

w.c.s. - Development #485

Affichage complet d'un workflow

10 juin 2011 23:58 - Thomas Noël

Statut:	Fermé	Début:	10 juin 2011
Priorité:	Normal	Echéance:	
Assigné à:		% réalisé:	100%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	
Patch proposé:			
Description En suivant une démo faite par Victor à un client sur les workflows, je me suis dit qu'une page "vue complète" d'un workflow (affichant tous les statuts et toutes les actions de chaque statut) pourrait permettre de mieux appréhender l'idée, et surtout, sans doute, de mieux déboguer un workflow complexe lorsqu'on est admin. Voici donc une proposition "preuve de concept" en fichier attaché, d'une "complete view" sur un workflow, programmée vite fait dans le train du retour. Il faudrait sans doute revoir le style, c'est pas mon fort.			
Demandes liées: Lié à w.c.s. - Development #12996: vue étendue d'un workflow			
		Fermé	30 août 2016

Historique

#1 - 11 juin 2011 11:52 - Frédéric Péters

J'ai un truc pour faire passer le "réalisé à 0%", je le donne ? Allez...

L'idée d'un workflow, c'est qu'il y a parcours, qui n'est pas forcément passer sur chaque état les uns après les autres, et donc je me demandais dans quelle mesure visualiser ça, comment relier les états entre eux, comment utiliser les attributs "status" de certaines actions pour "tracer", "marquer" ça.

e.g.

```
Nouveau → En cours → Validé
                        → Refusé
```

Mais il y a des workflows plus compliqués, qui bouclent entre états, du coup je me demandais si le plus simple, ce ne serait pas à ta liste d'ajouter les pointeurs vers les états qui peuvent suivre. (et du coup ça fait que le "réalisé" ne passe pas à 0%, cool).

Et pour passer à l'étape de visualisation des transitions, on pourrait "juste" faire appel à graphviz, parce que c'est super, il existe une sortie "imagemap", <http://www.graphviz.org/doc/info/output.html#d:imap> (ou du SVG, on a dépassé les années 90...).

#2 - 11 juin 2011 15:50 - Thomas Noël

- % réalisé changé de 80 à 10

J'ai laissé 10% de réalisé pour la volonté qui nous anime ;)

Pour un rendu un peu plus graphique, je pencherai plutôt vers un diagramme de séquence. Pour un saut entre statut, il faut afficher qui a le droit de faire ce saut (on l'indique sur la flèche). Reste que graphviz n'a pas l'air de savoir faire ça nativement, faut googliser un peu (si y'a rien, on laisse tomber, y'a d'autres dev + importants)

#3 - 11 juin 2011 20:58 - Benjamin Dauvergne

J'ai fait mumuse:

<http://graphviz.dev.au-quotidien.com/admin/workflows/1/>

<http://perso.entrouvert.org/~bdauvergne/git/cgi?url=wcs-perso/log/&h=wf-graphviz>

#4 - 11 juin 2011 22:36 - Benjamin Dauvergne

Thomas Noël a écrit :

Pour un rendu un peu plus graphique, je pencherai plutôt vers un diagramme de séquence. Pour un saut entre statut, il faut afficher qui a le droit de faire ce saut (on l'indique sur la flèche). Reste que graphviz n'a pas l'air de savoir faire ça nativement, faut googliser un peu (si y'a rien, on laisse tomber, y'a d'autres dev + importants)

Un diagramme de séquence ça ne représente en général qu'une seule séquence parmi la multitude que plusieurs machines à état (il y a communication) peuvent produire. Ce qu'il faut ici c'est bien un graphe.

#5 - 12 juin 2011 01:43 - Thomas Noël

Benjamin Dauvergne a écrit :

J'ai fait mumuse:

<http://graphviz.dev.au-quotidien.com/admin/workflows/1/>

<http://perso.entrouvert.org/~bdauvergne/git/cgit.cgi?url=wcs-perso/log/&h=wf-graphviz>

Ca jette. Trop fort. J'ai quelques doutes sur la façon dont graphviz tentera d'afficher des trucs plus complexes (il a parfois des choix bizarre), mais je les balaye. Mais globalement, je pense Victor va te demander en mariage.

Sinon, peut-être, <http://networkx.lanl.gov/pygraphviz/> ? (ça semble bien maintenu)

#6 - 12 juin 2011 01:48 - Thomas Noël

Benjamin Dauvergne a écrit :

Un diagramme de séquence ça ne représente en général qu'une seule séquence parmi la multitude que plusieurs machines à état (il y a communication) peuvent produire. Ce qu'il faut ici c'est bien un graphe.

