



Système d'aide à la décision

Manuel d'utilisation avancée

2016 - 2017

VILLAIN Edouard

DEFONTE Véronique

ROLLAND Thomas

Université de Toulouse - Paul Sabatier

Introduction

Ce manuel est destiné aux administrateurs souhaitant utiliser le système d'aide à la décision sur le site WordPress et qui possèdent des connaissances informatiques.

Il est nécessaire d'avoir lu le manuel d'utilisation au préalable pour une bonne compréhension du manuel d'utilisation avancée.

Nous détaillons ici le fonctionnement et les spécificités internes du système pour qu'un administrateur puisse éventuellement modifier certaines parties de code et adapter d'autant mieux le système à son site WordPress.

L'exemple de base serait un administrateur qui souhaiterait fournir les cinq meilleurs résultats et non plus les trois comme actuellement implémenté.

Le projet a été développé de façon générique et réutilisable.

Tout administrateurs d'un site WordPress peut alors intégrer le système d'aide à la décision à son site web.

Informations générales

- ❖ Version 1.0
- ❖ Langages HTML, php, css, javascript
- ❖ Base de données wordpress mySql requise
- ❖ Widget associé au système d'aide à la décision
- ❖ Widget associé au formulaire de saisie de note
- ❖ Page de configuration administrateur
- ❖ Plugin d'aide à la recommandation sous forme d'AMAS
 - Adaptive Multi Agents System
- ❖ Apprentissage du système sur la durée

Architecture du plugin

Fichiers	Rôles
System.php	Fichier racine du plugin, appelé lors de l'activation du plugin sur wordpress Lance l'instanciation du plugin.
instanciation.php	Ajouter les widgets au site web et création / suppression des tables de la base de données
Configuration.php	Configurer la base de données utilisées par le système
EscapeWidget.php	Interface du widget d'aide à la décision
FeedBackWidget.php	Interface du widget de saisie de notes
AMAS_plugin.php	Plugin du système sous forme d'AMAS
launcher_AMAS_requete.php	Récupère les paramètres du widget de recommandation rentrés par l'utilisateur puis appelle l'AMAS pour traiter la requête et fournir les trois meilleurs résultats
launcher_AMAS_feedback_choice.php	Récupère le choix fait par l'utilisateur parmi les trois résultats, ainsi que les proposition puis appelle l'AMAS pour traiter le feedback de choix
launcher_AMAS_feedback_saisieNotes.php	Récupère la salle sélectionnée par l'utilisateur ainsi que les notes saisies puis appelle l'AMAS pour traiter le formulaire de saisie de notes
Launcher_Delete_Critere.php	Récupère le critère sélectionné par l'administrateur dans la page de configuration et le supprime de la base de données

Launcher_Delete_Salle.php	Récupère la salle sélectionnée par l'administrateur dans la page de configuration et le supprime de la base de données
Launcher_Delete_Theme.php	Récupère le thème sélectionné par l'administrateur dans la page de configuration et le supprime de la base de données
Launcher_Lien.php	Récupère le lien tapé par l'administrateur dans la page de configuration et le supprime de la base de données
Launcher_New_Critere.php	Récupère le critère tapé par l'administrateur dans la page de configuration et l'ajoute de la base de données
Launcher_New_Theme	Récupère le thème tapé par l'administrateur dans la page de configuration et l'ajoute de la base de données
Launcher_Note	Récupère les notes tapées par l'administrateur dans la page de configuration et modifie la base de données
Launcher_Salle	Récupère les informations sur la salle renseignées par l'administrateur dans la page de configuration et l'ajoute à la base de données
Launcher_Theme	Récupère les thèmes entrés par l'administrateur dans la page de configuration et l'ajoute aux salles
scripts.js	Contient les fonctions permettant de modifier l'affichage des widget mais aussi les appels des launchers
ScriptConfiguration.js	Contient les fonctions d'affichage et appels de launchers de la page de configuration
style.css	Contient les styles utilisés par les widgets

Détail sur la base de données

Sept tables sont ajoutées à la base de données lors de l'instanciation du plugin dont nous allons détailler le contenu ci dessous.

