

L'assistant pédagogique

"We are natural learners" (naturellement l'être humain apprend)

Alex Beard [conférence internationale sur la pédagogie](#).

Le conférencier évoque les résultats PISA ^[1] de différents pays, dans leurs contextes.

Il soutient cette idée : **créer des conditions personnalisées** facilite l'apprentissage.

Une personne qui choisit son programme et son rythme progresse naturellement mieux.

Rien ne remplace l'échange humain entre enseignant et apprenant.

L'**assistant numérique** ^[2] aide l'enseignant à créer des conditions personnalisées d'apprentissage.

Il donne aux **artefacts** pédagogiques ^[3] la force d'un **dialogue** personnalisé avec chaque apprenant.

L'enseignant conçoit l'artefact. L'assistant aide à sa mise en oeuvre.

Dans ce contexte nous allons :

- écrire un artefact et
- créer un assistant adapté.

Écrire un artefact

Voici un exemple de dialogue entre apprenant et assistant.

Les mots entre [] sont les étapes du dialogue.

Les mots précédés de # correspondent aux **intentions** du participant (ses choix).

[welcome]

Bonjour et bienvenue chez ABC.

Nous vous proposons de :

- préparer une descente en snowboard [#snowboard]
- accompagner Louis Armstrong au piano [#piano]
- écrire un mail au président de la république [#président]
- préparer le journal de 20h [#jt]

Que voulez-vous essayer ?

[rien]

D'accord. Ce n'est pas facile de ne rien faire.

[snowboard]

Pendant cette activité souhaitez-vous en savoir plus sur :

- l'anglais [#anglais]
 - le français [#français]
 - le sport et l'histoire [#histoire_sport]
 - la préparation psychologique [#psy_prepa]
 - la géométrie [#géometrie]
 - la programmation informatique [#programmation]
- ?

[snowboard_programmation]

Pourquoi pas, mais au fait, en informatique vous êtes plutôt débutant [#débutant], initié [#initié] ou incollable [#incollable] ?

[snowboard_programmation_initié]

[q1]

Ok, nous allons d'abord vous équiper d'une bonne planche.

J'en ai quelques-unes en réserve mais le programme de gestion des stocks n'est pas terminé. Il ne parvient pas à lister les planches. Le voici :

```
affiche_liste() {  
    for  
        print (planche);  
}
```

L'instruction `for` est vide, je vois plusieurs possibilités de la compléter :

r:

- for (int i=0; i<planches.length; i++) [#for_1]
- for planche in planches [#for_2]
- planches.forEach(planche => { [#for_3]
- for (Board planche : planches) { [#for_4]

Laquelle essayons-nous ?"

[for_1]

euh ... ça ne marche pas. La variable 'i' n'est pas utilisée dans le reste du programme.

une autre idée ?

[for_4]

Ah parfait, merci. Nous avons la liste des planches :

- Apo Selektä TwinTip 154 [#fin]
- Bataleon EvilTwin [#fin]

- Burton Custom X [#fin]

Les voici en image :

!(Apo)[<http://www.iesanetwork.com/n.auriac/snowfrance/images/test%20boardfreestyle.jpg>]

!(Bataleon)[<https://images.evo.com/imgp/700/9793/167101/bataleon-evil-twin-snowboard-2007-.jpg>]

!(Burton)[http://www.surfshop.fr/55412-large_default/custom.jpg]

Ces planches sont recommandées pour le freestyle.

Voici une vidéo en exemple : ![freestyle](https://www.youtube.com/watch?v=7G_nXpv8JbE).

Quelle planche souhaitez-vous ?

[fin] Très bon choix, c'est une planche facile et performante ...

Le dialogue suit un **scénario** constitué d'**alternatives**.

L'assistant détecte les **intentions** du participant (ses choix).

Ce scénario permet d'acquérir un **savoir-faire** en programmation.

Ce sont les éléments de notre artéfact que nous allons écrire: **savoir-faire**, **intentions**, **alternatives**, et **scénario**.

Le savoir-faire

Associés à une discipline, le **savoir-faire** utilise des **compétences** et des **connaissances**.

La discipline décrite dans le tableau ci-dessous combine chaque colonne pour former un savoir-faire, par exemple 'parler anglais en utilisant la conjugaison'.

