IDiOM

"the Interactive Dictionary Output Module"

David Crawley
Michael Sparandara

Table of Contents

Chapter 1: General Information 3
- ***1.1 Application*** *3*
- ***1.2 Audience, Aesthetic, and Design*** *3*
- ***1.3 Research*** *3*
- ***1.4 Technology*** *4*
- ***1.5 Usability Evaluations*** *6*
- ***1.6 Open Issues*** *6*
  - 1.6.1 Advanced Navigation/UI Features: 6
  - 1.6.2 Advanced DTMF/T-9 Word Input 7
  - 1.6.3 Advanced Database Logging and Web Interface 7
  - 1.6.4 Advanced Language Resources 7
  - 1.6.5 Commercial Implications 7
  - 1.6.6 Fun Features 8
- ***1.7 Call Examples*** *8*
  - 1.7.1 A User Gets a Definition and a Sentence 8
  - 1.7.2 DTMF Entry for a Speedy Word Lookup 9

Chapter 2: Application Behavior 10
- ***2.1 Call Flow*** *10*
- ***2.2 State Tables*** *11*
  - 900_Welcome 11
  - 1000_Spell_Word 11
  - 1100_Say_Word 12
  - 1200_Confirm_Word 12
  - 3000_Definition 13
  - 3400_Synonyms 14
  - 3500_Sentence 14
  - 4000_Restart 14
  - 9000_Goodbye 15
  - COMMANDS 15

Chapter 3: Recording Script 16

Appendix 18

Chapter 1: General Information

1.1 Application

The mission of IDiOM is to provide an easy-to-use, comprehensive dictionary to anyone with access to a phone. The application extracts a word from the caller and then provides him or her with a definition, synonyms, and contextual examples.

1.2 Audience, Aesthetic, and Design

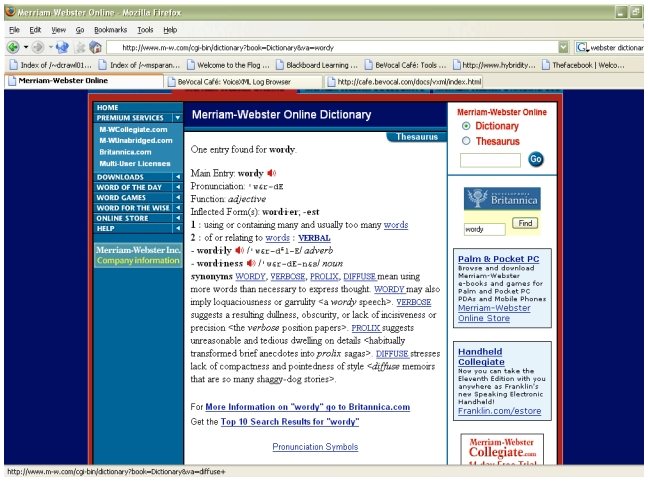
IDiOM is meant to be used by *anyone*. For this reason, it was important to design the system so that it would be as intuitive and easy to use for an 80 year-old man as it is for a middle-school girl.

Although IDiOM is not meant to replace a paper dictionary, it is a useful stand-in when one isn't available. Looking up a definition in a paper dictionary is a standard procedure; the user does not need prompting and instruction each time he or she uses it. The IDiOM system was designed with this concept in mind. Searching for definitions on the phone should be as logical as using a paper version. For this reason, a minimalist approach was taken in IDiOM's design; extraneous prompting is avoided wherever possible. The user is given very basic initial prompts, with additional information and instruction available in help prompts. Although first time users are usually able to follow the basic instructions provided in the initial prompts, if they do get stuck, help is only one word away. Repeat users are able to fly through the system getting their information as quickly as possible.

The search for a voice talent yielded the smooth, British-accented voice of Helen Lisanti. Ms. Lisanti's measured enunciation gives the application a scholarly, professional feel. Her speech, coupled with clips of Beethoven's fifth symphony during queue time, make users feel as if they are "in a library".

1.3 Research

Much research went into the development of IDiOM. In order to make use of advanced VXML capabilities integrated with dynamic scripting languages such as JavaScript and PERL, developers relied heavily on online documentation. The *World Wide Web Consortium* specifications on VXML and BeVocal's documentation section were invaluable resources for writing and debugging the application.



