

Agilingua, LLC.

アトム音声対話 SDK を使った開発プロセス

日本語版

Agilingua, LLC.

8/15/2010

コンテンツ

概要	3
ステップ1: データ収集	5
ステップ2: トランスクリプション	6
ステップ3: 文法開発	7
ステップ4: ダイアログ開発	8
ステップ5: ユーザビリティテスト	9
文法開発について	10
統合プログラムの開発	14

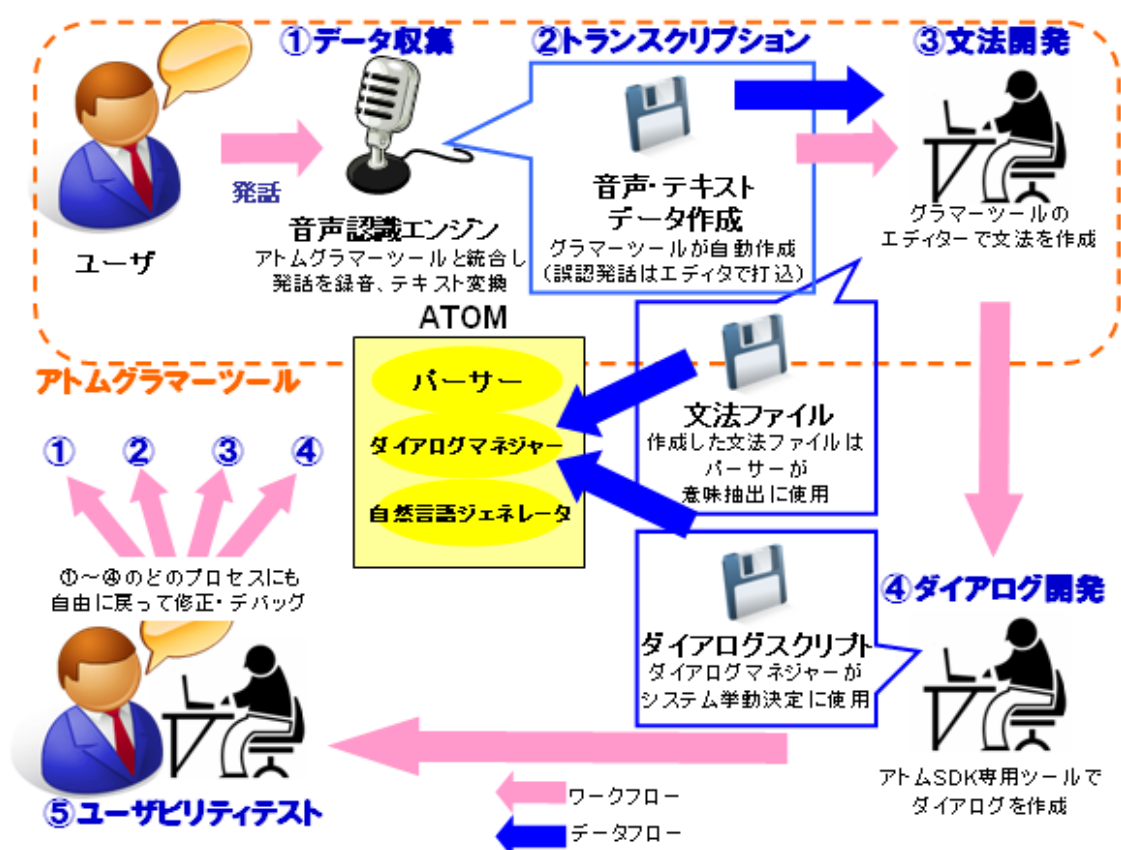
アトム音声対話 SDK を使用した開発プロセス

概要

アトム音声対話 SDK は、データ収集からテスト、デバッグまでのすべての開発サイクルを統括的に支援し、開発作業を簡易化、工数を削減する独自の CDD (Comprehensive Dialogue Development) システムを採用しています。

