



The Eidoserve Virtual Interview System: With Intelligent Service Agents

Description

A "Virtual interview" (or VI) is generally a method of simulating dynamic complex human interaction through automated intelligent Service Agents. In other words, a virtual Service Agent will respond to user stimuli as if the user is talking to another human being. The Service Agent may be a recorded version of a person, a virtual or graphical likeness of a person, or a completely fictitious person. Service Agents combine intelligent agents (software that does automated tasks or groups of tasks), a dynamic natural language user interface, artificial intelligence including the ability to learn and associate, advanced pattern matching, audio (speech) and visual recognition, to interact with a user. The VI system, and the Service Agent specifically, interprets inputs from users via text, and controlled speech (audio), and responds and interacts with the user with a combination of audio (pre-recorded or live), video (pre-recorded or live), human animation, images, documents, instant messaging, web cam, and/or many other communicative technologies.

The VI process may be used to educate and train, to provide support (technical and otherwise), to provide information, as a personal assistant, as an interface to data, to tell stories, and to report the news. Future applications can easily be extended for vending machines, toys, GPS systems, inventory data collection, in shopping carts, in e-commerce engines and in a variety of other applications. In short, any system which requires more than mere rote response to a given stimulus, may utilize the VI technology to enhance the user experience. The experience will be more normal and natural to the user.

The VI system and interface is readily adaptable to a wide variety of platforms and is not limited to any particular platform per se. Some of the most natural platforms for use with the VI system include computers, web interfaces, television, webtv, telephones (wired and wireless), mobile devices (*i.e.*, PDAs), directly connected video monitors, and any other audio and/or visual medium. The 'animated' interface is interesting because of it's extension to using the phone, and phone-calling the individual, and really acting like a personal assistant in many cases.

Some of the terms and concepts used to describe the present system include the following materials. These defined components may be a part



of the Virtual Interview system, and the definitions describe how each function may be used in the system.

Natural Language Interface –the ability to converse with a computer in ‘normal / conversational’ language syntax

Artificial Intelligence – the ability for the computer to make better, more educated decisions over time.

User Input – any information that is computed to the system by any method of data entry (keyboard, audio, visual, touch, sensor, etc.

User Session – the time period commencing when the user first begins to use the system (in any way) and running through the closing of the system.

Types of User Input – user input may be a question, a response to a question, some background information, a statement or any combination thereof.

Input Array – an array used to categorize user input by type and to store the system responses output to each user input.

Question Words – words that are used to signal that a user is attempting to ask a question to the system (e.g., who, what , when , where , and why).

Secondary Question Words – question words such as "can," "should," and "would" which typically signal a question when used at the beginning of a sentence.

Conditional Words – words such as "but," "if," "or," and "and" which may be stored in the input array so the system can dissect complex inputs.

Symbols - ?, . ! and any other type of symbol or punctuation which may be used by the system to delineate the end of questions / responses / statements.

Input Patterns – are pre-defined patterns used to execute functions and define the user input and break into semantics. For example an Input pattern could be spell checker or to identify a type of input.

System Responses – responses a user might have to a system question. User input may be only checked against the last system question asked. If no match is found, the system stops processing and offers a list of responses for the user to choose from. This is an opportunity for the system to learn. If the system still didn’t find a response after the second try, the system may start down the default response path.

System Questions – questions, responses or statements that the system delivers to the user which may or may not have supporting media associated with them.



Keywords – are preferably used in both questions and system responses. Key words are primarily used in free text searching.

Exact Match – occurs when the system attempts to match exactly a user input to a question or system response. This is done for normal questions types first with and without a subject match. Thereafter, the system checks for general question types with and without a subject match. Spaces may be stripped out of the input and case may be removed. Symbols may also be added to / removed from the input.

Question Types – are categories of questions such as "general," "stop," "dirty" or "normal," that are used to group questions when processing logic. General question types are typically questions that don't have a unique identifier or are general in nature. Normal question types are questions that can be uniquely defined. Stop question types are noise words. Dirty words are foul or unwanted language.

Subject / Last Subject - The system attempts to determine the last subject or current subject to which the user input relates. This is preferably done numerically.