Discipline	Compétence	Connaissance
anglais	comprendre	mots (vocabulaire)
	parler	structures (lexem, règle)
	lire	conjugaisons (règle)
	écrire	accords (règle)
	dialoguer	
...		

Dans cet exemple toutes les combinaisons sont possible.

Voici comment écrire les combinaisons possibles de cette discipline :

```
{
  "discipline": {
    "nom": "anglais",
    "compétences": [
      {
        "nom": "comprendre",
        "connaissances": [
          {
            "nom": "mots",
            "description": "le vocabulaire et l'orthographe"
          },
          {
            "nom": "structures",
            "description": "les phrases"
          },
          {
            "nom": "conjugaison",
            "description": "la conjugaison"
          },
          {
            "nom": "accords",
            "description": "règles d'accord en genre et en nombre"
          }
        ]
      },
      {
        "nom": "parler",
        "connaissances": [
          {
            "nom": "mots",
            "description": "le vocabulaire et l'orthographe"
          },
          {
```

```
        "nom": "structures",
        "description": "les phrases"
    },
    {
        "nom": "conjugaison",
        "description": "la conjugaison"
    },
    {
        "nom": "accords",
        "description": "règles d'accord en genre et en nombre"
    }
]
},
{
    "nom": "lire",
    "connaissances": [
        {
            "nom": "mots",
            "description": "le vocabulaire et l'orthographe"
        },
        {
            "nom": "structures",
            "description": "les phrases"
        },
        {
            "nom": "conjugaison",
            "description": "la conjugaison"
        },
        {
            "nom": "accords",
            "description": "règles d'accord en genre et en nombre"
        }
    ]
},
{
    "nom": "écrire",
    "connaissances": [
```

```
{
  "nom": "mots",
  "description": "le vocabulaire et l'orthographe"
},
{
  "nom": "structures",
  "description": "les phrases"
},
{
  "nom": "conjugaison",
  "description": "la conjugaison"
},
{
  "nom": "accords",
  "description": "règles d'accord en genre et en nombre"
}
],
{
  "nom": "dialoguer",
  "connaissances": [
    {
      "nom": "mots",
      "description": "le vocabulaire et l'orthographe"
    },
    {
      "nom": "structures",
      "description": "les phrases"
    },
    {
      "nom": "conjugaison",
      "description": "la conjugaison"
    },
    {
      "nom": "accords",
      "description": "règles d'accord en genre et en nombre"
    }
  ]
}
```

```
}  
}  
]  
}  
]
```

Pour la suite nous désignons un **savoir-faire** par une discipline, une compétence, et une connaissance.

Par exemple, ‘parler anglais en utilisant la conjugaison’ s’écrit :

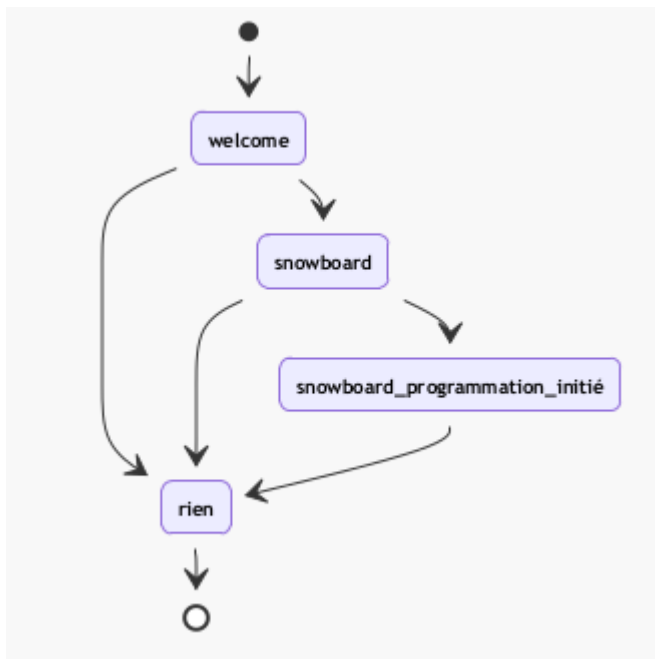
```
.anglais.parler.conjugaison
```

Passons maintenant à l’écriture des alternatives du dialogue.

Les alternatives

L’ensemble des possibilités d’un dialogue constitue un graphe ^[4] dont nous allons noter les étapes et les transitions.

