

Speech-Enabled .NET Framework Application for CIMS

by Murat SEMERCI
Çağdaş Kayra AKMAN

Instructor: Assoc. Prof. Levent ARSLAN
Project Advisor: Sibel MOYANO
Department of Electrical and Electronic Engineering

June 2004

SPEECH ENABLED .NET FRAMEWORK APPLICATION
FOR CIMS

by

Murat SEMERCI
Çağdaş Kayra AKMAN

Submitted to
Electrical and Electronic Engineering Department
as a requirement for the degree of
Bachelor of Science
in
Electrical and Electronic Engineering

BOĞAZIÇI UNIVERSITY

June 2004

To our families,

without them, we wouldn't be where we are now

Acknowledgement

We would like to express our sincere appreciation to our instructors for providing guidance and assistance during the development of the project, especially to Assoc. Prof. Levent ARSLAN and Sibel MOYANO.

Contents

Abstract.....	1
1. Introduction.....	2
2. Overview.....	3
3. Dialog Flow.....	5
4. Grammars.....	12
5. Prompts.....	14
6. Database Operations.....	15
7. Conclusion.....	20
8. References	21
9. Appendices.....	22
Grammars.....	22
Prompts.....	52

Abstract

In this project, an Interactive Voice Response system, speech-enabled voice-only version of Electrical and Electronic Engineering Department's Course Information Management System (CIMS), is developed. Microsoft's Speech Application SDK Beta 4 integrating into ASP.NET class library and Visual Studio .NET 2003 IDE is used in the project. User inputs are taken via a speech recognizer and information is conveyed to the user via a text-to-speech engine. Users can access the Grades, Announcements and Assignments pages of CIMS in the voice-only version.

1. Introduction

Speech Recognition and Text-to-Speech are two of the most important applications of Digital Speech Processing. One of the main motivations for mastering in this area is the goal to remove the obstacles in human-machine communication. The more humanlike the interface of computers or other machines/devices is, the more people will be able to use them. A more fundamental assumption is that it is in favor of all people if more and more individuals keep up with technological advances. Based on this assumption, digital speech processing research and development activities become very important from many points of views such as economical, social, political, educational perspectives, etc. Applications interacting with their users are available for a long time. It hasn't been very long that commercially justifiable products are available in the market or can be developed on demand.

Interactive Voice Response (IVR) systems are one such example that helps reduce the institutions their communication costs by automating repetitive tasks requiring labor force. Microsoft developed a software development platform, MS Speech Application SDK and Microsoft Speech Server, built on Microsoft's .NET Framework and Windows 2003 Server architectures. This platforms provides speech recognition and text-to-speech engines for the English language, and a library extending web development capabilities of ASP.NET so that voice-only and multimodal web applications can be designed, developed and deployed the same platform and the same IDE used for other .NET projects, namely Visual Studio .NET.

In our project, we aimed to design a speech-enabled voice-only web application that is accessed via PSTN and used only by spoken commands and prompts. CIMS, our department's Course Information Management System was the source system to be converted into a speech-enabled voice-only web site. User inputs were to be taken by a speech recognition engine and the prompts conveying information to the user were to be either composed from prerecorded phrases or converted to speech by a text-to-speech engine.

2. Overview

Boğaziçi University Electrical and Electronics Engineering Department Course Information Management System (CIMS) is a web site that offers a common platform for exchanging any kind of information between instructors, students and assistants. It is a password protected web site. Students can access to information related to the courses they are taking in that semester such as course syllabus, grades, announcements, assignments and course materials. Moreover, forums on a variety of topics, training information, calendar, email, and account management are some other functions on the web site.

In the voice-only version of CIMS, a small but essential subset of available functions is offered to users. Currently, the information available on the "Grades" and "Announcements" pages in the online version is accessible on the voice-only version. Moreover, users are prompted about new announcements.

The voice-only version of CIMS is used only by speaking out the necessary information asked by the system. No telephone key is pressed while browsing.

The project is developed using the Beta 4 version of the Microsoft Speech Application Software Development Kit (MS Speech App SDK). Speech Application SDK extends the capabilities of .NET Framework for multimodal and voice-only speech-enabled web applications. The SDK requires ASP.NET to be installed beforehand. Available classes and development tools can be used via Microsoft Visual Studio .NET integrated development environment. This project has been developed using Microsoft Visual Studio .NET 2003.

The project employs the conventional client/server architecture of the Internet. The user accesses a web browser via PSTN or local telephone network. This web browser acts as a client for the web server, which is called a Speech Server. Speech recognition and text-to-speech operations are realized on the client side. The server processes the information post back by the client and sends the necessary information to the client. The results generated by the speech recognition engine are returned to the server in form of SML output. On the server side, the database also

accessed by the online version of CIMS is used to retrieve the necessary information about the user or about the courses taken by the user that semester.

There are mainly two types of files in the project, ASP.NET (.aspx) and C# (.cs) files. .aspx files contain HTML, ASP, SALT and JScript codes. The corresponding .cs file - each aspx file has a code-behind page, another file, with the same file name but .cs extension - contains the server side code related to that page, such as initializations done upon loading the page or event handling functions upon data that are posted back by the loaded page at the client. Our project consists of several .aspx, and corresponding .cs, files accomplishing different tasks. The sequence of .aspx files loaded reflects the dialog flow.

SALT stands for Speech Application Language Tags, a markup language proposed by a number of IT companies and computer and network hardware manufacturers to World Wide Web Consortium as standard specifying API independent integration of speech into web applications. Speech Application SDK generates the necessary SALT codes via Visual C#.NET and extended ASP.NET component of .NET Framework. Since the .NET Framework is accessible to the full extend within C #classes, any task available on servers can realize and the results can be exchanged using the Internet with remote users either through conventional or speech-enabled web interfaces.

Database connections, extensive error handling and appropriate prompting, data processing based on the last access time, and the Intelligent Mode enabling users to retrieve information without imposing restrictions on uttered phrases are the main features that required much coding to keep the interface as simple as possible. Voice-only version of CIMS is compatible with the online version. Students' user names and passwords are also valid in the voice-only version. Students' last access dates are updated in both versions.

Development Environment:

Technologies used: ASP.NET, ADO.NET, Microsoft Speech Application SDK Beta 4

Development languages: mainly Microsoft Visual C#.NET; partly SALT, JScript coding was necessary.

IDE: Microsoft Visual Studio .NET 2003 Enterprise Architect Version

Database Server: Microsoft SQL Server 2000

Web Server: Microsoft Internet Information Services (IIS) 5.0

Workstation: Windows 2000 Professional Edition Service Pack 4 on IBM compatible PC with Intel Pentium III 800 MHz CPU and 256 MB (2*128 MB) RAM.

3. Dialog Flow

Voice-only version of CIMS is used in a very similar manner as the online version. First the username and the password, then the course ID and a menu item are asked. The order of submitting information to the web site is exactly the same as the online version.

This section contains detailed information on the dialog flow during browsing the CIMS web site.

1. Welcome Prompt:

It is a QA control only with a prerecorded welcome prompt that is played once. The user is informed about the system s/he has reached.

-Defined in: Default.aspx

-Next page loaded: UserName.aspx

2. UserName:

The user is asked to enter his/her user name. In case of CIMS, user names are students' school numbers. These may either 7, in case of students admitted to school before 2000, or 10 digits long. The associated grammar covers all possible 7 or 10 digit numbers.

Possible errors:

- i) No Recognition: User is prompted with an error message indicating that the user name was not recognized and asked to repeat the user name.
- ii) Invalid Input: User is prompted with an error message indicating that the recognized user name is not valid and asked to repeat the user name.

-Defined in: UserName.aspx

-Next page loaded: In case of valid input → Password.aspx

In case of invalid input → NoRecoError.aspx or InvalidInput.aspx
→UserName.aspx

3. Password:

The user is asked to enter his/her password. The associated grammar covers from 1 to 16 alphanumeric characters in any order. User should utter his/her password digit-by-digit and/or letter-by-letter.

Possible errors:

- i) No Recognition: User is prompted with an error message indicating that the password was not recognized and asked to repeat the password.
- ii) Invalid Input: User is prompted with an error message indicating that the recognized password is not valid and asked to repeat the password.

-Defined in: Password.aspx

-Next page loaded: In case of valid input → IntelModeSelect.aspx

In case of invalid input → NoRecoError.aspx or InvalidInput.aspx
→Password.aspx

4. Intelligent Mode Selection:

The user is asked whether s/he wants to use the intelligent mode. Intelligent mode permits the user to phrase any sentence to retrieve the desired information.

Possible errors:

- i) No Recognition: User is prompted with an error message indicating that his/her answer was not recognized and asked to repeat the answer.

-Defined in: IntelModeSelect.aspx

-Next page loaded: In case of valid input → Intelligent.aspx (if Intelligent Mode entered) or Course.aspx (if Intelligent Mode not entered)

In case of invalid input → NoRecoError.aspx → IntelModeSelect.aspx

5a. Course ID:

The user is asked to enter a course ID. A course ID is the 3-digit number used to identify the courses offered. Saying "EE" before a course ID is optional. The corresponding grammar covers all 3-digit numbers.

Possible errors:

- i) No Recognition: User is prompted with an error message indicating that the course ID was not recognized and asked to repeat the course ID.
- ii) Invalid Input: User is prompted with an error message indicating that the recognized course ID is not valid or the user is not registered to that course and asked to enter another course ID.

-Defined in: Course.aspx

-Next page loaded: In case of valid input → Menu.aspx

In case of invalid input → NoRecoError.aspx or InvalidInput.aspx
→Course.aspx

5b. Intelligent Mode:

The user is asked to phrase a sentence in which s/he is expected to say a course code or a course name, and one of the three menu items available in the voice-only version of CIMS, which are 'Grades', 'Assignments' and 'Announcements'. The order of these entries is not important. The associated grammar covers all possible course names, course codes and the three menu options provided.

Possible errors:

- i) No Recognition: User is prompted with an error message indicating that the recognition failed and asked to repeat the same information.

ii) Invalid Input: User is prompted with an error message indicating that the recognized course ID is not valid and asked to repeat the same information.

-Defined in: Intelligent.aspx

-Next page loaded: In case of valid input → CourseChanged.aspx → MenuChanged.aspx → IntelContinue.aspx

In case of invalid input → NoRecoError.aspx → Course.aspx

Note on XChanged.aspx pages: XChanged.aspx pages contain code sections used by both Intelligent and Normal modes. Corresponding X.aspx pages call the corresponding XChanged.aspx pages.

6a. Menu Item Selection:

The user asked to select one of the three possible menu items, which are "Grades", "Announcements" and "Assignments".

Possible errors:

i) No Recognition: User is prompted with an error message indicating that the recognition failed and asked to repeat the same information.

-Defined in: Menu.aspx

-Next page loaded: In case of valid input → Recent.aspx or NoRecent.aspx → ReadOld.aspx and/or Output.aspx → Another.aspx

In case of invalid input → NoRecoError.aspx or InvalidInput.aspx → Menu.aspx

7a. Returning to Menu

The user is asked whether s/he wants to return to menu.

Possible errors:

i) No Recognition: User is prompted with an error message indicating that the recognition failed and asked to repeat the answer.

-Defined in: Another.aspx

-Next page loaded: In case of valid input → Menu.aspx or Choice.aspx

In case of invalid input → NoRecoError.aspx→ Another.aspx

8a. Returning to Course Selection

The user is asked whether s/he wants to continue with another course.

Possible errors:

i) No Recognition: User is prompted with an error message indicating that the recognition failed and asked to repeat the answer.

-Defined in: Choice.aspx

-Next page loaded: In case of valid input → Course.aspx or GoodBye.aspx

In case of invalid input → NoRecoError.aspx→ Choice.aspx

8b. Another Query:

The user is asked whether s/he wants to make another Intelligent Mode query.

Possible errors:

i) No Recognition: User is prompted with an error message indicating that the recognition failed and asked to repeat the answer.

-Defined in: IntelContinue.aspx

-Next page loaded: In case of valid input → Intelligent.aspx or GoodBye.aspx

In case of invalid input → NoRecoError.aspx→ IntelContinue.aspx

9. Ending the Session

A prompt indicating that the session is ending is played.

-Defined in: GoodBye.aspx

Timing Issues During Dialog Flow:

The following timeouts are used for proper timing: Initial Timeout, Babble Timeout, End Silence Timeout, and Max Timeout.