■ アトム音声対話 SDK を使った開発プロセス

音声対話システムの開発には一般に ①データ収集、②トランスクリプション ③文法開発、④ダイアログ開発および ⑤ユーザビリティテスト の 5 つのプロセスが必要です。CDD はそれらすべてを統括してサポートします。下の図は CDD のワークフローを図示したものです。



* 図中オレンジ色の点線は、アトムグラマーツールがサポートするプロセス(1:データ収集、2:トランスクリプション、3:文法開発)を表しています。アトムグラマーツールはアジリングアのすべての SDK に含まれておりますが、単独でも販売している商品です。

ステップ1:データ収集

データ収集とは、システム開発、テスト、チューニングを目的に 自然音声言語データを収集するプロセスです。

ステップ2:トランスクリプション

トランスクリプションは録音した音声などの音声言語ソースをテキストに打ち込むプロセスです。

ステップ3:文法開発

文法開発は、聴き取る語彙および語彙パターンを音声認識装置が特定するプロセスです。

ステップ4:ダイアログ開発

ダイアログ開発は、ダイアログマネジャーのとるべきアクションを特定するプロセスです。

ステップ5:ユーザビリティテスト

ユーザビリティテストおよびチューニングは、会話ダイアログアプリケーションを逐次改良する反復作業です。文法開発およびダイアログ開発で規定したスペックの改良が目的です。

Step 1: データ収集

データ収集とは、システム開発、テスト、チューニングを目的に自然音声言語データを収集するプロセスです。つまり、開発するアプリケーションが出来上がったときにはユーザがどんなことをシステムに話さだろうかを予想し、その発話をどんどん録音していく作業です。データ収集は開発過程の当初のプロセスなので実行システムはありません。よってユーザ(ダイアログシステムのユーザ)に知られずに人的にダイアログシステムを擬態します(ウィザートオブオズ The Wizard of Oz より、これをウィザートオブオズシミュレーションと言います。このウィザートとはカーテンの後ろにかくれた人物によるシステム擬態を指しますが、以下、このシステム擬態をする人物をウィザードと表します)。

- データ収集のツール

アトム SDK を使ったデータ収集には二機のネットワークコンピュータが必要です。音声発話を記録するコンピュータにアトム SDK 内のデータ収集ソフトをインストールします。ウィザードは、記録されたユーザの音声発話を聴取、もしくはテキストを音声に変換するシステムによる音声発話の聴取というかたちで ユーザとインタラクションを確立します。データ収集ソフトはウェブサーバの機能を果たし、ウィザードからのインタラクションはすべてウェブブラウザを通して行います。

- データ収集プロセスの結果:

ステップ 1 の結果が、ユーザとウィザードとのインタラクションのコーパスとなります。インタラクションは保存し、そのタイムスタンプを記録します。ログファイルは XML フォーマットに保存し、標準ウェブブラウザで開けるようにします。ユーザの音声発話は記録し、ログファイルに対応させて保存します。記録した音声は音声認識装置の学習に、もしくはトランスクリプト後の文法開発に使用します。

- データ収集のサポート標準と API:

- EMMA:EMMA (Extensible MultiModal Annotation markup language) はユーザからの入力イベントを表すのに使用します。
- SAPI 4.0 および 5.1:SAPI 4.0 もしくは SAPI 5.1 対応音声合成エンジンはウィザードからのテキストを合成するのに使用します。

Step 2: トランスクリプション

トランスクリプションは録音した音声などの音声言語ソースをテキストに打ち込むプロセスです。トランスクリプト音声(トランスクリプションをともなう録音音声)は音声認識装置の音響モデルの学習に使用します。また、トランスクリプション単体は音声認識装置の言語モデルの学習、音声認識装置のグラマー開発、発話の理解に使用します。

- トランスクリプションのツール

トランスクリプションのツールは録音した音声ファイルに注釈を加えるのに使います。注釈はログファイルに保存します。

- トランスクリプションの結果:

トランスクリプションの結果が、録音した音声をトランスクリプトした段階でのユーザとウィザードとのインタラクションのコーパスとなります。トランスクリプションはログファイルに保存します。

トランスクリプト後のログファイルには音声対話システムの開発に必要なデータがすべて含まれ、これらは 文法開発 (step 3)、ダイアログ開発(step 4) および ユーザビリティテスト (step 5) に必要な各ツールに利用されます。

- トランスクリプションのサポートスタンダードと API:

- EMMA: EMMA (Extensible MultiModal Annotation markup language)は データ収集(step 1)で収集したユーザからの入力イベントを表すのに使用します。トランスクリプションツールは、`<emma:interpretation>`タグ以下に直接トランスクリプションを挿入します。

Step 3: 文法開発

文法開発は、聴き取る語彙および語彙パターンを音声認識装置が特定するプロセスです。また、自然言語理解ユニットは、認識されたテキストを発話の意味したところである意味表示に変換する際に意味注釈をとまなう文法を使用します。文法は、互換性のある様々なフォーマットに対応します。これらフォーマットは文脈自由文法を共通とします。アトム 音声 SDK は各文法フォーマットをサポートします。文法作成は、収集した自然発話言語と文法を一致させる反復作業です。文法と発話がまったく、もしくは部分的に一致しない場合は文法ルールを改定し、収集した発話がすべて文法に一致するようにします。文法開発についてはこの資料の 10 ページから詳しい説明があります。

- 文法開発のツール:

アプリケーションの開発者はアトム ボイスツールキット内のアトムグラマーツールを使い、step 1: データ収集 で製作したログファイルから発話を取り出すことができます。ログファイル内の文字カウント、ログファイル内容に対するテスト文法、カバレッジを求めることが可能です。認識されたテキストを意味表現に変換するため、文法に意味的注釈を加える必要があることがあります。

- 文法開発の結果:

文法開発の結果として、ひとつあるいは複数の文法ファイルが生成されます。文法ファイルは step 4: ダイアログ開発 のダイアログスクリプトに対応します。音声認識のため、開発した文法を N-Gram モデルに変換することもできます。

- 文法開発のサポートスタンダードと API:

- ECMA スクリプトコンパクトフレームワーク: ECMA スクリプトコンパクトフレームワークは、文法の意味解釈部分を特定するのに使用します。
- JSGF (Java Speech Grammar Format): JSGF は文法を表記できるフォーマットのひとつです。
- SISR (Semantic Interpretation for Speech Recognition): SISR は文法に注釈を加えることのできるフォーマットです。
- SRGS (Speech Recognition Grammar Specification) : SRGS は文法を表記できるフォーマットのひとつです。XML と ABNF がサポートされます。

- 専用拡張

以下の専用拡張で文脈自由文法と意味的ウェブスタンダード RDFS の表記を組み合わせることができます。

- RDFS 文法注釈: SISR に加え、RDFS の文法注釈もサポートされます。
- 意味文法: 意味文法の終端記号は RDFS で注釈を加えることができます。これによって意味文法をコンパイルチェックできます。このプロセスは XML スキーマや DTD を使った XML マークアップチェックと類似しています。スキーマプロセッサがスキーマ障害を提示するように、グラマープロセッサは RDFS スペックに障害が発生した際にそれを提示します。特に大きな文法を開発する際、これで開発に要する時間が短縮されます。

Step 4: ダイアログ開発

ダイアログマネジャーとは、ユーザの発話に対しシステムがどうリアクションを起こすか(音声で返答、回答、提案などをする、もしくはなんらかの挙動を起こす、など)を決定する、対話システムの頭脳ともいえる機構です。ダイアログ開発は、このダイアログマネジャーのとりべきアクションを規定するプロセスです。これらアクションは音声認識入力、タイムアウト、低信頼値などへの反応で特定されます。