Example of a prolix web-based dictionary entry, ineffective for TTS output

The majority of the research time was dedicated to finding open source resources for the system to use. Most importantly, a suitable dictionary had to be located. The Idiom system needs a dictionary that is easily queried to find definitions and contains definitions that are suitable to TTS

output (simple and not too wordy). Popular online dictionaries such as Dictionary.com and Merriam-Webster's Online were either too difficult too parse or had overly verbose definitions.

An appropriate dictionary was eventually located through Dict.org. Dict.org is a web interface that searches various open source dictionaries. One of these dictionaries, WordNet, suited the purposes of the IDiOM application. WordNet was developed by a team of cognitive language scientists at Princeton University. The dictionary contains over 152,000 different words linked together based on relationships between entries. The definitions provided are clear and concise, and synonyms and contextual examples are provided.

A second concern was finding a list of words to be searched in order to compile the grammar. Initially, the UNIX spell check dictionary was used; however, this list had some limitations. The UNIX list is based on the 1913 Webster's Unabridged Dictionary, which makes it extremely large. In fact, it includes over 240,000 words. Although this seemed like a good thing at first, the grammars generated from this list ended up being extremely large and contained plurals of nouns and conjugations of verbs. Also, since the list was written in 1913, many of the words included in the list are antiquated and more modern terms (internet, email, etc.) are not present. This created problems in recognition and dictionary querying.

A word list called "5Desk" was found at http://wordlist.sourceforge.net/. "5Desk" was created as a solution to the antiquated UNIX dictionary. It was generated by comparing the entries of five dictionaries and including words based on multiple inclusion in the different dictionaries. This technique effectively filtered out the antiquated words and extraneous entries that made the UNIX dictionary problematic. It also reduced the size of the list to around 60,000 words. This size was large enough to generate an effective grammar, but not so cumbersome that it would cause recognition problems.

1.4 Technology

The original release of IDiOM was run off a shared server, this presented a few challenges. Due to server limitations, it was impossible to run the WordNet database locally. This meant that a creative method for extracting dictionary definitions was necessary. Once a suitable submit format was created, it was possible to submit query words to the Dict.org website for definitions. The next challenge would be to parse the website and delicately extract only the definition and other relevant data.

To parse the website the website's source code was studied for patterns. It was determined that the definition was always enclosed in <pre> tags, special HTML tags which print text as formatted in the source. Simply put, IDiOM is able to go through the source of a resulting query search and then return the relevant block of text. This however, also had its troubles. The returned text was just that, text. There wasn't any tagging of the different elements of the dictionary entry (part of speech,

```
<pre>
       adj : of or relating to or denoting numbers; "a numeral
             adjective"; "numerical analysis" [syn: numerical, numeric]
       n : a symbol used to represent a number; "he learned to write
           the numerals before he went to school" [syn: number]
</pre>
```

Excerpt of source code from Dict.org

synonyms, etc.). This made it especially difficult to later separate these parts. A clever use of PERL's regular expression matching (regexp) was used. Using expression matching meant that developers were able to intelligently split this block of unformatted text into clear divisions of parts of speech as well as into separate numbered definitions. Once an internal representation of a "page in the dictionary" was created and properly tagged, PERL's next job was to output this into a VXML file. This was one of the easier parts of implementation, and followed a straight flow thru model, where markup was written in a serial fashion. PERL was also able to generate the many JavaScript functions that are used throughout the VXML.

IDiOM, like many of the leading speech recognition systems out there, has trouble understanding individual letters. This is because so many letters sound similar (B, C, D, E, G, P, T, V, Z). This is a huge pitfall for a system that relies so heavily upon the spelling of words. To overcome this problem, IDiOM not only returns the letter it was most confident in hearing, but five other letters it thinks the caller might have said. This comes with its own problems. In returning so many letters to the PERL script, the dynamically created grammar became increasingly large. To remedy this problem, if the system recognizes a letter with a high enough confidence level (higher than .8), then it will only return that one letter to the script. This effectively decreases the size of the grammar, which in turn makes recognition of the user's word much easier.

```
confidence: 0.62
utterance: d
inputmode: voice
interpretation: {
letter="d"
}
length: 6
[0]:{
confidence: 0.62
utterance: d
inputmode: voice
interpretation: {
letter="d"
}
}
[1]:{
confidence: 0.44
utterance: t
inputmode: voice
interpretation: {
letter="t"
}
}
[2]:{
confidence: 0.44
utterance: v
inputmode: voice
interpretation: {
letter="v"
}
}
…
```

Example of N-Best List

This technique of returning multiple letters to the grammar-generating script had fringe benefits as well. It allowed instant T-9 input compatibility. T-9 input is a system that allows a caller to enter letters on their keypad one by one by typing the associated number only once. This means that the application automatically figures out which words are to be added to the grammar by running all possible combinations of letters and comparing those to the word list.