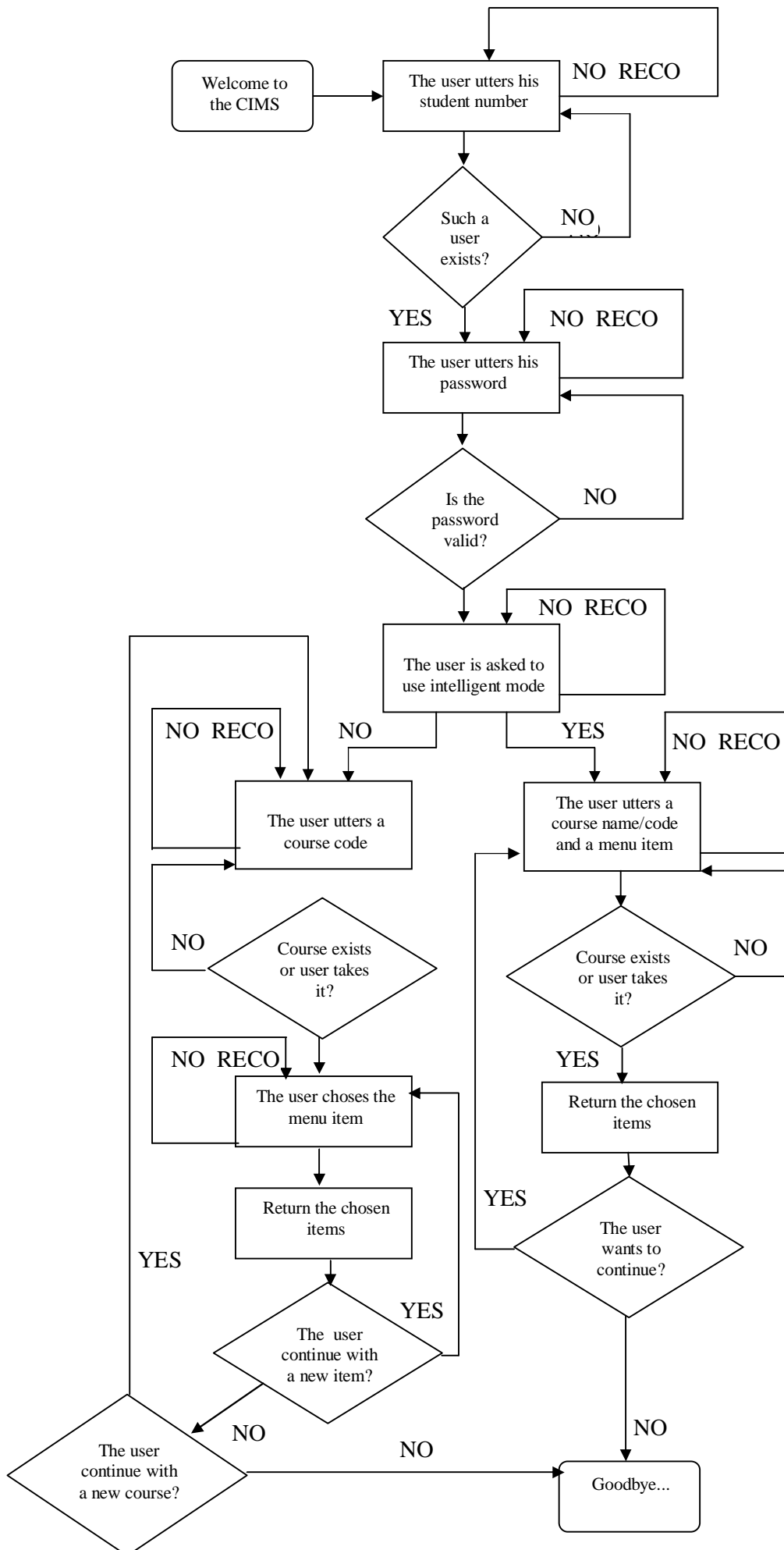
Initial Timeout limits the time passed between the start of recognition and detection of speech. Specified as 4000 msec for QA (Question-Answer) controls.

Babble Timeout limits the time passed between the detection of speech and returning the recognition result. Specified as at least 2500 msec. It is much longer where more than one word is expected.

End Silence Timeout limits the time passed between end of speech and returning the recognition result. Specified as 700 msec for QA controls except in Intelligent.aspx (1000 msec).

Max Timeout limits the time passed between the start of recognition and returning the recognition result. Specified as the sum of Initial and Babble Timeouts. It would have an effect if it were set a smaller value than the sum of those timeouts.

There are several prompts that only convey information to the user but don't ask for any input from the user. In order the program to execute without halting, 1 msec long Max Timeouts are set. So the recognition ends almost immediately and program flow continues uninterrupted.



Dialog Flow

4. Grammars

Speech recognition is the most essential technology that makes web applications speech-enabled. Without it, DTMF would be the only way of submitting information to voice-only applications. This would require a very structured architecture with many menus and submenus. Speech recognition, if wisely used, may reduce the complexity of such applications. Grammars are the most powerful tools to make use of a given speech recognition engine. They limit the possible words, numbers, phrases etc that a speech recognizer expects to "hear". This increases the likelihood that recognition occurs correctly.

This project uses several grammars. Most of them are designed for specific recognition tasks. Others use the Grammar Library provided by the MS Speech Application SDK. Grammars to recognize digits, "Yes", "No", "Cancel" phrases are among these grammars. A grammar can include references to other grammars. So a modular design is possible and it reduces development time.

Each grammar returns either the recognized text or another text related to the recognized text. Scripts are inserted into the grammars, which are, in fact, XML codes with nodes and subnodes using tags proposed by the World Wide Web Consortium's Speech Recognition Grammar Specification (SRGS) recommended standard. Results are returned within Semantic Markup Language (SML) tags, which is another XML base markup language developed as a standard communication method between a speech recognizer and the program using that recognizer.

This section explains the grammars used in the project. Detailed information is provided about the coverage of grammars and the information returned to the server using SML. A common feature of all the grammars used directly in the project is that the user can end the session immediately by saying "Cancel". QA (Question-Answer) controls in the project directly use the "YesNoCancel" grammar from the grammar library that covers one of the three possible answers in the grammar name. A single grammar file (CIMS.grxml) contains all of the grammars, but the Grammar Editor tool enables editing or even deleting of each grammar.

1. UserName:

This grammar uses two grammars from the grammar library of the Speech Application SDK. These grammars cover 7 or 10 digit numbers and return the recognized digit arrays as 7 or 10 digit numbers, not as a string array. These grammars support groupings of uttered numbers but users are asked to enter their usernames digit by digit without any groupings. The recognized number is returned to server within a subnode labeled "Username".

2. Password:

This grammar uses several grammars from the grammar library for alphanumeric characters. Since passwords up to 16 characters are allowed, the grammars referenced can be traversed up to 16 times.

Each grammar in the list adds the recognized digit or letter to a variable that is initialized as an empty string at the beginning of the grammar. The recognized password is returned within a subnode labeled as "Password".

3. Course:

This is a simple grammar that covers all possible 3-digit numbers. "e. e." at the beginning of the course code is optional. In either case, the course code without the possible "EE" is returned within a subnode labeled as "Course".

4. Menu:

This grammar covers the three available menu items: "Grades", "Announcements", and "Assignments". The answer is returned within a subnode labeled as "Menu".

5. Intelligent:

This grammar allows the user to utter any sentence or phrase including a course name or course code and one of the three menu items. The SKIP and WILDCARD elements in the Grammar Editor tool enable such a wide coverage. However, the likelihood of returning a wrong result is much greater than the other grammars used. An empty string variable is added to the recognized texts as one of the branches is traversed. There are two grammars used by this grammar rule: CourseCodes, CourseNames and Exams. (Exams extends this grammar to include words such as

"Quiz", "Midterm", "Final" instead of "Grades", but in the rest of this report only the three possible menu items will be regarded as the allowed entries.)

6. CourseCodes:

This grammar is identical to the Course grammar except it does not cover "Cancel". This is covered by the grammar Intelligent.

7. CourseNames:

This grammar covers the names of all of the courses offered by the EE Department and many possible variants. These variants are either different names as entered into CIMS or abbreviated forms as used by the students in case of long course names. The result consists of one of the menu items and a course code separated by a space. At the server, the returned string is parsed and processed.

8. Exams:

This grammar is designed to substitute the "Grades" menu item in the Intelligent grammar rule and contains most of the exam types that can be encountered in the grades section of CIMS.

5. Prompts

Prompts are the only way that the system can convey information to users. Some inform the user about the current status of the session; others ask for or convey some information. In the project, all of the static prompts are prerecorded, i.e. they are not played using the text-to-speech engine. Dynamic prompts that are formed from the information requested by the user are played using the text-to-speech engine at the client. Not all the static prompts are recorded during project development. Speech Application SDK's Prompts Database Management tool enables to mark parts of recorded prompts, which can be used then during run-time to compose other prompts. For example; the prompt "Please enter your user name" is marked as [Please enter your] [user name]. Here, the "Please enter your" part is combined with "password", recorded separately, to form a new prompt "Please enter your password" that is used during dialog flow but has not been recorded into the database. The advantage of this method is that not all of the possible prompts

have to be explicitly recorded, which may increase significantly as the application grows and becomes more complicated.

All of the prompts are played using QA controls, one of the Speech Application SDK classes extending ASP.NET. These controls are associated with variables updated by the client, prompt database, grammars used for recognition tasks, events raised either on the client or server side. Also several timeout are defined in the QA controls.

6. Database Operations

CIMS database is established on MS SQL SERVER 2000 Platform, which is a RDBMS (Relational Database Management System). Each relation has many attributes that are converted into tables and fields in the management system, respectively. Since EE Department has many entities (such as students, courses, announcements...), these entities and relationships (students take courses, grades are given for their exams....) are transformed into relations. As a result CIMS database is woven on a huge number of tables containing data about many different structures and relationships.

During the operations, the web site connects to the CIMS Database and queries some of these tables. The connections are done step-by-step depending on the function of the current page or choice of the user. To optimize the access number and operation time, each necessary table is queried and the needed fields are obtained only once.

When the user connects the site, at first, he or she is asked to enter the student number as the user name. Every student is an entity made of properties such as password, email, name, and surname... This entity was transformed as the USERS table. The student number is unique (the primary key) so that it is used as a predicate to query the table. The fields PASSWORD (holds the password of the user), LAST_ACCESS (the date the user has accessed lastly), and ID must be drawn from USERS to be used at the later stages. PASSWORD, clear from its name, used to check whether the user is the one he or she claims to be. LAST_ACCESS hold the last date the user has used the system. It is crucial on informing the user about most recent grades,

announcements and assignments. ID field is the unique identification number of the each user on the CIMS database and used as either primary key or foreign key on other tables. The operation is performed by the following SQL command:

```
SELECT PASSWORD, ID, LAST_ACCESS FROM USERS WHERE  
USERNAME ='userno'
```

where the userno is the variable that stores the student number of the user on the line.

The second database operation is checking firstly the user uttered a valid course and then in case it is a valid course, he or she takes the course (a student registered to the course). Two steps accomplish this. The step one is to search whether there exists such a course, or it is given in this semester (the course may be an existing course but it is given in the other semester). Querying the COURSES table in the CIMS does this search. The SQL query for this action is:

```
SELECT ID, Code FROM COURSES WHERE Enabled='1'
```

As a result of this query, all the courses, which are given in the current semester, (Enabled =1) are obtained. By a means of loop, the course code given by the user is sought through all the valid courses. If one of the courses matches the user's code, the ID value of the course is taken for using in other operations. The ID is unique for each course, and it is used as identification number of the course in other tables. The second step is to search if the user is really one of the course students. The table Students holds the student-course pair. The student taking a certain course must have a tuple in this table. This operation is accomplished by the following SQL query:

```
SELECT ID FROM Students WHERE CourseID= 'Session["COURSEID"]' AND  
UserID=(string)Session["st_id"]'
```

CourseID attribute in the Students relation is the same as the ID attribute in the COURSES relation, which has been stored in the Session["COURSEID"] variable previously. UserID is the

ID value drawn from the USERS table, which has been saved in the Session["st_id"]. The purpose of this query is to check if there exists such a pair.

The next operations are performed depending on the user's choice. The user can want to learn his/her grades, announcements and assignments in the course. In case the user wants to get aware of the assignments in the course, the table Assignments must be queried. The command to achieve this is:

```
SELECT UploadDT, DueDate FROM Assignments WHERE  
CourseID='Session["COURSEID"]' and Enabled='1'
```

UploadDT holds the date the assignment is uploaded. DueDate is the deadline of the assignment. The number of the assignments is returned to the user. The ones that are out of date are not returned. If the assignment is not enabled then it means that the lecturer does not want the students to reach the assignment anymore.

The user might choose to learn the announcements in the course. The announcements are drawn from the Announcements table. They are read to the user. The SQL query for getting them is:

```
SELECT Title, Description, StartDate FROM Announcements WHERE  
CourseID='Session["COURSEID"]' and Enabled='1'
```

StartDate is used in determining the recent announcements since the last access of the user. Title field holds the title of the announcement. Description field is announcement, itself.