Oui, en fait il faudrait presque un "mix" qui donne quand même une certaine idée de l'ordre chronologique dans lequel les choses peuvent se passer. Même si c'est vrai que des aller-retour sur un statut, ça met tout par terre. Bon, bref, digraph fait le boulot àmha.

#7 - 12 juin 2011 08:24 - Frédéric Péters

Je suis aussi curieux de voir si pygraphviz pourrait être utilisé, même si pour le peu de code que cela représente actuellement ce n'est sans doute pas absolument nécessaire (surtout si la version dans Lenny n'est pas suffisante).

Plus important par contre, ce serait de laisser à la CSS le soin d'appliquer du style aux éléments du SVG, il s'y retrouve notamment des "style="font-family:Times New Roman;font-size:14.00;", ce qui n'est pas fabuleux (du tout). (et les *salmon* ...).

Aussi ce serait peut-être judicieux de ne pas avoir les libellés complets, qui doivent s'étirer sur deux lignes (avec en conséquence un \n qui apparaît tel quel dans le tooltip (tooltips qui n'amènent pas plus d'infos, par ailleurs)); par exemple « Choix "Accepter" par destinataire » pourrait très bien être résumé en « Accepter ».

#8 - 12 juin 2011 14:49 - Thomas Noël

Frédéric Péters a écrit :

Aussi ce serait peut-être judicieux de ne pas avoir les libellés complets, qui doivent s'étirer sur deux lignes (avec en conséquence un \n qui apparaît tel quel dans le tooltip (tooltips qui n'amènent pas plus d'infos, par ailleurs)); par exemple « Choix "Accepter" par destinataire » pourrait très bien être résumé en « Accepter ».

MMhh je pense au contraire qu'il faut bien qu'on voit **qui** a le droit de cliquer sur les choix, quel(s) rôle(s) a le droit de faire bouger un statut. C'est ça qui permet de bien comprendre un workflow (en tout cas à Caen c'était ça le truc : quand est-ce que le médecin peut cliquer, quand tel autre le peut, etc).

En fait, quelque part, ça serait sympa de permettre quelques paramétrages d'affichage du workflow.

#9 - 12 juin 2011 15:07 - Frédéric Péters

Ok, mais un affichage comme ça laisse des libellés trop longs, peut-être qu'on pourrait imaginer colorier les arcs différemment suivant les rôles (et inclure une légende sur le côté) ?

#10 - 12 juin 2011 17:21 - Benjamin Dauvergne

Frédéric Péters a écrit :

Plus important par contre, ce serait de laisser à la CSS le soin d'appliquer du style aux éléments du SVG, il s'y retrouve notamment des "style="font-family:Times New Roman;font-size:14.00;", ce qui n'est pas fabuleux (du tout). (et les *salmon* ...).

Aussi ce serait peut-être judicieux de ne pas avoir les libellés complets, qui doivent s'étirer sur deux lignes (avec en conséquence un \n qui apparaît tel quel dans le tooltip (tooltips qui n'amènent pas plus d'infos, par ailleurs)); par exemple « Choix "Accepter" par destinataire » pourrait très bien être résumé en « Accepter ».

Je suis tout à fait d'accord sur les deux points. Pour le premier on pourrait remplacer le `color="salmon"` par un `id="page-subject"` qui forcerait un certain ID pour l'élément central du graphe permettant de le styler en css. Malheureusement j'ai l'impression que c'est difficile de lui dire de ne pas émettre du tout l'attribut `style` sur les polygones, même en enlevant toutes mes déclarations, il envoie quand même un `style`. Je ne sais pas si en CSS on peut forcer le style ensuite. Il y a la possibilité de filer une feuille de style XSLT à graphviz pour pré-traitement avant la sortie du fichier SVG.

Pour les tooltips je peux forcer le tooltip à vide.

Mais en fait je suis actuellement limité par la version de graphviz sur ma machine et sur macke (2.20) je ne peux donc tester ni l'attribut `"id"`, ni l'attribut `"tooltip"` (qui existe en 2.20 mais qui semble ignoré par la sortie svg). En squeeze graphviz est en version 2.26, il y a des chances que ça marche.

- <http://www.graphviz.org/content/attrs#did>
- <http://www.graphviz.org/content/attrs#dstylesheet>
- <http://www.graphviz.org/content/attrs#dtooltip>

Concernant les noms, on pourrait faire comme cela:

- `label = item.label or item.description`
- `color = color_for_role(item.by) if item.by else "default"`

Pareil ici au lieu d'utiliser une couleur ce serait bien de pouvoir passer une classe CSS mais malheureusement je ne vois pas comment faire avec graphviz, sauf à utiliser une feuille de style XSLT (mais là je passe mon tour) qui convertirait les `style="filled:xyz"` en `class="color-xyz"`.

