Avenir des logiciels de montage vidéo libres

En milieu professionnel

12 septembre 2011

Thibault Saunier

Collabora Ltd. And EPSI Lyon

Table des matières

Chapitre 1

Analyse du monde de l'édition vidéo professionnel

Contents

Le montage vidéo professionnel est un domaine très vaste, et l'on peut s'attendre à ce que la palette de fonctionnalités nécessaires à la creation des différents formats d'œuvres audiovisuelles varient fortement en fonction du type de contenu. Afin d'étudier les possibilités d'avenir des logiciels libres dans ce domaine, il nous faut définir, pour en connaître les différents besoins :

– les cas d'utilisation (plus communément appelées use cases)
– les fonctionnalités qui en découlent

Nous allons donc définir les principaux cas d'utilisation en fonction des différents formats de productions audiovisuelles et ainsi en déduire les fonctionnalités nécessaires pour répondre à ces cas d'utilisations.

Ensuite nous analyserons la base commune des fonctionnalités nécessaires à la réalisation de ces différents types de production. Pour finir nous verrons s'il existe une diversité dans les besoins, et essayerons de trouver les fonctionnalités qui sont propres à chaque type de production. Cette première analyse a pour but de clarifier les besoins des professionnels afin de déterminer par la suite quels sont ceux auxquels les logiciels libres répondent déjà, ceux auxquels on peut prétendre répondre dans un futur proche, et ceux qui sont hors du scope actuel des technologies libres.

1.1 Les bases de l'édition vidéo

Tout d'abord, il est évident que, pour qu'un logiciel de montage puisse répondre aux besoins des professionnels, les fonctionnalités basiques de l'édition vidéo non linéaire doivent être couvertes. Cette partie a pour but de définir quelles sont ces fonctionnalités, et de les expliquer succinctement :

1.1.1 Définition des termes techniques

Du fait de l'importance des termes suivant pour la compréhension de ce document, il est nécessaire qu'ils soient définis au sein même de celui-ci.

Les Footages Les footages correspondent à toutes les sources brutes qui ont été enregistrées et a partir desquelles le monteur va créer le rendu final de l'œuvre audiovisuelle.

Les clips Les clips correspondent dans les faits à un footage édité (retouche des couleurs, modification de la durée, ajout d'effets...) par le monteur afin de l'utiliser dans le contexte précis d'une œuvre finale.

Les templates Dans l'édition video, on parle de template pour définir un moule de montage. Il permet au monteur de monter très rapidement des oeuvres en s'assurant que le rendu rentre dans un cadre défini précédemment.

Colorimétrie (retouche des couleurs) En édition vidéo la colorimétrie est l'art de retoucher les couleurs, les étalonner au travers des différents clips.

Les effets vidéos Les effets vidéo sont des effets visuel qui permette de modifier l'image d'une vidéo de manière simple (à l'inverse des effets spéciaux qui modifie la vidéo de manière plus complexe.

Fig. 1.1: Les keyframes

Les keyframes Les keyframes définissent le début et la fin d'une animation, en particulier dans le cadre d'effet, de texte en mouvement au dessus d'une vidéo (dans le cadre de titres, sous-titres).

Speed control et time remmaping Le speed control permet de modifier la vitesse de lecture d'un clip dans la timeline (ralentir ou accélérer). Le time remapping est une technique avancée de speed control, et permet de changer la vitesse de lecture de partie de clip, et ainsi d' accélérer ou de ralentir des parties d'un même clip. Cette technique est couplée au keyframes afin d'obtenir le résultat souhaité.

Gestion des Footages Un logiciel d'édition vidéo doit permettre d'importer les Footages à partir desquels on veut faire le montage, c'est à dire les fichiers vidéos, audios, et images avec lesquels on travaille. Il doit être possible de prévisualiser ces clips.

1.1.2 Definition du concept d'édition timeline

La timeline est la partie de l'interface dans laquelle on va disposer les différents clips. Il s'agit du concept de base de l'édition vidéo non linéaire. Dans le cadre de l'édition timeline quelques fonctions sont absolument indispensables, et il est nécessaire de comprendre ces différents concepts pour comprendre la suite de ce document :

Découpages des clips La technique du decoupage de clip permet de diviser un footage en plusieurs parties afin de pouvoir les utiliser de manière indépendante.

Fig. 1.2: Spliting

Unlinking de la piste audio et de la piste vidéo Le fait de "delier" les clips permet de gérer de manière désynchronisée le son et la video.

Fig. 1.3: Unlinking

Gestion des "in point" et "out point" des clips Permet de définir la partie d'un footage à utiliser dans le montage final. Cela permet donc de redéfinir la longueur d'un clip dans la timeline, en ne jouant pas le debut ou la fin de celui-ci.

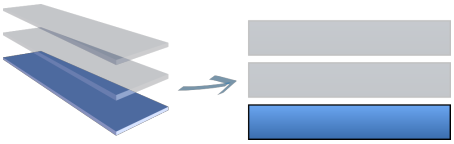

Fig. 1.4: Les layer

Notion de layer La notion de layer est essentielle dans l'édition video avancée dans la timeline : mixer plusieurs sources et ajouter des titres dépend de cette fonctionnalité. Afin de comprendre, il est plus simple de faire la comparaison avec de la peinture sur verre : on superpose plusieurs vitres les unes au dessus des autres, chacune de ces vitres représentant un layer. C'est la superposition des dessins de chacune des plaques qui va nous révéler le résultat final. De plus, il existe dans les layers une notion d'opacité , ce qui permet d'atténuer ou de révéler intégralement les dessins des plaques inférieures.

1.2 Définition du marché par segment

Dans un premier temps, nous allons définir et analyser les différents formats de productions audiovisuelles professionnelles. Nous avons interviewé un certain nombre de monteurs professionnels, afin de lister leurs besoins, (annexes 1) et en essayant de couvrir le maximum de champs de l'édition vidéo. Nous avons pu récolter des informations provenant de monteurs de clips vidéos, de courts métrage, de publicités de documentaire, de série télévisé, et de reportages.

La littérature dans la matière (En particulier [3]) nous propose de faire une nette distinction entre les deux segments du marché que sont :

- le monde du contenu post-produit : il s'agit de contenu dont la qualité de montage final est très importante. Celui-ci peut être de courte durée, tels que les clips vidéos ou publicités, ou bien de longue durée, tels que les films ou les séries télévisées. Mais il faut toutefois faire une différence entre ces derniers puisque la qualité du rendu final des films implique d'autres standards en terme de montage
- le monde de la production diffusée : il s'agit du contenu retransmis à la fois, sur internet, et sur les chaînes de télévisions et dont la création et la retransmission rapide impliquent des moyens spéciaux qui permettent de créer et retransmettre le contenu dans un temps restreint, voir en direct.

Certes les deux mondes ont des contenus différents, mais surtout ils ont des contraintes différentes, ce qui implique des divergences importantes en terme de besoin de fonctionnalités. Nous allons donc nous intéresser à ces deux domaines et découper notre analyse à partir de cette distinction. Tout d'abord, nous nous intéresserons aux fonctionnalités logicielles nécessaires à la production de contenu post-produit, par la suite, nous analyserons les besoins intrinsèques à la production de contenu visant le monde de la vidéo diffusée. Puis nous essayerons de voir où se situe la frontière entre ces deux mondes afin de pouvoir par la suite nous rendre compte de ce que l'investissement sur ces marchés implique pour les logiciels de montage vidéo libres.

Le monde du contenu post-produit

Le monde du contenu post produit est assez vaste, au première abord il peut apparaître comme étant tout le contenu qui n'est pas diffusé instantanément. Dans les faits, la distinction

est plus complexe, et il s'agit d'œuvres audiovisuelles dont le temps de post production n'est pas un critère de première d'importance pour le choix des moyens mis en place sur ce sujet.

De ce fait, les formats suivants peuvent être considérés comme étant post produits :

Les courts métrages

Les courts métrages concentrent une histoire en moins de 35 minutes. Ils sont donc soumis à des contraintes importantes. Ils répondent à une exigence de concision et il est donc intéressant de se poser la question pour savoir si dans ce genre d'œuvre les monteurs utilisent des techniques qui permettent de les rendre plus dynamiques et si des fonctionnalités spéciales sont utilisées dans ce but.

Dans la production de ce type d'œuvre, les interviews nous ont révélé des fonctionnalités indispensables telles que :

- Transition (fading en priorité)
- Effets basiques (par exemple le passage en noir et blanc)
- Time remmaping
- Retouche des couleurs
- Création et ajout de génériques

Les publicités

La publicité peut s'apparenter au court métrage puisqu'il s'agit de création courte et généralement dynamique mais dont l'objectif est différent. Pour atteindre cet objectif (attirer des consommateurs), les monteurs utilisent des techniques spéciales mais les fonctionnalités du logiciel nécessaires restent identiques à celles du court métrage.

En revanche, la qualité du rendu est très importante : ainsi des logiciels spécialisés sont fréquemment utilisés afin de créer le contenu (audio, effets, images...).

Les clips vidéos

Le clip vidéo est un contenu visuel qui a pour but d'illustrer une musique. Ce type de vidéos utilise souvent beaucoup d'effets spéciaux, et demande à priori une très grande précision au niveau de la synchronisation entre le son et l'image. La track audio dans de telle production sera de préférence effectuée avec un logiciel dédié à cet effet. Pour résumer, les fonctionnalités nécessaires sont :

– Création de titres complexes (Titre en mouvement, etc...)
– Ajout de titres
– Ajout d'effets
– Utilisation avancé des keyframes
– Time remapping

Les films

La production cinématographique bénéficie de budgets beaucoup plus élevés. Les techniques employés dans le cadre de la post production sont plus complexes et permettent de gérer avec soin la qualité du rendu.

Il n'a pas été possible d'interviewer de monteur de film jusqu'à maintenant, mais le livre "The technique of film and video editing, History, Theory, and Practice" [1] est un bon point de départ pour comprendre le montage cinématographique et la très grande influence qu'il a sur les autres types de productions audiovisuelles. On peut considérer le film comme étant une oeuvre audiovisuelle par excellence.

Dans le monde du cinéma, le logiciel de montage vidéo est l'un des logiciels parmi un système connecté de logiciel de post production. Des spécialistes de différents domaines créent les parties du film, et le monteur a pour mission de lier tout ces éléments au travers du logiciel de montage. Les logiciels de post production sont entre autres :

– Éditeur de son
– Création d'effet
– Retouche d'image
– Création d'animation
– ...

Les logiciels à visée professionnelle ne sont donc pas forcément utilisables dans le monde de la création cinématographique. Il conviendra de faire une réelle différence entre ces deux univers du montage vidéo.

Ce qui résulte du fait que le logiciel de montage vidéo à proprement parler ne demande pas vraiment de fonctionnalités très évoluées, la base de l'édition et la possibilité d'organiser l'immense quantité de Footages de manière efficace semblent être le point essentiel des logiciel d'édition vidéo pour répondre de manière efficace aux besoins des professionnels dans ce domaine.

Les autres logiciels de post production sont bien évidemment aussi nécessaires afin de permettre de faire le montage de films. Ce document n'a pas pour but de détailler ces autres logiciels.

Une autre caractéristique de la production cinématographique, qui est une conséquence directe de l'impératif de qualité irréprochable, réside dans le fait que les logiciels de montage doivent permettre de visualiser chaque image du film de manière très précise (le montage de film se fait dans certain cas en choisissant chaque image depuis un tableau de frames).