The user inevitably, mostly, would want to learn his/her grades in the course. This is the hardest task to do. Because there can be many grades announced (homeworks, laboratory reports, exams...) for each students. The grades can be obtained after two table accesses. The first table Exams is created to distribute a unique ID for each grade type (homework, exam...) for each course. This ID is a key element in the Grades table together with the ID of the user. The queries are:

```
SELECT NAME, ID FROM Exams WHERE CourseID = 'Session["COURSEID"]'  
    SELECT Grade, dateAdded FROM Grades WHERE StudentID = 'Session["st_id"]'  
    AND ExamID = 'Session[strGrades]'
```

In the first query, the grades ID are returned and saved in the Session[strGrades] variable to be benefited while questioning the Grades table . StudentID is the ID of the user, which was drawn at the beginning of the session. Grade field contains the grade of the student. dateAdded field used to return the user the most recent grade given.

The last operation is to update the last access date of the user while ending the session. This operation is necessary perform operations in the next session of the user. We connect to the database, and set the last access date as the current time. The command is:

```
UPDATE USERS SET LAST_ACCESS = GETDATE() WHERE ID = 'Session["st_id"]'
```

Regular Expressions

Regular expressions are a powerful, flexible and efficient tool for text processing and string manipulations. They provide very considerable features such as specific pattern matching, string splitting and substring replacement. By means of the Regex class, in a very fast manner a great amount of text can be parsed to a certain character pattern, manipulated and restructured.

During the evolution of the project, the main difficulty was that the lecturers are free to name the entity in any way they want. For instance; HW 3, hw #3 or homework 3 can be used to name the same entry. Usage of prerecorded prompts at the client side requires that the entries have a predefined form. However, CIMS does not restrict the lecturers in labeling the entries.

Regex class enabled us to overcome this problem. Different strings, having a certain pattern specified by the programmer, can be manipulated, or replaced to have the same output. Instead

of direct conversion of the name of the entry by the text-to-speech engine, it was decided to use the Regex class to convert the label of the entity to a predefined noun.

Firstly, we examined all the Grades name fields of the Exams table in the CIMS database. It was observed that homework certainly includes the letters h and w, and an at least one digit long number. There can be any number of letters spaces or escape characters before, between and after these letters. Also there can be any number of non-digit characters before and after the number. If there are any letters before the letter h, these correspond to meaningful words such as Spice, Matlab, etc. For example: Hw#1, homework 1, HW 1, Spice Homework 1, etc. The words before the letter h should be kept.

In order for the string manipulation to be done correctly, a number of patterns covering all of the grade label variations had to be designed. The following Regex instantiation matches all the labels containing at least 0 characters other than h or H followed by h or H, at least 0 characters other than w or W, then w or W followed by any number characters.

```
Regex rhw = new Regex(@"^[^hH]*(h|H)[^wW]*(w|W).*");
```

All the entry labels that are matched to this pattern are manipulated according to the following regular expression.

```
Regex.Replace(name, @"(?<preamble>\w*\s*)(H|h)\w*(W|w)([^\d]*  
    (?<no>\d*)).*", "{$preamble}"+"homework"+"${no}");
```

"(?<preamble>\w*\s*)" : stores all the characters and spaces up to h or H.

(H|h)\w*(W|w)([^\d])* : skips h or H, all the letters up to and including w or W and all the nonnumeric characters and white spaces.

(?<no>\d*)).*" : stores all consecutive numbers and skips nonnumeric characters and white spaces.

"\${preamble}"+"homework "+"\${no}" : a new string conforming to the predefined form is created.

The same mechanism is valid for quizzes, labs, midterms, finals, assignments and experiments.

7. Conclusion

Although speech processing is not a recent area of research from an academic point of view, its applications are still not widely used, but more and more systems employing speech recognition, text-to-speech or both technologies are commercially available in the software market. But not only telecommunications or IT sectors, which seem to be the most natural areas of applications, in general, every aspect of daily life, every business or industrial activity that requires human-machine interaction may be considered as a potential application area for speech processing technologies. Within this conceptual framework, this project gave us the opportunity to become familiar with a powerful speech-enabled web applications development platform. During the development of this project using the Beta 4 version of the SDK, Microsoft announced the final release (Ver. 1.0) of the Speech Application SDK. Only this example may by itself suffice to support the remarks made about the popularity of this research and development area.

There are many fringe benefits we enjoyed that deserve to be mentioned here. .NET Framework, Visual C#.NET, ASP.NET, ADO.NET and Visual Studio .NET 2003 IDE are among the emerging technologies and accompanying development environments which are proposed as new standards for desktop and web application development by Microsoft. Whether these technologies will be successful or we will be professionally using them is an unknown, but the experience we gained and the knowledge we accumulated will hopefully more than compensate for the time we spent in front of computer monitors.

8. References

2310B Developing Microsoft ASP.NET Web Applications Using Visual Studio .NET, Microsoft Co., 2002

T. Archer and A. Whitechapel, *Inside C#*, Washington: Microsoft Press, 2002

J. Liberty, *Programming ASP.NET*, Sebastopol, CA: O'Reilly, 2003

G. Malcolm, *Programming Microsoft SQL Server 2000 with XML*, Washington: Microsoft Press, 2002

Microsoft Developer Network Library, msdn.microsoft.com/library

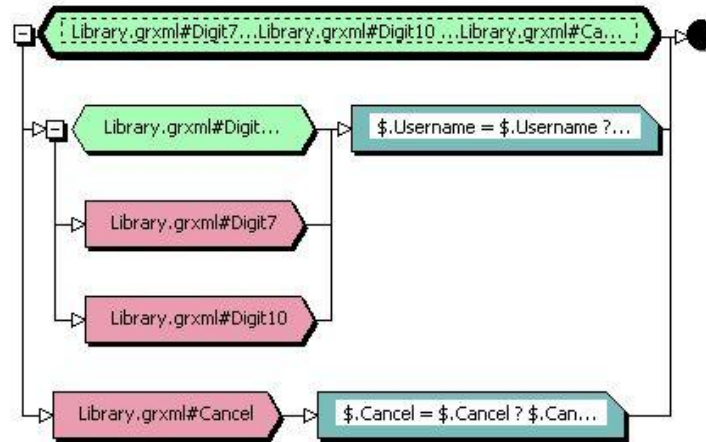
J. Prorise, *Programming Microsoft .NET*, Washington: Microsoft Press, 2002

J. Sharp and J. Jagger, *Microsoft Visual C#.NET Step by Step*, Washington: Microsoft Press, 2002

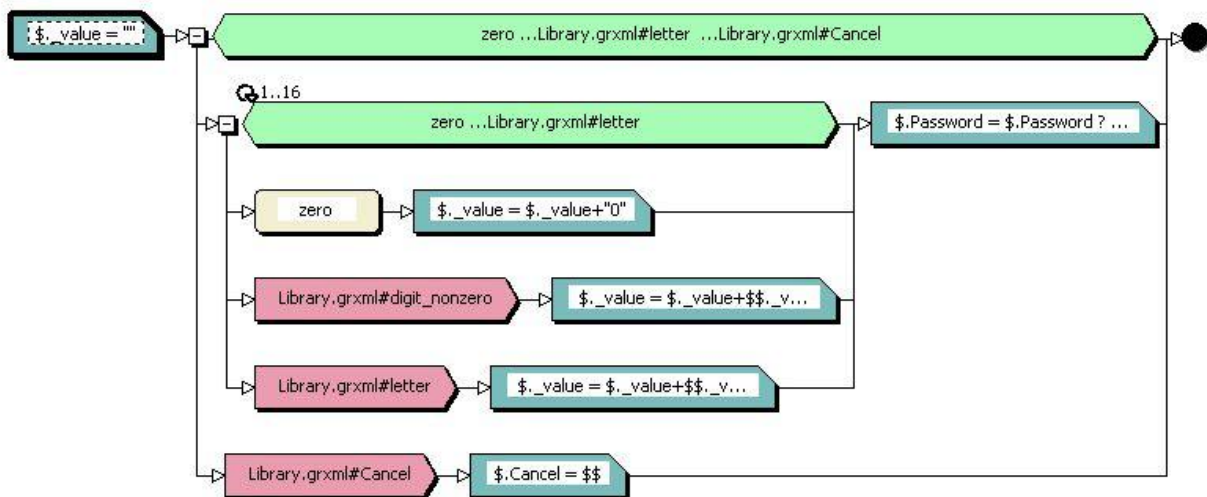
J. Webb with Microsoft Corporation, *Developing Web Applications with Microsoft Visual Basic.NET and Visual C#.NET*, Washington: Microsoft Press, 2003

