



Parla! Projetando em AutoCAD e Revit através de comandos falados.

José Luis Menegotto – Escola Politécnica da Universidade Federal de Rio de Janeiro

A palestra tem por objetivo mostrar como integrar as tecnologias TTS (*Text to Speech*) e de Reconhecimento de voz (*Speech Recognizer*) em plataforma .NET para criar aplicativos com interface de comandos falados. Será explicada como construir uma gramática de comandos e ações específicas. Serão mostrados exemplos de comandos por fala desenvolvidos para AutoCAD e Revit. No primeiro operando sobre entidades geométricas abstratas e no segundo sobre um modelo com elementos dentro do domínio da construção civil.

Objetivo de aprendizado

Ao final desta palestra você terá condições de:

- Integrar sistemas de voz aos seus aplicativos.
- Pensar cenários em que a tecnologia pode ser aplicada.
- Aproveitar a interface de voz para extrair informação de modelos complexos

Sobre o Palestrante

Arquiteto formado pela Facultad de Arquitectura Diseño y Urbanismo da Universidad Nacional de Buenos Aires, Argentina (1989). Estudou Desenho Assistido por Computador na cátedra do Arq. Arturo Montagú, no CentroCAO da FADU-UBA. Doutorado em Arquitetura e Urbanismo pela Universidade Federal de Rio de Janeiro (2009), para o qual desenvolveu o aplicativo Caixa de Música que traduz formas geométricas em formas sonoras e musicais. Foi professor de desenho na Pontifícia Universidade Católica de Rio de Janeiro PUC-Rio entre 1999 e 2009. Atualmente é professor do Departamento de Expressão Gráfica e do Programa de Pós-graduação em Engenharia Urbana PEU da Escola Politécnica da UFRJ. Incorporou o conceito de BIM e o aplicativo Revit como temas de pesquisa. Desenvolveu o aplicativo Cadtools arquitetura para AutoCAD (Registrado no INPI em 1991). Atuou como consultor na área de CAD desenvolvendo aplicativos destinados ao projeto de arquitetura e engenharia, como o DuctilCAD e PredialCAD para a empresa Saint-Gobain Canalização. Realizou diversos trabalhos nos campos da modelagem tridimensional e visualização. Ganhou o 1° prêmio TecnaCAD '93 com o modelo da Opera de Sydney. Autor do Livro O desenho digital. Técnica e arte. Interciencia, 2000.

jlmnegotto@poli.ufrj.br



Introdução

Duas técnicas que relacionam o campo da Linguística com o da Tecnologia da Informação têm chamado a atenção de pesquisadores desde a década de 1950. São elas as técnicas de reconhecimento automático de fala (ASR) *Automatic Speech Recognition* e a conversão de texto em voz (TTS) *Text to Speech* (Anusuya *et al.* 2009). Os programas de reconhecimento automático de fala (ASR) ou, simplesmente, reconhecimento de voz, procuram definir funcionalidades que confirmem ao computador a capacidade de reconhecer as locuções emitidas por um agente humano.

No sentido inverso, programas TTS procuram conferir ao computador a capacidade de traduzir informação ingressada como texto a sons sintetizados que simulam vozes humanas. No estágio de desenvolvimento atual existem bibliotecas de APIs que colocam a disposição esse tipo de funcionalidades e que podem ser integradas aos programas AutoCAD e Revit, como a todos os aplicativos que utilizem o Net Framework como plataforma de desenvolvimento.

Desde uma perspectiva tradicional da disciplina de Desenho e Projeto Assistido por Computador (CADD) pode-se observar a necessidade crescente dos projetistas contarem com a assistência de ferramentas e interfaces complementares que lhes permitam manipular modelos cada vez mais complexos, sem correr riscos de produzir alterações acidentais. A tarefa de agregar e gerenciar os objetos construtivos no modelo, é uma tarefa que demanda muito cuidado. A integração da base de dados gráfica em aplicativos como Revit, que permite a manipulação de todas as informações geométricas e lógicas do projeto, pode ser apontada como uma vantagem em relação aos aplicativos CAD tradicionais. Podemos, no entanto, enxergar uma dificuldade nessa possibilidade integradora, decorrente da dificuldade de se manterem constantes os níveis de concentração mental necessários para gerenciar com eficiência um banco de dados que torna-se cada vez maior e mais rico em vínculos. As interfaces por voz permitem não “tocar” no modelo para operar sobre ele.