Free Text Search – a weighted and ranked search based on keywords with and without subject.

Drive word Search – is a specific Free Text Search that is done after the first Free Text Search is done when there is more than one result returned. The system checks the input for question or secondary question words to determine direction words. Then, a Free Text Search is done on Drive words for the records returned from that first Free Text Search and that match the Direction words found.

Drive words – are typically words such as verbs that can't be uniquely defined as keywords but may be helpful in narrowing in on the input.

Direction words – Are question words and secondary question words

Questions – questions a user might ask which are defined by keywords, directions, words, and drive words.

Responses – are responses that are linked to questions a user might ask. They are preferably linked to supporting media.

Supporting Media – are documents, audio, video, links and other multimedia that are linked to both responses and system questions to facilitate a more intuitive user experience.

Goals – are, in a sense, "next steps" that define what the system or user is trying to do for responses and system questions. Goals are the building blocks for interaction within the system; they define flow of a conversation and can lead a user down a path. Goals may have one or



more stages, and each stage can have one or more actions and each action can have one or more rules.

Stages – generally speaking, are group actions which allow the system to be set up to process actions in order and halt until an action is complete. All stages are processed until an action or rule fails.

Actions – execute functions and interaction. They can have one or more rules associated with them. All actions are executed for a given stage until an action or rule fails.

Rules – are checks and guidelines for actions. They can parse user data, compare values, execute functions and more. All rules are executed for a given action until one fails.

Learning – the system learns in various ways. Learning means the system learns based on interaction with the user. The system can learn automatically or controlled. See logic section for specifics on learning.

Rating – users can rate each response from the system, and that information is stored for each user. If a user rates a response below the minimum level required for the interview, they will not see that response again and the next response, if applicable, is given to the user. This is an opportunity for the system to learn.

DataSources –generally speaking, every type of data source is stored here. The system can pull on virtually an type of data to give its answers. For example there are the system tables such as questions and system responses, the web, general knowledge base, any link or document, any odbc (*open database connectivity*) compliment system such as Siebel or People Soft.

Domains and Sub Domains – gives the system the ability to differentiate between similar terms in separate domains, as well as to show some users one set of answers and another set of users a different set. The example that makes sense here is a call center: the support tech will need more technical language while the customer will need information in more layman's terms.

Default Path – is the path that determines what to say to the user when the system can't figure out what the user is trying to communicate to the system. This is an opportunity for the system to learn.

Customer User Functions and SQL – The administrative user for the VI can create customer functions and SQL to allow the system to interact with other systems.

Administrative Site – the 'behind the scenes' maintenance is accomplished by going thru an administrative entrance to the site. It



provides access to such things as the logic systems and the data arrays housed within the system.

Interface – the graphical user interface for the SI. Each item within the interface is an object and is driven from a database.

Instant Messaging - (one to one or one to many) also push and receive of docs, links, and video

Web Cam - (one or two way) video and audio feed across the Internet.

Animation – the ability to film/record/synthesize the minimum required amount of an individual and then have dynamic animated movement and speech of that individual driven from a database table.

Speech Recognition – the ability to have audio or speech via microphone or other multimedia device and have it invoke the system as a user input.

Gesture Recognition – the ability to capture gestures via web cam or other multimedia device and have it invoke the system as a user input.

Email Interface – the ability to interface with the system via email.

Discussion Relating To The VI System Logic

The logic behind the VI is broken down into different types of interaction from the user: (1) initial use of the system; and (2) subsequent user sessions. An exemplary system methodology for each of these scenarios will now be discussed.

1. Initial Use Of The System By A User

- a) User is prompted to login if that is required, or the system creates the user id automatically
- b) Information about the user and their machine / internet connection are detected and stored, if required.
- c) An introductory response from the system is delivered, which may collect the user's name and other information or simply say "hello" and ask what the VI can do for them.
- d) If the user enters the VI from a place where they can enter a question, the system attempts to answer the user question.
- e) Once this process is complete, the VI is ready to process any user inputs, as the requests are received.
- f) A user session is created so that the system can group goals / questions / patterns and track usage of the system.