9. Appendices

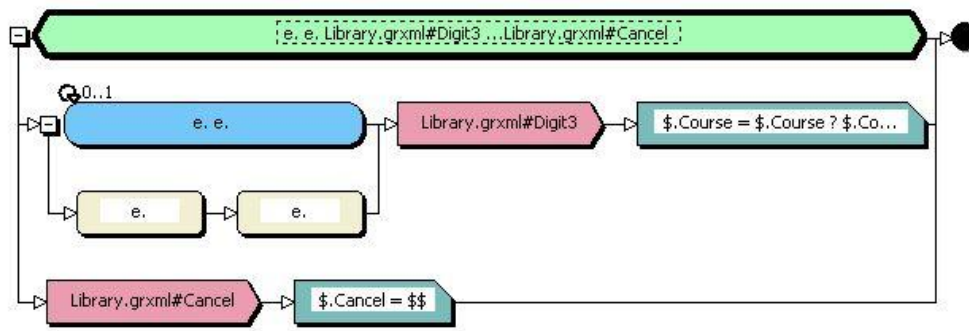
Grammars



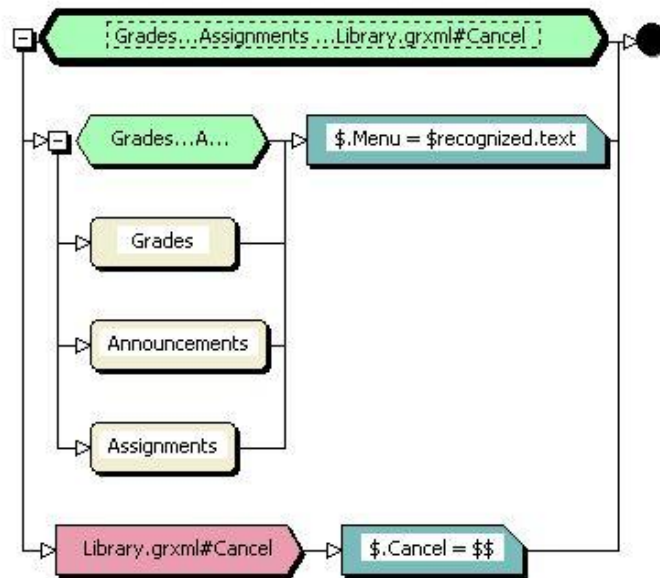
UserName



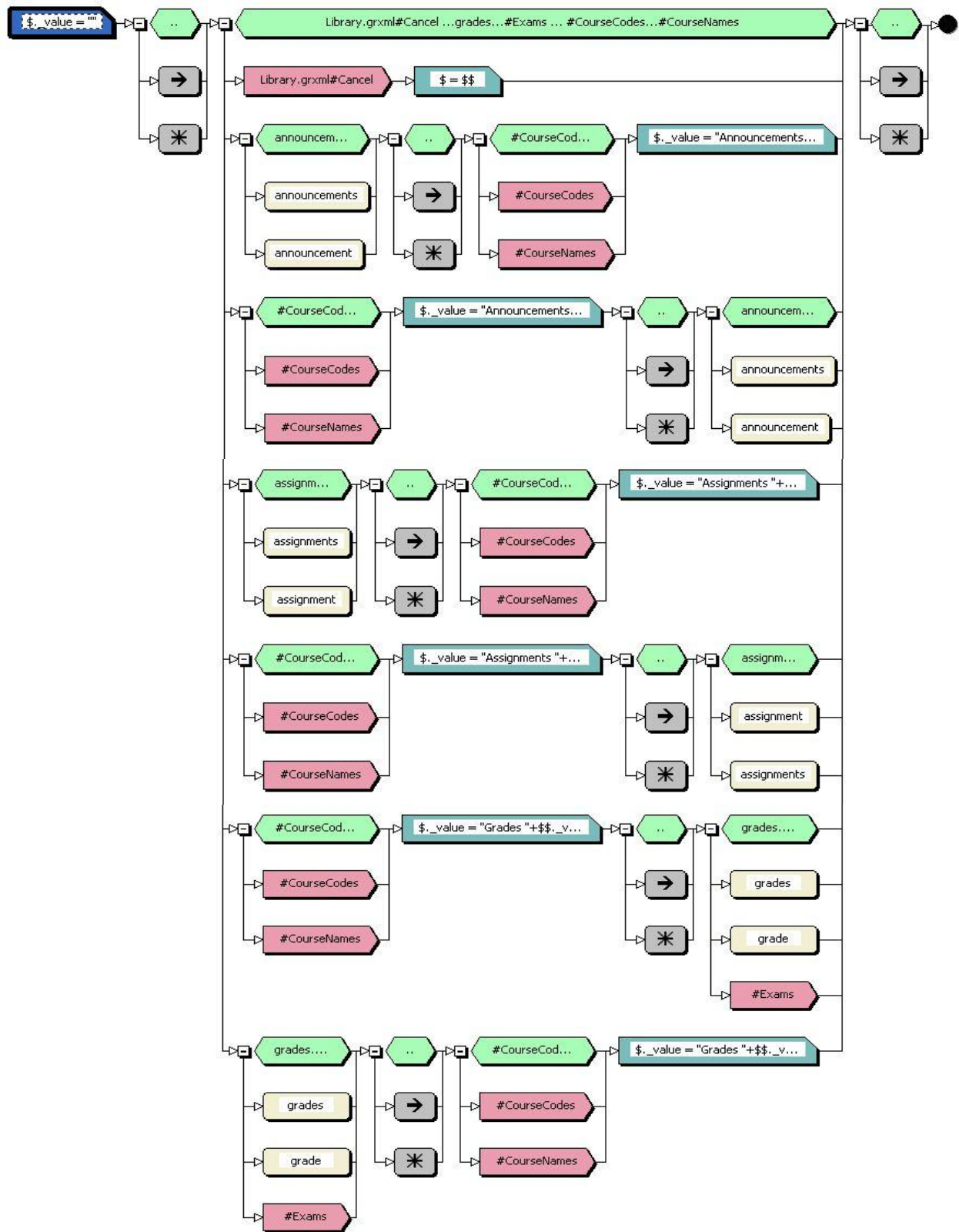
Password



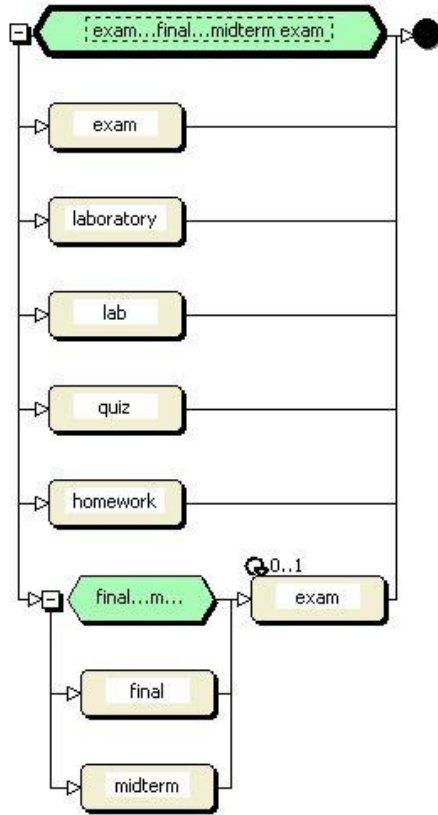
Course



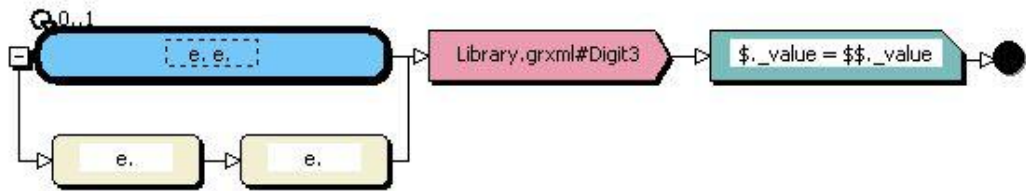
Menu



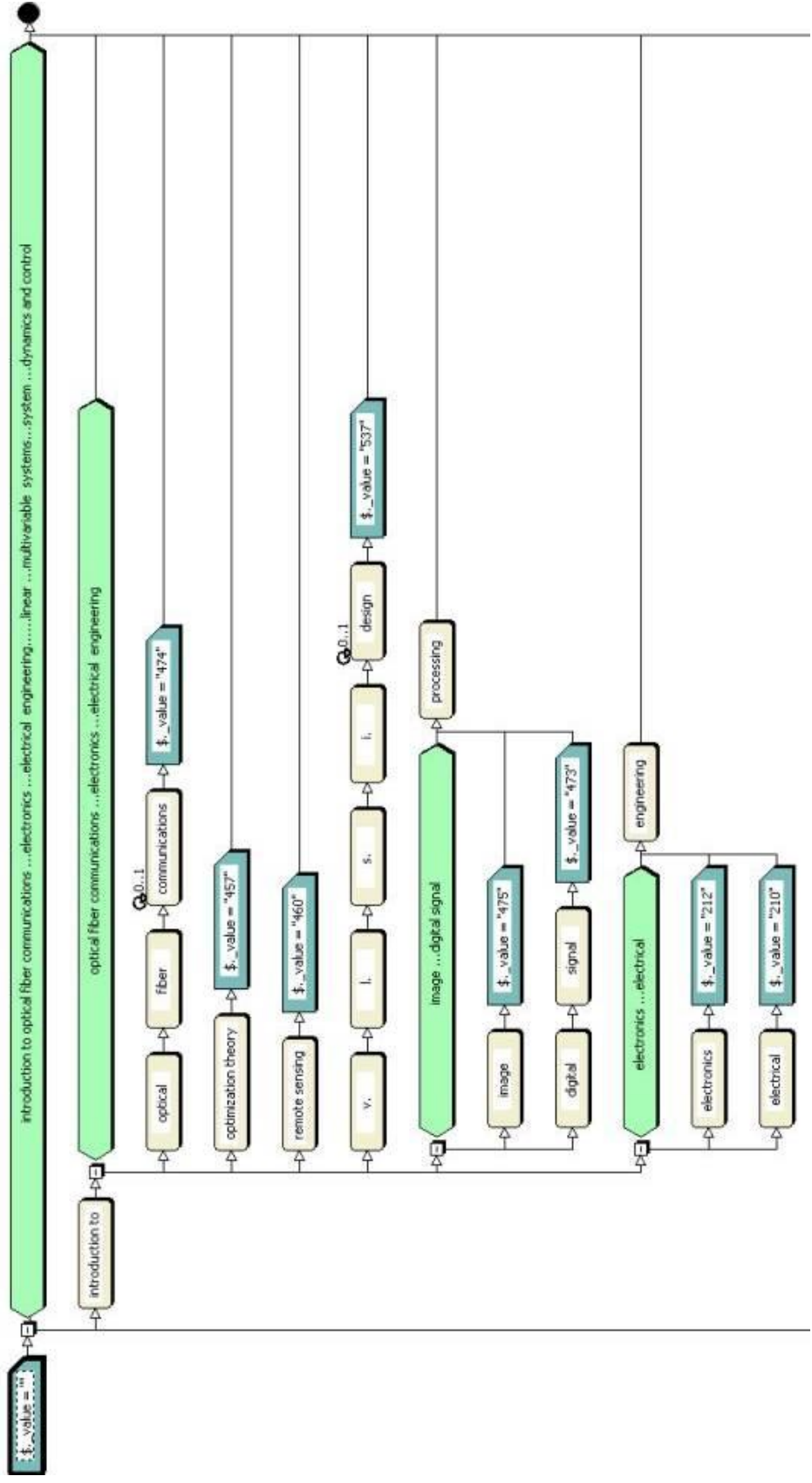
Intelligent



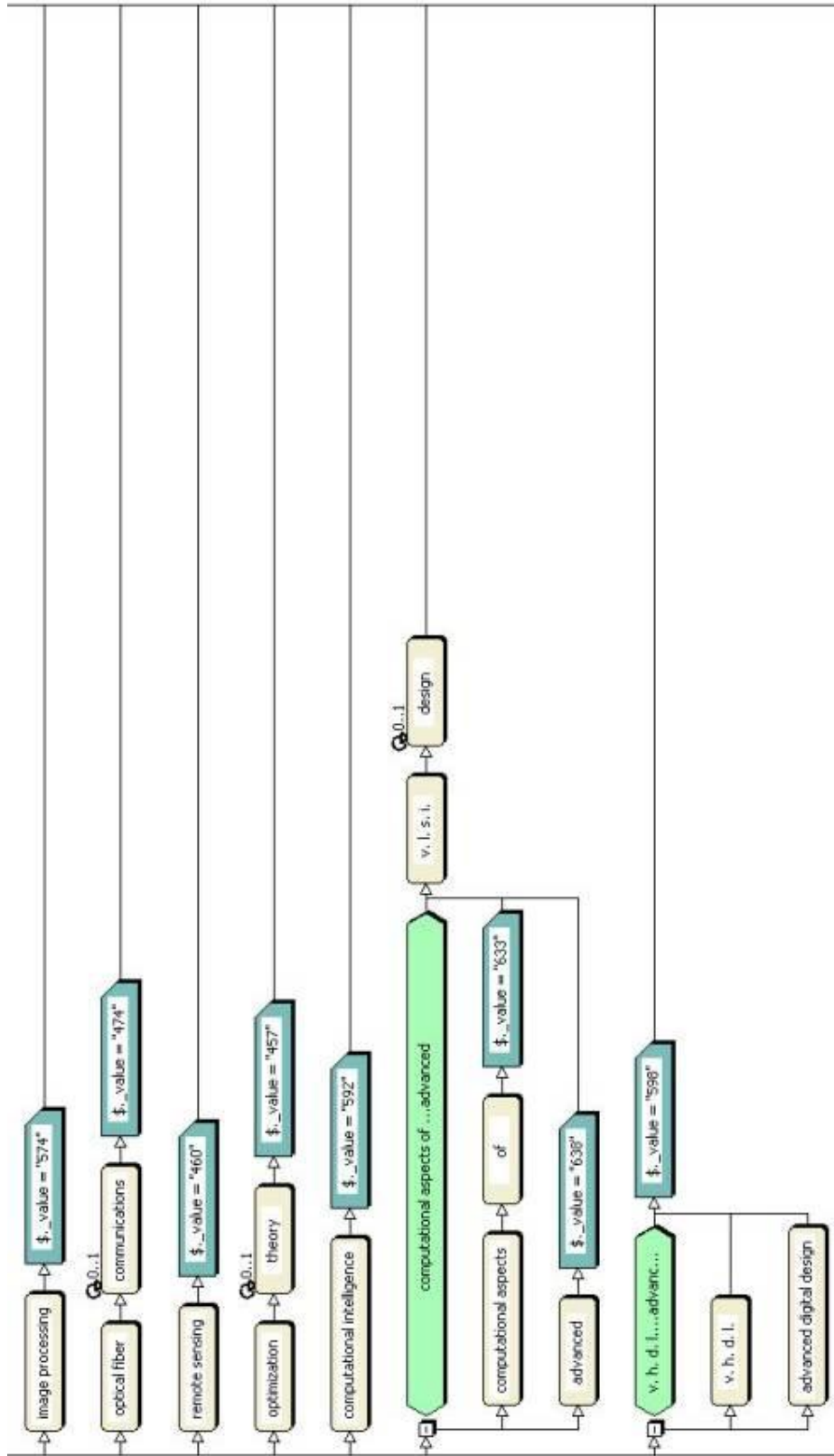
Exams

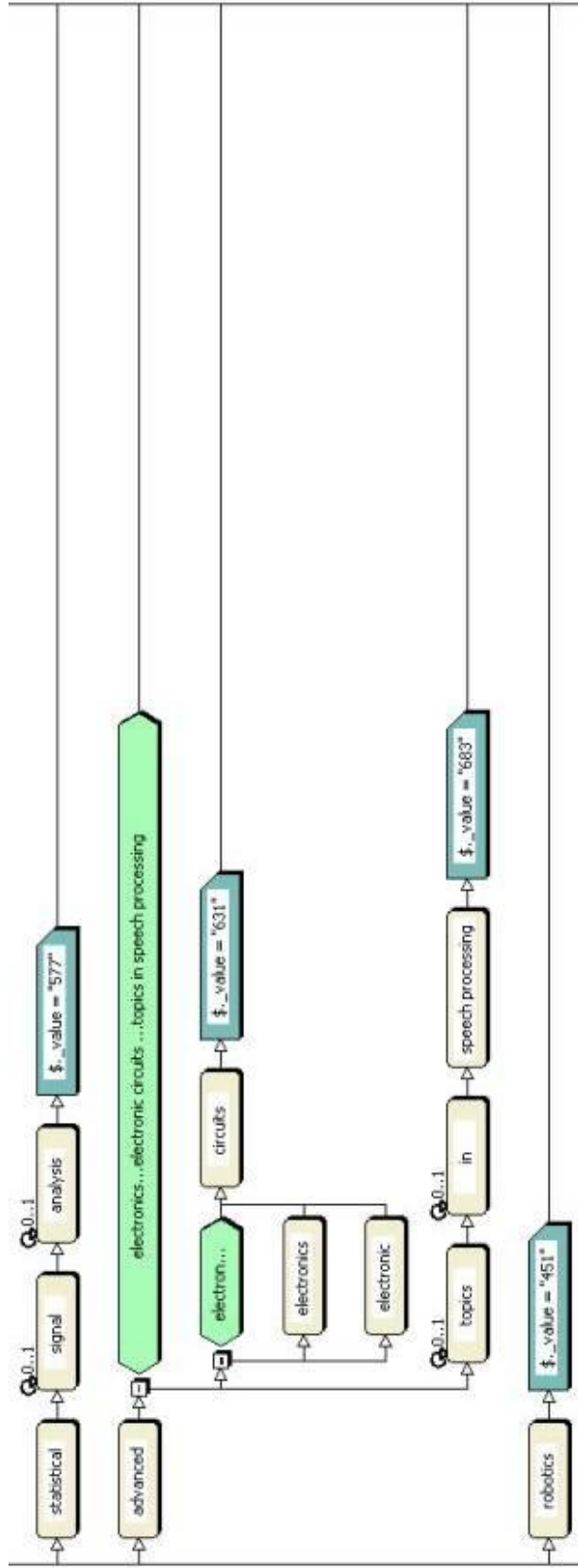


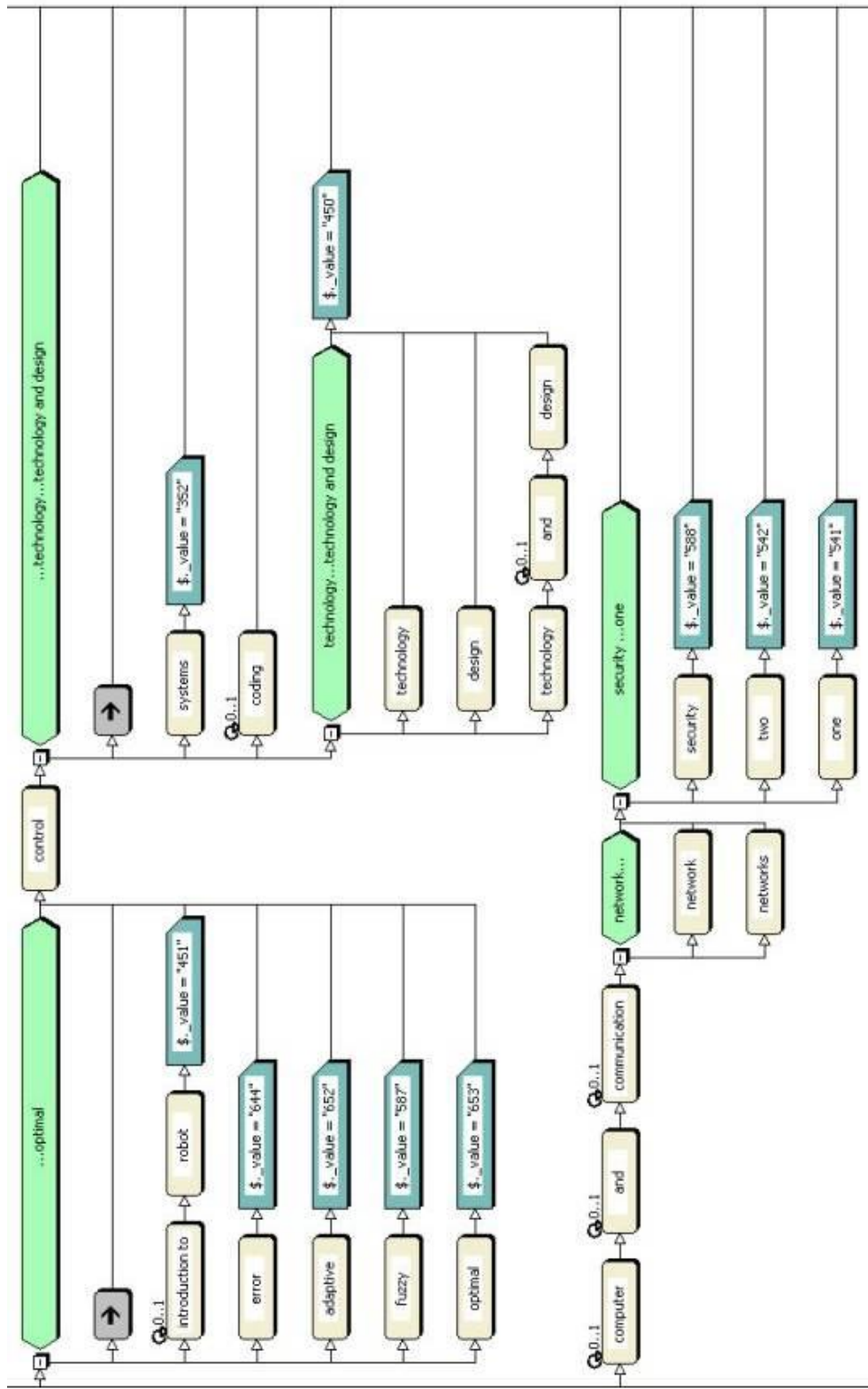
CourseCodes

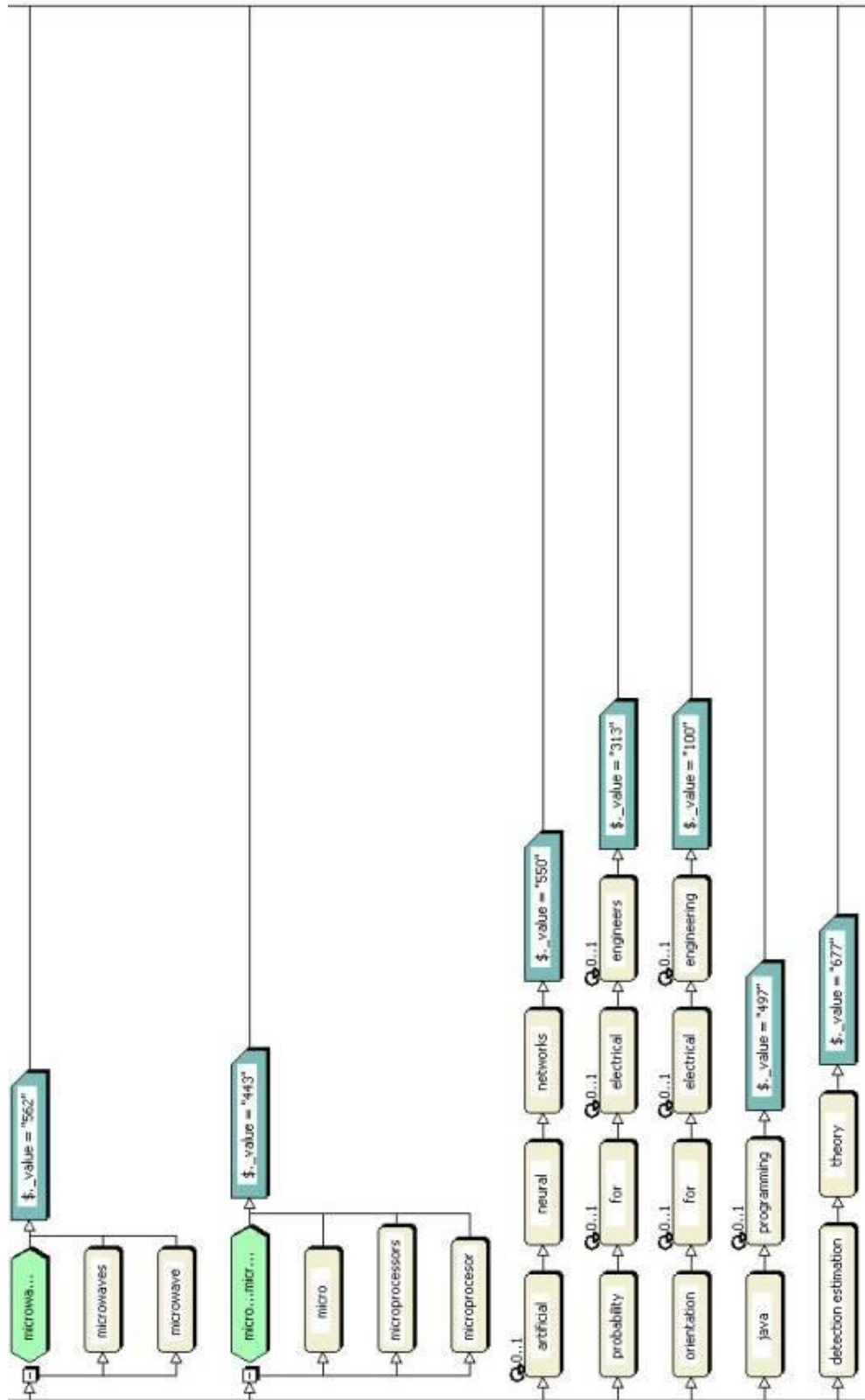


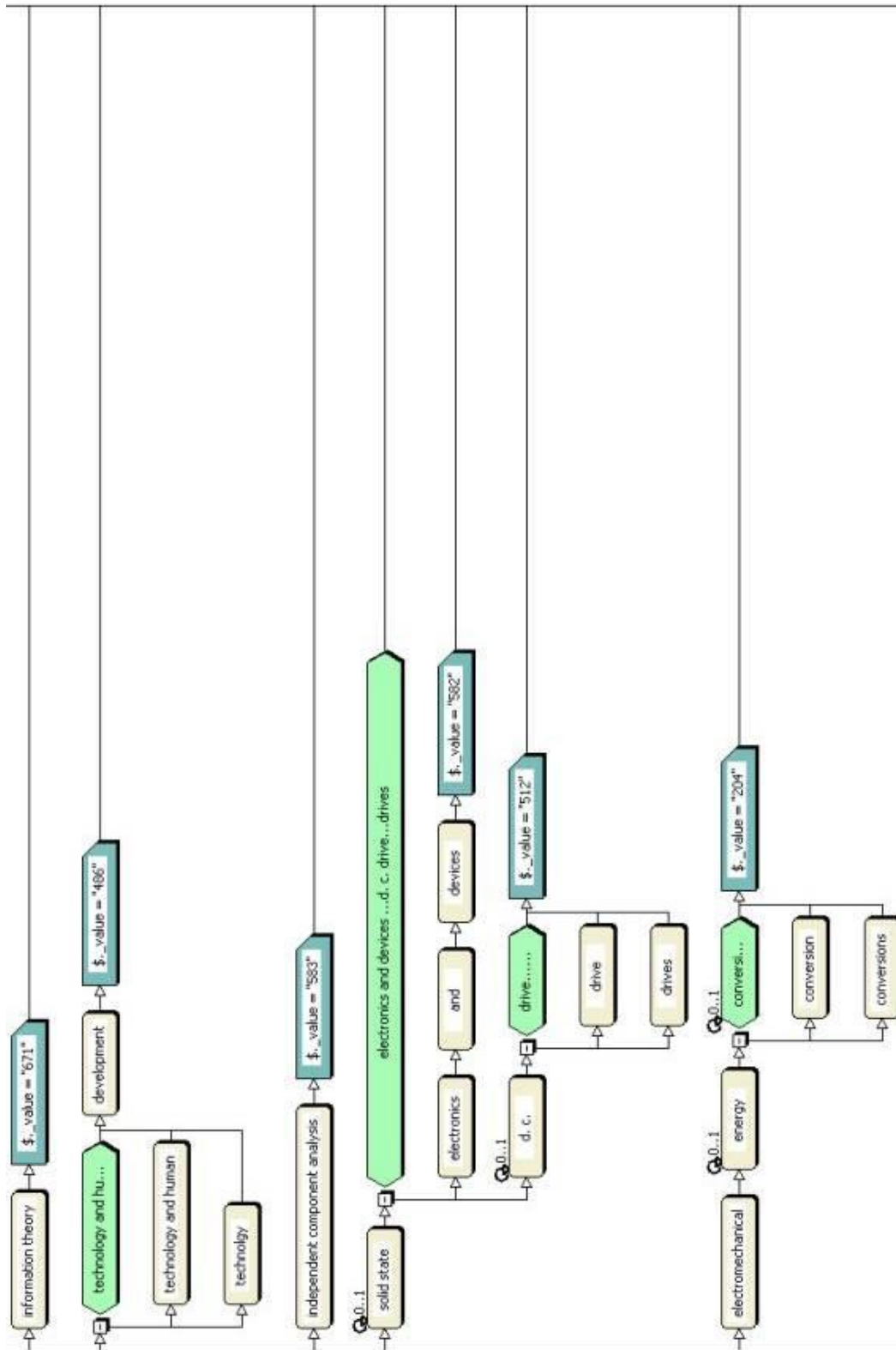
CourseNames 1/11

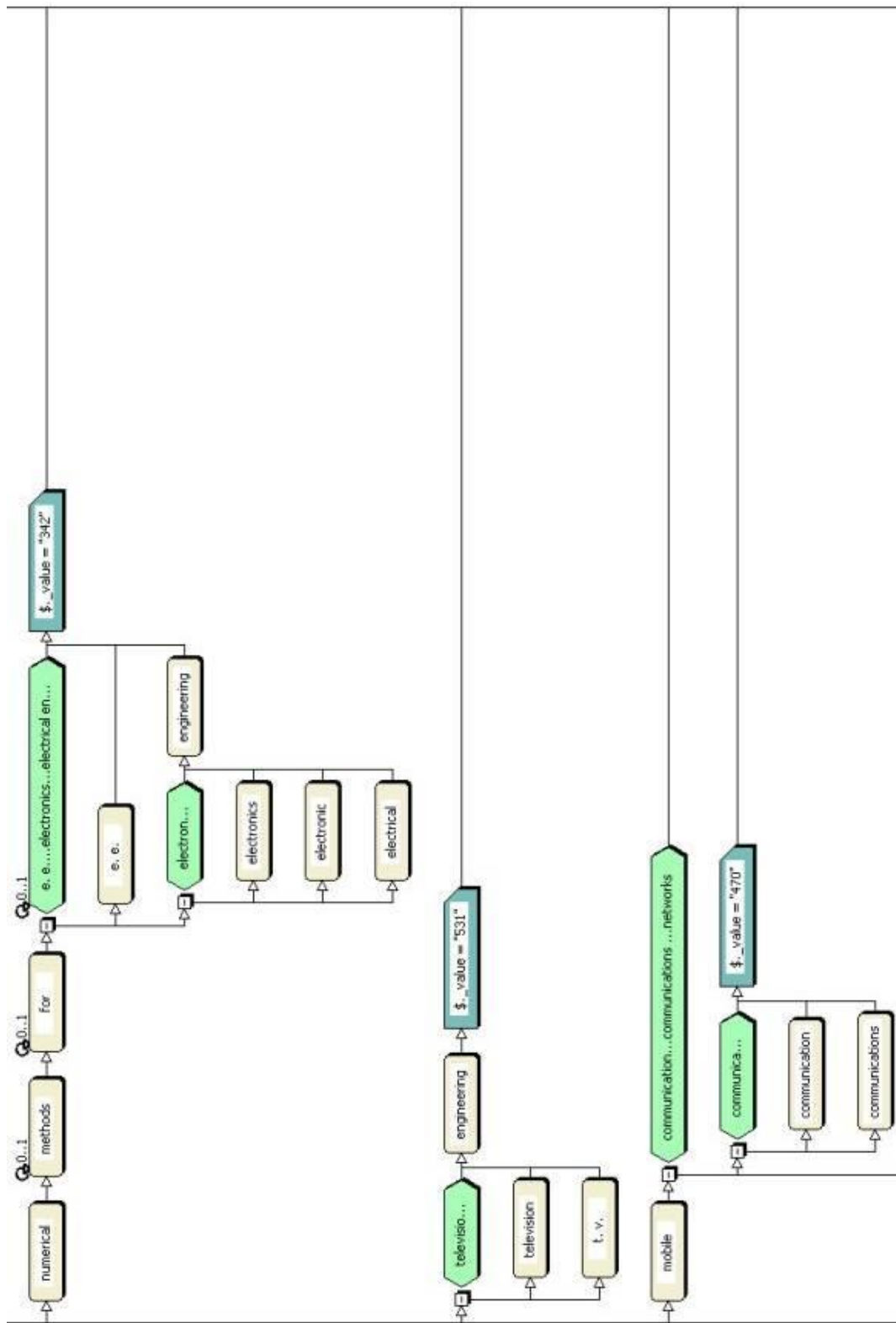


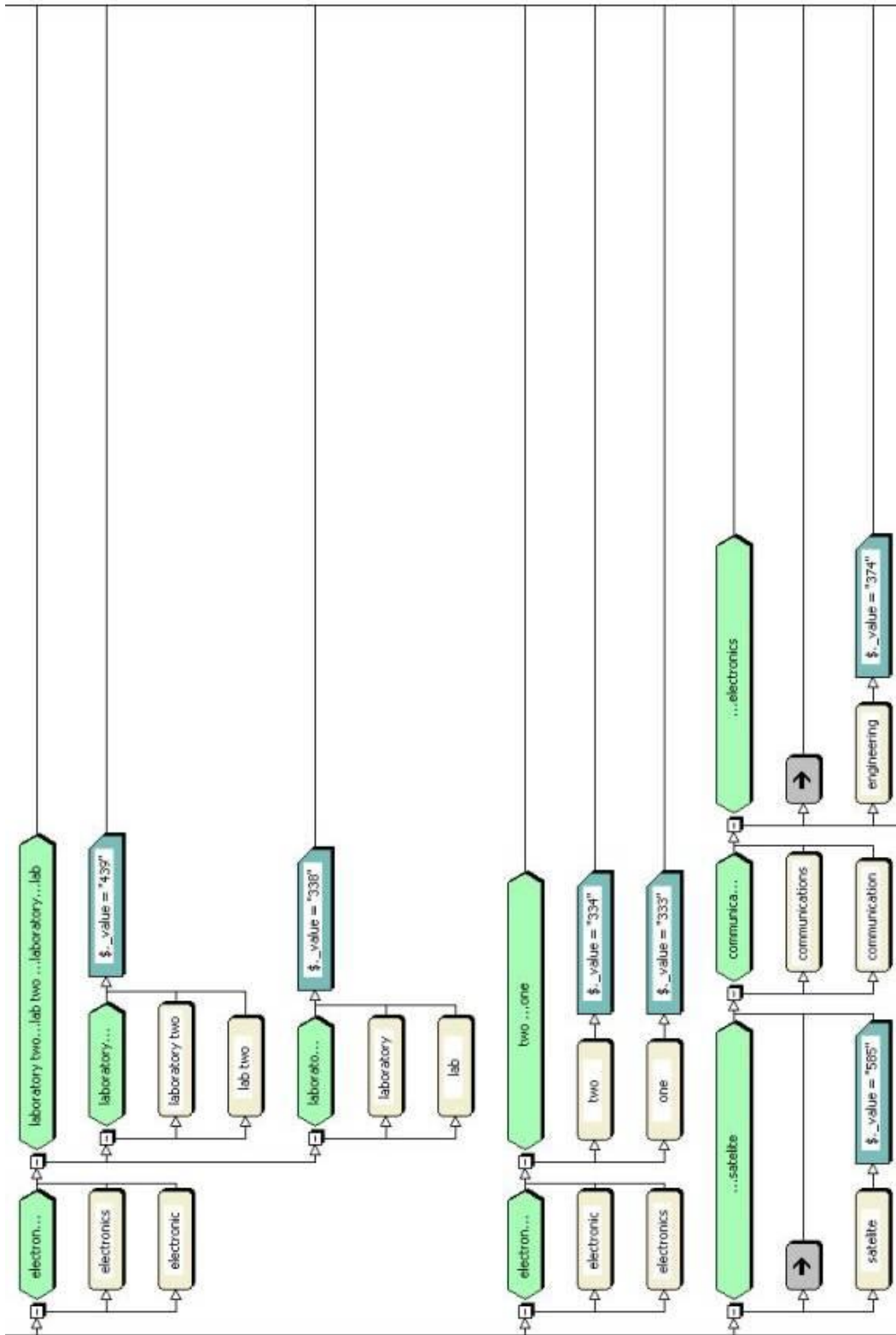












CourseNames 10/11


```

<grammar xmlns:sapi="http://schemas.microsoft.com/Speech/2002/06/SRGSExtensions" xml:lang="en-US"
tag-format="semantics-ms/1.0" version="1.0" mode="voice"
xmlns="http://www.w3.org/2001/06/grammar">
  <rule id="Menu" scope="public">
    <one-of>
      <item>
        <?MS_Grammar_Editor GroupWrap?>
        <one-of>
          <item>Grades</item>
          <item>Announcements</item>
          <item>Assignments</item>
        </one-of>
        <tag>$.Menu = $recognized.text</tag>
      </item>
      <item>
        <?MS_Grammar_Editor GroupWrap?>
        <ruleref uri="Library.grxml#Cancel" type="application/srgs+xml"/>
        <tag>$.Cancel = $$</tag>
      </item>
    </one-of>
  </rule>
  <rule id="UserName" scope="public">
    <one-of>
      <item>
        <?MS_Grammar_Editor GroupWrap?>
        <one-of>
          <item>
            <?MS_Grammar_Editor GroupWrap?>
            <ruleref uri="Library.grxml#Digit7"
type="application/srgs+xml" />
