**http://mitpress2.mit.edu/e-journals/LEA/ARTICLES/is_music.html**

*Leonardo Electronic Almanac*

Art – Science – Technology

Volume 7, No. 12

# Information Systems Development and Music: The Exploration of a Parallelism

Sasan Rahmatian, Ph.D.
Professor of Information Systems
The Sid Craig School of Business
Fresno, CA 93740-0007, U.S.A.
Email: sasan_rahmatian@csufresno.edu

## Abstract

The author examines numerous parallels between information systems and music. Roles played by individuals involved in the creation and performance of music are shown to correspond to parallel roles in the creation and implementation of information systems. Examples include correspondences between the user, in information systems, and the audience, in music. Similarly, many processes or elements---such as analysis, design and programming, among others---are common to both domains. For instance, thematic transformation in music composition can be seen to correspond to a form of programming logic in information systems. The author also notes that while many elements of information systems can be mapped onto musical functions, the analogy does not hold in all instances---in particular, the importance of subjectivity and interpretation in music does not have a parallel in information systems, where the complete objectivity of the computer and reproducibility of results are paramount.

The development of an information system and the creation and performance of a piece of music may be viewed as having a great deal in common. Information system roles such as user, analyst, designer, programmer and implementor can be mapped onto various music roles, such as audience, composer, conductor, performer, etc. (although not necessarily respectively). Going beyond personal roles, we can draw other analogies in regards the processes common to each domain, such as analysis, design, programming, testing, implementation and maintenance. Although the relationship is not an exact isomorphism, there are significant parallels. In exploring these parallels, I will present here the most common elements from the domain of information systems development, and for each explore the corresponding element from the realm of music.

## User

"The user" in information systems would correspond in the domain of music to the individuals who hear the music, i.e. the audience. In information system development, the user community can be either finite and well-defined or broad and ill-defined. The case of an information system tailored to the customized needs of a finite, well-defined user community would correspond to music commissioned by an individual (or by a group) for a particular purpose, such as a court function in past centuries, or a special event/inauguration in more modern times. The case of an information system designed for a broad and ill-defined user community (e.g. generic, off-the-shelf software) would correspond to music that is composed with no particular audience in mind. A further analogy is this: What starts early on as a customized program (such as very early accounting programs) may later spawn generic versions if the user base is broad enough to economically justify it. Likewise, a commissioned piece of music may later become accessible to a wider public due to demand.

## User Requirements

In the case of commissioned music, it is easy to see the parallel with information systems: the user requirements are the parameters within which the commissioned musical piece is defined, such as length, form, style, instrumentation and even emotional content (such as in film music). In the case of non-commissioned music, there are no such parameters, causing an apparent breakdown in the analogy.

But the question arises: In the absence of a well-defined user community in an information system situation, how are computer applications developed? The answer is something along the following lines: An unfulfilled need in the community is identified (hence defining a market), the size of that market is estimated, and the specific needs of that market are anticipated. In the case of music, the analogy does break down when we realize that music has an expressive nature. In certain situations (call it the "solitary scenario"), music would be composed for the sheer pleasure of the composer, even if there were no audience to hear it, just as an individual would write poetry just for the personal joy of expressing oneself. However, with professional musicians who depend for their livelihood on the public performance or recording of their music, this solitary scenario does not apply. A professional musician, hoping to "sell" to a market, needs to have a clear sense of the market, the size of the market and the demands of that market, just as in the information system case.

## Functionality

In information systems development, a system can perform one or more of several broad types of functions, such as transaction processing, management reporting, decision support, etc. Likewise music supports broad categories of functionality, such as identification (advertising jingle), support (film music), arousal (military march), relaxation (elevator music), romance (serenade), dance (ballet), etc.

## Analysis/Design

Systems analysis is the process of determining user requirements. Systems design is the process of creating detailed blueprints of the major components of the system aimed at meeting user requirements. Analysis and design can both be understood in terms of processes and products.

### The Processes of Analysis/Design

The processes of analysis and design involve a critically important element: modeling. Information system analysts and designers use graphic models to represent preliminary ideas. They then refine these models in a series of iterations. With each iteration, a model becomes more and more precise. Computer aided software engineering (CASE) tools are often used to electronically store the initial model. Once the model is digitized, it becomes a relatively painless task to modify it.

Likewise, with music systems, composers initially sketch preliminary ideas and then refined them in stages. With the aid of modern computer technology, these sketches are digitized, making it easier for later refinement.

### The Products of Analysis/Design

The products of analysis/design typically include user interface, database, program structure and program logic.