Bien que ne demandant pas vraiment de fonctionnalités très avancées, la création de film a des besoins assez évoluées en ce qui concerne le logiciel de montage :

- Organisation très avancée des Footages
- Création et ajout de générique
- Passerelles avec le reste des logiciels de post production
- Preview de chaque frame dans le détail

Les séries télévisées

Le niveau de qualité des séries télévisées n'étant pas aussi élevé que pour le montage des films, les traitements sont la plupart du temps réalisés directement dans le logiciel de montage même. Cela implique un nombre de fonctionnalités plus important avec comme nécessité :

- Création et ajout de titre
- Création et ajout de générique
- Retouche des couleurs

Les documentaires

Le documentaire est en général assez sobre en terme de montage. Il réside en général dans le logiciel de montage, mais ne demande pas de fonctionnalités spéciales. Les fonctionnalités utilisées pour produire ce type d'œuvre sont :

- Création et ajout de titre
- Création et ajout de génériques
- Retouche des couleurs
- Utilisation des keyframes
- Transition smpte

Le monde du contenu diffusé

La plupart du contenu post produit est par la suite diffusé, la différence que l'on fait ici entre ces deux types de production réside dans le temps de la post production. Dans le cas des journaux télévisés, émission de télé, la post production est soit totalement inexistante (dans le cas du direct), soit très courte, dans le cadre de reportages, jeux télévisés et autres types de production visant spécifiquement la télévision.

Les émissions télévisées

Suivant leur mode de production, les émissions de télévision peuvent être classées soit dans le contenu post-produit soit dans le contenu diffusé. Elles sont en général diffusées très rapidement après la création du contenu (si ce n'est en direct) et c'est la raison pour laquelle nous les considérons comme du contenu diffusé. De plus le fait qu'elles soient produites exclusivement pour la diffusion (aucune commercialisation matérielle n'en est faite), cette classification paraît naturelle.

Du fait de leur temps de production très réduit, les principales fonctionnalités en terme de logiciel de montage sont :

- Fonctionnalité de template qui permet d'avoir un cadre général de montage de présentations, au moment voulu et ainsi faire le montage en direct
- Titres

Bien évidemment, dans le cadre de la création de template, les transition “smpte” et les effets simples sont généralement utilisés. Mais il n'est pas rare que les template à proprement parler ne soient pas créés dans le logiciel de montage, mais plutôt dans d'autres logiciels de création de contenu audiovisuel.

Évènements spéciaux (sportif, d'actualité...)

En principe ce type de production audiovisuelle n'est pas post-produit. Il s'agit de production instantanée, et pour ce type de contenu, l'outil de montage non linéaire doit permettre de donner une impression de contenu post-produit alors qu'il n'en est rien. Les fonctionnalités nécessaires sont assez similaires à celles dont on aurait besoin pour produire des émissions de télévision.

De plus, l'acquisition étant aussi fait en direct, il doit être possible d'intégrer le logiciel du montage dans le système de capture d'image et de son.

De même que pour les émissions de télé, les template sont généralement produits avec des logiciels dédiés à cet effet.

1.2.1 Analyse des fonctionnalités communes

On s'aperçoit donc que de nombreuses fonctionnalités sont communes aux différents types d'œuvres. Il convient de détailler chacune de ces fonctionnalités afin de nous rendre compte de ce qu'elles impliquent en terme de logiciel de montage.

Création et ajout de titre

Cette fonctionnalité est utilisée dans la création de plusieurs types de contenu :

- Séries télévisés
- Documentaires
- Clips vidéos

Bien que cette fonctionnalité est utilisée dans différents types de contenu, le logiciel sera le résultat de différents paramètres. Par exemple, dans une série télévisée le travail sur les titres sera assez limité : on aura souvent une vidéo en arrière-plan et un titre qui fera un fondu arrière. En revanche dans le cadre de clips vidéo, on verra fréquemment le titre en mouvement sur le

rythme de la musique par exemple. Il est nécessaire de tout mettre en oeuvre pour répondre à la diversité de ces besoins mais il sera plus difficile aussi bien en terme de backend qu'en terme d'interface utilisateur de répondre aux besoins les plus spécifiques.

Création et ajout de générique

La création de générique est une fonctionnalité indispensable, à laquelle de nombreux monteurs (en particulier professionnels) font appel. Cette fonctionnalité en terme de backend est similaire à celle des titres puisqu'il s'agit ni plus ni moins d'ajouter du texte au dessus d'un fond qu'il soit animé ou non. Mais en terme d'interface utilisateur, , il s'agit de deux fonctionnalités différentes puisque par définition, le générique est un texte qui défile dans une très grande majorité des cas, de haut en bas.

Cette fonctionnalité est l'une des plus basiques si l'on veut pouvoir répondre aux besoins des professionnels. Elle est utilisée dans la plupart des créations vidéo et doit être à priori standardisée et simple à utiliser dans l'interface utilisateur afin que la mise en place des génériques (déjà écrits) soit effectuée de manière simple et rapide par les monteurs.

Gestion des Keyframes :

Les keyframes sont utilisées dans bien des domaines, mais dans beaucoup de cas, elles sont utilisées avec parcimonie. Elles permettent dans une vidéo, d'animer les propriétés d'éléments ajoutés par le monteur (effets, texte, etc...). Il apparaît donc nécessaire d'avoir une gestion minimale des keyframes, en particulier pour une gestion fine des couleurs, mais leur utilisation est rarement vraiment avancée.

Dans la création de clips en particulier, afin de dynamiser la vidéo, les monteurs utilisent de manière intensive les keyframes.

1.2.2 Fonctionnalités spécifiques

Dans les faits, les fonctionnalités utilisées sont assez similaires bien que les œuvres finales soient totalement différentes.

Quelques fonctionnalités sont apparues comme vraiment propres à la création d'un type d'oeuvre en particulier.

Visualisation image par image : Dans le cadre de la création de film, la prévisualisation de chaque frame de manière précise semble être une fonctionnalité essentielle. Cela signifie que le logiciel de montage doit permettre de voir de manière simple chaque frame des vidéos présentes dans la timeline. Cette fonctionnalité est aussi utile dans le cadre de la création d'autres oeuvres, mais elle est indispensable dans le cas de films, permettant ainsi de s'assurer de la qualité du résultat. En effet, lors de la création d'un film, chaque frame doit être contrôlée. Dans d'autres types d'œuvres, les exigences et les moyens étant moins élevées, une telle fonctionnalité n'est pas indispensable.

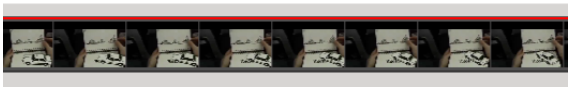

Fig. 1.5: Visualisation frame par frame

Gestion avancée des Footages

Dans le cadre de productions longues, un des problèmes auquel doit répondre de manière satisfaisante le logiciel d'édition est la gestion et la classification des Footages. C' est en particulièrement vrai pour les films et les séries télévisées. Dans ces types de productions le nombre d'heures de Footages peut être très grand, et le monteur doit dans un premier temps établir une classification des Footages. Le logiciel de montage doit, pour répondre aux besoins des monteurs, permettre de les ordonner de manière précise et bien pensée.

Intégration dans un écosystème de logiciel de post production

Dans le cadre de création de films en particulier, on constate que le logiciel de montage doit s'intégrer dans l'écosystème de logiciel de post production. C'est généralement possible si ce logiciel de montage respecte les quelques standards de la post production d'oeuvre audiovisuelle comme par exemple le Material eXchange Format

Time remapping

Le time remapping, comme précédemment indiqué, est particulièrement utilisé dans la création de contenu court. Il permet d'accélérer, où ralentir une partie d'un clip pour le rendre l'œuvre la plus dynamique possible.

Gestion des templates

La création de contenu non post produit demande des fonctionnalités particulières. La fonctionnalité qui apparaît comme clé pour répondre aux besoins liés à ce type de produit est la création de template. Par exemple la création de journaux télévisés ou autres évènements sportifs demande une gestion avancée de “moule“, ou template, qui permet de simplement lier les contenus des différentes caméras à un moment donné de la retransmission.

Cette fonctionnalité n’est pas exclusivement utilisée dans la création de contenu en direct, mais elle est très largement utilisée dans tout ce qui est contenu destiné à la télévision.

En conclusion, on a constaté que le champ de fonctionnalité est vaste, la plupart de ces fonctionnalités sont génériques et leur utilisation est,par conséquent, commune à différents types d’œuvres.En revanche ce qui varie particulièrement est la finesse d’implémentation et le niveau d’utilisation qu’en fait le monteur.

1.3 Comparaison des principaux logiciels présents sur le marché de l'édition vidéo professionnel, et analyse des manques et risques du marché

Il conviendra d'analyser précisément les logiciels existants, qu'ils soient propriétaires ou libres. Cette partie a pour but de rendre compte de l'état actuel du marché des logiciels d'édition vidéo qui ont pour principal public les professionnels. Cette étude portant principalement sur les logiciels libres, ceux-ci seront évidemment inclus dans cette analyse bien que l'on puisse considérer que à cause de leur manque de maturité, ils n'y aient pas totalement leur place.

Dans cette optique, on analysera les points clés des logiciels. Tout d'abord on comparera les fonctionnalités des logiciels, la manière dont elles sont gérées, et on essayera d'avoir l'avis de professionnels sur ces fonctionnalités et leur implémentation dans les différents logiciels. Ensuite on regardera le prix de ces logiciels,on verra en quoi cela peut être un argument de poids pour les logiciels libres et leur éventuelle prise de part de marché. Par la suite nous nous concentrerons sur la documentation, livres et autres tutoriels disponibles pour ces différents logiciels, et nous verrons quels supports sont offerts aux professionnels pour ces logiciels.

1.3.1 Historique du marché

Les tout premiers logiciels d'édition non linéaire ont vu le jour dans le début des années 70. A cette époque les solutions de stockages de données étant très limitantes, les premiers logiciels de montage vidéo non linéaire effectivement utilisables ont vu le jour en 1989, ceux-ci étaient basés sur les disques durs pour ce qui est du stockage. C'est cette année là que "Editing Machines Corp." et Avid ont mis sur le marché les logiciels de montage vidéo non linéaire ainsi que le matériel qui permettait son utilisation. Une fois de plus, les limitations en terme de stockage de données (accès limité à 50 Gigabytes à la fois maximum), rendait l'utilisation des systèmes de montage non linéaire inutilisables dans le domaine du cinéma, et même dans de nombreux cas de la télévision. C'est en 1992 que cette limitation a été surmontée, il était alors possible d'accéder jusqu'à 7 Terabytes de données à la fois, ce qui rendait envisageable le montage de production longue de manière informatique. C'est en 1993 que Avid prend avantage de cela et s'impose comme leader mondial, remplaçant les équipements de montage de pellicules 35mm de toutes les principales maisons de production cinématographiques dans le monde. Avid a été le leader incontesté du marché de l'édition professionnel jusqu'en 2003, date à laquelle Final Cut Pro a été considéré comme une bonne alternative à Avid par les grands acteurs de l'édition vidéo professionnel.

1.3.2 Définition des plus grands acteurs du marché

Logiciels commerciaux

Avid Media Composer : Leader historique du marché du logiciel de montage non linéaire professionnel. Il s'agit du produit phare de Avid Technology publié en 1989. Depuis, ce logiciel a joué un rôle essentiel dans le développement de ce marché.

Avid Symphony : Evolution de Avid Media Composer, il s'agit d'une version plus complète en terme de fonctionnalités qui a pour but de répondre aux besoins des monteurs de productions longues tels que les documentaires et les séries télévisées.

Final cut pro : Logiciel de montage intégré dans la suite de logiciels de post-production de Apple, Final Cut Studio. Il s'agit d'un logiciel de montage orienté à la fois professionnel et création de film. Il est de nos jours très utilisé et est devenu l'un des leaders mondial du marché.

Adobe Premiere Pro : Logiciel de montage de la suite Adobe Creative suite, il s'agit du logiciel d'édition vidéo à visée professionnelle de Adobe System. Il est à la fois adapté pour la création de contenu diffusé, mais aussi de contenu post produit

Logiciel en cours de libération :

lightworks : Logiciel de montage actuellement commercial, très puissant, et offrant des fonctionnalités uniques, il permet de faire parallèlement la production diffusée et la création post-produit. Ce produit est particulier car ses créateurs ont décidés de libérer le code source [6], et ainsi de créer une communauté de développeurs pour en faire un projet de logiciel libre.