PERL was also crucial in creating the list of search strings that was matched against the wordlist. To create the comprehensive wordlist, each letter returned from either the DTMF or N-Best list was permutated with all other letters returned. This created a list of words that overcompensated for what the user said. This enormous number of permutations did not cause problems because when the list was filtered against the valid wordlist grammar and the string length requirement, 90% of the strings disappeared. This created a suitable grammar that was ready to handle pretty much anything the user could say. The depth of search is also fully scalable and can be changed at any time for a deeper or shallower N-Best letter lookup.

1.5 Usability Evaluations

1.5.1 Description

Formal and informal usability testing was performed in private and public settings. Usability testing was used to fine tune the application at all stages of development. Almost all minor usability flaws were immediately coded and fixed in the application. It is important to note that most users were able to able to find their word with no instruction from the testers. Users often expressed surprise and amazement that the application was not only functional but easy to use. No one, unless instructed to, had to use a help prompt. 90% of all user complaints were due to the fact that a word was not found. As the application evolved these complaints steadily declined. By the end of testing, there were very few cases of unfound words.

1.5.2 Usability Testing Problem Table

Problem Description	Solution	Technical Difficulty (1-5)	Usability Impairment (1-5)	Fixed?
Failure to recognize the word resulted in hangup.	Send user back to entry state.	2	4	Yes
Letters weren't recognized correctly.	Added N-best results to grammar generator.	4	5	Yes
Letters still weren't recognized correctly	Added depth to N-best results	1	3	Yes
User didn't know to say "done" when entering a three letter word.	Include done in initial prompts	1	1	No
Wording of TTS around definitions was awkward	Changed wording	2	2	Yes

1.6 Open Issues

IDiOM, as it exists today, is a completely functional and comprehensive dictionary. That does not mean that it is close to being done. The designers at IDiOM are constantly thinking creatively on how to make this application even better. The future looks bright for IDiOM. Below is a short list of where IDiOM could be in just a few short months.

1.6.1 Advanced Navigation/UI Features:

To make IDiOM soar, callers need to be able to get to the definition they want quickly. That means that IDiOM's navigational system needs to be revamped to offer faster access to all data provided. For instance, in the future a user may be able to speak the number of the definition they are looking for, or even jump to a particular part of speech. As a corollary to that, by adding a tone before each definition that has synonyms or usage examples, users would immediately know what they can ask for. This would alleviate the case of a caller asking for usage examples or synonyms when none are present.

Furthermore, the system relies heavily on barge-in. If you want a usage example of definition number three you must say "sentence" or "use in a sentence" during definition number three. In the future IDiOM will allow you to say both "sentence three" or "previous sentence" if you have already passed the intended definition. The system already has a next and previous system of navigation implemented. These core technologies would most likely affect repeat users who are familiar with the system and know what they are looking for more than novice users. In addition, hot word recognition will be implemented so that a caller could repeat the definition to another person as he or she hears it without barging in on the prompt.

Another intuitive feature would be for the system to use the synonym list to create a grammar of search terms. This would allow a caller listening to synonyms to instantly jump to another definition, letting the caller quickly get more information without having to input another word to the system.

1.6.2 Advanced DTMF/T-9 Word Input

Instead of relying upon only the first four letters, and then saying a word, IDiOM will soon allow a caller the option to enter the whole word in using DTMF. This would remove the necessity of the user saying the word in many cases, and would further improve the recognition of the system by limiting the grammar to very few possible words. This would also remove any remaining problems IDiOM still has concerning homonyms. For example, the words "Sunday" and "sundae" would never be confused when using T-9 input.