Ensuite en attribuant de façon persistante une classe de couleur à chaque rôle, on aurait alors des classes prédéfinies `color1` à `color20`, permettant de style la palette comme on veut, ça couvrirait 99,9% des besoins.

#11 - 14 juin 2011 18:18 - Benjamin Dauvergne

J'ai ajouté du code utilisant `ElementTree` pour nettoyer le contenu SVG avant affichage, il peut maintenant être stylé indépendamment, j'ai ajouté les règles CSS nécessaires dans `dc2/admin.css`:

- <http://perso.entrouvert.org/~bdauvergne/git/cgit.cgi?url=wcs-perso/commit/&h=wf-graphviz&id=addfa8493792f8639c669ff7dd4374ca89e3da64>
- <http://perso.entrouvert.org/~bdauvergne/git/cgit.cgi?url=wcs-perso/commit/&h=wf-graphviz&id=7c8758a89cce682e6c7cc758725845c350186fa2>

Par contre j'ai du faire marche arrière concernant l'utilisation de l'attribut couleur pour définir la classe des éléments, car on ne peut pas mettre ce qu'on veut comme couleur, il faut que la couleur soit présente dans un des catalogues de couleurs reconnu par graphviz :(Par conséquent j'ai laissé la couleur `"salmon"` et je la recherche explicitement pour appliquer la classe `"page-subject"` à l'élément concerné.

J'ai aussi remplacé les labels un peu longs par un truc un peu plus court quand c'est possible:

- <http://perso.entrouvert.org/~bdauvergne/git/cgit.cgi?url=wcs-perso/commit/&h=wf-graphviz&id=93e2ef637c9f9b98b2d7b1f96db321e5b8dcc615>
- <http://perso.entrouvert.org/~bdauvergne/git/cgit.cgi?url=wcs-perso/commit/&h=wf-graphviz&id=7c8758a89cce682e6c7cc758725845c350186fa2>

L'idée d'attribuer une couleur en fonction du rôle il faut l'oublier pour la même raison que celle de l'attribution des classes, ou alors il faut choisir une dizaine de couleur comprise dans le catalogue de couleur de X Windows et leur attribuer une classe équivalente `role-1`, `role-2`, etc.. qu'on pourra styler.

#12 - 15 juin 2011 10:37 - Thomas Noël

Benjamin Dauvergne a écrit :

Par contre j'ai du faire marche arrière concernant l'utilisation de l'attribut couleur pour définir la classe des éléments,

en revanche on peut mettre ce qu'on veut dans le `fontname`. on pourrait remplacer le `fontname=` par `class=`, et zou... ou bien ?

--

Thomas

#13 - 15 juin 2011 10:59 - Benjamin Dauvergne

redmine@entrouvert.com écrivait:

Benjamin Dauvergne a écrit :

Par contre j'ai du faire marche arrière concernant l'utilisation de l'attribut couleur pour définir la classe des éléments,

en revanche on peut mettre ce qu'on veut dans le `fontname`. on pourrait remplacer le `fontname=` par `class=`, et zou... ou bien ?

Excellente idée !

#14 - 06 avril 2012 12:18 - Thomas Noël

- Version cible mis à Au-quotidien 2012.3

#15 - 06 avril 2012 13:30 - Thomas Noël

- Priorité changé de Bas à Normal

#16 - 16 août 2012 11:39 - Frédéric Péters

- Statut changé de Solution déployée à Résolu (à déployer)

- % réalisé changé de 10 à 100

Je viens d'intégrer tout ça et d'y apporter une série de petites modifications, principalement esthétiques. Par contre je n'ai pas vraiment réussi à avoir la main sur le style des éléments en passant par des classes, j'ai donc laissé ça ainsi pour le moment. Il y aura sans doute des ajustements à faire quand ce sera exposé à une plus grande série de workflows, on verra alors.

#17 - 20 novembre 2013 11:57 - Frédéric Péters

- Statut changé de Résolu (à déployer) à Fermé

#18 - 26 août 2015 16:53 - Thomas Noël

- Version cible Au-quotidien 2012.3 supprimé

#19 - 30 août 2016 17:45 - Frédéric Péters

- Lié à Development #12996: vue étendue d'un workflow ajouté

Fichiers

workflow-complete-view.patch	2,5 ko	10 juin 2011	Thomas Noël
------------------------------	--------	--------------	-------------