Logiciels libres :

Cinelerra : Logiciel de montage et de compositing libre développé principalement par la société Héroïne [1] et intégré par la société LMA[2]. Il s'agit d'un logiciel de montage non linéaire avec de très nombreuses fonctionnalités. Principalement créé pour le création de contenu

[1]Heroine : http ://heroinewarrior.com/
[2]LMA : http ://lmahd.com/

diffusé, il permet aussi de répondre aux besoins de la production de contenu post-produit. Il s'agit du seul logiciel libre utilisé en milieu professionnel.

Kdenlive : Logiciel de montage libre s'intégrant dans la suite logicielle de l'interface graphique KDE. Ce logiciel de montage est assez complet et peut répondre aux besoins des monteurs de contenu post-produit.

PiTiVi : Logiciel de montage libre encore basique mais en plein développement. Ce logiciel a pour but de répondre aux besoins du plus grand nombre, et en particulier à ceux des professionnels de la création de contenu, qu'il soit post produit ou non.

1.3.3 Fonctionnalités

Tout d'abord, il convient de voir quelles fonctionnalités existent chez les différents acteurs du marché. Afin de faciliter la lecture et avoir une vision globale de ce qui ce fait nous avons dessiné une représentation sur un diagramme en toile d'araignée. D'après les interview et l'analyse précédemment faites, les axes suivants ont été choisis :

- Gestion des formats de fichiers
- Integration dans un écosystème de post production
- Gestion des template
- Gestion des footage
- Colorimétries
- Effets
- Transitions
- Support multi-platform

Dans le schéma suivant le niveau et la qualité d'implémentation a été prise en compte. Les retours utilisateurs ont aussi une place importante dans cette évaluation. La plupart de ces évaluations portent sur des données non quantifiables, pour cette raison, aucune échelle précise n'est donnée. Par exemple, l'ergonomie ne peut être quantifiée, seul le ressenti des utilisateurs peut être analysé, et c'est ce travail qui a été effectué.

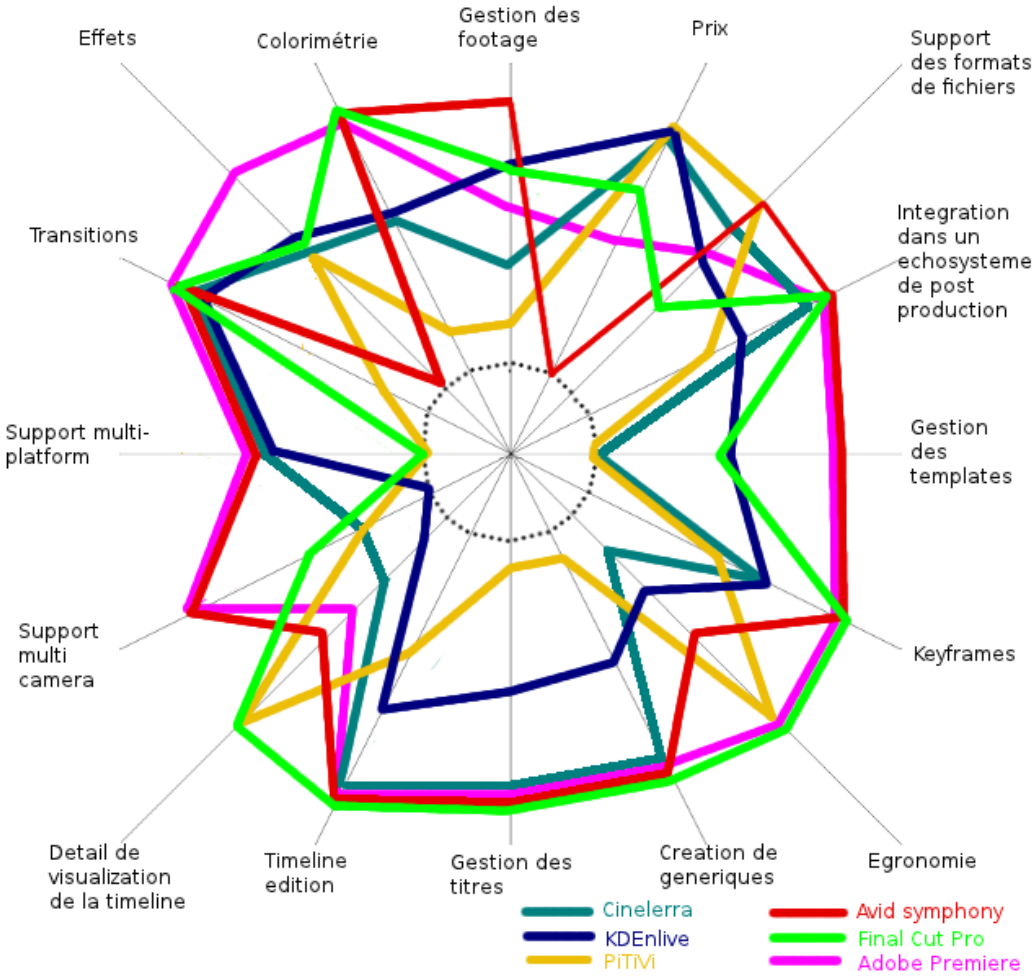

Fig. 1.6: Comparaison des fonctionnalités des logiciels leaders sur le marché

Ce schéma nous permet d'expliquer très facilement que seul le logiciel libre Cinelerra est une place sur le marché, les lacunes de PiTiVi et Kdenlive en terme de fonctionnalité rend leur utilisation impossible en milieu professionnel.

1.4 Visions du marché par les professionnels du montage

L'importance de la position du logiciel sur le marché

En ce qui concerne le choix du logiciel de montage dans une entreprise, la connaissance que les monteurs ont du logiciel qu'ils utilisent est essentiel. C'est pour cela que dans de nombreux cas le premier facteur de choix est la position de ce logiciel sur le marché de l'édition vidéo. En effet, les différentes industries veulent être sûres qu'il est possible de trouver de la main d'oeuvre compétente sur le logiciel qu'elle utilise. Mais ce n'est pas seulement la disponibilité de personnes compétentes qui est importante, il est également indispensable que ces dernières puissent se former et qu'elles aient accès facilement aux informations concernant les nouvelles résolutions des problèmes posés par un logiciel au travers des différentes versions.

Comme l'a souligné M. Faure lors de son interview (annexe 3), dans son cas, il ne serait pas envisageable de changer de logiciel à moins que le marché n'évolue. C'est à dire que l'un des critères de choix dans son cas est la position de tel logiciel sur le marché. Cela est principalement une question de crédibilité, et donc afin d'éviter la prise de risque, les professionnels préfèrent prendre la référence du marché.

Mais cela est principalement vrai dans les structures de taille moyenne. Dans le cadre de petits structures, comme l'a souligné M. Veri (annexe 1) durant son interview, il serait plus logique, dans un souci de se différencier de ses concurrents, de choisir un logiciel différent de celui de référence. Bien évidemment ce logiciel devra répondre convenablement à leurs besoins et permette de satisfaire les demandes de leurs clients. Dans son cas, M. Veri estime qu'un tel logiciel n'existe pas et que la référence du marché (Final Cut Pro) est en réalité la meilleure option.

Dépendance vis-à-vis du créateur

Un souci important des professionnels du montage qui ressort des interviews est le changement de l'expérience utilisateur : l'arrivée d'une nouvelle version est redoutée car elle demande un temps d'adaptation pour la maitriser. Ceci peut être illustré par la dernière version de Final Cut Pro, qui a été très critiquée [4] : Apple a décidé de changer le workflow des professionnels de l'édition vidéo dans la dernière monture de leur logiciel phare de ce secteur, et cela a très mal été perçu par les professionnels. Toutefois pour M. Faure, ce problème n'est pas trop grave, puisqu'il considère que dans le cas où la transition à cette nouvelle version s'avère compliquée (trop coûteuse, formation du personnel trop onéreuse, perte de temps pour le maitriser) , ils

ont toujours avec Apple a toujours l'option de conserver la version courante. Mais cela n'est valable que sur le court terme, puisqu'il n'est pas envisageable, pour des raisons de sécurité et de support, d'utiliser de manière commerciale un logiciel non maintenu et non supporté par l'entreprise éditrice. Pour M. Veri en revanche, cela pose aussi un problème important, et ayant lui même utilisé cette nouvelle version, il considère qu'il serait préférable pour son entreprise de trouver une alternative.

Dans le cadre de structures importantes, la dépendance vis-à-vis des éditeurs est parfois considérée potentiellement dangereuse, et donc à éviter . Par exemple, Dreamworks Picture, a décidé de créer et de maintenir leur propre suite logiciel [**?**] de post production. Leur principal objectif est d'avoir la garantie que leurs logiciels répondent bien à leurs besoins précis en maitrisant leur évolution. Ils ne dépendent plus d' un éditeur externe et pour conforter cette indépendance, cette entreprise a décidé d'utiliser Linux comme système d'exploitation (distribution Red Hat), ce qui lui garantit un grande liberté. De plus, la partie "Dreamworks animations" est cliente de la société LMA, ce qui signifie que très probablement celle-ci utilise Cinelerra dans leur processus de post-production. De même les sociétés de télévision Française TF1 et Canal+ sont clientes de cette même entreprise, ce qui montre bien que Cinelerra est utilisé de manière professionnel. Bien que d'après M. Faure, TF1 semble abandonner petit à petit Cinelerra au profit de Final Cut Pro.

1.5 Bilan

Nous constatons donc que dans un marché vaste et diversifié seuls quelques logiciels permettent à l'heure actuelle de répondre aux besoins de la grande majorité des utilisateurs.

Les professionnels sont plutôt satisfaits des logiciels commerciaux existants dont ils disposent avec cependant certaines limites dûes principalement au fait que ces logiciels sont fermés et édités par une seule entreprise (externe) . Les fonctionnalités de ces logiciels visent à répondre à la majorité des besoins du marché, mais ils ont des difficultés pour s'adapter à des besoins spécifiifiques d'entreprises .

Partant de ce constat on peut se demander si l'utilisation de logiciels ouverts et édités par de nombreuses entreprises ne pourraient pas prendre une place plus importante sur le marché, en offrant de nouvelles perspectives aux acteurs du marché du montage vidéo. Les éléments suivants peuvent être considérés comme des éléments importants sur un marché fermé, et contrôlé par un nombre très restreint d'entreprises :

Plus grande autonomie vis-à-vis de la société éditrice (développeur/designer) :

Grâce à l'utilisation de logiciel libre, il est possible de garantir et maintenir le logiciel en interne tout en profitant du fait que des personnes externes à l'entreprise, d'une part le développent, et d'autre part, le connaissent. Cela a pour conséquence de réduire les coûts d'une manière très importante tout en gardant un contrôle certain sur le développement du logiciel. C'est très certainement pour ces raisons que Dreamworks, TF1 et bien d'autres se sont mis à utiliser des logiciels libres.

Possibilité de collaboration et de communication avec les développeurs :

Le développement des logiciels libres étant en principe fait publiquement, les utilisateurs (entreprises qui font le montage vidéos) peuvent voir l'évolution du logiciel au fur et à mesure de sa progression. Cela leur permet aussi de donner leur avis sur les directions à prendre.

Réduction de coût :

Comme l'ont souligné M. Veri et M. Hachemi au cours de leur interview, dans le cadre de petites structures la réduction des coûts est un souci important. En effet, les frais liés à l'achat de licences pour les logiciels de montage constituent une charge financière élevée compte tenu des prix de ces licences. L'utilisation de logiciel libre permettrait donc, dans la très grande majorité des cas, de réduire d'une façon significative les coûts de montage, puisque le code source est libre d'accès.

Possibilité d'adaptation aux besoins précis de l'entreprise :

L'ouverture du code et sa mise à disposition à tous ouvre la possibilité d'effectuer des adaptations dans le core même du logiciel et ainsi de développer des versions spécialement adaptées aux besoins de l'entreprise utilisatrice.