1.6.3 Advanced Database Logging and Web Interface

A web interface for IDiOM is already being designed and implemented. The interface will allow users to view logs of recently searched words and their definitions. It will also incorporate the word-linking technology of WordNet to allow users to explore other words connected to the searches they have made. The database will be automatically generated using caller-ID information, so that the user can log in using their 10 digit phone number on the IDiOM website. This logging technology will also be used in the voice application, allowing users the option to say or choose from a list of recently searched words at the beginning of the call to rehear definitions.

1.6.4 Advanced Language Resources

IDiOM already provides the mechanism for delivering linguistic content. The next step is allowing users to listen to antonyms, hear spellings, and use the other language resources provided by the WordNet2.0 dictionary. Furthermore, the word input system could be used to query other linguistic databases providing the caller with foreign language dictionaries, encyclopedia entries, search engine results, and more. The possibilities are limitless.

1.6.5 Commercial Implications

Eventually IDiOM technology could be ported to suit the needs of different dictionary makers. Perhaps someday, every dictionary will be sold with an accompanying 800-number for mobile access.

1.6.6 Fun Features

Fun features could soon find their way into IDiOM. These could include, but are not limited to, a word of the day, as well as a spelling bee function. These features would show that a speech interface can be just as complete as its visual equivalents. In fact, a spelling bee function is something IDiOM's paper and online counterparts could never successfully do.

1.7 Call Examples

1.7.1 A User Gets a Definition and a Sentence

Speaker	Dialogue	Notes
System	Welcome to IDiOM! Let's get started.	
System	Please say or enter the first letter of the word you're searching for.	Word is spelled to narrow down grammar choices.
Caller	B.	
System	Now the second letter.	
Caller	L.	
System	Now the third.	
Caller	A.	
System	And the fourth.	
Caller	D.	
System	Now, say the word you're looking for.	Grammar dynamically generated based on spelling of the word.
Caller	Blade.	
System	Did you say blade?	Word selection confirmed.
Caller	Yeah.	
System	There are nine definitions for blade. At any time during a definition you can say "synonyms" or "use in a sentence".	
System	As a noun, 1: especially a leaf of grass or the broad portion of a leaf as distinct from the petiole	
System	2: a dashing young man	
System	3: something long and thin resembling…	
Caller	(barge) Previous.	Navigation commands to move between definitions.
System	2. a dashing young…	
Caller	(barge) Sentence.	
System	Gay young blades bragged of their amorous adventures.	Contextual example of definition.
System	Definition 2: a dashing young…	