          </item>
          <item>
            <?MS_Grammar_Editor GroupWrap?>
            <ruleref uri="Library.grxml#Digit10"
type="application/srgs+xml" />
          </item>
        </one-of>
        <tag>$.Username = $.Username ? $.Username : {}; $.Username._value =
$$._value</tag>
      </item>
      <item>
        <?MS_Grammar_Editor GroupWrap?>
        <ruleref uri="Library.grxml#Cancel" type="application/srgs+xml"/>
        <tag>$.Cancel = $.Cancel ? $.Cancel : {}; $.Cancel._value =
$$._value</tag>
      </item>
    </one-of>
  </rule>
  <rule id="Course" scope="public">
    <one-of>
      <item>
        <?MS_Grammar_Editor GroupWrap?>
        <item repeat="0-1">
          <item>e.</item>
          <item>e.</item>
        </item>
        <ruleref uri="Library.grxml#Digit3" type="application/srgs+xml"/>
        <tag>$.Course = $.Course ? $.Course : {}; $.Course._value =
$$._value</tag>
      </item>
      <item>
        <?MS_Grammar_Editor GroupWrap?>
        <ruleref uri="Library.grxml#Cancel" type="application/srgs+xml"/>
        <tag>$.Cancel = $$</tag>
      </item>
    </one-of>
  </rule>
  <rule id="Password" scope="public">
    <tag>$. _value = ""</tag>
    <one-of>
      <item>

```

```

<?MS_Grammar_Editor GroupWrap?>
<item repeat="1-16">
  <?MS_Grammar_Editor GroupWrap?>
  <one-of>
    <item>
      <?MS_Grammar_Editor GroupWrap?>
      <item>zero</item>
      <tag>$.value = $.value+"0"</tag>
    </item>
    <item>
      <?MS_Grammar_Editor GroupWrap?>
      <ruleref uri="Library.grxml#digit_nonzero"
type="application/srgs+xml"/>
      <tag>$.value = $.value+$.value</tag>
    </item>