Voici un exemple :



Le code ci-dessous permet de générer le graphique ci-dessus

```
stateDiagram
    [*] --> welcome
    welcome --> rien
    welcome --> snowboard
    snowboard --> snowboard_programmation_initié
    snowboard --> rien
    snowboard_programmation_initié --> rien
    rien --> [*]
```

Chaque mot ou suite de mots séparés par _ est une étape.

L'astérisque * représente le participant en début et en fin de dialogue.

La flèche --> représente une transition possible entre deux étapes.

Voici l'écriture du graphe des alternatives pour le début de notre artefact :

```
{
  "alternatives": {
    "*": "welcome",
    "welcome": "snowboard",
    "welcome": "rien",
    "snowboard": "snowboard_programmation",
    "snowboard": "rien",
    "snowboard_programmation": "snowboard_programmation_initié",
    "snowboard_programmation": "rien",
    "snowboard_programmation_initié": "q1",
    "rien": "*"
  }
}
```

Après avoir choisit son **activité** le participant entre dans les alternatives spécifiques de l'artefact :

```
{
  "alternatives": {
    "*": "q1",
```



```

    "q1": "rien,for_1,for_2,for_3,for_4",
    "for_1": "*",
    "for_2": "*",
    "for_3": "*",
    "for_4": "fin",
    "fin": "*",
    "rien": "*"
  },
}

```

Chaque paire d'étapes, notée "etape A": "etape B", est une transition possible.

Passons maintenant à l'écriture des différentes intentions possibles (les choix offerts aux participants).

Les intentions

L'intention est contextuelle, elle dépend de l'alternative.

Dans notre exemple, l'intention '[#initié]' n'a de sens que dans une alternative où le participant a déjà choisit une discipline.

Voici l'écriture des intentions pour le début de notre artefact :

```

{
  "intentions": [
    {
      "welcome": {
        "rien": "aucune, rien, non, négatif",
        "snowboard": "glisse, planche, board, surf",
        "piano": "musique, instrument, clavier, clavecin",
        "président": "trump, obama, macron, hollande",
        "jt": "journal, 20 heures, JT"
      },
      "snowboard": {
        "rien": "aucune, rien, non, négatif",
        "anglais": "anglais, américain, US, UK, AUS, Common Wealth, Sha
kespeare",

```

```

    "français": "langue, france",
    "histoire_sport": "sport, histoire, passé",
    "psy_prepa": "mental, psychologie, préparation",
    "géometrie": "mathématique, géométrie, pythagore",
    "programmation": "ordinateur, informatique, programme, logique
  }
},
  "snowboard_programmation": {
    "débutant": "1, débutant, premier, faible, pas bon",
    "initié": "2, initié, deuxième, moyen",
    "incollable": "3, incollable, expert, fort, très bon"
  },
}
]
}

```

Après avoir choisit son **activité** le participant entre dans les intentions spécifiques :

```

{
  "intentions": [
    {
      "q1": {
        "rien": "aucune, rien, non, négatif",
        "for_1": "premier, un",
        "for_2": "deuxième, deux",
        "for_3": "troisième, trois",
        "for_4": "quatrième, quatre",
      },
      "for_4": {
        "rien": "aucune, rien, non, négatif",
        "fin": "premier, un, deuxième, deux, troisième, trois, bataleo
n, burton, apo, custom, eviltwin, selekta",
      },
    }
  ]
}

```

Chaque intention est définie par un ensemble de mots.

Cela permet à l'assistant de reconnaître le choix du participant, lors de l'analyse vocale notamment.

En général cette liste reste dans la **sémantique** de l'intention à définir.

Il ne reste plus qu'à écrire le scénario.

Le scénario

L'assistant pourra générer la première partie du dialogue de manière automatique lorsque nous aurons écrit les scénarii spécifiques à chaque activité.

L'écriture d'un scénario commence donc là où le participant entre en activité.