Paralelamente ao projeto colaborativo, tornou-se real a possibilidade de um único profissional, altamente qualificado nas diversas áreas, projetar ou assumir para si várias das disciplinas de uma obra complexa. Pensamos que essa forma de encarar o projeto exige interfaces e ferramentas que liberem o projetista da fadiga mental que significa ter de lidar com o modelo, memorizando, operando e associando comandos específicos e procedimentos de projeto. As interfaces por voz parecem ser uma boa alternativa para ajudar a reduzir essa fadiga. Talvez seja desejável que a interface dos aplicativos gráficos evoluam eliminando paulatinamente muitos dos elementos presentes atualmente, como botões, janelas, browsers, menus, etc. Nesse sentido, as interfaces por voz podem dar a sua contribuição.

Além do controle a distância do modelo e do controle de comandos, a tecnologia de SR/TTS permite a criação de serviços de envio de mensagens gravadas, que podem conter relatórios falados com os dados constantes no modelo. Esses relatórios podem ser intercambiados entre os projetistas através das redes de comunicação disponíveis.

Outra situação onde as interfaces por voz mostram o seu potencial, é no auxílio de profissionais que sofreram algum tipo de lesão e ficaram impossibilitados de executar ações motoras. Eles podem se beneficiar de interfaces que prescindam do uso de dispositivos de controle como o *mouse* e o teclado. Com o foco voltado para auxiliar pessoas portadoras de necessidades especiais podem ser encontradas experiências bem sucedidas, como a conduzida e relatada por Corine Bickley, Hong Z. Tan e David Horowitz, no início da década de 1990. Esses pesquisadores conceberam e testaram uma interface de reconhecimento de voz para AutoCAD, permitindo a um arquiteto quadriplégico voltar a projetar utilizando comandos de voz (Bickley *et al.*, 1993). Empresas de software como Infoquest Technologies e Enact Technology também desenvolveram aplicações para executar comandos de voz em AutoCAD, inclusive apresentando-as em eventos anteriores da Autodesk University (2005).

Vemos valores na possibilidade de continuar integrando tecnologia de voz (SR e TTS) em AutoCAD e agora também em Revit. O foco do aplicativo experimental de TTS/ASR que apresentamos na AU Brasil 2014 diz respeito a fornecer uma interface de fala para que o projetista possa interagir com o modelo, sem a mediação de cursores e solicitando-lhe informação de um modo mais próximo da linguagem natural, evitando ter de conhecer os comandos específicos do programa. Procura-se reduzir a pressão mental que implica a memorização de comandos e processos, permitindo, ao menos em teoria, uma interação mais livre e confortável para o projetista, liberando-o para pensar nas questões concretas do projeto.

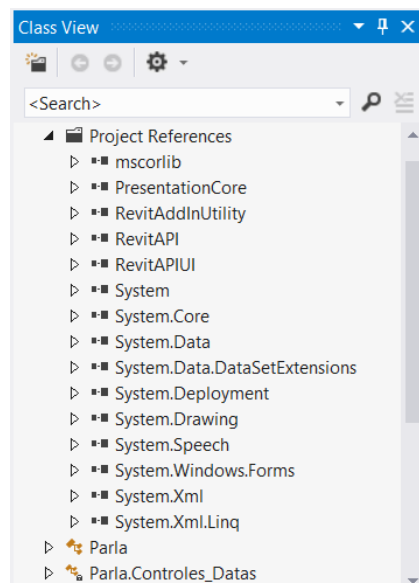
Com essa finalidade, foram desenvolvidas conjuntos de gramáticas que direcionam as requisições feitas através de solicitações com uma linguagem mais natural para instruções específicas, tanto em AutoCAD como em Revit. Essas gramáticas podem ser cumulativamente customizadas pelos usuários uma vez que sendo armazenadas em arquivos XML, podem ser incrementadas com novas locuções de acordo com as preferências de fala dos usuários.