Elles sont supprimées uniquement à la désinstallation du plugin, et non la simple désactivation.

- ❖ wp_system_recommandation_configuration
 - (ID, increment_feedback_choix, nb_max_feedback_choix_jour, increment_feedback_saisienotes, nb_max_feedback_saisienotes_jour)
- ❖ wp_system_recommandation_criteres
 - (ID, Name)
- ❖ wp_system_recommandation_themes
 - (ID, Name)
- ❖ wp_system_recommandation_salles
 - (ID, Name, lien, theme)
- ❖ wp_system_recommandation_notes
 - (ID, Note, #id_criteres, #id_salle)
- ❖ wp_system_recommandation_log_feedback_choix
 - (ID, #id_salle, Date, IP, Modifications)
- ❖ wp_system_recommandation_log_feedback_saisienotes
 - (ID, #id_salle, Date, IP, Modifications)

Nous pouvons remarquer les clés primaires soulignées, et les clés étrangères avec un '#' devant le nom du champs.

De plus nous pouvons noter que les thèmes d'une salles sont stockées dans la base de données dans le champs theme de la table wp_system_recommandation_salles qui n'est autre qu'un varchar(512), autrement dit une chaine de caractère contenant la concaténation des thèmes séparés d'un point virgule.

De même, les modifications enregistrées dans les tables de log des feedback sont de la forme suivante $critere_1;modification_1;; \dots;; critere_n;modification_n$. Un double point virgule sépare les critères et la modification effectuée sur ce critère et un simple point virgule sépare l'id du critère de sa modification.

Détail sur la page de configuration

Le code de cette partie est contenu dans le fichier Configuration.php. L'interface est composé de plusieurs tableaux.

Nom salle	Prix	Décor	Thème	Lien	
Toulouse	17.652	22.852	Gestion des thèmes	http://monsie.toulous	Supprimer
Lyon	20.828	35.828	Gestion des thèmes	http://monsie.lyon.fr	Supprimer
Paris	39.8	45.5	Gestion des thèmes	http://monsie.paris.fr	Supprimer
			Gestion des thèmes		

Le premier tableau liste toutes les salles présentent, ainsi que leurs notes pour chaque critères, les thèmes et enfin les liens, présent dans la base de données. La modification d'un champs, suivis de l'action cliqué sur le bouton correspondant, permet de modifier les informations dans la base de données. Une salle peut être supprimée de la base de données. Toutes les informations correspondant à cette salle seront supprimées de toutes les tables. Enfin une salle peut être ajoutée à la base de données en fonction des informations saisies dans le tableau.

Détails des widgets

Détails du widget d'aide à la décision

Ce widget, dont le code est contenu dans le fichier EscapeWidget.php, est composé de plusieurs tableaux et blocs qu'il est possible d'afficher ou non via le javascript contenu dans le fichier script.js.

Voici le détail de l'implémentation du widget

Titre du widget bulle d'aide

Critères	Important	Pas important	Valeur
critère_1	Barre de slide pour sélectionner le poid sur critère_1		valeur_1
...	...		
critère_c	Barre de slide pour sélectionner le poid sur critère_n		valeur_c

Sélection des thèmes (menu déroulant) ↓	
Thème_1	↓
...	
Thème_t	

Expertise	Faible	Elevée	Valeur
Expertise	Barre de slide pour sélectionner l'expertise		valeur_expertise

Lancer la recherche

Lorsqu'un utilisateur lance une recherche, le tableau des résultats devient visible ce qui lui permet de choisir un choix parmi les trois le sélectionnant via les radioboutons. Les liens ne sont jamais affichés, ils sont uniquement présent pour être utilisés lorsque l'utilisateur clique sur le bouton choisir.

Nom salle	Score	Choisir	Lien (jamais affiché)
Salle_i	Score_i %		Lien_i
Salle_j	Score_j %	●	Lien_j
Salle_k	Score_k %		Lien_k

Choisir

Lorsque l'utilisateur clique sur le bouton choisir, une fenêtre s'ouvre et affiche le contenu du lien de la salle en question. Un feedback est aussi lancé pour permettre l'apprentissage du système.