    <item>
      <?MS_Grammar_Editor GroupWrap?>
      <ruleref uri="Library.grxml#letter"
type="application/srgs+xml"/>
      <tag>$.value = $.value+$.value</tag>
    </item>
  </one-of>
</item>
<tag>$.Password = $.Password ? $.Password : {}; $.Password.value =
$.value</tag>
  </item>
  <item>
    <?MS_Grammar_Editor GroupWrap?>
    <ruleref uri="Library.grxml#Cancel" type="application/srgs+xml"/>
    <tag>$.Cancel = $$</tag>
  </item>
</one-of>
</rule>
<rule id="Dummy" scope="public">
  <ruleref uri="Library.grxml#Alphanuml" type="application/srgs+xml"/>
  <tag>$.Dummy = $.Dummy ? $.Dummy : {}; $.Dummy.value = "dummy"</tag>
</rule>
<rule id="CourseNames" scope="public">
  <tag>$.value = ""</tag>
  <one-of>
    <item>
      <?MS_Grammar_Editor GroupWrap?>
      <item>introduction to</item>
      <one-of>
        <item>
          <?MS_Grammar_Editor GroupWrap?>
          <item>optical</item>
          <item>fiber</item>
          <item repeat="0-1">communications</item>
          <tag>$.value = "474"</tag>
        </item>
        <item>
          <?MS_Grammar_Editor GroupWrap?>
          <item>optimization theory</item>
          <tag>$.value = "457"</tag>
        </item>
        <item>
          <?MS_Grammar_Editor GroupWrap?>
          <item>remote sensing</item>
          <tag>$.value = "460"</tag>
        </item>
        <item>
          <?MS_Grammar_Editor GroupWrap?>
          <item>v.</item>
          <item>l.</item>
          <item>s.</item>
          <item>i.</item>
          <item repeat="0-1">design</item>
          <tag>$.value = "537"</tag>
        </item>
      </one-of>
    </item>
  </one-of>
</rule>

