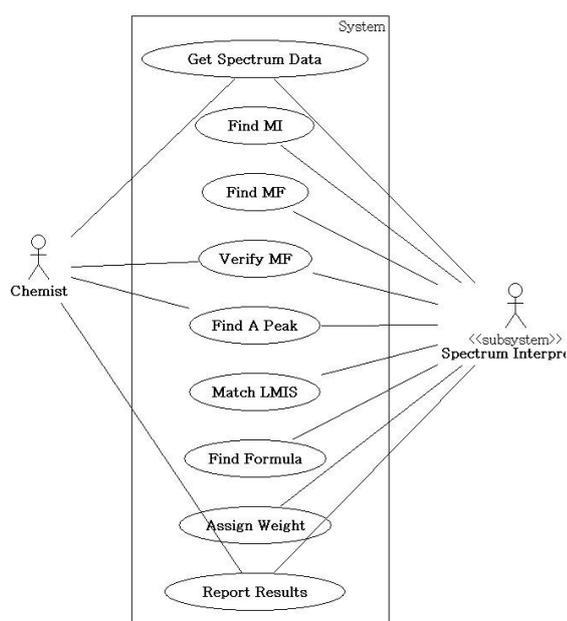


Fig. 2. Use case diagram of the system.

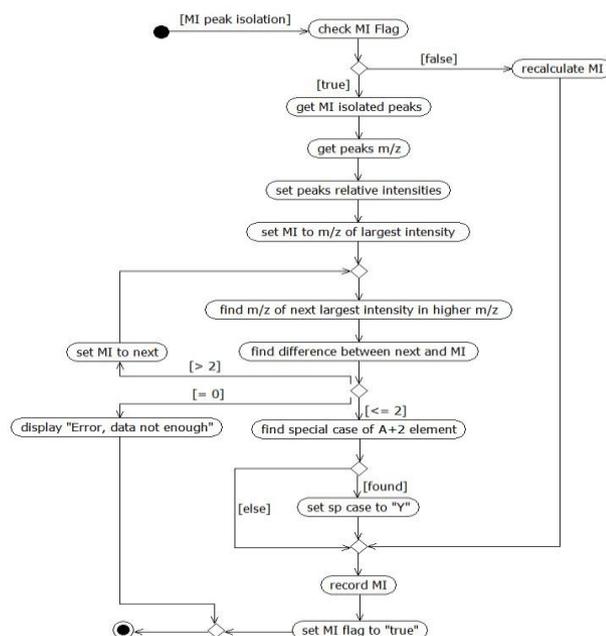


Fig. 3. Activity diagram for determining MI peak.

##### 4.2. Activity Diagrams for the System

To document the dynamic aspects of the system, activity and sequence diagrams were built. With the use case description, processes for the system were identified. Activity diagrams described the flow of

information to perform the process. Some of the activity diagrams are described here. The activity diagram for finding LMIS matches (Fig. 3) starts with setting the number of C atoms from the molecular formula. Then the series of values are formed using the general formula from one carbon to the number of carbons from the MF.

### 4.3. Analysis Level Class Diagram for the System

To represent the static structure of the system, the class diagram of Fig. 4 was developed. From the user requirements, static entities such as MF, MI, Fragment, Element were identified as classes. Their characteristics were taken as attributes for the classes. That helped in building the analysis level class diagram (see Fig. 4).

