

demande de soumission

**Bonification d'un système de gestion des abonnements
Drupal 7 – Ubercart**

Groupe TRAQ

**contact
Rémi Guertin
rguertin.com@gmail.com**

Québec, mai 2016

1. Le contexte

Le Groupe TRAQ – Transport sur rail au Québec – est un OBNL qui, depuis 15 ans, fait la promotion du transport ferroviaire ainsi que la promotion du patrimoine ferroviaire du Québec. Outre son colloque annuel, le Groupe publie une revue bimestrielle : *Rail Québec*.

L'année dernière, le Groupe s'est doté d'un nouveau site internet : www.groupe-traq.com. Le site est plutôt un fouillis, notamment parce que certains bénévoles n'ont pas toujours respecté les consignes relativement à la mise en ligne de contenus. Toutefois, le Groupe TRAQ peut dorénavant compter sur une archiviste bénévole qui saura mettre de l'ordre dans le site; le site est développé à la hauteur de nos moyens (bénévolat).

Le site a été réalisé avec Drupal 7; nous sommes en mesure d'effectuer plusieurs opérations avec Drupal : création de types de contenus, taxonomie, mise à jour de modules, *Backup and Migrate*, *Views*, *Rules* (de base), etc. Pour la gestion des abonnements (« rôles »), nous avons retenu *Ubercart*. Toutefois, le système n'est pas parfait et manque de convivialité. De plus, nous ne pouvons pas gérer l'ensemble de nos abonnements en ligne pour diverses raisons; c'est l'objectif de la présente soumission.

2. Le projet : bonifier la gestion des abonnements et des membres

Les abonnements sont actuellement gérés à l'aide d'une feuille Excel. Si cette façon de faire présente une certaine efficacité, elle demeure relativement lourde. Il faut par exemple prendre en note le mode de paiement, sa date de réception et la date à laquelle l'abonnement se termine. De plus, nous sommes confrontés à une pléthore de moyens de paiement (chèque, espèces, carte de crédit (avec la machine manuelle), par notre site, Accès D).

Tous les deux mois, nous postons (avec la revue) un rappel pour les abonnements, car certains de nos membres n'ont pas internet tandis que d'autres oublient de renouveler leur adhésion (note : une semaine avant la fin d'un abonnement *Ubercart* envoie un courriel de rappel à ceux qui ont payé par le site).

Nous utilisons *Ubercart* à la fois pour la vente de « rôles » et pour notre boutique en ligne. C'est la simplicité d'*Ubercart* qui nous a incités à faire ce choix; la vente de « rôles » avec *Commerce* implique une quantité trop importante de modules complémentaires (mises à jour de modules plus longues). De plus, notre boutique en ligne reste simple : vente de revues (anciens numéros), de livres, quelques objets souvenirs.

Le système actuel fonctionne : les membres peuvent acquérir un compte – un « rôle » Drupal – qui donne accès à certaines fonctionnalités. De plus, l'acheteur reçoit des courriels de confirmation et un reçu. Toutefois, le processus d'achat manque de franchement de clarté (renouvellement versus nouvel abonnement, etc.). Aussi, des membres aimeraient pouvoir faire un don au moment où ils renouvellent leur abonnement. Nous voulons gérer les abonnements à même notre plate-forme (accessible à plus d'un gestionnaire à la fois) et il serait bien de pouvoir être payé par chèque *via* la plate-forme (sous-module).

Il est à noter qu'un individu qui s'ouvre un compte sur la plate-forme sans acquérir de compte ne peut accéder aux fonctionnalités du site. Il faut acheter un « rôle » pour commenter les « nœuds » ou pour participer au forum. Les trois « rôles » donnent accès aux mêmes fonctionnalités, à l'exception du compte « corporatif » qui donne accès à un « bottin » de compagnies.

Bref, nous aimerions bonifier notre boutique en ligne de façon à centraliser la gestion des membres et surtout de façon à simplifier quelque peu cette gestion, en dépit d'une pléthore de modes de paiement.