Os componentes TTS e RS

System.Speech é uma biblioteca de componentes .net API com um reconhecedor de fala para Windows 7. Pode ser embarcada em clientes ou em desktop. Aceita treinamento para reconhecimento de diversos usuários, incluindo uma gramática para ditado (*dictation grammar*).

Microsoft.Speech é similar à biblioteca *System.Speech*. Faz parte de vários produtos para servidores e pode ser encontrado na API de desenvolvimento do UCMA, *Unified Communications Managed* da Microsoft. Não requer treinamento de fala.

No projeto do aplicativo desenvolvido selecionamos o System.Speech adicionando como referência ao Project References.



Biblioteca System Speech referenciada no projeto do aplicativo

Referenciando NameSpaces no projeto

Depois de referenciar a biblioteca System.Speech, declaramos como *using* os seguintes Namespaces que permitem funcionalidades específicas.

System.Speech.Recognition: específico para realizar o processo de reconhecimento da fala (RS).
System.Speech.Synthesis: específico para realizar a transformação de textos em áudio (TTS).
System.Speech.AudioFormat: específico para manipulação de arquivos de áudio gerados.

```
using System.Speech.Recognition;
using System.Speech.Synthesis;
using System.Speech.AudioFormat;
```

Uma vez que preparamos as referências e os *Namespace* que serão necessários podemos iniciar o processo de preparar as classes do nosso aplicativo. Embaixo segue um exemplo no qual se utiliza um construtor para a classe RecFala. É necessário criar um objeto de sintetizador ([SpeechSynthesizer](#)) para retornar as mensagens faladas e um objeto para o mecanismo de reconhecimento ([SpeechRecognitionEngine](#)). Devemos também carregar os drivers dos mecanismos de reconhecimento, através do **Painel de Controle** na seção de **Facilidade de acesso** do Windows.

```
// Construtores
public RecFala(string aplic, string idi, bool dice, UIApplication app, Application uia, Document doc, UIDocument uin)
{
    Idioma      = idi;
    Aplicativo  = aplic;
    App         = app;
    UIa         = uia;
    Doc         = doc;
    Uin         = uin;
    Escutou    = "-";
    Accion     = "....";
    switch (Idioma)
    {
        case "es-ES": Idioma_msg = "Reconocedor español";    break;
        case "en-US": Idioma_msg = "English Recognition";    break;
        case "pt-BR": Idioma_msg = "Reconhecedor português"; break;
        default:      Idioma_msg = "Reconocedor desconocido"; break;
    }
    Sinte = new SpeechSynthesizer();
    Sreci = Prepara_Reconocedor(Idioma);
    if (Sreci != null && dice == true)
    {
        RecEn = new SpeechRecognitionEngine(Sreci);
        Decir(Sreci, Idioma, Sinte, Idioma_msg, true);
    }
    if (Sreci != null) RecEn = new SpeechRecognitionEngine(Sreci);
}
// Métodos
public RecognizerInfo Prepara_Reconocedor(string idioma)
{
    RecognizerInfo info = null;
    foreach (RecognizerInfo ri in SpeechRecognitionEngine.InstalledRecognizers())
    {
        if (ri.Culture.Name.Equals(idioma))
        {
            info = ri;
            break;
        }
    }
    return info;
}
```

A seguir se mostra um método para ajustar diversas características da fala permitidas pelo sintetizador. Podem ser ajustadas as velocidades de fala (Rate), a ênfase de fala ([PromptEmphasis](#)) e o idioma utilizado pelo sintetizador. A função que executa a fala é a função `Speak()`.

```
public void Decir(RecognizerInfo ri, string idioma, SpeechSynthesizer sinte, string mensaje, bool diz)
{
    PromptBuilder cultura = new PromptBuilder(new System.Globalization.CultureInfo(idioma));
    sinte.Rate = Taxvoz;
    cultura.StartVoice(ri.Culture);
    cultura.AppendText(mensaje, PromptEmphasis.Moderate);
    cultura.EndVoice();
    if (diz == true)
        sinte.Speak(cultura);
}
```