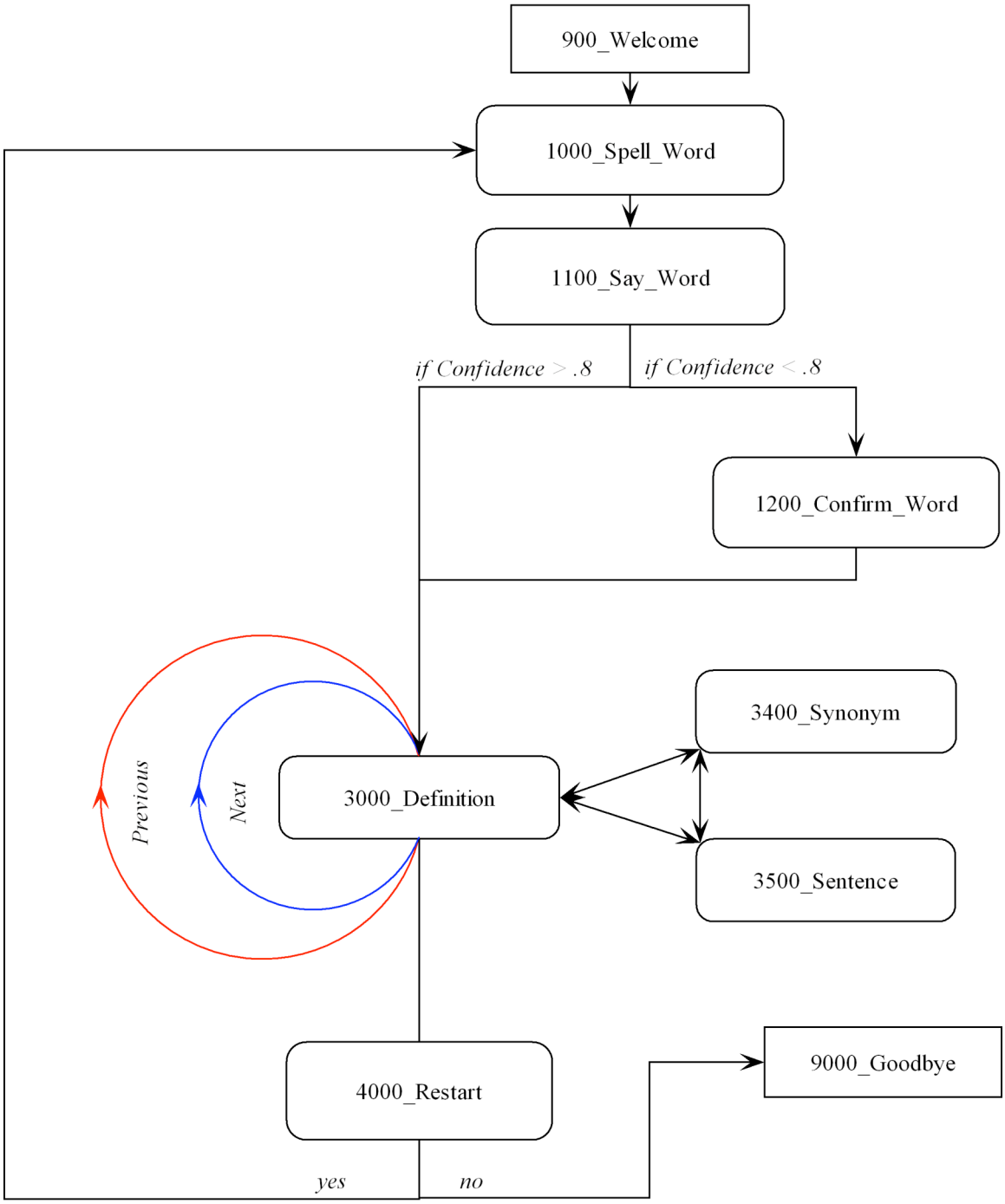
Caller	(hang up)	

1.7.2 DTMF Entry for a Speedy Word Lookup

Speaker	Dialogue	Notes
System	Welcome to IDiOM! Let's get started.	
System	Please say or enter the first letter of the word you're searching for.	
Caller	(DTMF)3…5…3…7.	Optional DTMF letter entry for speed and accuracy.
System	Now say the word you are looking for.	
Caller	Elephant	
System	There are 2 definitions for elephant.	
System	At any time during a definition you can say "synonyms" or "use in a sentence".	
System	As a noun: 1: five-toed pachyderm	
System	2: the symbol of the Republican Party; introduced in cartoons by Thomas Nast in 1874	
System	There are no more definitions for elephant.	
System	Would you like to look up another word?	
Caller	(hang up)	

Chapter 2: Application Behavior

2.1 Call Flow



2.2 State Tables

900_Welcome

Play Prompt
Entering from
Incoming Call

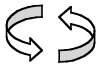
Prompts

Name	*Condition*	*Wording*
9001	*Always*	Welcome to IDiOM! Let's get started.

Condition	*Action*
Always	GOTO: 1000_Spell_Word

Module Settings
No barge-in

1000_Spell_Word



Speech Input
Entering from
900_Welcome, 4000_Restart

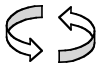
Prompts

Type	*Name*	*Wording*
Initial	10001	Please say or enter the first letter of the word you're searching for.
	10002	Now the second letter.
	10003	Now the third.
	10004	And the fourth.
Timeout 1	10005	Say the first letter, or enter it on your phone's keypad. For more information say "help".
Retry 1	10006	I'm sorry I didn't understand you. Please repeat the letter or enter on your phone's keypad. For more information say "help".
Help	10007	You are using IDiOM, the Interactive Dictionary Output Module. The first step in finding your definition is spelling the first four letters of the word you are looking for. If you don't know how to spell the first four letters, just spell as much of it as you can and then stop spelling. If your word is less than four letters, say "done" when prompted for the next letter. You can either say or enter each letter on your phone's keypad when asked. For example, to enter the letter H, you could press the 4 key one time or say "H". Let's try it.