Dans le cadre de projets commerciaux,le développement et l'adaptation de différentes versions ne sont faites que par l'entreprise éditrice. On peut cependant, même dans ce cadre-là, modifier, ajouter des fonctionnalités grâce aux systèmes d'extension logicielle, mais la flexibilité offerte par ce système reste limitée à ce que la société éditrice veut bien mettre à disposition des développeurs externes. (En terme d'API .

On voit que des logiciels libres sont utilisés en milieu professionnel, mais cela reste marginal. Les opportunités que ces logiciels peuvent apporter au monde professionnel de la post-production étant importantes, nous allons voir comment utiliser le potentiel de ces logiciels dans ce domaine.

Chapitre 2

Analyse des opportunités des technologies libres dans le domaine de l'édition vidéo et prévisions

Contents

Maintenant que les besoins et que les solutions existantes ont été analysées on rendra compte de la situation actuelle des technologies libres et de leurs communautés. Il est aussi important de chercher les raisons qui expliquent que ces logiciels ne sont pas plus utilisés par les professionnels. Puis, nous essayerons d'envisager les solutions possibles qui permettraient de remédier à cette situation.

Dans cette partie, nous analyserons la différence entre les manières d'envisager la création de logiciel et nous verrons quels sont les avantages et inconvénients de ces fonctionnements. Par la suite nous nous concentrerons sur les frameworks existants pour faire une analyse technique de ces technologies. Puis, nous analyserons les communautés qui portent ces différents projets afin de déterminer les points forts et les points faibles de chacun des projets. Pour finir, nous tirerons les conclusions de cette analyse afin de trouver des solutions aux défis qu'est la création d'un logiciel libre de montage vidéo.

2.1 Etat actuel de l'offre de logiciel libre

Le schéma suivant permet de résumer facilement la situation :

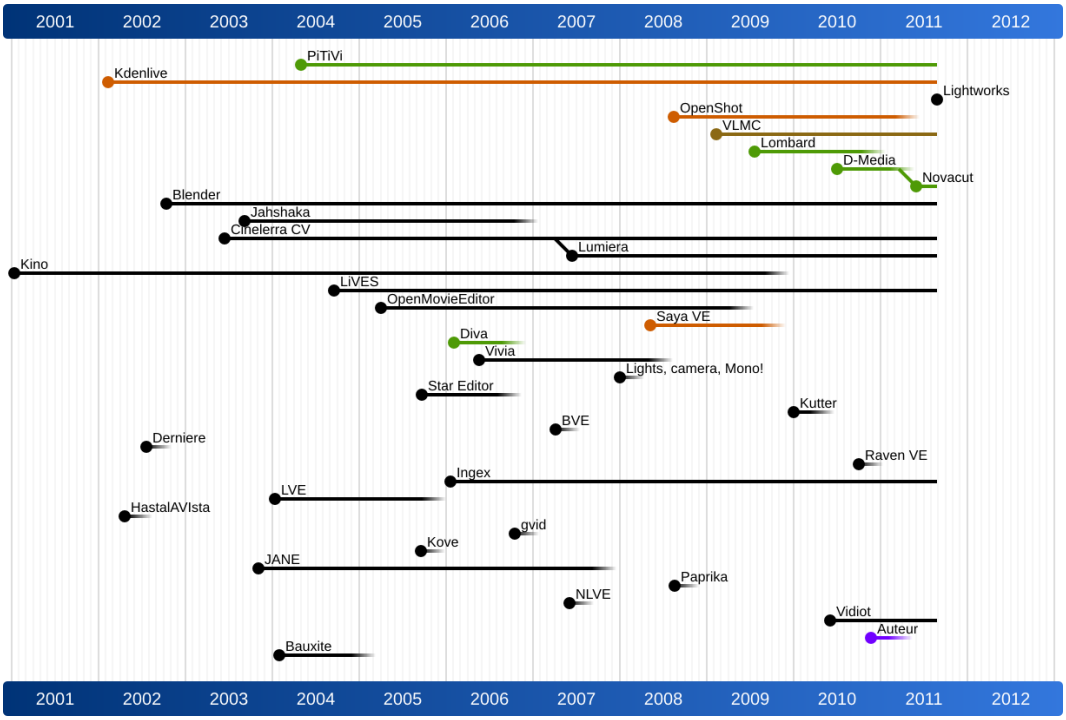

Fig. 2.1: Open source video editors timeline (Auteur : Jean-François Fortin, PiTiVi designer)

On constate donc que de nombreux projets de logiciel libre de montage vidéo on vu le jours ces 10 dernières années et dont les objectifs sont différents. On peut distinguer deux types de public visés par ces projets :

– Les amateurs
– Les professionnels ou semi professionnels

Les amateurs de montages vidéo

Plusieurs projets libres permettent ou visent à répondre aux besoins des amateurs, mais à l'heure actuelle ils ne sont que partiellement adaptés à ces besoins. Parmi les logiciels dont l'objectif est de permettre de créer des montages simples on distingue :

– openshot : Logiciel présentant de nombreuses fonctionnalités, mais dont la qualité d'implémentation présente des faiblesses.

– kino : Logiciel avec un nombre de fonctionnalités limité permettant de faire des petit montages avec efficacité.
– Vidiot qui vise la production de vidéo amateur simple

Mais il existe des logiciels ayant pour objectif de répondre aux besoins plus avancés en particulier à ceux des professionnels (précédemment présenté dans le cadre de la définition des plus grands acteurs du marché) qui peuvent être utilisés dans le cadre de montage amateur. Leur utilisation (et c'est plus particulièrement le cas du logiciel Cinelerra) demeure complexe.

Un nouveau projet a aussi récemment vu le jour, dont la finalité est assez différente des logiciels actuellement présents. Il s'agit de Novacut, qui permet aux créateurs de films et séries web de faire le montage de manière collaborative à travers d'Internet en partageant les ressources (footage).

Le fait que Cinelerra ne soit packagé [1] dans aucune distribution Linux [2] montre que même ce logiciel est le seul à avoir réussi à prendre une part de marché dans le milieu de l'édition professionnelle, celui-ci n'a pas réussi à rassembler les développeurs et utilisateurs standard de logiciel libre.

En définitive aucun projet n'a encore réussi à s'imposer et ainsi regrouper les développeurs au sein de projets majeurs. Dans d'autre domaines, cela a été le cas, par exemple dans le domaine des lecteurs vidéo, Vlc a su surpasser ses concurrents, et ainsi supplanter le marché des lecteurs vidéo, qu'il soit libre ou non. Dans le domaine des environnements de Bureau graphique, KDE et Gnome sont arrivés à un stade où leur supériorité technique, et en terme de fonctionnalités, fait d'eux des plateformes de référence.

Il est donc intéressant de se demander quelles technologies et quels logiciel(s), pourraient se voir attribuer cette place dans le monde de l'édition vidéo libre. Nous allons donc analyser les logiciels et les technologies libres les plus avancés, (précédemment mentionnés dans le cadre de l'analyse de marché : Cinelerra, Kdenlive et PiTiVi). Nous verrons ainsi s'ils ont le potentiel de pouvoir un jour rivaliser avec les logiciels propriétaires sur le marché très fermé du montage vidéo professionnel.

[1] Packagé : fait que qu'un paquet, une archive (fichier compressé) comprenant les fichiers informatiques, les informations et procédures nécessaires à l'installation d'un logiciel sur un système d'exploitation au sein d'un agrégat logiciel, en s'assurant de la cohérence fonctionnelle du système ainsi modifié ait été créé. Source : Wikipedia

[2] Une distribution Linux, appelée aussi distribution GNU/Linux pour faire référence aux logiciels du projet GNU, est un ensemble cohérent de logiciels, la plupart étant logiciels libres, assemblés autour du noyau Linux. Source : Wikipedia

NB : Il aurait été intéressant d'analyser le logiciel lightworks, en voie de libération, mais à l'heure actuelle, aucun code n'a été libéré, et par conséquent, celui-ci ne peut pas faire partie de cette analyse.

2.2 Technologies

Pour faire une analyse technique des produits permettant de faire de l'édition vidéo, il est nécessaire d'analyser le "core" des logiciels, c'est à dire la partie du logiciel où les opérations d'édition sont effectivement réalisées. Dans ces domaines, il existe deux façon de procéder :

– Création d'un logiciel monolithique

– Création d'un framework

2.2.1 Technologies monolithiques VS technologies modulaires, frameworks

2.2.1.1 Logiciels monolithiques

Le conception monolithique dans le cadre des logiciels d'édition vidéo, consiste à développer au sein d'un même entité de code :

– la partie graphique et la partie de calculs permettant la gestion de tout ce que l'édition non linéaire implique

– L'interface utilisateur.

Par le terme logiciel monolithique, il faut réaliser que le logiciel peut utiliser des librairies externes, mais le core de ce même logiciel, et la logique d'édition linéaire à proprement parler sont directement élaborés à l'intérieur du logiciel et non par une librairie ou framework externe. Cela a pour principal avantage de présenter une conception simplifiée pour les raisons suivantes :

Les logiciels professionnels (commerciaux) utilisent très probablement tous ce mode de fonctionnement (même si vraisemblablement, en interne il ont un core qui ressemble fortement à un framework).Dans le monde des logiciels libres,les développeurs de Cinelerra ont décidé d'utiliser ce mode de fonctionnement.

On peut voir plusieurs conséquences immédiates de ce mode de développement :

– Les développeurs n'ont pas la nécessité de penser en terme d'interface publique de programmation (API), et n'ont pas à garantir la stabilité de celle-ci : le risque réside dans le fait que la qualité de l'architecture ne soit pas optimale car la création d'API oblige les développeurs/architectes à réellement analyser les besoins de manière plus large dès le début de la conception. Dans le cas où l'on ne crée pas d'interface publique de programmation vouée à être réutilisée, le risque est que le travail de design et d'architecture ne soit pas réalisé, et

que le code grandisse de manière anarchique avec les différents développeurs qui font des extensions au fur et à mesure de leurs besoins.

– Les développeurs n'ont besoin de penser l'architecture seulement pour les cas d'utilisation qui sont liés à ce même logiciel : ils n'ont pas à voir au delà de ces use cases.
– Les erreurs en terme de design n'ont pas d'incidences aussi graves que dans le cas d'un framework.

On se rend compte que cette manière de faire a pour principal avantage le fait que le logiciel peut être développé plus rapidement puisque le core du logiciel, et donc le code qui implémente la logique de l'édition non linéaire, est conçue avec pour seul cas d'utilisation, celui du logiciel. Cependant, de nombreux inconvénients existent à cause de la nature monolithique du design :

Besoins en main d'oeuvre considérables :

Dans le cadre de logiciel d'édition vidéo, le code à produire est considérable, comme le montre les statistiques (Annexes 2). Le logiciel Cinelerra à lui seul fait plus d'un million de lignes. Un tel volume de code est difficile à maintenir et requiert des ressources importantes en terme de main d'oeuvre. Le fait que le logiciel soit monolithique implique que celui-ci va être utilisé seulement par ce logiciel, et par conséquent, les développeurs ne peuvent pas compter sur d'autre utilisation de ce code pour améliorer et développer le core du logiciel.

Réutilisabilité :

L'un des inconvénients de cette manière de faire est que le code présent à l'intérieur du logiciel n'est pas réutilisable directement par d'autres projets. On le considère comme "individualiste", situation qu'il convient d'éviter dans le cadre du développement de logiciel libre afin de ne pas multiplier les efforts, et dupliquer le code.

Cette façon de faire a été utilisée par le projet Cinelerra. Ce logiciel est le plus avancé en terme de fonctionnalités dans l'offre des logiciels libres de montage . On peut penser que son architecture monolithique explique ce développement plus abouti, bien qu'il y ait évidemment de nombreux autres facteurs qui interviennent, en particulier le fait que ce logiciel ait été développé par la société Heroine Virtual pour ces besoins en tant que professionnel.

2.2.1.2 Utilisation de frameworks

L'autre possibilité est de séparer en deux parties bien distinctes l'implémentation de la logique de l'édition, lecture, encoding vidéo (core logiciel), de la partie graphique, interaction avec l'utilisateur final.