- g) The system gets info needed for that interview (can break out later)

2. Subsequent Use Of The System By The Same User

- a) If the VI is configured to review or build a rapport with the user, it does so; otherwise a simple welcome back message is given to the user.
- b) The VI determines if the user had any reminders set up or unfinished goals, and, if so, asks if the user would like to pick up where they left off.
- c) A new user session is created.
- d) The system gets info needed for that interview (can break out later)

After the login and welcome above, the general methodology used by the VI system to interact with users involves a variety of different processes. Some of those processes are outlined below.

1) Determining types of user input and building the Input Array

At this point, the user is prompted to input either a response to a question the system asked or a question from the user. The user input is split into an array of statements/responses and questions. The system splits the input -- assuming if the user has a question word in the input that the question goes until the next question word or a symbol. If there is a secondary question word found at the beginning of the user input or following a symbol, the system also assumes that this is a question.

Any part of the input that doesn't have a question word or begin with a secondary question word is split out as a response or pattern. Any conditional words such as "and," "or," "if," or "but" are also stored with each element of the array that have them contained. Each element of the array is processed by the system. Should an input array element have a goal associated with it, the processing is stopped because there is input needed from the user. The system can be configured to remember all inputs and go back and process them.

The system also checks to see if the user inputs are all stop words. If so, the user is given the stop word default system question and is asked to give us more context. The system checks if dirty words are allowed in the interview, and the user is given the dirty default if it's on and the input contains dirty language.



2) Checking for System Responses and Patterns matches

At this point the system has built our Input Array and checks to see if the system is expecting a response from the user and checks for matching input patterns. If all inputs are not text (*i.e.*, no question or secondary question words or there is no previous system question in the user session log or no system responses mapped to that system question), then the system does not do a system response search.

Again, note that if a goal is found, processing stops to get input from the user.

- The system checks for input patterns (see section 10 Input patterns, below).
 - Replace misspelled words and acronyms, contractions, other bad phrases
 - Check for specific patterns *i.e.*, search, tell me more, etc and execute those functions. For certain types of patterns a response can be generated and a goal initiated.

- The system checks the user session log for a failed goal and grabs the last system question ID and attempts
 - Do an exact match (see section 4 Exact Match below) search on the system responses.
 - If no exact match, the system does a free text (see section 5 Searching below) search on keywords of system responses.
 - If there is a match, the system checks to see if there is a goal and does the goal then processes previous failed goals if they exists.
 - If there is no system response match, processing stops and the user is given a list of responses the system was expecting for the user to choose from. The system stops all processing at this point. This is an opportunity for the system to learn.

3) Checking for Question Matches

If the system reaches this step in the process, the user input is either a question or a variable input from the user, like the user's name, for example. The system again processes each input array element until complete or a goal is found.



- The system performs an exact match (see section 4 Exact Match below) on normal and general question with subject and without subject breaking out as soon as the system finds one
- If there is no exact match, the system does free text (see section 5 Searching below) searching on keywords
- If only one match, the system returns that as the match
- If more than one match, the system looks at direction words (what, where, etc)
- Take the subset of keywords match and direction words match and do a free text on drive words
- If nothing found here, the system does a general question check where the system counts word matches. If the count is over the system acceptable %, the response and/or system question is displayed. If the count is below, the system responds with something like "I am not sure this is what you were trying to say but..." and then gives the user a list of what it thought were close questions to their particular issue. Based upon the user's choice, this is an opportunity for the system to learn.
- If nothing found at this point, the system checks for a previous failed goal, which means the system was expecting a variable user input, such as name, etc. If there is one, it is executed and the system moves to the next element in the input array.
- The system now checks to see if other data sources are enabled for this interview. Other data sources could consist of a dictionary, encyclopedia, a web link, a knowledgebase, an existing system such as PeopleSoft or Siebel. This system was designed to be able to pull from any data source
- If the system reaches this point the system don't know what the user is trying to convey to the system and start down the default path (see section 8 Default Path below).

4) **Exact Match**

The system does an Exact Match check in the following order. Note the process is the same for questions and system responses and each is done separately in the logic when the function is called

- Strips all spaces from the current input array element
- Removes case from all characters.
- Add and remove symbols.