De plus, les résultats sont alors cachés pour éviter que l'utilisateur ne puisse choisir plusieurs salles. Il faudra alors recommencer une recherche (qui aura pris en compte les éventuelles modifications du feedback précédent) si l'utilisateur souhaite à nouveau utiliser le système d'aide à la décision.

Détails du widget de saisie de notes

Ce widget, dont le code est contenu dans le fichier FeedBackWidget.php, est composé de plusieurs tableaux et blocs qu'il est possible d'afficher ou non via le javascript contenu dans le fichier script.js.

Voici le détail de l'implémentation du widget

Titre du widget bulle d'aide

Sélection de la salle (menu déroulant) ↓
Salle_1
...
Salle_t

Critères	Important	Pas important	Valeur
critère_1	Barre de slide pour sélectionner le poid sur critère_1		valeur_1
...	...		
critère_c	Barre de slide pour sélectionner le poid sur critère_n		valeur_c

Expertise	Faible	Elevée	Valeur
Expertise	Barre de slide pour sélectionner l'expertise		valeur_expertise

Noter la salle

La sélection d'une salle fait apparaître le tableau des critères, de l'expertise ainsi que le bouton pour noter la salle.

Lorsque l'utilisateur clique sur le bouton noter la salle, le formulaire de saisie de note est envoyé et un feedback sera alors traité par l'AMAS.

De plus, le tableau des critères est alors de nouveau caché pour éviter que l'utilisateur puisse exécuter plusieurs fois le formulaire de saisie de notes trop facilement.

Détails sur le plugin

Détails de l'AMAS

```
<?php
class agent_interface {
    public function agent_interface_getListIdSalle($listeTheme);
    public function agent_interface_allocation_salles($listIdSalle);
    public function agent_interface_traiter_requete($listePoid, $listeTheme);
    public function agent_interface_get_three_max_res($results);
    public function agent_interface_traiter_feedback_saisieNotes($nomSalle, $expertise, $listePoid);
    public function agent_interface_log_feedback_saisieNotes($idSalle);
    public function agent_interface_traiter_feedback_choix($idSalle_choisie, $expertise, $listePoid, $idSalleProposees);
    public function agent_interface_log_feedback_choix($idSalle_choisie);
    public function agent_interface_get_dynamix_marge($expertise);
    public function agent_interface_GetIP();
}

class agent_salle {
    public function agent_salle_allocation_all_criteres();
    public function agent_salle_get_note_salle($idSalle, $listePoid);
    public function agent_salle_modify_note_salle($idSalle, $listePoid, $expertise, $marge, $type_feedback);
}

class agent_criterie {
    public function agent_criterie_get_note_criterie($idSalle, $idCriterie, $poid);
    public function agent_criterie_modify_note($idSalle, $idCriterie, $poid, $marge, $expertise, $type_feedback);
}
```

L'AMAS est codé en php dans un unique fichier qui se décompose en trois classes correspondant aux trois agents, à savoir l'agent interface, les agents salle et les agents critères.

Vous constatez sur le screen précédent du détail des fonctions qui composent ces classes.

Vous pouvez noter les fonctions d'allocation des classes agent_interface et agent_salle de part l'architecture et l'organisation interne de l'AMAS. En effet l'agent interface est appelé

en premier dans la chaîne du traitement des requêtes utilisateurs et feedback. Cet agent appelle lui même des agents salles, qui eux même appellent des agents critères.

L'agent interface possède trois fonctions principales.