O reconhecimento de voz pode ser realizado em modo de ditado ou controlado por gramáticas previamente definidas. No caso de trabalhar no modo ditado, é extremamente recomendável treinar o reconhecedor no ambiente Windows, para que o sistema possa entender, graças a uma amostragem de sons previamente coletada, os diversos perfis de vozes que devem ser reconhecidas, uma vez que cada usuário tem um timbre e impoção de voz singular. No aplicativo experimental apresentado foi utilizado o modo de reconhecimento controlado por gramáticas, no qual as locuções estão previamente estabelecidas de modo textual em arquivos XML especialmente formatados.

A construção de uma gramática.

A representação da estrutura gramatical dos exemplos mostrados segue as recomendações publicadas pelo consórcio W3C. O consórcio determina a especificação SRGS (*Speech Recognition Grammar Specification*) com a organização da estrutura sintática que os desenvolvedores devem respeitar ao criar os seus controles de voz. As gramáticas podem ser escritas utilizando o formato *Augmented BNF* (ABNF) ou o formato XML. Embaixo segue um pequeno exemplo em XML de um trecho de gramática na qual estão definidas várias locuções possíveis que o mecanismo de reconhecimento entenderá, direcionando como saída (out =) uma única corrente de caracteres ("A_Z001") que será utilizada para acionar uma instrução específica no aplicativo que se está controlando.

```
<item> <one-of>
    <item>setar vista                </item>
    <item>seta vista                 </item>
    <item>sete vista                 </item>
    <item>por favor sete a vista    </item>
    <item>prepare vista             </item>
    <item>preparar vista            </item>
    <item>mostrar vista             </item>
    <item>quero ver                 </item>
    <item>gostaria de ver a vista  </item>
    <item>mostre me                 </item>
</one-of> <tag> out = "A_Z001"; </tag>
</item>
```

Exemplo de membro de uma gramática simples.

No exemplo, cada locução está encerrada numa estrutura simples `<item>` Locução `</item>` e agrupada numa estrutura `<one-of>` `</one-of>` que permite ao reconhecedor escolher dentre várias locuções possíveis. Veja que alguns dos itens apresentam uma forma de expressão mais direta enquanto outros são preparados de modo mais coloquial. Como os itens podem ser aninhados, essa estrutura é inserida numa gramática maior com o conjunto de gramáticas que direcionam para a execução de diversos outros comandos.

Assim pode ser formado um léxico completo de expressões que servirão para direcionar os comandos do AutoCAD e Revit. Para evitar que todas as locuções estejam num único arquivo, elas podem ser

distribuídas em arquivos XML específicos separando-as por tipo de ação ou procedimento. Durante uma seção de reconhecimento as gramáticas podem ser carregadas e descarregadas a qualquer momento. Embaixo um exemplo de função que serve para carregar no mecanismo de reconhecimento uma gramática específica `RecEn.LoadGrammarAsync(gramatica)`.

```
public Grammar Carga_Gramatica_Arquivada(string arquivo, string nome)
{
    Grammar gramatica = new Grammar(@"C:\JLMenegotto\Academia\04_Pesquisa\Parla\" + arquivo);
    gramatica.Name = nome;
    RecEn.LoadGrammarAsync(gramatica);
    return gramatica;
}
```

Notamos que a separação de gramáticas em arquivos específicos torna o processo de reconhecimento mais eficiente e menos sujeito a ambiguidades semânticas, pois permite repetir a mesma palavra ou locução em contextos diferentes.

Ativação do reconhecimento em AutoCAD.