3. L'objet de la soumission

A. Boutique en ligne

1. Rendre le processus d'abonnement (achat/renouvellement d'un « rôle ») plus clair.
2. Permettre l'ajout d'un don lors d'un abonnement et/ou d'un second compte;
3. Vérifier, ajouter et configurer le module pour Poste Canada (selon les coûts);
4. Permettre l'inscription au colloque; faire la liste des inscrits et l'imprimer.

Notes

1. Actuellement, le processus d'acquisition d'un nouveau compte est quelque peu ambigu. Il est nécessaire de faire en sorte que les choses soient plus claires. Actuellement, le processus d'ouverture de compte et d'achat/renouvellement d'un compte n'est pas clair. La facture comporte des liens qui ne sont pas clairs et il appert que les membres n'ont pas accès à leur facture en ligne (question de « permissions »?). Les membres doivent pouvoir mettre à jour par eux-mêmes, à partir de leur profil, les coordonnées qui apparaissent sur leur facture/compte. Nous avons ajouté le module pour les coordonnées avec pour résultat que les membres doivent inscrire deux fois leurs coordonnées.

2. Il y a trois types d'abonnements au Groupe TRAQ : « soutien » (sans revue); « régulier » (avec revue); « corporatif » (avec revue). Certains membres vont acquérir deux comptes : « soutien » pour un ami et « régulier » pour eux-mêmes. De fait, il faudrait avoir la possibilité de « faire un autre achat avant de passer à la caisse » (essentiellement une question d'ajustement de la boutique). Il est à noter que des membres ont fait l'acquisition séparément de deux comptes (« rôles ») sans erreurs de la part du système.

Pour les dons, il y a quatre modules qui pourraient être utilisés – *Ubercart donation product*, *Ubercart Add donation*, *Ubercart Donation incentive*, *UC Variable Price* –, mais deux peuvent répondre à deux cas de figure spécifiques : l'utilisateur qui veut seulement faire un don; celui qui veut faire un don au moment de faire un achat (*donation product* et *Add donation*). Ces deux modules existent en version D7, mais vont devoir être vérifiés puisque cette version a été élaborée par la communauté (version de développement). Éventuellement, redonner à la communauté avec une note que ces mises à jour ont été payées par le Groupe TRAQ.

La question est de savoir si ces modules peuvent ou non interférer entre eux et avec l'acquisition d'un « rôle ». La facture émise par *Ubercart* comporte des liens ambigus.

3. Le module pour Poste Canada (drupal.org/project/uc_canada_post) tient compte de leur nouvelle API. Toutefois, et selon les coûts que cela représente, comme il s'agit d'une version de développement, il serait intéressant de vérifier le module et de redonner à la communauté une « version 1.0 ». Une autre approche serait d'utiliser un tarif uniformisé (sous-module *flat rate*) advenant le cas où le module de Poste Canada pose problème.

4. Quelques semaines avant le début du colloque, nous adaptons le contenu d'un bloc (date, titre, etc.) que rendons visible en page d'accueil. Le bloc contient des indications sur le colloque ainsi que des liens pour télécharger le programme. Il contenait également un lien vers le service *Eventbrite* (gestion des inscriptions). Nous voulons dorénavant gérer les inscriptions à partir de notre plate-forme.

Les inscriptions sont assez simples dans la mesure où il nous faut un « produit » pour vendre trois types d'inscriptions (6, car il y aura une version anglaise) : assister aux 2 jours du colloque; le premier jour seulement ; la dernière journée seulement (nous sommes en mesure de créer ces « produits »). Passé une certaine date (qui change chaque année), le prix augmente (retardataires).

Que nous soyons obligés de changer le prix manuellement à la fin de la période d'inscription « standard » ne nous pose pas de problèmes dans la mesure où c'est quelque chose qui est à faire qu'une seule fois par année. L'important est de faire en sorte que les congressistes

reçoivent un courriel confirmant leur inscription, un courriel qui doit contenir un reçu (déjà émis par *Ubercart*). Un autre courriel doit être envoyé à une tierce personne pour l'avertir des nouvelles inscriptions. L'émission de ces courriels peut se faire probablement par *Rules*. Quant à la facture, il faudrait ajouter un champ pour le « titre » d'une personne au sein de sa compagnie (informations recueillies par *Ubercart*).