Le framework

La grande différence entre la conception monolithiques et la création d'un framework réside dans le le fait que dans le cadre d'un framework, on développe une API autour du core du logiciel. Cela résulte dans le fait que le core est un programme (librairie) externe, réutilisable par n'importe quel autre application. On peut considérer que les avantages des frameworks sont les inconvénients des applications monolithiques et vice-versa. L'avantage principal des frameworks sur une conception monolithique est la possibilité de partager un même code à travers de multiples applications. Celà permet de réunir les efforts au travers, dans notre cas précis, de tout type d'application multimedia.

Dans le cadre de l'édition vidéo, on peut encore distinguer deux manière d'envisager son développement :

- Utiliser un framework multimedia généraliste, et créer les outils nécessaire au montage au dessus de celui-ci
- Créer un framework spécialement orienté montage vidéo

Dans le monde du logiciel libre, ces deux manières d'envisager le développement d'un framework multimedia ont été abordées par les deux projets de framework leader sur ce segment :

- MLT qui se définit comme étant un "Framework multimedia design et développé pour le broadcasting télévisé."
- Gstreamer qui se définit comme étant un "framework multimédia basé sur la notion de pipeline" ce qui lui permet de nombreux types d'applications multimedia tels que des lecteurs multimédia, des logiciels de broadcasting, des logiciels de montage vidéo..."

Au dessus de ces frameworks, plusieurs applications (interfaces graphiques) de montage vidéo se sont développées.

- PiTiVi : utilise le Framework multimedia GStreamer
- Kdenlive openshot utilisent le framework orienté édition et broadcasting MLT.

Dans le cadre des Frameworks, nous nous intéresserons en particulier à l'analyse de ceux-ci puisque les notions relatives à l'édition vidéo, et la gestion de toute la partie multimédia est réalisée par ceux-ci. Les logiciels d'édition ne sont à priori que de simples interfaces graphiques basées sur ces frameworks. Dans les faits, l'implémentation actuelle de PiTiVi n'est pas qu'un simple interface graphique au dessus de GStreamer, mais une partie de la logique d'édition vidéo est actuellement réalisée dans le logiciel même (ceci est en train de changer avec la migration [2] vers gstreamer-editing-services[5]).

2.3 Analyse technique

Dans cette partie nous allons analyser la structure intern des trois logiciels précédemment définis : Cinelerra, PiTiVi et Kdenlive.

2.3.1 Cinelerra :

Cinelerra est développé en C++ et utilise par conséquent la paradigm objet. Il est distribué sous licence GPL Version 2 ou plus.

2.3.1.1 Documentation du code

Au niveau de la documentation, celle-ci est inexistante et le code lui-même ne contient que très peu de commentaires. Il est donc très compliqué de comprendre le fonctionnement et les relations entre ces centaines de milliers de lignes de code. L'analyse de son fonctionnement est par conséquent assez complexe, et il est possible que cette analyse contienne des imperfections.

2.3.1.2 Structuration du code

En terme de structure, le code de Cinelerra est décomposé en 3 partie :

- Lecture, rendering audio vidéo : ce code est principalement contenu dans les dossiers "quicktime", "thirdparty" et "libmpeg3".
- Effets audios et vidéos : Ceux-ci sont développés comme plugins, et le code est donc présent de le dossier "plugins"
- Edition vidéo non linéaire et interface graphique : ce code est contenu dans un seul et unique dossier, "cinelerra"
- Système de plugins : Aussi développé dans le dossier "cinelerra"

Cette structure semble être assez limité puisqu'il convient en théorie de décomposer le code par petites parties, alors que dans le cadre de Cinelerra, le dossier "cinelerra" contient non moins de 1000 fichiers et 207789 lignes de code.

2.3.1.3 Lecture, rendering

Dans le cadre de la lecture audio et video, Cinelerra fait appel à diverses librairies :

– ffmpeg : Solution compete, cross plateforme d'enregistrement, lecture, conversion de flux audio et vidéo. Il inclue libavcodec, librairie leader dans le domaines des coder/decoder. Il s'agit du core de la lecture audio et vidéo de Cinelerra.
– faac/faad : AAC audio encoder/decoder
– x264 : h264 encoder
– libdv : DV codec
– ...

Toutes ces librairies sont utilisées dans le but de lire et écrire des fichiers multimedia. Afin de standardiser, et permettre l'utilisation de ces libraries de manière similaire au sein du logiciel, les développeurs de Cinelerra ont élaboré au cas par cas des ponts entre ces librairies et le reste du logiciel (Fichier dans le dossier quicktime).

– Quicktime 4 Linux : supporte en particulier les formats DV, les codecs H.264 et AAC, et implémente des éléments de conversion d'espaces colorimétrique (colorspace conversion)
– Libmepg3 : supporte la plupart des formats du "Mpeg Picture Motion Group" et permet l'édition vidéo en utilisant ces formats bien qu'il ne soit pas conçu pour ce type d'utilisation.

2.3.1.4 Effets audio et vidéo

Afin de permettre la création d'effets, Cinelerra utilise du code provenant de deux librairies :

– ladspa, Linux Audio Developers Simple Plugins API : Librairie d'effets audios qui contient une multitude de plugins.Il s'agit d'une API très simple et extrêmement flexible qui théoriquement permet la création de plugins permettant n'importe quelle manipulation et transformation du son. Dans les faits, certaines fonctionnalités ne sont pas implémentées pour éviter de complexifier le core de la librairie.
– frei0r : Framework minimalist multi-platform de création d'effets vidéo. Il permet la création d'effets à travers de plugins. Il s'agit du standard de fait en terme d'effets vidéo dans le milieu des logiciels libres. Cette librairie a été élaborée par de nombreux développeurs issus de différentes communautés de logiciels libres en relation avec le multimedia. De nombreux plugins existent et sont stables, mais il présentent un inconvénient assez important concernant ce set d'effet, ils supportent uniquement l'espace de couleur RGB. Cela a pour conséquence que dans le cas ou le flux vidéo n'est pas dans cette espace de couleur, une conversion d'espace colorimétrique est nécessaire afin de les utiliser. Un autre inconvénient de cette librairie réside dans le fait que les effets sont réalisés de manière logicielle, alors qu'à l'heure actuelle, l'utilisation de la carte graphique permettrait de tirer partie de manière

beaucoup plus intéressante dans l'application d'effets sur les vidéos.

Afin de permettre l'utilisation d'effets, les développeurs de Cinelerra on mis en place un système de plugins. En terme d'implémentation, Cinelerra reprend le code de ces librairies dans un set de plugins Cinelerra en ajoutant l'implémentation de l'interface graphique qui permet la configuration de ces effets.

2.3.1.5 Interface Graphique

L'interface graphique est développée en utilisant directement le server X sans aucune librairie graphique au dessus. Ceci a pour conséquence d augmenter le code à produire mais permet de controller intégralement le projet sans dépendre de ces librairies. Dans le cadre de Cinelerra, cela est logique car ce logiciel est développé quasi intégralement en interne.

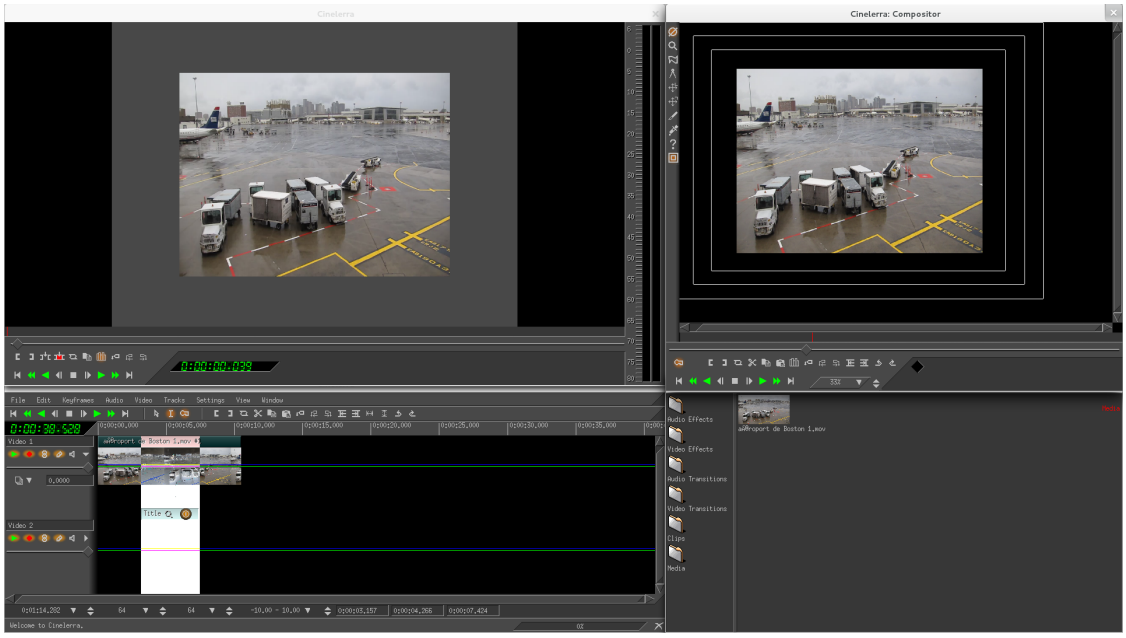

Fig. 2.2: Interface graphique de cinelerra

Structure de l'interface graphique : L'interface graphique de cinelerra est composée de quatre fenêtres principales :

– La timeline (en bas à gauches sur le screenshot) : cette partie permet de gérer un grand nombre d'opérations sur le contenu de la timeline.

– La fenêtre de ressource (en bas à droite sur le screenshot) : dans cette fenêtre l'utilisateur peut accéder au différents footages qu'il a importé. Il peut aussi accéder au différent effets, transitions...
– Le fenêtre de preview (en haut à gauches sur le screenshot) : cette fenêtre permet la prévisualisation des footages avant de les importer dans la timeline
– La fenêtre de composition (en haut à droite sur le screenshot) : cette fenêtre permet d'effectuer des opérations sur les vidéos et de prévisualiser la timeline.

2.3.1.6 Edition non linéaire

Conception

Dans Cinelerra, l'interface utilisateur et la logique de l'édition vidéo sont deux parties intégralement interdépendante. Au sein du code, il n'est pas possible de savoir quelle partie est plutôt liée à l'interface graphique et quelle partie dû peine de dissocier ces deux parties qui sont conceptuelement complètement différentes.

Accélération materiel

Lorsque openGL est présent sur le système, Cinelerra est en mesure de l'utiliser directement, dans la mesure où cette fonctionnalité ait été activée lors de la compilation. La fonction de compositing est ainsi accélérée, ainsi que la gestion des effets vidéo openGL.

2.3.2 Kdenlive

Comme précédemment énoncé, Kdenlive utilise le framework orienté montage et broadcasting MLT. Dans cette partie, nous allons dans un premier temps analyser ce framework.

2.3.2.1 Framework multimedia orienté montage : MLT

Panorama de la technologie

Le framework MLT est écrit en C et offre une API stable simple et minimaliste. Il est basé sur aucun library (seulement POSIX)

et le standard C, C99). MLT bien qu'écrit en C, utilise le paradigme de la programmation orienté objet en implémentant en interne le concept d'objet. Ce framework est modulaire,

et conçu pour permettre le développement de nouveaux composants. Il permet l'utilisation des différents core des processeurs pour faire les calculs, afin d'utiliser au mieux les processeurs modernes. Il est aussi cross platform et peut être utilisé sur les principaux systèmes d'exploitation : Linux, BSD, OS X et windows.

MLT est distribué sous licence LGPL Version

Concepts de base

Réseau de service

Le framework MLT est basé sur le concept de réseau de service sur lequel on distingue trois entités (classes) clés : producteur, filtre et consommateur. Ces différentes classes sont toutes des sous-classe de la classe appelé "service".