Para que o processo de reconhecimento aconteça dentro do editor de AutoCAD e Revit, devemos preparar um controlador de eventos associado à classe de reconhecimento. Esse controlador dispara o processo que interpreta o som da fala emitida pelo usuário.

```
public void Ouvir()
{
    ..
    ..

    dbw = HostApplicationServices.WorkingDatabase;
    Grammar GR1 = Carga_Gramatica_Arquivada("Autocad_Todas.xml", "GR1"); //Carrega a gramática
    RecEn.SetInputToDefaultAudioDevice(); //Setar saída de áudio
    RecEn.SpeechRecognized += new EventHandler<SpeechRecognizedEventArgs>(Interpretar_Fala);
    acadApp.Idle += new EventHandler(Accion_de_Inicial);
    RecEn.RecognizeAsync(RecognizeMode.Multiple); //Iniciar reconhecimento
    ..
    ..
    ..
}
```

O reconhecimento é um processo que ficará ativo em segundo plano durante uma seção de desenho ou modelagem.

Ativação do reconhecimento em Revit.

Em Revit, o processo é similar, apenas variando o controlador `acadApp.Idle` por um evento `App.Idling` devemos preparar um controlador de eventos associado à classe de reconhecimento. Esse controlador dispara o processo de entender a fala do usuário.

```
public Result Execute(ExternalCommandData commandData, ref string message, ElementSet elements)
{
    ..
    ..
    RecFala voz = new RecFala("Revit", "es-ES", true, App, Rvt, Doc, Uin);
    Grammar GR1 = voz.Carga_Gramatica_Arquivada("Revit_Todas.xml", "GR1");
    voz.RecEn.SetInputToDefaultAudioDevice(); // Setar saída de áudio
    voz.RecEn.SpeechRecognized += new EventHandler<SpeechRecognizedEventArgs>(voz.Interpretar_Fala);
    App.Idling += new EventHandler<IdlingEventArgs>(voz.Activar_Acciones);
    voz.RecEn.RecognizeAsync(RecognizeMode.Multiple); // Iniciar reconhecimento
    ..
    ..
    return Result.Succeeded;
}
```

Interpretar a fala.

O objeto reconhecedor, ativado através do controlador de eventos `EventHandler` (`RecEn.SpeechRecognized`) devolverá o resultado ouvido (`recres = e.Result;`) cuja semântica deverá ser passada a uma outra função de interpretação. Aqui preparamos uma função `RS_Semantica` para derivar o resultado e executar a ação correspondente no aplicativo (AutoCAD ou Revit). Segue trecho da função:

```
public void RS_Semantica(RecognitionResult recres)
{
    string itens = "-";
    if (recres.Semantics != null)
    {
        foreach (KeyValuePair<String, SemanticValue> child in recres.Semantics)
        {
            string llave = child.Key;
            string valor = child.Value.Value.ToString();
            switch (llave)
            {
                case "salida": Accion = valor; itens = itens + " | " + Accion; break;
                case "letras": Letras = valor; itens = itens + " | " + Letras; break;
                case "numero": Numero = valor; itens = itens + " | " + Numero; break;
                case "accion": Accion = valor; itens = itens + " | " + Accion; break;
                case "objeto": Objeto = valor; itens = itens + " | " + Objeto; break;
            }
        }
    }
}
```

Como aplicar o reconhecimento.