Il faut une « vue » (ou un module selon la complexité) qui fait la liste des congressistes qui se sont inscrits. Cette liste doit pouvoir être exportée en Excel et être imprimable à partir du site; elle doit contenir les données suivantes : nom, prénom, courriel, société/compagnie, titre, téléphone, coordonnées; déjà, *Ubercart* recueille certaines données lors d'un achat. Aussi, il faut pouvoir contrôler qui a accès à cette liste.

L'impression est nécessaire, car le jour du colloque, les responsables ont besoin de contrôler les congressistes à leur arrivée.

B. Gestion des abonnements

1. Avoir un « tableau de bord » facilitant la gestion des membres (recherche, ajout, renouvellement, retrait, etc.). Ce tableau de bord doit être jumelé avec *Ubercart* puisque c'est la boutique en ligne qui gère les abonnements (vente de « rôles »);
2. Faire en sorte que la navigation entre le tableau de bord les comptes soit *fluides* (prévoir peut-être des fenêtres contextuelles (*pop-up*) donnant accès à certaines fonctionnalités pour éviter les va-et-vient entre un compte et le tableau de bord;
3. Ce tableau de bord doit permettre de faire la liste des membres qui doivent renouveler leur abonnement dans les deux mois à venir (à partir d'une date donnée);
4. il faut pouvoir imprimer les factures *Ubercart* des membres qui doivent renouveler leur abonnement (impression « en vrac » de toutes les factures figurants dans la liste du point 3);
5. il faut pouvoir imprimer les noms et adresses de chaque membre devant renouveler (point 3) sur des étiquettes (selon un gabarit standard). Il existe au moins deux modules d'impression d'étiquettes qui pourraient peut-être faire l'affaire (sinon faire un gabarit en CSS?);
6. Produire un pense-bête pour aider le gestionnaire à s'y retrouver dans *Ubercart* et dans la gestion des abonnements (tableau de bord).

L'interface doit présenter une certaine convivialité et une certaine lisibilité tout en minimisant lorsque possible l'utilisation de module sur mesure (ex. : les *pop-up* peuvent être faits avec *Colobox node* ou encore avec la fonction intégrée dans *Ctools*, etc.). Le module *Total Control Admin Dashboard* pourrait représenter un point de départ pour la création du « tableau de bord ».

Certaines données doivent être protégées, car celui qui gère les abonnements n'est pas celui qui fait les impressions (notamment, les numéros de carte de crédit puisque nous utilisons toujours une « machine manuelle »). Il faut noter que pour l'heure, nous utilisons le module *User Permissions* (qui peut être remplacé par un autre).

4. Notes complémentaires

Pour des raisons d'économie, nous avons une préférence pour l'emploi de modules existants; cela va faciliter la mise à jour. De même, nous avons une préférence pour l'emploi de *Views*, de *Flag*, de *Rules* et des fonctionnalités de *Ctools* avant la programmation d'un module (lorsque possible bien entendu). Bref, dans la mesure du possible, éviter de réinventer la roue.

Inclure un « plan d'ajustements » si la bonification de notre système de gestion devait produire des « messages d'erreurs » à l'usage (*bogues*).

Les modules utilisés

Advanced forum

Back up and migrate

Colorbox / colobox node

Comment notify

Entity API

Entity tokens

Field formatter settings API

Hide block by role (sera probablement inutile avec un module de dons)

Job Scheduler

Librairies

Pathauto

Reg With Pic

Simple Google Maps

Token

Transliteration

User restrictions

Address field

Entity Reference

Entity Reference View Widget

Grid Field Formatter
User Permissions
Captcha
Honeypot
Calendar (possiblement avec le mauvais format de date; D8)
Date, Date all day, Date API, Date Pop up, Date Views
Video Filter
Feeds, Feeds Admin UI, Feeds XLS
CKEditor (bug un peu)
External Links
jQuery Update
Entity Translation, Entity translation update
Localization update
Imagecache Actions, Canvas actions
Rules, Rules UI
Metatag
Ubercart : Cart, order, Product, Store, Payment, Reports, Roles, Add to Cart Tweaks, CSV export,
Payment method pack, Paypal
Et bien entendu : Views.