- Checks for an exact match for normal question types or system responses with subject
- Checks for an exact match for normal question types or system responses without subject
- Checks for an exact match for general question types or system responses with subject
- Checks for an exact match for general question types or system responses without subject

5) **Searching**

The system does a free text search on questions and system responses. The Free Text search is based on the stripping of stop words or noise words, weighting, close meanings of words, keywords and / or drive words and rank. The Free Text search happens in the following order. Note for Free Text Search on system responses it stops after step one.

- A Free Text Search is done based on the grammartized sentence configuration (nouns / subject at this point)
- If more than one result is found the system determines directions words in the input. If one result is found that result is given.
- The system then does another Free Text Search on the subset of the grammartized sentence (drive words, verbs at this point – then direction words (adjectives) after this).
- If results are found the top ranked response is given
- If no results are found the top Free Text ranked result is given
- If no results are found in the Free Text the system proceeds to general question search.
- General question search takes a look at the user input and does a word count based on questions with subject and without subject / nouns / verbs / adjectives, etc.
- The system takes the first result that is greater than the system defined minimum it gives that result.
- If the word count is below the system minimum the system states it's not sure it has the right answer to what they were asking, gives the result, and asks if it answered their question. This is an opportunity for the system to learn.



6) **Processing Goals, Stages, Actions, Rules**

The system brain is in the processing of Goals, Stages, Actions, and Rules.

Basically any interaction with the system can have a Goal associated to it. Goals are executed until an action fails or a rule is broken. If a Goal fails the system knows that and takes action to get the user to input what it needs to pass the Goal. Stages are used to order the way actions are processed by the system. Each Stage can have one or more Actions associated with it. Actions are anything from custom functions to executing queries to a database. Each Action can have one or many Rules that are anything from custom functions to comparisons. All Actions and Rules are executed until they fail.

7) **Session / Answer Logs**

The User Session and User Answer logs are critical to maintain state within the system, track performance, and maintenance of the system. The User Session log stores all current activity in order for the system to execute goals and interact intelligently with the user. The User Answer log stores all interaction once it is complete or terminated. It gives the system the ability to view history and create future patterns.

8) **Default Path**

The Default Path is used by the system when the system can't determine what the user is trying to communicate. The Default Path is an opportunity for the System to learn.

- Check to see if there is a previous goal or failed goal and give the Goal Default, which is a default response specific to that goal.
- If no previous goal or failed goal the system checks to see if this is the first time it has reached a default on and for last subject of the previous question asked and give subject specific default. For example the system knows that it was talking about cars but didn't quite understand what was said, could you please rephrase or try one of these. Here the system also suggests based upon related question patterns of all users what others asked next or previous based on their last question.



- If this is the second time default was found or no subject could be determined the user would receive the system default which can be set up with a goal to ask them if they want to be contacted when an answer is found to their question or search the web. The system can be set up to create a case ticket at this point and the users input is stored in a table for admin personal to review. The system can also be set up to email a transcript of the conversation to the user.

9) **Learning and Rating**

There are several points at which the system can learn either freely or controlled by and admin, of course learning can be turned on or off per interview. The key to learning is looking at what users ask or say to the system and how they respond to the systems response to that input. The system can also learn word associations per interview or user.

Additionally patterns and probabilities can be leveraged across the interview. For example the system dynamically determines FAQ's in real-time. Another example is the system stores patterns of questions asked and goals gone down. Currently the system learns the following ways

- Dynamically determines FAQ's in real-time
- Stores and reviews probabilities of question patterns and goals
- Stores rating info and tailors responses based on user feedback dynamically. Also can remove all together based on interview minimum criteria or flag admin automatically for review
- When the system goes down the Default Path the first time it associates the user input and their next user input with the question / response that the next input matched with. This can be configured to be automatic or based on admin review.
- When the system expects a user input (i.e. the system question has system responses mapped to it, and can't match the user input to a system response) the system gives the user choices. At this point, the system can associate the users last input to the response that they choose when given options. This can be configured to be automatic or based on admin review.