#### *User Interface*

A major difference between a computer application and a piece of music is that the former contains an internal, hidden set of components behind the one component with which the user interacts directly, i.e. the user interface. In music, there are no hidden components; every single note is transparent and audible to the user. The entire piece is the user interface. While this is generally true, it is "more true" of relatively simple pieces than of complex ones. In a complex orchestral piece consisting of over 100 instruments, it is very difficult---and perhaps not even desirable---to hear every single detail, especially at those times when a large number of instruments play at the same time. In such situations, what listeners tend to perceive are layers of sound, one on top of another, some more accessible ("foreground"), others less so ("background color"). The most accessible layer is the melody, which stands out on top of everything else going on at the same time. This melody can be interpreted as the interface between the listener and the piece of music. The melody is the face the music shows to the listener. It is what the listener remembers after all the background materials die off.

### *Database*

In an information system, the database is the system's memory. As such, its overall structure and contents are invisible to the user. Various subsets of its contents are visible to specific users, depending on the tasks they have to perform. Again, in music, there is no invisible (inaudible) component.

But if we reconceptualize the database as that which makes the system "remember" and then use at a later time something that happened at an earlier time, then we can think of a motif as serving the same purpose in music. A motif is a theme that is introduced, then later elaborated on and developed. It is a thread that runs through a piece of music, as if to remind the listener of where the music has come from and pointing to where it is possibly headed. Evoking memories of events past is sometimes done in more explicit ways, such as at the beginning of the last movement of Beethoven's Ninth Symphony, where themes from earlier movements are momentarily remembered and explicitly quoted.

### *Program Structure*

The analogy between program structure and music structure is quite strong. A well-designed, modularized computer program can be broken down from top to bottom, yielding smaller and smaller programs at each step, until we reach the lowest level, which consists of individual lines of code. The structure of the overall program refers to how all these programs of various sizes at different levels hang together as a unified whole. Similarly, in music, a long symphony may be broken down into movements, each movement into sections, each section into smaller segments, etc., until we reach the lowest level, consisting of individual tones. *"Musicians are trained to always bear in mind both the minute details of a musical moment and the overall structure of the piece that surrounds the moment. This ability is absolutely critical to systems work, where a failure in thinking at either the micro or macro level is equally disastrous"* [1].

There is yet another parallel: in studying various program structures, one often encounters a situation in which the same module (subprogram) is called on by more than one parent module (larger program), because of its versatility and usefulness. That is why such modules are referred to as "utility modules." Likewise, in music, one sometimes encounters situations where a theme is called on in more than one parent module (or movement). The *idee fixe* in Hector Berlioz's *Symphonie Fantastique* is a prominent example of this phenomenon.

### *Program Logic*

Program logic refers to the sequence of steps taken to convert the given data into the information produced at the end. Music has its sense of logic, too. But, as in program logic, there is not a single logic that applies to all pieces of music, but rather patterns of such logic. For instance, a developmental logic used by Franz Liszt throughout his music is thematic transformation, where he states a series of themes and then submits each one to various recognizable transformations in rhythm, dynamics, etc. A fugue is another example of musical logic.

The program logic analogy can be taken much deeper. Structured programming has been proven to be exclusively based on three control structures: sequence, selection and iteration [2]. Likewise, music composition is based on three dimensions: melody, harmony and rhythm. Any piece of music can thus be conceptualized as a dynamic, unidirectional line in this three-dimensional space, consisting of a permutation of the three basic elements at any given moment. Going one step further, the three control structures of programming can be nested in various ways (such as selection within selection, selection within iteration, iteration within selection, iteration within iteration within iteration, etc.) to provide versatile structures capable of coping with higher levels of complexity. Likewise, in music, nesting is possible: melodies incorporating harmonic sequences, rhythms embodying melodic fragments, etc.

### Programming

Programming corresponds to composing. The output of the programming activity is a program written in a particular language. The output of the composing activity is a music score, also written in a particular language. A computer program is a set of instructions for performing a specific task, expressed in a language understandable to the computer performing it. Likewise, a music score is a set of instructions for performing a piece of music, expressed in a language understandable to the musicians performing it. A well-written program is designed first before it is actually written. In the same vein, a well-written piece of music is designed first before it is documented, or notated on paper.

### Hardware

The work performed by an information system is often done as a collaborative effort between people and computer hardware. Likewise, performing a piece of music involves a collaborative effort between performers and their hardware, i.e. their musical instruments. In some instances, an information system may be a completely manual (non-computerized) system. Likewise, we may conceptualize a vocal performance as a manual, hardware-free (instrumentless) system.

### Testing

Testing a computer program before it is put into production is parallel to rehearsing a piece of music before it is actually performed in a concert. The parts that do not work correctly are identified and subjected to alternative treatments (such as different syntax in information systems and different fingering in music).

## Implementation

If by "implementation" we mean running a system of computer programs that would produce meaningful output for the user, then this would parallel the performance of a piece of music.