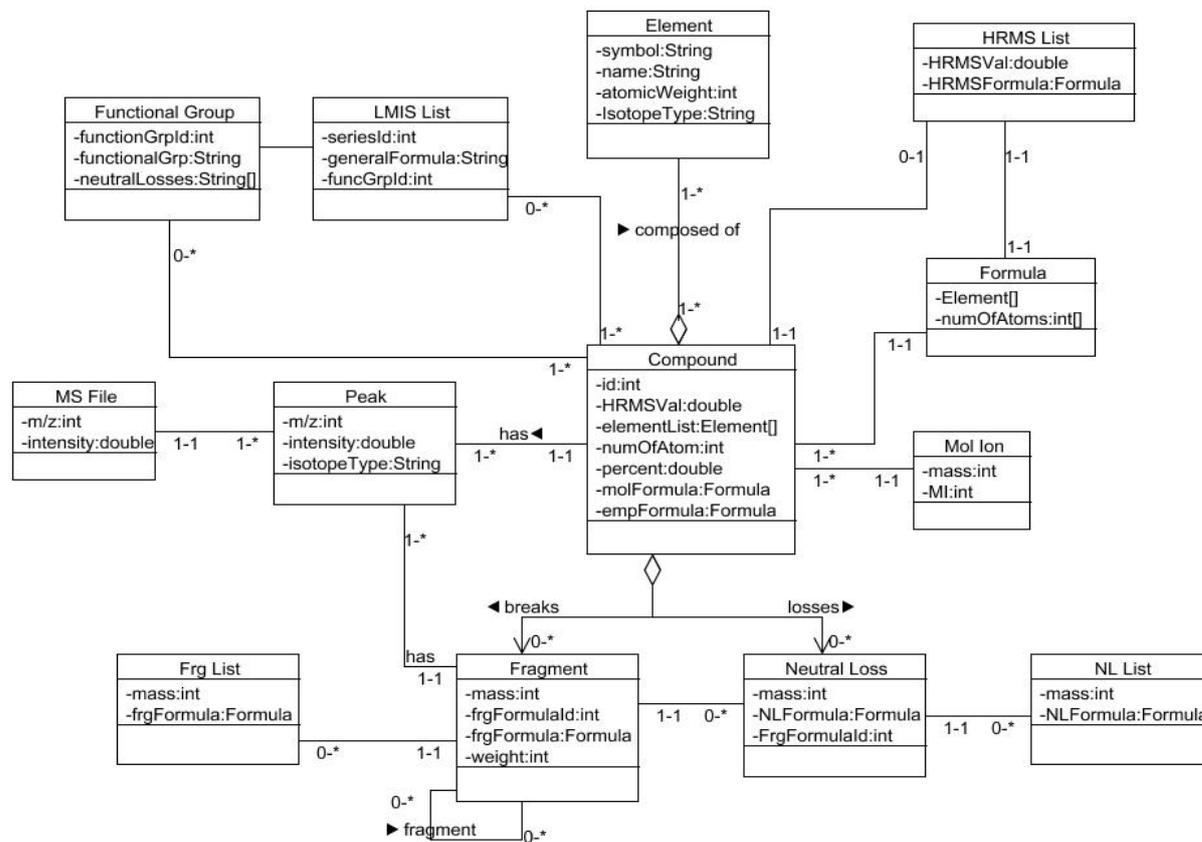


Fig. 4. Analysis level class diagram for the system.

## 5. Requirement Elicitation

The experts played a major role in the problem description, and the expected services of the system. The development process started with the requirement engineering where the system requirements were gathered from the user by interviews. The services evolved into the use cases of the use case diagram.

The experts provided tutorials to the developers on mass spectrum and its interpretation. The tutorials provide descriptions of the important features of spectrum, which is used in the interpretation, such as importance of the highest m/z to determine the MI and the base peak that is used to normalize all the peak's intensities. The common users usually do not have expertise in the field of mass spectrometry. It is difficult to interpret the enormous amount of data without an automated system. Due to these difficulties, many users are not able to use mass spectrum efficiently and end up using mass spectrum only for verification of molecular weight or verification of structural information obtained from other sources such

as NMR, IR, etc.

Notes were compiled throughout the interviewing process. Nouns were listed as candidate classes for constructing the class diagram. The properties of these nouns became the candidate attributes of the classes. As the field was new to the developers checks were made with the experts to verify the interpretation. Supplemental research on mass spectrum complimented the tutorial sessions. The major problem identified with the interpretation of mass spectrum data is that there is no standard procedure to solve a particular spectrum. Experts interpret the data based on their knowledge and experience.

### **5.1. Models Role in Communication**

Models played a major role in the research to facilitate the communication of ideas with developers and user. UML models are easy to understand even for people with little knowledge of software development. From the specifications provided by the experts, use cases were developed. The realizations for the use cases were done through activity diagram. Similar to use case diagrams, activity diagrams are relatively easy to explain. The use case diagram provided a comprehensive picture of the functional system requirements. Through the use case diagram, user could visualize their role in the system to be developed such as providing spectrum data and helping in MI identification.

The activity diagrams explained the steps required to perform the desirable result. The experts got a better picture of how the system will process the data. They related their interpretation of spectrum with the system's processing. In the activity diagram for finding functional groups for the compounds, the experts provided important information on tracking the last match value in the series. This helped in determining the match formula for the series. During the sequence diagram development, details were added to achieve the tasks described in the activity diagram. A number of difficulties in the realizations of some functions were uncovered.

Models provided the overall picture of the system without addressing implementation issues (software engineering principle of separation of concern). Models allowed the experts to visualize what the system would provide.

### **5.2. User Validation**

As the system was not implemented, final system validation could not be performed. The system validation was restricted to the capture of the system requirements. Experts input were taken throughout the research work. Steps for the interpretation were explained to the user in detail and the feedbacks were taken into account. The experts were kept informed on the progress of the system development. The steps in deciphering mass spectrum were verified by the experts. Each module function was explained to the experts and its role in the whole process in achieving the result. The experts also agreed on a user assistance process to control the number of molecular formulas to be analyzed. From the activity diagrams, some known mass spectrum data were analyzed to determine the results. The results were acceptable to the experts.

### **5.3. Report on User Exit Interview**

At the end of the model development process, the experts were asked to provide their feedback on the system development process. They found interviews and tutorials useful in providing the developers important information about the requirements and specifications for the proposed system.