- ダイアログ開発のツール:

アプリケーション開発者はアトム ボイスツールキット内のツールを使い、ECMAScript の専用版にダイアログスクリプトを書き込むことができます。アプリケーション開発者はこの専用版を利用してダイアログマネジャーが順守すべきアクションや条件を規定し、また文脈依存文法を対応させます。

SQL に互換性を持つデータベースはダイアログマネジャーと統合させます。自然言語理解のプロセスおよびその認識に由来する意味的解釈は、簡便に SQL クエリに変換することができます。

- ダイアログ開発の結果

ダイアログ開発の結果、ひとつまたは複数のダイアログスクリプトファイルが生成されます。ダイアログスクリプトファイルおよび文法ファイルは、モジュラーアプリケーション開発のためにまとめておきます。

- ダイアログ開発のサポートスタンダードと API:

- ECMA スクリプトコンパクトフレームワーク: ECMA スクリプトコンパクトフレームワークは、ダイアログプロセスアルゴリズムを特定するのに使用します。その専用フレームワークは、アクションおよび条件を規定するのに使用します。

- SAPI 4.0 および 5.1: SAPI 4.0 もしくは SAPI 5.1 対応音声合成エンジンは、ダイアログマネジャーが生成したテキストを合成するのに使用します。

- SQL データベースのサポート: SQL データベースはダイアログマネジャーから直接アクセスできます。ここで得たデータセットはテキスト化され読み上げることができます。

- 専用拡張

専用拡張を利用し、意味的解釈プロセスのアウトプットと条件を一致させます。この条件とアクションは専用 ECMAScript を使用して規定します。

Step 5: ユーザビリティテスト

ユーザビリティテストおよびチューニングは、会話ダイアログアプリケーションを逐次改良する反復作業(テスト・デバッグ)です。step 3: 文法開発および step 4: ダイアログ開発で規定したスペックの改良とシステムを安定させることが目的です。ユーザビリティテストの際、ユーザはダイアログシステムがすでに完成しているものと想定してこれと対話します。すべてのアクションを記録します。

- ユーザビリティテストのツール

アプリケーション開発者が問題あるインタラクションを確認できるよう、アトム 音声対話 SDK に付属のツールでログに注釈を加えます。

- ユーザビリティテストの結果

ユーザビリティテストの結果、複数のログファイルが生成されます。これらログファイルは音声認識装置からのトランスクリプションを含んでいることを除き、Step 1: データ収集 の際に生成したファイルと類似します。正しく処理できないインタラクションもここで注釈され、確認されます。

- ユーザビリティテストのサポートスタンダードと API:

- EMMA: EMMA (Extensible MultiModal Annotation markup language)はユーザからの入力イベントを表すのに使用します。

文法開発について

- 音声認識および自然言語理解のための文法開発

文法開発(Grammar Development)とは、ユーザの音声発言を音声認識ソフトウェアに言葉として認識させるためのものです。DVDレコーダーを音声で対話的な操作をするための文法を例に考えてみましょう。ユーザが以下のような音声コマンドを発することをDVDレコーダーは理解できなければならないとします。

「今晚'アグリーベティ'録画しといて」

「今日の'アグリーベティ'を録画してください」

- 文法フォーマット

文法はルールに従って作成され、機能します。文法のルールは、ユーザが話す(だろうと予測される)フレーズのリストのようなものと考えると理解しやすいでしょう。上記の例は文法のルールには以下のように表記できます。

```
<RecordShow> = 今晚'アグリーベティ'録画しといて  
                | 今日の'アグリーベティ'を録画してください
```

;

ここで、ふたつめのフレーズの頭にある縦線|は選択肢であることを意味します。つまり、ユーザが上の言い方、下の言い方のどちらで話しても認識ソフトは同じ意味として理解する、と言うことをあらわします。ここで多様な言い方を入力することで、ユーザが制限のない、自然で自由な話し方をできるようになります。

- ルールを参照させる(ルール・リファレンス)

DVDレコーダーとして、ひとつの番組だけでなく他のテレビ番組も録画できる機能が必要です。そのために、すべてのテレビ番組の名前を網羅したリスト、<ShowName>を新しく作成し、<RecordShow>のルールから参照(リファレンス)させます。

```
<RecordShow> = 今晚 <ShowName> 録画しといて  
                | 今日の<ShowName>を録画してください
```

;

```
<ShowName> = アグリーベティ  
            | グレイズ・アナトミー
```

| ヒーローズ

| ...