On peu schématiser le concept de réseau de service le plus simple de la manière suivante :

Fig. 2.3: Schéma du concept de producteur, filtre, consommateur

Le producteur a pour rôle de produire des données (lire un fichier audio, vidéo...) et de les faire passer au consommateur qui lui est connecté. Le consommateur a pour but de faire passer ces données (output datas) à la carte son, device vidéo, un autre fichier, où retransmettre à travers d'un moyen de télécommunication (broadcasting). Le filtre qui n'est pas obligatoire pour lire des donnés multimedias, permet de modifier les données (par exemple produisant un effet vidéo/audio), il peut aussi être connecté à plusieurs producteurs, avec comme exemple la production d'une transition entre ceux-ci.

Mais cela ne permet pas la creation d'éditeur de vidéo non linéaire, mais seulement la lecture et création de fichier. Pour permettre cette fonctionnalité, la classe multitrack a été mise en place. Celle-ci permet de gérer plusieurs producteurs et filtres les uns à la suite des autres. La multitrack contient plusieurs producteurs et les dispose les uns à la suite des autres.

Ces mêmes producers peuvent aussi provenir d'une playlist qui est un concept différent, et celles-ci peuvent aussi être ajoutées directement à un multitrack, et enregistrées sous forme de fichier de playlist (dans les différents standards existants).

Grâce à la création de ces réseaux de service, il est possible de créer des timelines complexes, et ainsi créer des logiciels d'édition vidéo non linéaires. On peut donc schématiser les réseaux de services de la manière suivante :

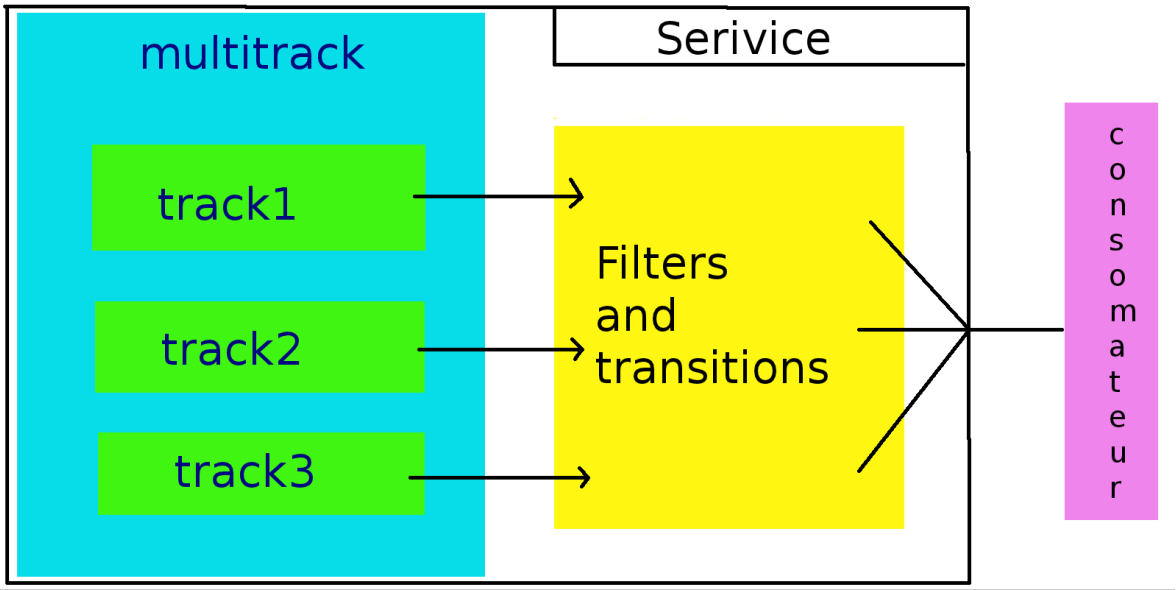

Fig. 2.4: Schéma simplifié d'un service MLT (point de vu interne et non utilisateur du framework)

Modularité

2.3.2.2 Documentation du code

Mlt possède une documentation accessible en ligne qui permet de connaitre les différents services existants. Le code est plutôt bien documenté, chaque fonction de l'API possède une description. Aussi, des documents texts dans les source (dossier docs/) permette au développeurs d'avoir un aperçu de la technologie. De plus il existe un outil en ligne de commande (melt) qui permet de tester de manière simple le framework. Le gros soucis étant qu'il est difficile de trouver une documentation claire de toute l'API, celle-ci ne semble pas être généré automatiquement lors de la compilation. Dans les sources de MLT, il n'existe pas vraiment d'exemple expliquant chacune des fonctionnalités une par une, il faut pour cela regarder le

code de melt, qui permet d'avoir un aperçu de comment utiliser l'API comme elle a été pensé par les développeurs.

Dans MLT, les de producteur, consommateurs où filtres sont des classes extern implémenté sous form de plugins. Certains sont implémentés directement dans MLT tel que :

– Le filtre "transition" : permet la création de transition
– Le filtre mono : permet de convertir un flux audio en mono
– Le filtre resize : Permet de redimensionner une vidéo
– ...

Cela est fait grace à des libraries externes en implémentant des modules externes afin de faire le lien (wrapper)

entre ces librairies et l'API précédemment présenté. Les principales libraries actuellement utilisables à travers de ce framework sont :

– libav, libdv et libvorbis pour ce qui concerne des codecs et muxers/demuxers
– frei0r pour les effets vidéos
– ladspa pour les effets audios
– ...

Ces différents services disponibles permettent de lire un nombre considérable de formats multimedia, en particulier grâce à libav, et d'effectuer de très nombreuses operations principalement grâce aux plugins frei0r pour la vidéo et ladspa pour l'audio.

Gestion des services

Le "repository" Pour connaitre les différent services présents sur le système, l'utilisateur a à sa disposition une classe Repository. Celle-ci offre une API simple listant les différents services par type (consommateur, producteur ou filtre). Pour illustrer, le simple code suivant permet de récupérer la liste de tous les effets présents sur le système :

```
mlt_repository repo = mlt_repository_init (NULL) /* We just use the
standard module path */ mlt_properties filters = mlt_repository_filters
(repo); /*
```

Les "factory" Dans le but de simplifier la création de services de différents types, MLT utilise le design pattern de la factory. On peut ainsi créer n'importe quel type de service de manière simple en utilisant les différentes factories existantes. Bien que tous les objets d'un réseau de service soient des descendants de la classe Service, les développeurs ont décidé de créer 4 méthodes différentes de la classe factory :

– mlt_factory_producer : permet d'instancier des producteurs
– mlt_factory_filter : permet d'instancier des filtres
– mlt_factory_transition : permet d'instancier des transitions
– mlt_factory_consumer : permet d'instancier des consommateurs

Par exemple, pour créer un effet "invert" depuis MLT il suffit de faire :

```
filter = mlt_factory_filter ( "invert", "my-invert-effect");
```

Il sera ensuite possible d'ajouter ce service au réseau de service.

Les données dans un réseau de services

La classe Frame

Les données qui transitent dans un réseau de service sont contenues dans des objets de type frame. Celles-ci contiennent à la fois les données audio et les données vidéos. Les producteurs peuvent setter ces données grâce aux méthodes mlt_frame_set_audio et mlt_frame_set_video.

Relations entre les différents services en terme de flux de données Afin commencer le flux de données dans un réseau de service, le consommateur est celui qui fait la demande de donner (pull datas) les données depuis le service sur lequel il est connecté. Les autre services réagissent en fonction,et forment une chaine jusqu'au producteur qui produit les données, et les fait suivre au service suivant et ainsi de suite. Ce processus peut être schématisé de la manière suivante :

Phase	producteur	Filtre	Consommateur
1			Request frame
2		Réception de la demande Demande la frame à son tour	
3	Réception de la demande Génération de la frame à la position Mise à jours de la position Mise à disposition de la frame		
4		Réception de la frame Update de la frame Mise à disposition	
5			Réception et process de la frame

Cette manière de fonctionner implique que seul le consommateur est le moteur du réseau. Il est celui qui doit faire partie d'un thread séparé et les appels à la fonction get_frame doivent être faits dans une boucle principale. Ces appels s'arrêteront au moment où le flux a été terminé (EOS).

Accélération materiel)

Au niveau de l'accélération materiel, MLT supporte le decoding avec acceleration matériel pour les cartes graphiques Nvidia (via VDPAU), et pour l'affichage mais pas le compositing vidéo accéléré.

Fonctionnalités haut niveau (High level features)

La petite API qu'offre ce framework comporte des méthodes et des fonctions haut niveau et permet de répondre à des besoins spécifiques de l'édition vidéo de manière simple pour l'utilisateur.

Génération de waveform/thumbnail depuis une frame

Le framework offre par example une fonction permettant la génération (sous forme d'image) des waveform de la partie audio d'une frame, et de thumbnail depuis la partie audio. Dans la partie précédente, nous avons constaté que la visualisation avancée de chaque frame était une fonctionnalité très utile en particulier dans le cadre de la création de films.

Serialization et deserialization de projets

Il intègre un système de serialisation, deserialization, ce qui permet de sauvegarder facilement les projets, avec les différents tracks, effets, transition... dans le cadre d'application de montage vidéo.

Le framework MLT offre des bindings haut niveau pour les languages : C++, C#, Java, Lua, Perl, PHP, Ruby, TCL et Python. Cela permet à un plus grand nombre de développeurs d'envisager l'utilisation de MLT dans leurs applications. C'est grâce aux bindings C++ que Kdenlive a pu être développé au dessus du framework MLT

Fonctionnalités

Cette analyse technique du projet MLT nous permet de constater que son fonctionnement est simple, et son API petite, et facile à prendre en main. Il permet de répondre à différents besoins de base des professionnels en terme de fonctionnalités :

- Ajout de titres et génériques : A travers du module développé par la communauté Kdenlive : QImage.
- Gestion des keyframes : possible dans les modules implémentant cette fonctionnalité, pas de solution générique au niveau du core de MLT
- Visualisation image par images et waveform : Directement accessible à travers du core de MLT.
- Visualisation image par image et waveform : Directement accessible à travers du core de MLT.

La principale lacune en terme de fonctionnalité est l'impossibilité de faire du time remapping, mais il est tout de même possible de gérer le contrôle de la vitesse de lecture des clips, ce qui est un point essentiel.

2.3.2.3 Logiciel de montage vidéo basé sur MLT : Kdenlive

Kdenlive est l'éditeur vidéo créé par la communauté en charge du bureau libre KDE. Ce logiciel est donc écrit en C++ utilisant le framework graphique QT ainsi que les kdelibs qui forment le framework permettant la création d'application intégré au bureau du même nom. L'analyse technique de ce logiciel n'a pas vraiment d'intérêt car on est en présence d'une interface graphique tirant partit du framework MLT.