```

```

        <?MS_Grammar_Editor GroupWrap?>
        <one-of>
            <item>
                <?MS_Grammar_Editor GroupWrap?>
                <item>image</item>
                <tag>$.value = "475"</tag>
            </item>
            <item>
                <?MS_Grammar_Editor GroupWrap?>
                <item>digital</item>
                <item>signal</item>
                <tag>$.value = "473"</tag>
            </item>
        </one-of>
        <item>processing</item>
    </item>
    <item>
        <?MS_Grammar_Editor GroupWrap?>
        <one-of>
            <item>
                <?MS_Grammar_Editor GroupWrap?>
                <item>electronics</item>
                <tag>$.value = "212"</tag>
            </item>
            <item>
                <?MS_Grammar_Editor GroupWrap?>
                <item>electrical</item>
                <tag>$.value = "210"</tag>
            </item>
        </one-of>
        <item>engineering</item>
    </item>
</one-of>
</item>
<item>
    <?MS_Grammar_Editor GroupWrap?>
    <item>image processing</item>
    <tag>$.value = "574"</tag>
</item>
<item>
    <?MS_Grammar_Editor GroupWrap?>
    <item>optical fiber</item>
    <item repeat="0-1">communications</item>
    <tag>$.value = "474"</tag>
</item>
<item>
    <?MS_Grammar_Editor GroupWrap?>
    <item>remote sensing</item>
    <tag>$.value = "460"</tag>
</item>
<item>
    <?MS_Grammar_Editor GroupWrap?>
    <item>optimization</item>
    <item repeat="0-1">theory</item>
    <tag>$.value = "457"</tag>
</item>
<item>
    <?MS_Grammar_Editor GroupWrap?>
    <item>computational intelligence</item>
    <tag>$.value = "592"</tag>
</item>
<item>
    <?MS_Grammar_Editor GroupWrap?>
    <one-of>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <item>computational aspects</item>
            <item>of</item>
            <tag>$.value = "633"</tag>
        </item>
    </one-of>
</item>

```

```

                <?MS_Grammar_Editor GroupWrap?>
                <item>advanced</item>
                <tag>$.value = "638"</tag>
            </item>
        </one-of>
        <item>v. l. s. i.</item>
        <item repeat="0-1">design</item>
    </item>
<item>
    <?MS_Grammar_Editor GroupWrap?>
    <one-of>
        <item>v. h. d. l.</item>
        <item>advanced digital design</item>
    </one-of>
    <tag>$.value = "598"</tag>
</item>
<item>
    <?MS_Grammar_Editor GroupWrap?>
    <item>digital</item>
    <one-of>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <ruleref special="NULL"/>
        </item>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <item>modulation</item>
            <tag>$.value = "643"</tag>
        </item>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <item>communication</item>
            <tag>$.value = "477"</tag>
        </item>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <one-of>
                <item>systems</item>
                <item>design</item>
            </one-of>
            <tag>$.value = "140"</tag>
        </item>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <item>electronics</item>
            <tag>$.value = "432"</tag>
        </item>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <item>control</item>
            <tag>$.value = "552"</tag>
        </item>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <one-of>
                <item>
                    <?MS_Grammar_Editor GroupWrap?>
                    <item>speech</item>
                    <tag>$.value = "578"</tag>
                </item>
                <item>
                    <?MS_Grammar_Editor GroupWrap?>
                    <item>signal</item>
                    <tag>$.value = "570"</tag>
                </item>
            </one-of>
            <item>processing</item>
        </item>
    </one-of>
</item>
<item>

```

```

        <?MS_Grammar_Editor GroupWrap?>
        <item>statistical</item>
        <item repeat="0-1">signal</item>
        <item repeat="0-1">analysis</item>
        <tag>$.value = "577"</tag>
    </item>
    <item>
        <?MS_Grammar_Editor GroupWrap?>
        <item>advanced</item>
        <one-of>
            <item>
                <?MS_Grammar_Editor GroupWrap?>
                <one-of>
                    <item>electronics</item>
                    <item>electronic</item>
                </one-of>
                <item>circuits</item>
                <tag>$.value = "631"</tag>
            </item>
            <item>
                <?MS_Grammar_Editor GroupWrap?>
                <item repeat="0-1">topics</item>
                <item repeat="0-1">in</item>
                <item>speech processing</item>
                <tag>$.value = "683"</tag>
            </item>
        </one-of>
    </item>
    <item>
        <?MS_Grammar_Editor GroupWrap?>
        <item>robotics</item>
        <tag>$.value = "451"</tag>
    </item>
    <item>
        <?MS_Grammar_Editor GroupWrap?>
        <one-of>
            <item>
                <?MS_Grammar_Editor GroupWrap?>
                <ruleref special="NULL"/>
            </item>
            <item>
                <?MS_Grammar_Editor GroupWrap?>
                <item repeat="0-1">introduction to</item>
                <item>robot</item>
                <tag>$.value = "451"</tag>
            </item>
            <item>
                <?MS_Grammar_Editor GroupWrap?>
                <item>error</item>
                <tag>$.value = "644"</tag>
            </item>
            <item>
                <?MS_Grammar_Editor GroupWrap?>
                <item>adaptive</item>
                <tag>$.value = "652"</tag>
            </item>
            <item>
                <?MS_Grammar_Editor GroupWrap?>
                <item>fuzzy</item>
                <tag>$.value = "587"</tag>
            </item>
            <item>
                <?MS_Grammar_Editor GroupWrap?>
                <item>optimal</item>
                <tag>$.value = "653"</tag>
            </item>
        </one-of>
        <item>control</item>
        <one-of>
            <item>
                <?MS_Grammar_Editor GroupWrap?>

```

```

        <ruleref special="NULL"/>
    </item>
    <item>
        <?MS_Grammar_Editor GroupWrap?>
        <item>systems</item>
        <tag>$.value = "352"</tag>
    </item>
    <item repeat="0-1">coding</item>
    <item>
        <?MS_Grammar_Editor GroupWrap?>
        <one-of>
            <item>technology</item>
            <item>design</item>
            <item>
                <?MS_Grammar_Editor GroupWrap?>
                <item>technology</item>
                <item repeat="0-1">and</item>
                <item>design</item>
            </item>
        </one-of>
        <tag>$.value = "450"</tag>
    </item>
</one-of>
</item>
<item>
    <?MS_Grammar_Editor GroupWrap?>
    <item repeat="0-1">computer</item>
    <item repeat="0-1">and</item>
    <item repeat="0-1">communication</item>
    <one-of>
        <item>network</item>
        <item>networks</item>
    </one-of>
    <one-of>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <item>security</item>
            <tag>$.value = "588"</tag>
        </item>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <item>two</item>
            <tag>$.value = "542"</tag>
        </item>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <item>one</item>
            <tag>$.value = "541"</tag>
        </item>
    </one-of>
</item>
<item>
    <?MS_Grammar_Editor GroupWrap?>
    <one-of>
        <item>microwaves</item>
        <item>microwave</item>
    </one-of>
    <tag>$.value = "562"</tag>
</item>
<item>
    <?MS_Grammar_Editor GroupWrap?>
    <one-of>
        <item>micro</item>
        <item>microprocessors</item>
        <item>microprocesor</item>
    </one-of>
    <tag>$.value = "443"</tag>
</item>
<item>
    <?MS_Grammar_Editor GroupWrap?>
    <item repeat="0-1">artificial</item>

```

```

        <item>neural</item>
        <item>networks</item>
        <tag>$.value = "550"</tag>
    </item>
    <item>
        <?MS_Grammar_Editor GroupWrap?>
        <item>probability</item>
        <item repeat="0-1">for</item>
        <item repeat="0-1">electrical</item>
        <item repeat="0-1">engineers</item>
        <tag>$.value = "313"</tag>
    </item>
    <item>
        <?MS_Grammar_Editor GroupWrap?>
        <item>orientation</item>
        <item repeat="0-1">for</item>
        <item repeat="0-1">electrical</item>
        <item repeat="0-1">engineering</item>
        <tag>$.value = "100"</tag>
    </item>
    <item>
        <?MS_Grammar_Editor GroupWrap?>
        <item>java</item>
        <item repeat="0-1">programming</item>
        <tag>$.value = "497"</tag>
    </item>
    <item>
        <?MS_Grammar_Editor GroupWrap?>
        <item>detection estimation</item>
        <item>theory</item>
        <tag>$.value = "677"</tag>
    </item>
    <item>
        <?MS_Grammar_Editor GroupWrap?>
        <item>information theory</item>
        <tag>$.value = "671"</tag>
    </item>
    <item>
        <?MS_Grammar_Editor GroupWrap?>
        <one-of>
            <item>technology and human</item>
            <item>technolgy</item>
        </one-of>
        <item>development</item>
        <tag>$.value = "486"</tag>
    </item>
    <item>
        <?MS_Grammar_Editor GroupWrap?>
        <item>independent component analysis</item>
        <tag>$.value = "583"</tag>
    </item>
    <item>
        <?MS_Grammar_Editor GroupWrap?>
        <item repeat="0-1">solid state</item>
        <one-of>
            <item>
                <?MS_Grammar_Editor GroupWrap?>
                <item>electronics</item>
                <item>and</item>
                <item>devices</item>
                <tag>$.value = "582"</tag>
            </item>
            <item>
                <?MS_Grammar_Editor GroupWrap?>
                <item repeat="0-1">d. c.</item>
                <one-of>
                    <item>drive</item>
                    <item>drives</item>
                </one-of>
                <tag>$.value = "512"</tag>
            </item>
        </one-of>
    </item>

```

```

        </one-of>
    </item>
    <item>
        <?MS_Grammar_Editor GroupWrap?>
        <item>electromechanical</item>
        <item repeat="0-1">energy</item>
        <item repeat="0-1">
            <?MS_Grammar_Editor GroupWrap?>
            <one-of>
                <item>conversion</item>
                <item>conversions</item>
            </one-of>
        </item>
        <tag>$._value = "204"</tag>
    </item>
    <item>
        <?MS_Grammar_Editor GroupWrap?>
        <item>numerical</item>
        <item repeat="0-1">methods</item>
        <item repeat="0-1">for</item>
        <item repeat="0-1">
            <?MS_Grammar_Editor GroupWrap?>
            <one-of>
                <item>e. e.</item>
                <item>
                    <?MS_Grammar_Editor GroupWrap?>
                    <one-of>
                        <item>electronics</item>
                        <item>electronic</item>
                        <item>electrical</item>
                    </one-of>
                    <item>engineering</item>
                </item>
            </one-of>
        </item>
        <tag>$._value = "342"</tag>
    </item>
    <item>
        <?MS_Grammar_Editor GroupWrap?>
        <one-of>
            <item>television</item>
            <item>t. v.</item>
        </one-of>
        <item>engineering</item>
        <tag>$._value = "531"</tag>
    </item>
    <item>
        <?MS_Grammar_Editor GroupWrap?>
        <item>mobile</item>
        <one-of>
            <item>
                <?MS_Grammar_Editor GroupWrap?>
                <one-of>
                    <item>communication</item>
                    <item>communications</item>
                </one-of>
                <tag>$._value = "470"</tag>
            </item>
            <item>
                <?MS_Grammar_Editor GroupWrap?>
                <item>networks</item>
                <tag>$._value = "589"</tag>
            </item>
        </one-of>
    </item>
    <item>
        <?MS_Grammar_Editor GroupWrap?>
        <item repeat="0-1">
            <?MS_Grammar_Editor GroupWrap?>
            <one-of>
                <item>

```

```

                <?MS_Grammar_Editor GroupWrap?>
                <item>current mode</item>
                <item>analog</item>
                <tag>$._value = "599"</tag>
            </item>
            <item>electrical</item>
            <item>electric</item>
        </one-of>
    </item>
    <one-of>
        <item>circuit</item>
        <item>circuits</item>
    </one-of>
    <one-of>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <ruleref special="NULL"/>
        </item>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <item>one</item>
            <tag>$._value = "201"</tag>
        </item>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <item>two</item>
            <tag>$._value = "202"</tag>
        </item>
    </one-of>
</item>
<item>
    <?MS_Grammar_Editor GroupWrap?>
    <one-of>
        <item>op amp</item>
        <item>op amps</item>
    </one-of>
    <item repeat="0-1">and</item>
    <item>applications</item>
    <tag>$._value = "437"</tag>
</item>
<item>
    <?MS_Grammar_Editor GroupWrap?>
    <item>design</item>
    <item repeat="0-1">with</item>
    <one-of>
        <item>i. c.s</item>
        <item>integrated circuits</item>
    </one-of>
    <tag>$._value = "438"</tag>
</item>
<item>
    <?MS_Grammar_Editor GroupWrap?>
    <one-of>
        <item>electronics</item>
        <item>electronic</item>
    </one-of>
    <one-of>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <one-of>
                <item>laboratory two</item>
                <item>lab two</item>
            </one-of>
            <tag>$._value = "439"</tag>
        </item>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <one-of>
                <item>laboratory</item>
                <item>lab</item>
            </one-of>
        </item>
    </one-of>
</item>

```

```

        <tag>$.value = "338"</tag>
    </item>
</one-of>
</item>
<item>
    <?MS_Grammar_Editor GroupWrap?>
    <one-of>
        <item>electronic</item>
        <item>electronics</item>
    </one-of>
    <one-of>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <item>two</item>
            <tag>$.value = "334"</tag>
        </item>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <item>one</item>
            <tag>$.value = "333"</tag>
        </item>
    </one-of>
</item>
<item>
    <?MS_Grammar_Editor GroupWrap?>
    <one-of>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <ruleref special="NULL"/>
        </item>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <item>satelite</item>
            <tag>$.value = "585"</tag>
        </item>
    </one-of>
    <one-of>
        <item>communications</item>
        <item>communication</item>
    </one-of>
    <one-of>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <ruleref special="NULL"/>
        </item>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <item>engineering</item>
            <tag>$.value = "374"</tag>
        </item>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <item>laboratory</item>
            <tag>$.value = "479"</tag>
        </item>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <item>electronics</item>
            <tag>$.value = "433"</tag>
        </item>
    </one-of>
</item>
<item>
    <?MS_Grammar_Editor GroupWrap?>
    <one-of>
        <item>e. m. c.</item>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <item>electromagnetic compatibility</item>
            <item repeat="0-1">
                <item>and</item>