- Traiter les requêtes utilisateurs, à savoir lorsque l'utilisateur demande une aide à la décision
 - Récupérer les salles en fonctions des thèmes sélectionnés par l'utilisateur
 - Allouer un agent salles pour chacune de ces salles
 - Demander à chacun de ces agents la note de leur salle en fonction des poids de la requête
 - Trier et ne garder que les trois meilleurs résultats
 - Récupérer les informations complémentaires de ces trois salles
 - Retourner le résultats
- Traiter le feedback lorsque l'utilisateur choisit un choix parmi ceux proposés par le système d'aide à la décision
 - Log dans la base de données des informations du feedback de choix
 - Vérifie le nombre d'utilisation de ce feedback de cet utilisateur, pour cette salle, et sur cette journée
 - Si l'utilisateur n'a pas utilisé trop de fois ce type de feedback
 - Appel de l'agent salle concerné et modification de ses notes
 - Si l'utilisateur a utilisé trop de fois ce type de feedback
 - Une alerte est levé pour le prévenir qu'il n'y a pas de modification des notes
 - Ouverture du lien de la salle choisie dans une nouvelle fenêtre
- Traiter les formulaire de saisie de notes
 - Log dans la base de données des informations du feedback de saisie de notes
 - Vérifie le nombre d'utilisation de ce feedback de cet utilisateur, pour cette salle, et sur cette journée
 - Si l'utilisateur n'a pas utilisé trop de fois ce type de feedback
 - Appel de l'agent salle concerné et modification de ses notes
 - Si l'utilisateur a utilisé trop de fois ce type de feedback
 - Une alerte est levé pour le prévenir qu'il n'y a pas de modification des notes

Les agents salle possèdent deux fonctions principales.

- Récupérer la notes de la salle en fonction des poids sur critères
 - Appel des agents critères pour récupérer les notes modulés par les poids fournis par l'utilisateur puis calcul le score globale de la salle
- Modifier les notes des critères d'une salles en cas de feedback

- Appel des agents critères pour appliquer une modification des notes selon un type de feedback

Les agents critères ne possèdent que deux fonctions.

- Récupérer la note de son critère d'une salle modulé par le poids sur critère
 - Interroge la base de données pour obtenir la note du critère de la salle en question, et multiplie celle ci par le poids avant de retourner le résultat à l'agent salle concerné.
- Modifier la note de son critère d'une salle
 - Deux comportement selon le type de feedback
 - Récupère la base de l'incrément selon le type de feedback et module ce dernier par l'expertise de l'utilisateur
 - Vérifie que la note fournie par l'utilisateur n'est pas dans la marge d'erreur calculée dynamiquement et modifie la base de données en additionnant ou soustrayant de la note présente dans la base de donnée de l'incrément calculé précédemment

Informations complémentaires

Détails sur l'expertise

L'expertise est utilisé à plusieurs reprise dans ce système et permet de moduler les modifications de la base de données.

Elle est renseignée par l'utilisateur via une barre de slide et $expertise \in [1..100]$.

- ❖ Calcul d'une marge dynamique
 - Une note de la base de données n'est modifiée que si la note que l'utilisateur souhaiterait donner s'éloigne d'au moins une certaine marge
 - Plus un utilisateur est expert, moins cet éloignement est grand
 - Moins un utilisateur est expert, plus cet éloignement est grand
 - Actuellement $marge = 20 - (expertise / 10)$
 - $marge \in]10, 20]$
- ❖ Calcul d'un pas d'incrément dynamique selon le type de feedback
 - Plus un utilisateur est expert, plus le pas d'incrément est grand
 - Moins un utilisateur est expert, plus le pas d'incrément est petit
 - Récupère $incrément_{base}$ dans la base de données
 - Actuellement $pas = incrément_{base} * (expertise / 100)$
 - $pas \in]0, 1]$

Au final, on obtient une expertise qui permet bien de moduler les modifications des notes.

Un utilisateur expert peut modifier plus facilement d'un plus grand pas les notes, contrairement à l'utilisateur novice qui aura une plus grande marge et un plus petit pas d'incrémentation.

Contacts

Nom	Rôle	Contact
VILLAIN Edouard	Chef de projet	systeme.aide.decision@gmail.com
DEFONTE Véronique	Responsable communication	systeme.aide.decision@gmail.com
ROLLAND Thomas	Responsable technique	systeme.aide.decision@gmail.com