Capture d'écran

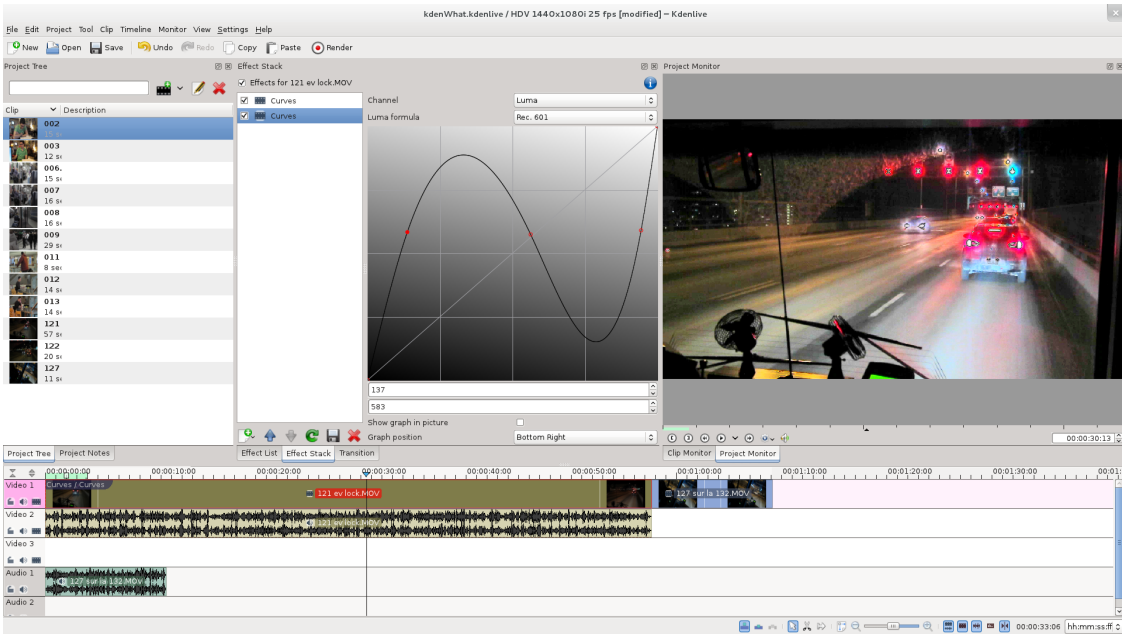

Fig. 2.5: Screenshot de Kdenlive

L'interface graphique de Kdenlive est composé d'une seule fenêtre présentant quatre éléments majeurs :

- La timeline en bas
- La gestion des footages, en haut à gauches
- La gestion des effets et transitions, au centre en haut. La configuration de ceux-ci se fait dans cette même partie.
- Le previewer, en haut à droite

Ce screenshot montre que Kdenlive présente de nombreuses fonctionnalités à l'utilisateur bien que son interface soit assez épuré.

2.3.3 PiTiVi

Comme précédemment énoncé, PiTiVi utilise le framework multimedia GStreamer. Dans cette partie, nous nous concentrerons sur l'analyse de ce framework.

2.3.3.1 Framework multimedia : GStreamer

Panorama de la technologie

GStreamer un framework multimedia basé sur le concept de pipeline écrit en C. L' API de ce framework est plus complet que celle de MLT puisque les use-cases auxquels celui- ci cherche à répondre sont plus nombreux. En effet, la communauté GStreamer a pour objectif de créer un framework open source permettant de répondre au plus grand nombre possible de cas d'utilisations ayant un lien avec le multimedia (depuis les téléphones mobiles jusqu'aux renders farms en passant par les logiciels de montage video). Ce framework, bien qu'écrit en C, utilise le paradigme objet, à travers de la librairie Glib. Cette librairie implémente de nombreuses API et en particulier la notion d'objet en C grace au module GObject. La Glib permet aussi la gestion des threads, des signaux... De très nombreux projets de logiciels libres l'utilisent, en particulier le projet d'interface graphique Gnome. GStreamer offre une API stable aussi bien pour les développeurs de plugins que pour les développeurs d'applications. Il est hautement multi threaded, et est thread safe, c'est a dire que la mémoire partagée par les différents threads ne peut pas être corrompue dans le cas ou plusieurs threads voudraient la modifier en même temps(en utilisant le système d'exclusion mutuel).

Le schéma suivant permet de représenter graphiquement l'architecture globale du framework :

GStreamer est distribué sous licence LGPL Version

2.3.3.2 Documentation du code

Gstreamer utilise la syntaxe de gtk-doc afin de permettre la génération automatique de la documentation à partir du code source. Celà signifie que l'API est complètement documenté et accessible sur internet. Afin de permettre d'obtenir des information sur les différent éléments, un outil gstreamer a été développé (gst-inspect). Il existe aussi des livres libres (pas toujours actualisé) permettant d'aider les développeurs soit à développer des applications en utilisant GStreamer (Gstreamer manual) où des plugins/éléments (Plugins Writer Guide). Il est aussi possible de créer des pipeline en ligne de commande grace à l'outil 'gst-launch'. En terme

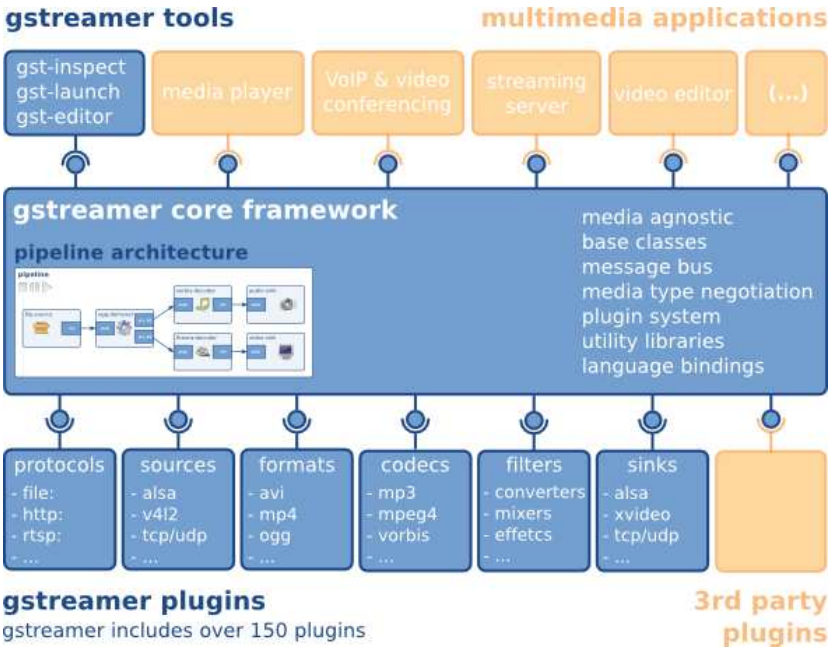

Fig. 2.6: Architecture du framework GStreamer

d'exemple, les développeurs essaye de tenir à jours des example pour chaque API nouvellement créé, mais cela n'est pas toujours le cas.

Concepts de base

Le concept de pipeline

La grande majorité des classes dans GStreamer dérive de la classe GstElement qui est l'élément de base dans un pipeline, lui même GstElement. Un schéma permet d'expliquer cette notion de pipeline assez facilement :

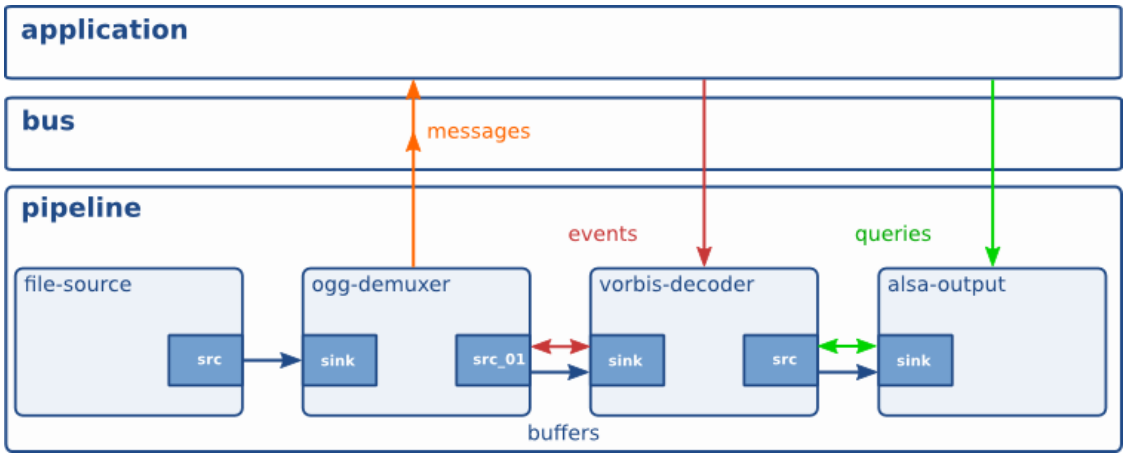

Fig. 2.7: Un pipeline GStreamer permettant la lecture d'un fichier audio ogg ayant pour codec vorbis (ce schéma prend en compte d'autres notions importantes du framework).

On constate que l'architecture est assez similaire à celle MLT, avec pour grande difference le fait que les éléments composants le "réseau de service" sont tous dans un même élément : le pipeline. Cet élément GStreamer execute un thread séparé à partir du moment où l'on cherche a jouer son contenu. Dès lors, les données transitent entre les différents élément gstreamer, a commencé par l'élément source. Dans l'exemple donne par ce schéma, il s'agit d'un file-source, et terminant par le sink, dans notre exemple alsa-output. Les éléments entre les deux se chargent du demuxing (ogg-demux), et du decoding (vorbis-decoder).

Une autre différence réside dans le fait que l'utilisateur de GStreamer a un contrôle plus important que dans le cas de MLT sur les composants de la chaines d'éléments permettant la reproduction de contenu multimedia. Dans GStreamer chaque tâche est effectuée par un élément, c'est à dire qu'un service MLT sera décomposé en plusieurs élément GStreamer. Dans le cadre d'un producteur en particulier, chez GStreamer, cela est effectué à travers un élément pour chaque étape du coding/decoding/muxing/demuxing, et dans certain cas, du parsing du bitstream. Cette décomposition en petits éléments effectuant une tâche a le gros avantage de permettre la réutilisabilité de chacun d'eux et ce, dans les différents contextes d'applications multimédia.

Communication entre les éléments

La communication entre ces différents éléments se fait au travers des GstPad. Il convient de représenter un élément (dans ce cas un demuxer, qui permet de recevoir un flux en entrée

de le décomposer afin d'avoir deux sorties, l'une contenant les données audios, et l'autre avec les données vidéos), de la manière suivante :

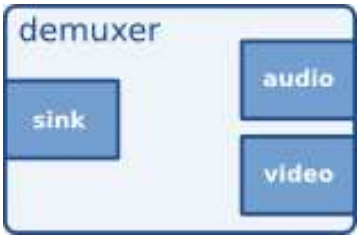

Fig. 2.8: Représentation graphique d'un GstElement permettant le demuxing.

Sur ce schéma, on constate la présence d'un sink pad, nommé "sink", et deux source pad, nommé "audio" et "video". Ces objets (GstPad), permettent la connection entre les différents élément. Il peuvent être comparé au "prise" et "port" dans le monde réel. Afin de brancher une prise dans un port, il convient de s'assurer que ceux-ci soient compatibles, c'est ce rôle qu'ont les caps, assurer la compatibilité de la communication entre différents éléments.

Modularité

GStreamer est extrêmement modulaire, et l'on peut l'étendre facilement en créant de nouveaux éléments. Il suffit d'ajouter ces élément (sous forme de plugins) sur le système pour que l'utilisateur puisse tirer partie des nouvelles fonctionnalités. La communauté GStreamer offre des collections de plugins contenant un très grand nombre d'élément. Ces set de plugins sont classés selon des critères précis :

- base : Plugins de grande qualité et très bien maintenus. Il offre aussi des classes de base afin de faciliter la creation d'éléments.
- good : Plugins de qualité, ayant une licence LGPL.
- Ugly : Plugins de qualité, ayant des problèmes au niveau de la licence.
- bad : Plugins de moins grande qualité, ayant des problèmes au niveau de la licence. Il s'agit d'éléments dont le code a été moins testé et risque ainsi de présenter davantage de bugs.

De nombreuses librairies sont wrapper dans ces différentes collections de plugins, par exemple, dans "bad", les librairies d'effets ladspa et frei0r y sont incluses et dès qu'elles sont installées sur le système, elles permettent à n'importe quelle application basée GStreamer de bénéficier de leurs fonctionnalités. Ces plugins sont distribués sous forme de librairies partagées, et peuvent donc être chargées à la volée, durant l'exécution du framework.

En terme de lecture de contenu multimedia, les développeurs GStreamer conseillent l'utilisation de la collection de plugins "gst-ffmpeg" qui permet de bénéficier de toutes les codec, muxer et demuxer de cette librairie. D'autres librairies tel que libogg, libdv, x264... sont wrapper dans les différentes collections de plugins précédemment présentées.

Gestion des éléments

La gestion des éléments est très comparable à ce qui est fait dans le framework MLT, c'est à dire, que l'on a une classe GstRegistry (comparable à la classe repository de MLT), et des GstFactory qui permettent la fabrication facile d'éléments.