```

```

                <item>interference</item>
            </item>
        </one-of>
    <tag>$.value = "597"</tag>
</item>
<item>
    <?MS_Grammar_Editor GroupWrap?>
    <item>electromagnetic</item>
    <item repeat="0-1">
        <item>field</item>
        <item>theory</item>
    </item>
    <tag>$.value = "363"</tag>
</item>
<item>
    <?MS_Grammar_Editor GroupWrap?>
    <one-of>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <ruleref special="NULL"/>
        </item>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <one-of>
                <item>signals</item>
                <item>signal</item>
            </one-of>
            <ruleref special="GARBAGE"/>
            <tag>$.value = "373"</tag>
        </item>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <item>linear</item>
            <one-of>
                <item>
                    <?MS_Grammar_Editor GroupWrap?>
                    <ruleref special="NULL"/>
                </item>
                <item>
                    <?MS_Grammar_Editor GroupWrap?>
                    <item>multivariable</item>
                    <tag>$.value = "454"</tag>
                </item>
            </one-of>
        </item>
    </one-of>
</item>
<one-of>
<one-of>
    <item>systems</item>
    <item>system</item>
</one-of>
<one-of>
    <item>
        <?MS_Grammar_Editor GroupWrap?>
        <ruleref special="NULL"/>
    </item>
    <item>
        <?MS_Grammar_Editor GroupWrap?>
        <item>theory</item>
        <tag>$.value = "453"</tag>
    </item>
    <item>
        <?MS_Grammar_Editor GroupWrap?>
        <item>design</item>
        <item repeat="0-1">with</item>
        <item>microelectronic</item>
        <item>devices</item>
        <tag>$.value = "632"</tag>
    </item>
    <item>
        <?MS_Grammar_Editor GroupWrap?>

```

```

                                <item>dynamics</item>
                                <item repeat="0-1">and</item>
                                <item>control</item>
                                <tag>$.value = "352"</tag>
                            </item>
                        </one-of>
                    </item>
                </one-of>
            </rule>
            <rule id="Exams" scope="public">
                <one-of>
                    <item>exam</item>
                    <item>laboratory</item>
                    <item>lab</item>
                    <item>quiz</item>
                    <item>homework</item>
                    <item>
                        <?MS_Grammar_Editor GroupWrap?>
                        <one-of>
                            <item>final</item>
                            <item>midterm</item>
                        </one-of>
                        <item repeat="0-1">exam</item>
                    </item>
                </one-of>
            </rule>
            <rule id="Intelligent" scope="public">
                <tag>$.value = ""</tag>
                <one-of>
                    <item>
                        <?MS_Grammar_Editor GroupWrap?>
                        <ruleref special="NULL"/>
                    </item>
                    <item>
                        <?MS_Grammar_Editor GroupWrap?>
                        <ruleref special="GARBAGE"/>
                    </item>
                </one-of>
                <one-of>
                    <item>
                        <?MS_Grammar_Editor GroupWrap?>
                        <ruleref uri="Library.grxml#Cancel" type="application/srgs+xml"/>
                        <tag>$ = $$</tag>
                    </item>
                    <item>
                        <?MS_Grammar_Editor GroupWrap?>
                        <one-of>
                            <item>announcements</item>
                            <item>announcement</item>
                        </one-of>
                        <one-of>
                            <item>
                                <?MS_Grammar_Editor GroupWrap?>
                                <ruleref special="NULL"/>
                            </item>
                            <item>
                                <?MS_Grammar_Editor GroupWrap?>
                                <ruleref special="GARBAGE"/>
                            </item>
                        </one-of>
                    </one-of>
                </one-of>
                <item>
                    <?MS_Grammar_Editor GroupWrap?>
                    <ruleref uri="#CourseCodes"
type="application/srgs+xml"/>
                </item>
                <item>
                    <?MS_Grammar_Editor GroupWrap?>
                    <ruleref uri="#CourseNames"
type="application/srgs+xml"/>
                </item>
            </item>

```

```

        </one-of>
        <tag>$_value = "Announcements "+$$._value</tag>
    </item>
    <item>
        <?MS_Grammar_Editor GroupWrap?>
        <one-of>
            <item>
                <?MS_Grammar_Editor GroupWrap?>
                <ruleref uri="#CourseCodes"
type="application/srgs+xml"/>
            </item>
            <item>
                <?MS_Grammar_Editor GroupWrap?>
                <ruleref uri="#CourseNames"
type="application/srgs+xml"/>
            </item>
        </one-of>
        <tag>$_value = "Announcements "+$$._value</tag>
        <one-of>
            <item>
                <?MS_Grammar_Editor GroupWrap?>
                <ruleref special="NULL"/>
            </item>
            <item>
                <?MS_Grammar_Editor GroupWrap?>
                <ruleref special="GARBAGE"/>
            </item>
        </one-of>
        <one-of>
            <item>announcements</item>
            <item>announcement</item>
        </one-of>
    </item>
    <item>
        <?MS_Grammar_Editor GroupWrap?>
        <one-of>
            <item>assignments</item>
            <item>assignment</item>
        </one-of>
        <one-of>
            <item>
                <?MS_Grammar_Editor GroupWrap?>
                <ruleref special="NULL"/>
            </item>
            <item>
                <?MS_Grammar_Editor GroupWrap?>
                <ruleref special="GARBAGE"/>
            </item>
        </one-of>
        <one-of>
            <item>
                <?MS_Grammar_Editor GroupWrap?>
                <ruleref uri="#CourseCodes"
type="application/srgs+xml"/>
            </item>
            <item>
                <?MS_Grammar_Editor GroupWrap?>
                <ruleref uri="#CourseNames"
type="application/srgs+xml"/>
            </item>
        </one-of>
        <tag>$_value = "Assignments "+$$._value</tag>
    </item>
    <item>
        <?MS_Grammar_Editor GroupWrap?>
        <one-of>
            <item>
                <?MS_Grammar_Editor GroupWrap?>
                <ruleref uri="#CourseCodes"
type="application/srgs+xml"/>
            </item>
        </one-of>
    </item>

```

```

        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <ruleref uri="#CourseNames"
type="application/srgs+xml"/>
        </item>
    </one-of>
    <tag>$. _value = "Assignments "+$$._value</tag>
    <one-of>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <ruleref special="NULL"/>
        </item>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <ruleref special="GARBAGE"/>
        </item>
    </one-of>
    <one-of>
        <item>assignment</item>
        <item>assignments</item>
    </one-of>
</item>
<item>
    <?MS_Grammar_Editor GroupWrap?>
    <one-of>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <ruleref uri="#CourseCodes"
type="application/srgs+xml"/>
        </item>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <ruleref uri="#CourseNames"
type="application/srgs+xml"/>
        </item>
    </one-of>
    <tag>$. _value = "Grades "+$$._value</tag>
    <one-of>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <ruleref special="NULL"/>
        </item>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <ruleref special="GARBAGE"/>
        </item>
    </one-of>
    <one-of>
        <item>grades</item>
        <item>grade</item>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <ruleref uri="#Exams" type="application/srgs+xml"/>
        </item>
    </one-of>
</item>
<item>
    <?MS_Grammar_Editor GroupWrap?>
    <one-of>
        <item>grades</item>
        <item>grade</item>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <ruleref uri="#Exams" type="application/srgs+xml"/>
        </item>
    </one-of>
    <one-of>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <ruleref special="NULL"/>
        </item>







































































```















































































```

        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <ruleref special="GARBAGE"/>
        </item>
    </one-of>
    <one-of>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <ruleref uri="#CourseCodes"
type="application/srgs+xml"/>
        </item>
        <item>
            <?MS_Grammar_Editor GroupWrap?>
            <ruleref uri="#CourseNames"
type="application/srgs+xml"/>
        </item>
    </one-of>
    <tag>$. _value = "Grades " + $. _value </tag>
</item>
</one-of>
<one-of>
    <item>
        <?MS_Grammar_Editor GroupWrap?>
        <ruleref special="NULL"/>
    </item>
    <item>
        <?MS_Grammar_Editor GroupWrap?>
        <ruleref special="GARBAGE"/>
    </item>
</one-of>
</rule>
<rule id="CourseCodes" scope="public">
    <item repeat="0-1">
        <item>e.</item>
        <item>e.</item>
    </item>
    <ruleref uri="Library.grxml#Digit3" type="application/srgs+xml"/>
    <tag>$. _value = $. _value </tag>
</rule>
</grammar>

```

Prompts

Transcription	Display Text	Has Wave	Has Alignments	Wave
[Welcome to CIMS]	Welcome to CIMS			Rec0000.wav
[Electrical and Electronic Engineering Department Cou...]	Electrical and Electronic E...			Rec0001.wav
[Please enter your] [user name]	Please enter your user name			Rec0002.wav
[digit by digit]	digit by digit			Rec0003.wav
[letter by letter]	letter by letter			Rec0004.wav
[password]	password			Rec0005.wav
[or]	or			Rec0006.wav
[Please select one of the following menu items]	Please select one of the f...			Rec0007.wav
[Grades]	Grades			Rec0008.wav
[Announcements]	Announcements			Rec0009.wav
[Assignments]	Assignments			Rec0010.wav
[Please enter the course code]	Please enter the course c...			Rec0011.wav
[Thanks for using CIMS]	Thanks for using CIMS			Rec0012.wav
[Good Bye]	Good Bye			Rec0013.wav
[No such user exists]	No such user exists			Rec0015.wav
[Incorrect password]	Incorrect password			Rec0016.wav
[You are not taking this course]	You are not taking this co...			Rec0017.wav
[Such a course does not exist]	Such a course does not exist			Rec0018.wav
[Such a menu item is not available]	Such a menu item is not a...			Rec0019.wav
[spice]	spice			Rec0020.wav
[matlab]	matlab			Rec0021.wav
[experiment]	experiment			Rec0022.wav
[homework]	homework			Rec0023.wav
[exam]	exam			Rec0024.wav
[final]	final			Rec0025.wav
[midterm]	midterm			Rec0026.wav
[report]	report			Rec0027.wav
[lab]	lab			Rec0028.wav
[quiz]	quiz			Rec0029.wav
[Database connection cannot be established]	Database connection can...			Rec0040.wav
[no announcements made] [since your last access]	no announcements made ...			Rec0014.wav
[there are no grades announced]	there are no grades anno...			Rec0041.wav
[no announcements have been made]	no announcements have ...			Rec0043.wav
[in this course]	in this course			Rec0044.wav
[there are no announcements]	there are no announceme...			Rec0045.wav

[there are no assignments]	there are no assignments			Rec0046.wav
[user name] [was not recognized]	user name was not recog...			Rec0047.wav
[course ID was not recognized]	course ID was not recogni...			Rec0048.wav
[menu item]	menu item			Rec0049.wav
[your answer]	your answer			Rec0050.wav
[one]	one			Rec0030.wav
[two]	two			Rec0031.wav
[three]	three			Rec0032.wav
[four]	four			Rec0033.wav
[five]	five			Rec0034.wav
[six]	six			Rec0035.wav
[seven]	seven			Rec0036.wav
[eight]	eight			Rec0037.wav
[nine]	nine			Rec0038.wav
[ten]	ten			Rec0039.wav
[eleven]	eleven			Rec0051.wav
[twelve]	twelve			Rec0052.wav
[thirteen]	thirteen			Rec0053.wav
[fifteen]	fifteen			Rec0054.wav
[sixteen]	sixteen			Rec0055.wav
[seventeen]	seventeen			Rec0056.wav
[eighteen]	eighteen			Rec0057.wav
[nineteen]	nineteen			Rec0058.wav
[twenty]	twenty			Rec0059.wav
[thirty]	thirty			Rec0060.wav
[forty]	fourty			Rec0061.wav
[fifty]	fifty			Rec0062.wav
[sixty]	sixty			Rec0063.wav
[seventy]	seventy			Rec0064.wav
[eighty]	eighty			Rec0065.wav
[ninety]	ninety			Rec0066.wav
[hundred]	hundred			Rec0067.wav
[is] [announced but your grade is not available]	is announced but your gra...			Rec0068.wav
[fourteen]	fourteen			Rec0042.wav
[Do you want to use the intelligent mode]	Do you want to use the in...			Rec0069.wav
[Do you want to continue]	Do you want to continue?			Rec0070.wav
[What would you like to do]	What would you like to do?			Rec0071.wav
[do you want to listen to other grades]	do you want to listen to o...			Rec0072.wav
[do you want to listen to other announcements]	do you want to listen to o...			Rec0073.wav