Option	*Vocabulary*	*DTMF*	*Action*	*Confirm.*
Letter	`Alphabet`	Yes	If < 4 letters GOTO: 1000_Spell_Word If 4 letters GOTO: 1100_Say_Word If > 2 letters AND no input GOTO: 1100_Say_Word	*Never*
Done	`Done`		If "done" GOTO: 1100_Say_Word	*Never*

Commands
See default settings, as specified in ""

1100_Say_Word



Speech Input

Entering from

1000_Spell_Word

Prompts

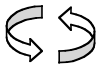
Type	*Name*	*Wording*
Initial	11001	Now, say the word you're looking for.
Timeout 1	11002	Please say the word you are looking for or say "help" for more information.
Retry 1	11003	I'm sorry, I didn't understand you. Please say the word that you are looking for again.
Retry 2	11004	I'm sorry; I'm just not hearing you well. It might help me figure out what you're trying to say if you spell the word again. If you're in a loud environment, try typing the letters on your phone's keypad instead of saying them.
Help	11005	The next step in finding your definition is saying the word you are looking for. If you don't know exactly how to pronounce your word, say it as best as you can.

Option	*Vocabulary*	*DTMF*	*Action*	*Confirm.*
Word	(Dynamically Generated)	No	If confidence < .8 GOTO: 1200_Confirm_Word ELSE GOTO: 3000_Definition	*If necessary*

Commands

See default settings, as specified in ""

1200_Confirm_Word



Speech Input

Entering from

1100_Say_Word

Prompts

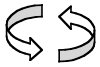
Type		*Name*	*Wording*
Initial	First Word	12001	Did you say…
		12002	*<First N-best Result>*?
	Second Word	12003	How about…
		12004	*<Second N-best Result>*?
	Third Word	12005	*<Third N-best Result>*?
	Fourth Word	12006	*<Fourth N-best Result>*?
Timeout 1		12007	Did you say…
			<Current N-best result>?
Retry 1		12008	Did you say…
			<Current N-best result>?
Help		12009	The computer has narrowed down your word to a few possibilities. You will now be asked whether or not you said each one of these possibilities. Say "yes" to confirm that you said a word, or "no" to hear the next word.
Respell	If no words confirmed	120010	It might help me figure out what you're trying to say if you spell the word again. If you're in a loud environment, try typing the letters on your phone's keypad instead of saying them.

Option	*Vocabulary*	*DTMF*	*Action*	*Confirm.*
Confirm	Boolean	No	If "true" GOTO: 3000_Definition If "false" GOTO: 1200_Confirm_Word	*If necessary*

Commands
See default settings, as specified in ""

Module Settings

3000_Definition



Speech Input

Entering from
1100_Say_Word, 1200_Confirm_Word, 3000_Definition, 3400_Synonyms, 3500_Sentence

Prompts

Type		Name	Wording
Initial	In no definitions found	30001	I'm sorry I couldn't find a definition for
		30002	*<the word>*
	Always, if definitions are found.	30003	There are
		30004	*<definition count>*
		30005	Definitions for
		30006	*<the word>*
		30007	At any time during a definition you can say "synonyms" or "use in a sentence".
	Before noun definitions	30008	As a noun
	Before adjective definitions	30009	As an adjective
	Before adverb definitions	300010	As an adverb
	Before verb definitions	300011	As a verb
	After last definition	300012	There are no more definitions for
		300013	*<the word>*
		300014	Would you like to look up another word?
Definition	Always, if definitions are queued	300015	*<parsed definition from WordNet>*
Timeout 1		300016	GOTO: 3000_Definition, next definition
Retry 1		300018	GOTO: 3000_Definition, repeat definition
Help		300019	You're listening to definitions for
		300020	*<the word>*
		300021	You can navigate through these definitions by saying "previous" or "next", or get more information on a particular definition by saying "use in a sentence" or "synonyms" at any time.