## Subjectivity

There is an element in music that does not correspond to anything in information systems: interpretation. Music is essentially a triadic relationship between the composer, the audience and the intermediary, known as the performer. The challenge facing the performer is to understand the musical intentions of the composer and convey them to the audience in performance. However, in reality each individual is unique with distinctive life experiences. No two performers understand the composer's message in exactly the same way. The subjective elements associated with the performer's mind and emotional makeup act as a filter through which the composer's intentions travel before they reach the audience. In listening to various performances of the same piece of music, we witness a situation in which the same input (notes) produces different outputs (performances). These individual differences, collectively referred to as interpretation, constitute the very human essence of music.

By contrast, the computer is an entirely objective medium of execution. Looking at a management report produced by an information system, one cannot make any inferences regarding the specifics of the computer system that produced it. The very point of automation is to delegate the human task to the machine, thus eliminating the human's role from the process. While information systems do not encompass the kind of subjectivity associated with the composer-performer-audience relationship, they do involve a different sense of interpretation. In developing and using information systems, we tend to run into the following forms of subjectivity:

### Systems analyst's understanding of user requirements.

Consider the various ways in which one may understand the word "efficiency" (as in "we need a more efficient system"). Depending on the desired result, the analyst or the user may interpret "efficiency" (the reduction of resources in producing a certain level of results) in various ways. This variety in interpretations would arise from the ambiguity as to which type of resource need to be reduced: human, financial, material or time. Whereas the user may have in mind a more cost-effective system, the analyst may hear a request for a faster system.

### User interpretation of information.

Different users may interpret the same piece of information in different ways, depending on their interests and viewpoints. For instance, a bank's annual report may be interpreted one way by the stockholder (when he or she notices the bad debts incurred), another way by the investor (when he or she notices that despite losses, the bank made money and paid handsome dividends), and still a third way by a regulator (who looks for compliance with regulatory ratios).

## Maintenance

Maintaining an information system involves revising it to meet the needs of new, emerging circumstances. This would correspond to revision, transcription or arrangement in music. In classical music, revision is a common phenomenon. Composers are known to revise their music years or even decades after first creating them. Transcription involves the arrangement of a piece of music for a different medium and is often undertaken in response to new, emerging needs. For instance, many large-scale orchestral pieces have been transcribed for the piano because the assembly of a large orchestra is a far more daunting task than securing a piano. Also, many pieces written for older, simpler instruments have been arranged to take advantage of the opportunities offered by their modern counterparts. For instance, a number of pieces by J.S. Bach have been substantially modified (notably by the Italian composer and pianist Ferruccio Busoni) to fit the opportunities offered by the modern piano.

## Management

To the extent that an information systems manager is involved in the planning, organizing, staffing, motivating and controlling of new systems development projects, a conductor can be seen as the information systems manager's musical analog. Like an information systems manager, a conductor (usually in the combined role of music director) has to make critical decisions pertaining to the planning and organizing of a concert season, staffing (specifically, selecting guest artists), motivating the players during rehearsals in the face of formidable technical difficulties and grueling schedules, and supervising the project as a whole. The conductor is indeed a manager.

| *MIS* | *MUSIC* |
|---|---|
| **1. USER** | **Audience (Commissioned/Generic)** |
| **2. USER REQUIREMENTS** | **Specified Parameters** |
| **3. FUNCTIONALITY** | **Identification (commercial jingle)**<br>**Support (film music)**<br>**Arousal (military march)**<br>**Relaxation (elevator music)**<br>**Romance (serenade)**<br>**Etc.** |
| **4. ANALYSIS/DESIGN** | |
| **4.1. PROCESS** | **Modeling**<br>**Stepwise refinement** |
| **4.2 PRODUCT** | |
| ***4.2.A.* User interface** | **Foreground (Melody)**<br>**Background (Harmony)** |
| ***4.2.B.* Database** | **Motif** |
| ***4.2.C.* Program Structure** | **Functional Decomposition (Hierarchy)**<br>**Modularization** |
| ***4.2.D.* Program Logic** | **Specific (Fugue, Thematic Transformation, etc.)**<br>**Generic (Three control structures)** |
| **5. PROGRAMMING** | **Notating** |
| **6. HARDWARE** | **Musical Instrument** |
| **7. TESTING** | **Rehearsal** |
| **8. IMPLEMENTATION** | **Performance** |
| **9. ??????** | **Interpretation(Composer → Performer → Audience)** |
| **10. MAINTENANCE** | **Revision/Transcription/Arrangement** |
| **11. MANAGEMENT** | **Conducting** |

## References

[1] R. Hunter, "Listen to the Music," *InformationWEEK* (August 28, 1989), p.48.