- When the system finds a general search match below the system minimum % matched word count needed, the system asks if this helped. If the user says yes, the system can store the last user input with that response. This can be configured to be automatic or based on admin review.
- The system also uses bayes theorem to learn.
- There are a number of reports and calculations that can be conducted on the User Answer Logs to allow the system to learn.
- When a user disagrees with a response the system gives, the system asks what the user thinks it should be and associates that user input with the response they disagreed with. It can be configured to replace that response order over time based on rating or approval from users. This can also be configured to be automatic or based on admin review.

The system, if rating is on for the interview, can store ratings per user on responses. So if a user rates an item below the interview defined minimum rating they would not see that response again. On the admin side this response is then flagged for review. The system can be configured to automatically remove the response if minimum interview criteria are met, so no user would ever see that item again. Rating applies to all responses normal or learned.

10) **Input Patterns**

Input patterns revolve around advanced semantic understanding, advanced patterns, bayes theorem and probabilities for question and goals and responses.

11) **Data Sources**

The system was designed to integrate with any odbc compliant data source. The system can be set up to order the processing of the data sources. Also a particular data source can be used only for specific questions thru custom functions and SQL. The idea here is that this system can process any type or format of data some examples include but are not limited to.

- System Tables



- CRM and ERP solutions (Siebel, PeopleSoft, SAP for example)
- Web Sites
- Links
- Documents
- Video
- General Knowledge Base

12) **Custom Functions**

Custom functions allow the administrator of the system to set up ad hock and on demand functions. These functions can be used to write to/retrieve from a database, interact with another system, send a message, compare data, and create rules.

13) **Templates and Leverage VI data**

The system has the ability to create templates set of knowledge, questions, responses, system responses, system questions, goals, stages, actions, and rules. The templates can be used in other systems and interviews. Additionally the system data can be used to for other systems, interviews, marketing, sales, supply forecasting, and patterns / trends.

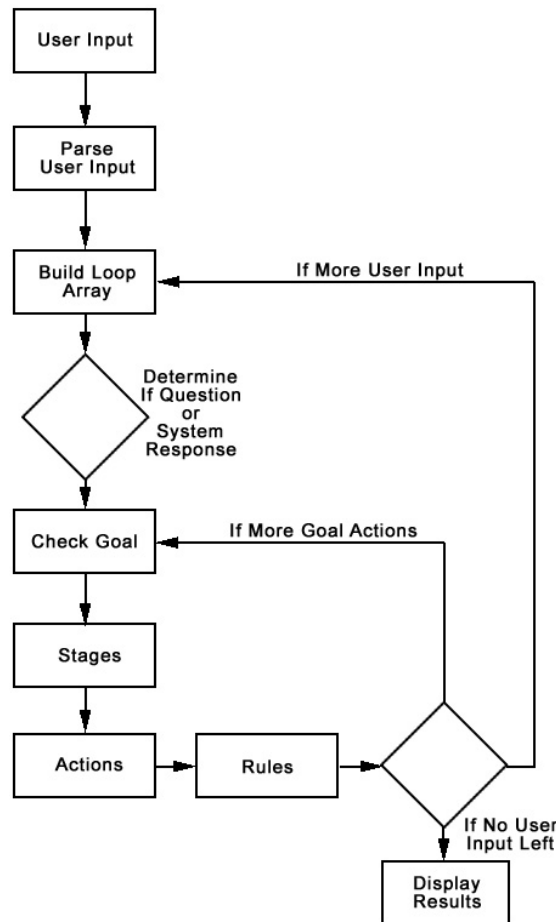
14) **VI Interface – see Appendix 1**

15) **Administrative Side – see Appendix 2**



Figures

This is a high-level logic flow diagram, which illustrates the high-level system flow. The system takes in the user input and pareses to determine what the user is communicating. Each group of the user input is put into the loop array. The system determines if the user input is a question or a response to a question. Then the system checks to see if the result has a goal, stage, action and rule and executes until an action or rule fails. This process is repeated until no more user inputs are in the loop array or an action or rule fails. The system then process all system outputs, prepares, and then displays to the user.