Option	Vocabulary	DTMF	Action	Confirm.
Synonym	`Synonym`	No	GOTO: 3400_Synonyms	*If necessary*
Sentence	`Sentence, use in a sentence`	No	GOTO: 3500_Sentence	*If necessary*
Previous	`Previous`	No	GOTO: 3000_Definition, previous definition	*If necessary*
Next	`Next`	No	GOTO: 3000_Definition, next definition	*If necessary*

Commands
See default settings, as specified in ""

Module Settings

3400_Synonyms

Play Prompt

Entering from
3000_Definition

Prompts		
Name	**Condition**	**Wording**
Synonym	*Always*	*<parsed synonyms from WordNet>*

Condition	Action
Always	GOTO: 3000_Definition, currently queued word

Module Settings
No barge-in

3500_Sentence

Play Prompt

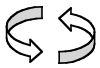
Entering from
3000_Definition

Prompts		
Name	**Condition**	**Wording**
Sentence	*Always*	*<parsed sentence from WordNet>*

Condition	Action
Always	GOTO: 3000_Definition, currently queued word

Module Settings
No barge-in

4000_Restart



Speech Input

Entering from
3000_Definition

Prompts		
Type	**Name**	**Wording**
Initial	40001	There are no more definitions for
	40002	*<the word>*
	40003	Would you like to look up another word?

Option	Vocabulary	DTMF	Action	Confirm.
Choice	`Boolean`		If true GOTO: 1000_Spell_Word If false GOTO: 9000_Goodbye	*If necessary*

Commands
See default settings, as specified in ""

Module Settings

9000_Goodbye

Play Prompt

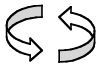
Entering from
4000_Restart

Prompts		
Name	**Condition**	**Wording**
Goodbye	*always*	Thank you for using IDiOM.

Condition	Action
Always	Disconnect

Module Settings
No barge-in

COMMANDS



Speech Input	Command Vocabulary

Commands	DTMF	Action (unless re-specified by the module)	Confirm.
"Help"	no	Play the Help prompt as specified for the module.	*Never*

Confirmation Prompts				
Command	**Name**	**Wording**	**Name**	**Synonym Wording**
No confirmation prompts needed				

Chapter 3: Recording Script

Begin Script

9001

Welcome to IDiOM! Let's get started.

10001

Please say or enter the first letter of the word you're searching for.

10002

Now the second letter.

10003

Now the third.

10004

And the fourth.

10005

Say the first letter, or enter it on your phone's keypad. For more information say "help".

10006

I'm sorry I didn't understand you. Please repeat the letter or enter it on your phone's keypad. For more information say "help".

10007

You are using IDiOM, the Interactive Dictionary Output Module. The first step in finding your definition is spelling the first four letters of the word you are looking for. If you don't know how to spell the first four letters, just spell as much of it as you can and then stop spelling. If your word is less than four letters, say "done" when prompted for the next letter. You can either say or enter each letter on your phone's keypad when asked. For example, to enter the letter H, you could press the 4 key one time or say "H". Let's try it.

11001

Now, say the word you're looking for.

11002

Please say the word you are looking for or say "help" for more information.

11003

I'm sorry, I didn't understand you. Please say the word that you are looking for again.

11004

I'm sorry; I'm just not hearing you well. It might help me figure out what you're trying to say if you spell the word again. If you're in a loud environment, try typing the letters on your phone's keypad instead of saying them.

11005

The next step in finding your definition is saying the word you are looking for. If you don't know exactly how to pronounce your word, say it as best as you can.

12001

Did you say [watermelon]?

12003

How about [apple]?

12008

The computer has narrowed down your word to a few possibilities. You will now be asked whether or not you said each one of

these possibilities. Say "yes" to confirm that you said a word, or "no" to hear the next word.

30001

Here is the listing for

30002

As a noun

30003

As an adjective

30004

As an adverb

30005

As a verb

30006

As a pronoun

30007

Definition [one]:

30008

You can say "next", "use in a sentence", or "synonyms"

30009

Say "previous", "next", "use in a sentence", or "synonyms"

300010

There are no more definitions for [strawberry].

300011

Would you like to look up another word?

300012

I'm sorry, I couldn't find a definition for [kiwi].

300015

You're listening to definitions for [mango].

300016

You can navigate through these definitions by saying "previous" or "next", or get more information on a particular definition by saying "use in a sentence" or "synonyms" at any time.

300017

At any time during a definition you can say "synonyms" or "use in a sentence".

300018

Synonyms are [tangerine, clementine, and orange].

900001

Or, for more information, say "help".

End Script

Appendix

Call in information:

For the most up-to-date information regarding the IDiOM project, please visit:
http://www.cs.tufts.edu/~msparand/vxml/idiom

To call into IDiOM, please call:
1.877.33.VOCAL (1 877 338 6225)
User ID: 2456751
Pin: 1234