The experts felt the process of modeling was useful in this complex project. Models helped them in understanding what they can expect from the proposed system. They liked the use case diagram, as it was simple to understand and yet provided useful information about the services they could expect from the system. The analysis level class diagram experts in understanding the relationships between the entities of the domain. The activity diagrams were most helpful as they gave a clear understanding of how the system

would process the data.

Sequential diagrams were very complicated to understand and were not so useful in the communication process. Interfaces were very useful in describing users interaction with the system. It also helped them in explaining their role in the system and the data they will need to provide. Overall, the experts were satisfied with the model development process for the proposed system and have confidence that the proposed system would be a helpful tool in the spectrum interpretation.

## 6. Conclusion

Through model-driven design, a new way was identified in which mass spectrum data may be analyzed without the use of large databases of spectrums. The implemented system will be useful for common users of mass spectroscopy. This system will be able to provide them with information about the unknown compound in determination of the structural features of the compound. The models will be used for the implementation of the system, in any specific programming language.

### 6.1. Things Not Accomplished

The research had planned to include functionality to determine the structural formula of the compound. Because of the complexity of the project and project time constraints, the work was limit to determining the chemical formula of the compound and fragments

The problems encountered on this project were due to a lack of knowledge of the problem domain. Additionally, the complexity of the project was underestimated. In the process of building system models, the intricacies of the system were revealed. Difficulties in achieving some of the processes were identified. Changes had to be made in the system design to resolve some of these difficulties. Without the use of models, these issues would have been exasperated in the later stage of the system development process. While the exact requirements proposed could not be achieved, a sound alternative was implemented to overcome the deficiencies.

### 6.2. Future Work

Future work will involve the implementation of the system. As the project is complex, the implementation process might require involvement of the experts. UML models are platform independent models (PIM), which was advantageous, as there was no need to focus on particular programming language during the system design. However, before the implementation of the system platform specific models (PSM) would have to be developed [5]. This is done in order to ease the process of implementation. PSM can help in the slow transition of models into code.

## References

- [1] Booch, G., Rumbaugh, J., & Jacobson, I. (2004). The unified modeling language user guide. Addison-Wesley, Boston.
- [2] McLafferty, F. W., & Turecek, F. (1993). Interpretation of mass spectra, 4th Ed., University Science Books: Mill Valley.
- [3] Silverstein, R. M., & Webster, F. X. (1996). Spectroscopic identification of organic compounds, 6th ed. John Wiley & Sons, Inc., New York.
- [4] McLafferty, F. W. (1978). Computer identification of mass spectra. *Pure and Applied Chemistry*, 50, 197-203.
- [5] Grant, E., Dazell, T., & Reza, H. (2008). Towards UML platform specific models for programming languages. *Proceedings of IASTED International Conference on Advances in Computer Science and Technology (ACST 2008)*.

- [6] Pesyna, G. M., Vankataraghavan, R., Dayringer, H. E., & McLafferty, F. W. (1976). Probability based matching system using a large collection of reference mass spectra. *Analytical Chemistry*, 48, 1362-1368.
- [7] Kwok, K., Venkataraghavan, S. R., & McLafferty, F. W. (1973). Computer-aided interpretation of mass spectra. *Journal of the American Chemical Society*, 95, 4185-4194, 1973.
- [8] Pavia, D. L., Lampman, G. M., & Kriz, G. S. (2001). *Introduction to spectroscopy*. Third Edition, Brooks/Cole.
- [9] Object Management Group. (2006). OMG unified modeling language (UML) specification. *Object management Group*, Needham, Massachusetts, USA, UML 2.0 edition.
- [10] Shane, S., & Wojtek, K. (2003). Model transformation: The heart and soul of model-driven software development. *Computer Society*, 20(5), 42-45.
- [11] Lee, M., Kim, H., Kim, J., & Lee, J. (2005). *StarUML 5.0 Developer Guide*.



**Emanuel S. Grant** received his B.Sc. from the University of the West Indies, Barbados. He received his MCS from Florida Atlantic University, USA, and he received his Ph.D. from Colorado State University, USA, all in computer science. His research interests are in software development methodologies, formal specification techniques, domain-specific modeling languages, and model driven software development.

He is an associate professor in the Department of Computer Science at the University of North Dakota, USA, since 2008, where he started as an assistant professor in 2002. He is an adjunct professor at the Holy Angel University, Philippines, where he is conducting research in software engineering teaching with collaborators from HELP University College, Malaysia; III-Hyderabad, India; Singapore Management University, Singapore; Rochester Institute of Technology, Baylor University, Montclair State University, and University of North Carolina Wilmington of the USA; and the University of Technology, Jamaica.

Dr. Grant is a member of the Association for Computing Machinery (ACM), Upsilon Pi Epsilon (UPE), and the Institute of Electrical and Electronics Engineers (IEEE).