Voici l'écriture du scénario "snowboard_programmation_initié" :

```
{
  "scenario": {
    "code": "snowboard_programmation", "niveau": "initié",
    "etapes": [{
      "etape": "q1",
      "savoir-faire": "programmation.swift.boucle_for",
      "md": "
Ok, nous allons d'abord vous équiper d'une bonne planche.
J'en ai quelques-unes en réserve mais le programme de gestion des s
tocks n'est pas terminé. Il ne parvient pas à lister les planches. Le voici :
affiche_liste() {
  for
    print (planche);
}
}
L'instruction `for` est vide, je vois plusieurs possibilités de la
compléter:
- for (int i=0; i<planches.length; i++) [#for_1]
- for planche in planches [#for_2]
- planches.foreach( planche => { [#for_3]
- for (Board planche : planches) { [#for_4]
Laquelle essayons-nous ?"
    },
```

```
{  
    "etape":"for_1",  
    "savoir-faire":"programmation.swift.boucle_for",  
    "score":"-1",  
    "md": "
```

euh ... ca ne marche pas. La varibale 'i' n'est pas utilisée dans le reste du programme. une autre idée ?

```
"  
},  
{  
    "etape":"for_2",  
    "savoir-faire":"programmation.swift.boucle_for",  
    "score":"-1",  
    "md": "
```

Cela ne fonctionne pas. Il me semble que la bonne instruction devrait référencer la variable `planche` utilisée dans l'instruction `print`

```
"  
},  
{  
    "etape":"for_3",  
    "savoir-faire":"programmation.swift.boucle_for",  
    "score":"-1",  
    "md": "
```

Cela ne fonctionne pas. Avec cette ligne il manque une parenthèse fermante au programme.

```
"  
},  
{  
    "etape":"for_4",  
    "savoir-faire":"programmation.swift.boucle_for",  
    "score":"+1",  
    "md": "
```

Ah parfait, merci. Nous avons la liste des planches :

- Apo Seleakta TwinTip 154 [#fin]
- Bataleon EvilTwin [#fin]
- Burton Custom X [#fin]

Les voici en image :

```

        ! (Apo) [http://www.iesanetwork.com/n.auriac/snowfrance/images/test%20boardfreestyle.jpg]
        ! (Bataleon) [https://images.evo.com/imgp/700/9793/167101/bataleon-evil-twin-snowboard-2007-.jpg]
        ! (Burton) [http://www.surfshop.fr/55412-large_default/custom.jpg]
        Ces planches sont recommandées pour le freestyle. Voici une vidéo en exemple : ![freestyle](https://www.youtube.com/watch?v=7G_nXpv8JbE).
        Quelle planche souhaitez-vous ?
        "
    },
    {
        "etape": "fin",
        "niveau": "initié",
        "savoir-faire": "programmation.swift.boucle_for",
        "md": "
        Très bon choix, c'est une planche facile et performante ...
        "
    } ]
}
}

```

L'assistant lit à voix haute la partie 'md' de chaque étape.

Dans cette partie le point d'exclamation sert à donner un lien internet, généralement vers des ressources documentaires.

Un libellé donné entre parenthèses () sert de lien vers une adresse donnée entre crochets [].

Pour rappel les **intentions** sont nommées avec un dièse # en préfixe. Elles mènent à une étape du même nom.

L'assistant pourra utiliser l'information 'score' en la cumulant pour déterminer si un 'savoir-faire' est acquis ou non.

Créer l'assistant

Les 'chatbot pédagogiques' ^[5] existent depuis quelques années, certains sous forme de projets open-source.

Sont-ils faciles à utiliser ? Est-ce qu'il y a des standards ? Est-ce compliqué à créer ? De quoi avons nous besoin ?

Nous avons besoin :

- d'une conversion de la voix en texte
- de synthèse vocale
- d'analyse du langage naturel
- d'analyse sémantique
- d'une interface homme-machine

Apple propose ces 5 services, intégrés dans un environnement de programmation stable et bien documenté.

Ces conditions invitent à l'essai, au prototype. Nous allons utiliser les bibliothèque de programmation de 'Siri' [6].

1. Présentation du Programme International pour le suivi des acquis des élèves sur le [site de l'éducation nationale](#). ↩
2. Logiciel et appareil proposant un dialogue en langage naturel, par exemple une borne 'Alexa', 'Google home' ou 'Home Pod' ↩
3. Terme utilisé en pédagogie, définition sur le site [edutechwiki.unige.ch](#) ↩
4. Mathématiques. Définition sur [wikipedia](#) ↩
5. Exemples de chat bot pédagogique : <https://fr.snatchbot.me/education> , <https://tiny-coaching.com/> ↩
6. Application de commande vocale intégrée aux appareils d'Apple
[https://fr.wikipedia.org/wiki/Siri_\(logiciel\)](https://fr.wikipedia.org/wiki/Siri_(logiciel)) ↩