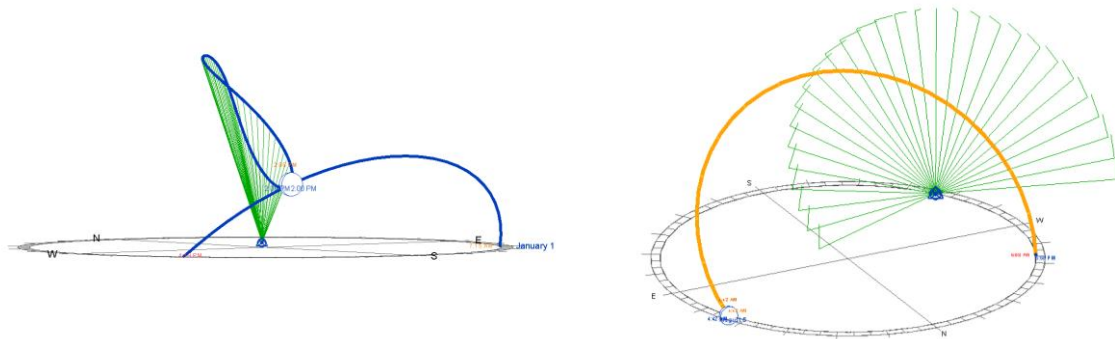
As bibliotecas para realizar o reconhecimento de fala dependem de drivers de reconhecimento para cada língua específica. Atualmente no ambiente Windows há disponível reconhecedores em inglês, francês, espanhol, alemão, japonês, chinês simplificado e chinês tradicional. Embora ainda não disponível para o português, pode-se utilizar o mecanismo em espanhol, usando locuções que sejam as mesmas para ambas as línguas (vista, planta, ampliar, etc.) ou declarando a locução em português desde que o seu fonema¹ exista em espanhol. Ainda que a palavra não exista o som será reconhecido. Desde que declaradas na gramática, as duas locuções seguintes são compreensíveis no interpretador espanhol.

¹ O *fonema* é o menor elemento sonoro capaz de estabelecer uma distinção de significado entre as palavras.

```
<item>quero ver </item>
<item>gostaria de ver a vista</item>
```

Já para o processo de transformação de Texto em fala (TTS) há vozes prontas e gratuitas em português com a variante brasileira.

Nesta experiência foram criadas algumas gramáticas específicas misturando locuções em espanhol e português, para executar ações de criação de *Grids* e *Levels*, contabilizar elementos construtivos, realizar operações de gerenciamento de vistas, setar *View Templates*, aplicar *Zooms*, ajustar o ponto de vista em vistas 3D, ajustar a *Section Box* e realizar estudos solares.



Esquerda: raios do Analema solar, durante um ano com intervalo semanal às 2.00 PM.
Direita: raios solares num dia com intervalo de 30 minutos.

Gramática de datas por anos,

```
<rule id="fechaa">
  <one-of>
    <item><one-of><item>atual </item><item>mundial de brasil </item></one-of><tag> out = "2014"; </tag></item>
    <item><one-of><item>o do mundial de brasil </item><item>mundial en brasil </item></one-of><tag> out = "2014"; </tag></item>
    <item><one-of><item>el del mundial de brasil</item><item>copa del mundo en brasil</item></one-of><tag> out = "2014"; </tag></item>
    <item><one-of><item>el actual </item><item>corriente</item><item>corrente </item></one-of><tag> out = "2014"; </tag></item>
    <item><one-of><item>año pasado </item><item>año anterior </item></one-of><tag> out = "2013"; </tag></item>
    <item><one-of><item>ano pasado </item><item>ano anterior </item></one-of><tag> out = "2013"; </tag></item>
    <item><one-of><item>próximo año </item><item>año que viene </item></one-of><tag> out = "2015"; </tag></item>
    <item><one-of><item>el de los juegos olímpicos</item><item>olimpiadas </item></one-of><tag> out = "2016"; </tag></item>
    <item> <rule ref special="GARBAGE"/> <tag> out = "..."; </tag> </item>
  </one-of>
</rule>
```

Gramática de horários para os estudos solares,

```
<rule id="fechah">
  <one-of>
    <item><one-of><item>siete de la mañana </item></one-of> <tag> out = "7"; </tag></item>
    <item><one-of><item>sete da manhã</item> </one-of> <tag> out = "7"; </tag></item>
    <item><one-of><item>ocho de la mañana </item></one-of> <tag> out = "8"; </tag></item>
    <item><one-of><item>oito da manhã</item> </one-of> <tag> out = "8"; </tag></item>
    <item><one-of><item>nueve de la mañana </item></one-of> <tag> out = "9"; </tag></item>
    <item><one-of><item>nove da manhã</item> </one-of> <tag> out = "9"; </tag></item>
    <item><one-of><item>diez de la mañana </item></one-of> <tag> out = "10"; </tag></item>
    <item><one-of><item>dez da manhã</item> </one-of> <tag> out = "10"; </tag></item>
    <item><one-of><item>once de la mañana </item></one-of> <tag> out = "11"; </tag></item>
    <item><one-of><item>once da manhã</item> </one-of> <tag> out = "11"; </tag></item>
  </one-of>
</rule>
```



```

<item><one-of><item>mediodía </item></one-of> <tag> out = "12";</tag></item>
<item><one-of><item>meiodía </item></one-of> <tag> out = "12";</tag></item>
<item><one-of> <item>una de la tarde </item></one-of> <tag> out = "13"; </tag></item>
<item><one-of> <item>una da tarde </item></one-of> <tag> out = "13"; </tag></item>