;

- 文法を安定させる・自由な発話に対応

ユーザがそれぞれに異なる言い方を自然に自由に話して音声操作できるようにするため、文法を豊かに作成しさまざまな言い方に対応できるように開発することがロバストな音声アプリケーション開発に重要です。たとえば、「・・・を録画して欲しい」と言う意味のさまざまな発話に対応させるには以下のように文法を作成します。

<RecordShow> = 今晚 <ShowName> <WantRecord>

| 今日の<ShowName><WantRecord>

;

<ShowName> = アグリーベティ

| グレイズ・アナトミー

| ヒーローズ

| ...

;

<WantRecord> = '録画' 'しといて'

| 'を' '録画' 'して' 'ください'

| 'を' '録画' 'して' 'ね'

| '録画' 'しとけ'

| ...

;

- コマンド追加

文法にはさらに多くのコマンドを追加することができます。たとえば、以下の<Please>の文法を加えることでさらに多様な言い方で録画を命令することができるようになります。

<WantRecord> = '録画'

| '録画' <Please>

| ...

;

<Please> = お願い

| お願いします

| をお願いね

| をして

| をしてください

| しといて

| ...

;

- 文法のチューニング

大きな文法を作成していくうちに、タグのよっては非文法的なコマンドになったりすることがあります。例えば録画予約の一覧を見たいときのために以下のような文法を作成したとします。

<ShowReserve> = 録画予約みせて

| 録画予約をみせてください

| 録画予約<Please>

;

この場合、「録画予約お願いします」などのユーザ発話については正しく録画予約番組の一覧を表示するためのコマンドとして機能しますが、「録画予約して」、「録画予約しといて」などの発話はユーザがある番組を録画するための予約をしたい意味などで使われることもあり、そのために作成したコマンド文法と重複が考えられ、文法的にこのコマンドには不適となります。

アトムグラマーツール には、このような文法エラーなどを自動的に検索、表示する機能がついており、文法作成の迅速化、作業の軽減に役立ちます。

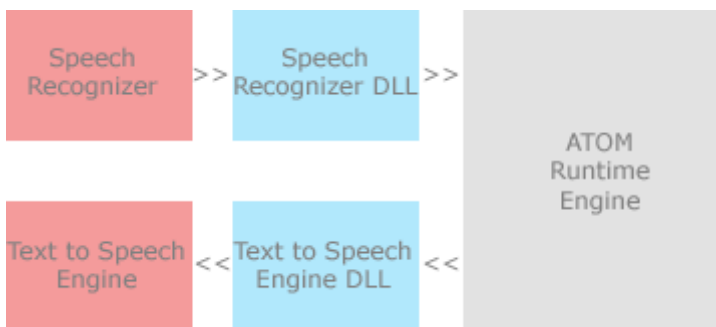
注：この例の文法フォーマットは解説をわかりやすくするため簡易化されています。

統合プログラムの開発

アトムランタイムエンジンには、SAPI 5.1 対応の音声認識、音声合成ソフトとの統合プログラムがすでに含まれています。また、アトムランタイムエンジンはニーズに合わせて任意の音声ソフトウェアと統合できるように C++ 言語で作成されている API が含まれており、簡便に統合プログラムを作成することによって任意の C または C++ API 対応音声認識、音声合成ソフトとも統合できます。