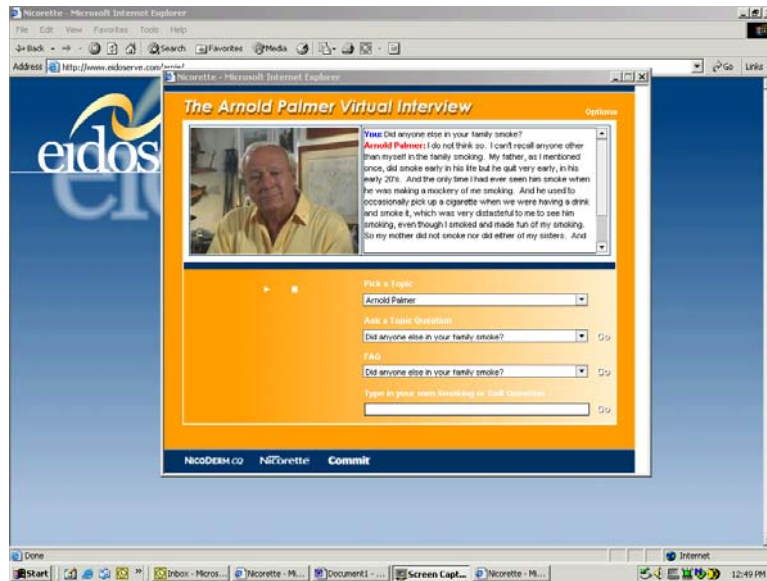




445 ft. pitt blvd. suite 501
pittsburgh, pa 15219
p 412.227.0254
f 412.227.0853
www.eidoserve.com



Appendix 1 VI graphical user interface

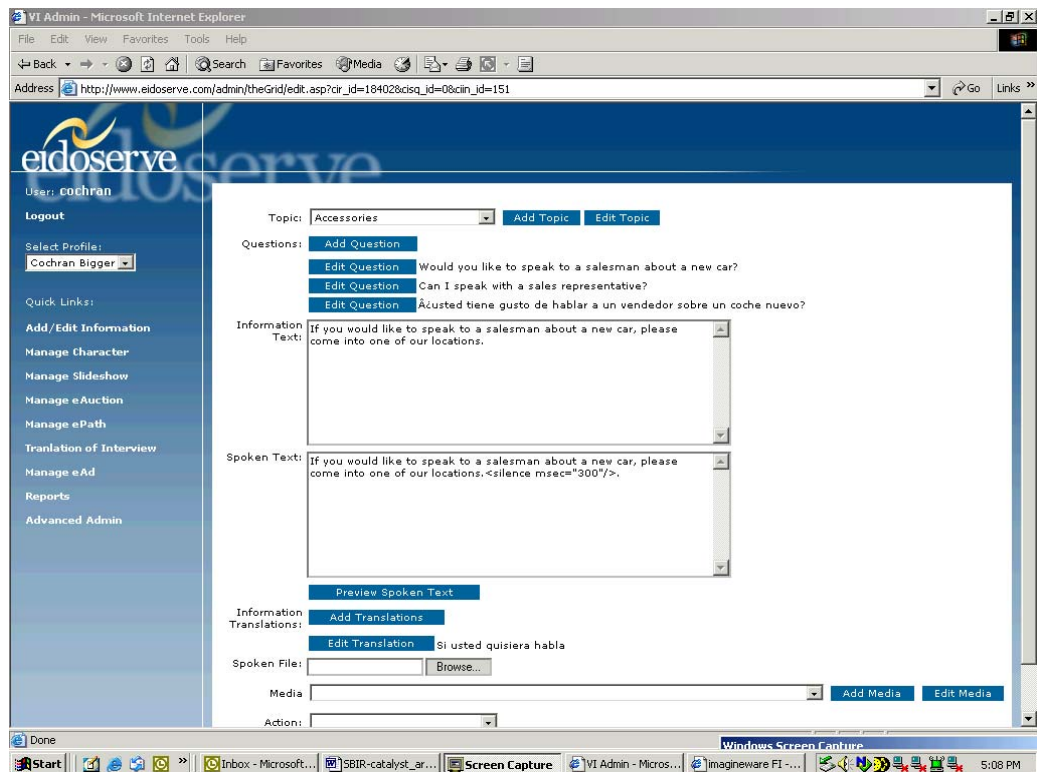




Appendix 2:

Administrative Interface:

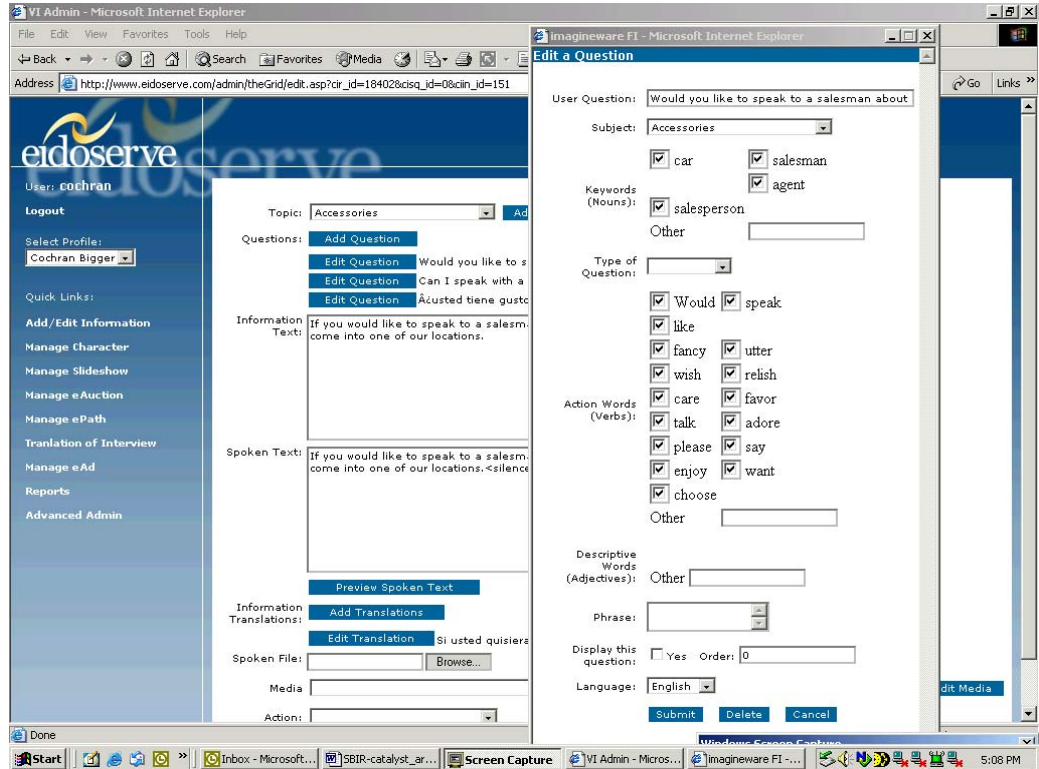
A couple views of the existing 'administrative' user interface to the Q-Engine that are being used by Eidoserve customers currently are shown below: The first one shows the many questions that are both machine and hand-generated – that lead a user to a specific response. These can even be translated into multiple languages – as you can see.



Notice the ability to add multi-media files as a method of rich-media delivery – as well as multi-lingual capabilities.



The following slide will show some of the automatically, machine-generated dictionary elements. Some of these deal with subjects, nouns, verbs, etc.



The issues surrounding an administrative side are two fold: (a) how do you validate / quantify that a question was answered correctly, and then (b) if you find that one was answered incorrectly, how do you notify the appropriate personnel for the correction. In terms of, how do we validate that the correct answer was given? Eidoserve automatically grades answers on a relevance score. This score is generated from algorithms that take in to account grammars, statistical analysis, and linguistic techniques. All situations can be reported to the user at any time. However, when the relevance score falls outside a range, different responses are triggered, and these situations can be tracked and notified. Notification can be sent via e-mail, as well as via phone. Yes, the output can actually trigger a phone call to a 'live person', where a text to speech engine can recount the problem and the solution that was given to the customer.