```

Algumas opções gramaticais para executar funções de manipulação de Zoom, de edição e criação de objetos geométricos.

```

<!--Acciones de tela-->
<item><one-of> <item>alejarse </item></one-of> <tag> out = "A_Z001"; </tag> </item>
<item><one-of> <item>achicar </item></one-of> <tag> out = "A_Z002"; </tag> </item>
<item><one-of> <item>acercar </item></one-of> <tag> out = "A_Z003"; </tag> </item>
<item><one-of> <item>aproximar </item></one-of> <tag> out = "A_Z004"; </tag> </item>
<item><one-of> <item>ampliar </item></one-of> <tag> out = "A_Z005"; </tag> </item>
<item><one-of> <item>focalizar </item></one-of> <tag> out = "A_Z006"; </tag> </item>
<item><one-of> <item>focalizar todo </item></one-of> <tag> out = "A_Z007"; </tag> </item>
<item><one-of> <item>focalizar toda </item></one-of> <tag> out = "A_Z008"; </tag> </item>
<item><one-of> <item>focalizar todos </item></one-of> <tag> out = "A_Z009"; </tag> </item>
<item><one-of> <item>focalizar todas </item></one-of> <tag> out = "A_Z010"; </tag> </item>
<item><one-of> <item>focalizar sala </item></one-of> <tag> out = "A_Z011"; </tag> </item>
<item><one-of> <item>focalizar compartimento </item></one-of> <tag> out = "A_Z012"; </tag> </item>
<item><one-of> <item>axonométrica </item></one-of> <tag> out = "A_Z013"; </tag> </item>
<item><one-of> <item>proyección paralela </item></one-of> <tag> out = "A_Z014"; </tag> </item>
<item><one-of> <item>vista 3d </item></one-of> <tag> out = "A_Z015"; </tag> </item>
<item><one-of> <item>perspectivar </item></one-of> <tag> out = "A_Z016"; </tag> </item>
<item><one-of> <item>perspectiva cónica </item></one-of> <tag> out = "A_Z017"; </tag> </item>
<item><one-of> <item>cámara en 3d </item></one-of> <tag> out = "A_Z018"; </tag> </item>
<item><one-of> <item>pan </item></one-of> <tag> out = "A_Z019"; </tag> </item>
<item><one-of> <item>correr </item></one-of> <tag> out = "A_Z020"; </tag> </item>
<item><one-of> <item>vista predefinida </item></one-of> <tag> out = "A_Z021"; </tag> </item>
<item><one-of> <item>vistas predefinidas </item></one-of> <tag> out = "A_Z022"; </tag> </item>
<item><one-of> <item>ir a la vista </item></one-of> <tag> out = "A_Z023"; </tag> </item>
<item><one-of> <item>setar la vista </item></one-of> <tag> out = "A_Z024"; </tag> </item>
<item><one-of> <item>mostrar vistas </item></one-of> <tag> out = "A_Z025"; </tag> </item>
<item><one-of> <item>setar vistas </item></one-of> <tag> out = "A_Z026"; </tag> </item>
<item><one-of> <item>vista cortada </item></one-of> <tag> out = "A_Z027"; </tag> </item>
<item><one-of> <item>límite la vista </item></one-of> <tag> out = "A_Z028"; </tag> </item>
<item><one-of> <item>aumentar caja </item></one-of> <tag> out = "A_Z029"; </tag> </item>
<item><one-of> <item>ampliar caja </item></one-of> <tag> out = "A_Z030"; </tag> </item>
<item><one-of> <item>aumentar recorte </item></one-of> <tag> out = "A_Z031"; </tag> </item>
<item><one-of> <item>ampliar recorte </item></one-of> <tag> out = "A_Z032"; </tag> </item>
<item><one-of> <item>diminuir caja </item></one-of> <tag> out = "A_Z033"; </tag> </item>
<item><one-of> <item>reducir caja </item></one-of> <tag> out = "A_Z034"; </tag> </item>
<item><one-of> <item>reducir recorte </item></one-of> <tag> out = "A_Z035"; </tag> </item>
<item><one-of> <item>ampliar recorte </item></one-of> <tag> out = "A_Z036"; </tag> </item>
<item><one-of> <item>reducir recorte </item></one-of> <tag> out = "A_Z037"; </tag> </item>
<item><one-of> <item>reducir recorte </item></one-of> <tag> out = "A_Z038"; </tag> </item>
<item><one-of> <item>reducir recorte </item></one-of> <tag> out = "A_Z039"; </tag> </item>
<item><one-of> <item>reducir recorte </item></one-of> <tag> out = "A_Z040"; </tag> </item>
<item><one-of> <item>reducir recorte </item></one-of> <tag> out = "A_Z041"; </tag> </item>
<item><one-of> <item>reducir recorte </item></one-of> <tag> out = "A_Z042"; </tag> </item>
<item><one-of> <item>reducir recorte </item></one-of> <tag> out = "A_Z043"; </tag> </item>
<item><one-of> <item>reducir recorte </item></one-of> <tag> out = "A_Z044"; </tag> </item>
<item><one-of> <item>reducir recorte </item></one-of> <tag> out = "A_Z045"; </tag> </item>
<item><one-of> <item>reducir recorte </item></one-of> <tag> out = "A_Z046"; </tag> </item>
<item><one-of> <item>reducir recorte </item></one-of> <tag> out = "A_Z047"; </tag> </item>
<item><one-of> <item>reducir recorte </item></one-of> <tag> out = "A_Z048"; </tag> </item>
<item><one-of> <item>reducir recorte </item></one-of> <tag> out = "A_Z049"; </tag> </item>
<item><one-of> <item>reducir recorte </item></one-of> <tag> out = "A_Z050"; </tag> </item>
</rule>