アトムランタイムエンジンを任意の音声認識、音声合成ソフトと統合するため、アトム API コールを統合する音声ソフトの API コールにあわせて (もしくは音声ソフトの API コールをアトム API コールにあわせて) 変換する必要があります。プログラムコードは DLL (Dynamic Link Library: ダイナミックリンクライブラリ) として変換に関与します。アトム音声対話 SDK およびレファレンス SDK には統合プログラム作成のためのサンプルプロジェクトが含まれており簡便ですばやい統合プログラムの作成を支援します。

こちらは DLL がアトムランタイムエンジンと音声認識・合成ソフトとの統合にどのように使用されるか図示したものです。



- 音声認識ソフトとの統合

アトムと音声認識ソフトとの統合において、DLL は以下のような機能への支援に関与します:

- * 音声認識ソフトのインスタンスの作成と破棄
- * ランタイムにおける文法の動的読み込み
- * 音声認識ソフトの起動と停止
- * 音声認識ソフトからアトムランタイムエンジンへのイベント送信
- * N-ベクトリストおよび信頼値を含む認識結果のアトムへの送信

- 音声合成ソフトとの統合

アトムと音声合成ソフトとの統合において、DLL は以下のような機能への支援に関与します:

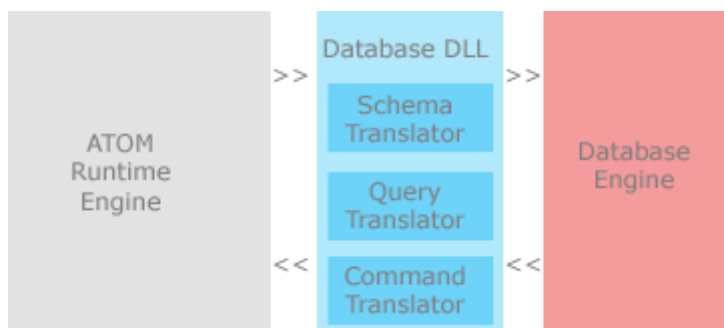
- * 音声合成ソフトのインスタンスの作成と破棄

- * 音声合成ソフトの起動と停止
- * 音声合成ソフトからアトムランタイムエンジンへのイベント送信

- データベースエンジンとの統合

アトムランタイムエンジンは、Microsoft 社の Access および SQLite データベースとはすでに統合されています。また、アトムランタイムエンジンは任意の C または C++ API 対応のデータベースエンジンとも統合できます。

アトムランタイムエンジンを任意のデータベースエンジンと統合するため、アトム API コールを統合するデータベースエンジンの API コールにあわせて(もしくはデータベースエンジンの API コールをアトム API コールにあわせて)変換する必要があります。プログラムコードは DLL(Dynamic Link Library:ダイナミックリンクライブラリ)として変換に関与します。アトム音声対話 SDK およびレファレンス SDK には統合プログラム作成のためのサンプルプロジェクトが含まれており簡便ですばやい統合プログラムの作成を支援します。



こちらは DLL がアトムランタイムエンジンとデータベースエンジンとの統合にどのように使用されるか図示したものです。クエリ、スキーマおよびコマンドトランスレーターはアトム API からのコールをデータベースの API に(またはその逆に)変換するために使用されます。

- データベースエンジンとの統合

アトムとデータベースとの統合において、DLL は以下のような機能への支援に関与します:

- * データベースエンジンのインスタンスの作成と破棄
- * クエリの実行、コマンドの挿入と削除、およびスキーマの作成

この資料やアトム音声対話 SDK に関するご質問までEメールにてお問い合わせください。

Agilingua, LLC.

24-65 38th Street #5D Astoria, NY 11103 USA

EMAIL: info@agilingua.com

WEBSITE : www.agilingua.com

FAQ: <http://www.agilingua.com/jp/qa/overview.php>

Agilingua, LLC. All rights Reserved. 2004-2010

2010 年 8 月 15 日改定

###