```

Conclusões.

A passagem do desenho tradicional para o desenho com ferramentas CAD afastou a régua, o lápis e o compasso das mãos dos projetistas. Talvez as interfaces de reconhecimento de voz possam, em algum momento, fazer com que o mouse e o teclado sofram o mesmo destino e sejam deixados de lado das estações de trabalho.

Apesar dessa declaração ser parte das especulações sobre o futuro, sabemos que locuções do tipo “Quero um edifício de 3 andares separado das divisas de acordo com o código municipal de Rio de Janeiro”, “Diga-me se alguma tubulação interfere na estrutura” ou “Qual é o melhor ângulo para esta janela?” podem, hoje em dia, ser interpretadas e traduzidas como comandos ou sequências de comandos, o que nos leva a concluir, com uma pequena margem de dúvida, que projetar falando em programas como AutoCAD ou Revit poderá tornar-se uma tarefa cada vez mais habitual.

Bibliografia

Anusuya, M.A., Katti. S. K. *Speech Recognition by Machine: A Review. International Journal of Computer Science and Information Security, (IJCSIS). Vol. 6, No. 3, 2009.*

Bickley, Corine, Hong Z. Tan, Horowitz, David. *Voice Input For Graphics And Text Creation: A Case Study.* Proceedings of the Eighth Annual Conference on Technology and Persons with Disabilities (1993).

[http://msdn.microsoft.com/en-us/library/gg145021\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/gg145021(v=vs.110).aspx)

<http://www.w3.org/TR/speech-grammar/>