Et l'édition vidéo ?

En effet, tout cela ne nous permet pas de faire de l'édition vidéo requiert la création de pipeline dynamique, c'est à dire qui permet de faire évoluer les éléments présents dans le pipeline au fil du temps. C'est pour cette raison que les éléments du plugins Gnonlin ont été développés. Il s'agit de 4 éléments principaux dont voici la définition

- GnlComposition : élément le plus important qui permet de faire évoluer le pipeline de manière dynamique
- Gnl[uri/file]Source : élément source qui a les propriétés nécessaires pour savoir à quel moment il doit être présent ou non dans le pipeline
- GnlOperation : élément qui permet l'application d'effet, transition etc... il a aussi les propriété permettant de savoir à quel moment il doit être présent ou non dans le pipeline

Mais ces éléments ne permettent pas de facilement créer des application de montage audio/ou vidéo. Le core actuel de PiTiVi utilise directement ces élément afin de gérer les notion propre à l'édition vidéo, mais les développeur se sont rendu compte que cela était trop complexe à utiliser, et par concéquent PiTiVi n'est pas une simple interface au dessus de GStreamer, mais introduit de plusieurs concepts supplémentaires au dessus de ces éléments.

Et plus récemment, les développeurs de GStreamer ont décidé de créer une nouvelle librairie basé sur GStreamer permettant de faciliter au maximum la création d'un logiciel d'édition vidéo basé sur GStreamer : gst-editing-services. L'objectif premier de cette librairie est d'implémenter les concepts même d'édition vidéo directement dans le backend. C'est à dire qu'au cœur de cette librairie, on retrouve les concepts de timeline, de layer, de transition, d'effets... Mais bien évidemment, celle-ci étant basée sur GStreamer, le concept the pipeline

continu d'exister, et il est très facile de créer un pipeline directement depuis une timeline. Un schéma permettant de montrer les concepts de base de cette librairies permet de simplifier la compréhension de son fonctionnement.

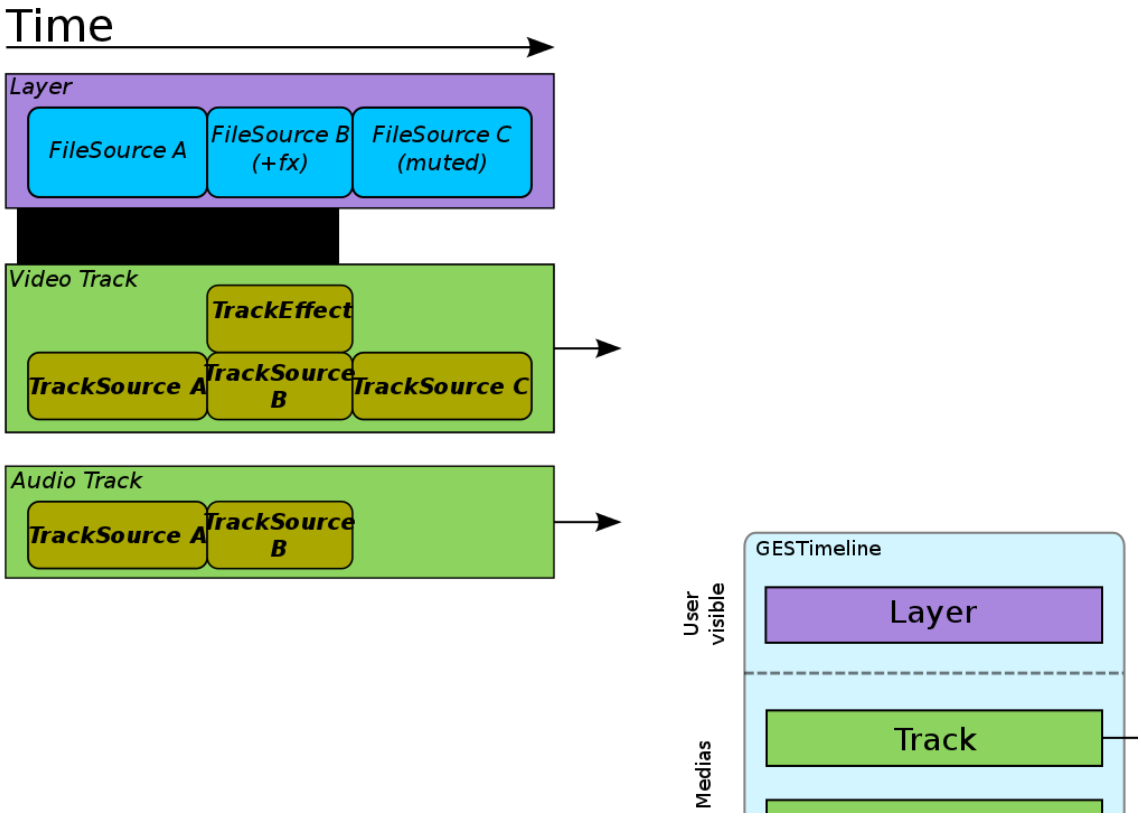

Fig. 2.9: Concepts introduits par GES

Ce schéma permet de representer graphiquement les différents concepts qui entre en jeux. Niveau utilisateurs, l'API de GES peut être utiliser de manière plus où moins avancé. C'est à dire que dans le cadre d'application simple d'édition vidéo, la notion de média est caché à l'utilisateur et celui-ci ne s'intéresse simplement à ce que sont les fichiers, effets etc... Mais il est possible de faire une utilisation avancé de cette API afin de créer un éditeur vidéo à visée professionnel.

Tout comme ce que l'on a pu constater dans MLT, GES offre des fonctionnalités de haut niveau tel que la génération de thumbnails à partir de la timeline, la serialisation, deserialization des timelines...

Dans les faits, GES est très comparable à ce qu'est le core actuel de PiTiVi. C'est donc naturellement que PiTiVi est en train de faire la migration vers les gst-editing-services.

Fonctionnalités

Cette analyse technique du projet GStreamer et GES nous permet de voir que GStreamer étant une technologie très complexe, les développeurs ont développé diverse outils afin de permettre la création de manière simplifié de logiciels de montage vidéo. Ce framework à tracer de GES permet de répondre aux différents besoins de base des professionnels en terme de fonctionnalités :

- Ajout de titres et génériques : GES offre des TextLayer et TitleSource qui permettent d'ajouter des titres de manière simple, en ce qui concerne les générique, cela est possible à travers de keyframes, mais cela n'est pas vraiment supporté à l'heure actuel. Il s'agit d'un manque majeurs pour répondre aux besoins des professionnels
- Gestion des keyframes : possible dans les modules implémentant cette fonctionnalité, pas de solution générique au niveau du core de MLT
- Gestion des keyframes : Le framework GStreamer offre la possibilité de créer des keyframes sur toutes les propriétés de ces élément à grace à la classe GstInterpollation
- Visualisation image par image : Directement accessible grace à GES.

2.3.3.3 Logiciel de montage vidéo basé sur GStreamer : PiTiVi

PiTiVi est l'éditeur vidéo supporté par la communauté GStreamer et la communauté Gnome. Il est écrit avec le language de programmation Python, et la libraries graphique Gtk+ .

Capture d'écran

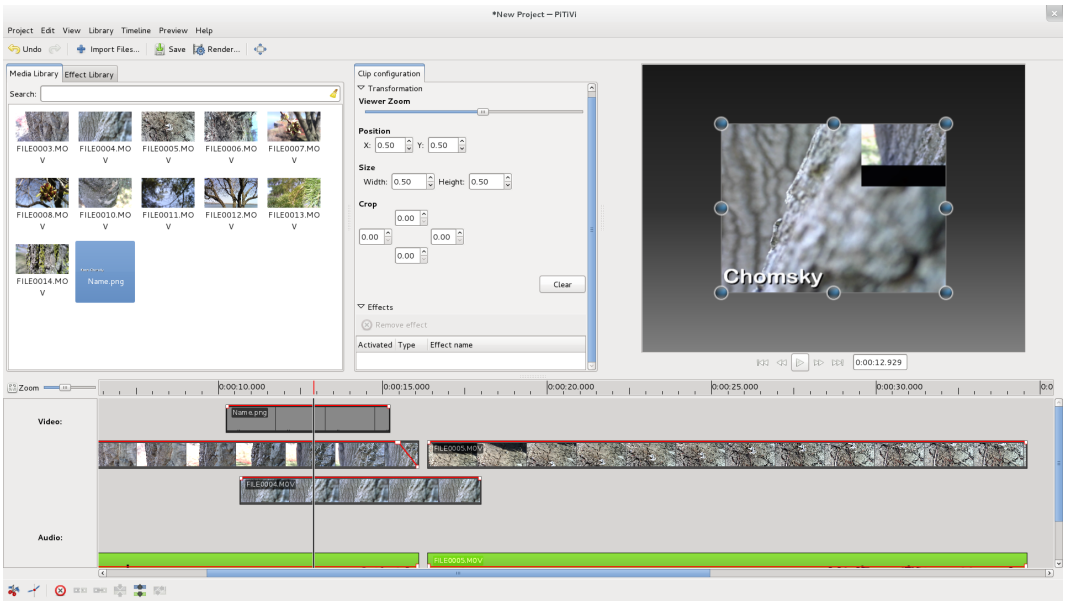

Fig. 2.10: Screenshot de PiTiVi

L'interface graphique de PiTiVi est aussi composé d'une seule fenêtre présentant quatre éléments majeurs :

- La timeline en bas
- La gestion des footages et effet, sous forms d'onglets, en haut à gauche
- La configuration des effets et transformations, au centre en haut.
- Le previewer, en haut à droite

Ce screenshot permet de voir l'interface simple et épuré de PiTiVi. Celà traduit en effet son manque de fonctionnalités, mais aussi le but des développeurs qui est d'offrir le strict nécessaire en terme d'interface. C'est à dire offrir les options seulement si celle-ci sont utiles à l'utilisateur.

2.4 Analyse des communautés

Afin d'analyser une communautés, quatres éléments majeurs sont à prendre en comptes :

- Le code source : il convient de mesurer le nombre d'auteur, la vitesse de son évolution d'évolution. Afin de mesurer cela, des outils statistiques existent, en particulier pour le système de gestion de code source git, et à l'outil gitstats. Celà permet d'analyser l'état de santé d'un communauté de logiciel libre.
- Les entreprises qui travail sur le projet : Les logiciels libre font à l'heure actuel actuel pleinement partit de l'économie du logiciel informatique. L'implication des entreprises dans les différentes communautés libre est un point essentiel qui permet permettant de comprendre la viabilité de celui-ci en particulier, comme c'est le cas dans cette analyse, au prêt des professionnels
- Le nombre de bugs rapportés, et l'analyse de la vitesse de fix de ceux-ci
- Communication entre les membres de la communauté

Pour les différent projets précédemment analyser, nous allons faire un état des lieux de leurs communauté selon les deux critère précédemment mentionnés.

2.4.1 Analyse du code source

Afin d'analyser le code, nous avons exécuté le script gitstats sur les différent repository git des projets :

- cinelerra : version communautaire du projet : http ://git.cinelerra.org
- MLT : http ://www.mltframework.org/gitweb/mlt.git ?p=mltframework.org/mlt.git ;a=summary
- GStreamer : http ://cgit.freedesktop.org/gstreamer/gstreamer/

Il est important de noter que ce tableau doit être analyser en prenant en compte les paramètres suivants :

- Le fait que le projet cinelerra ne soit pas réelement basé sur la communauté risque de faussé la comparaison avec les autres projets, mais il convient de garder en tête que ce projet est le seul projet présent sur le marché de l'édition professionnel.
- Le fait que cinelerra ne soit pas un framework mais une application complète d'édition vidéo doit être pris en compte
- Dans MLT, les plugins sont prise en compte, alors que pour GStreamer, on ne prend en compte que le core du famework (qui n'est qu'un très petit partie du framework)

Logiciel	Nombre de lignes	Nombre de développeurs	Nombre de commit le mois passé	Mois avec le plus grand nombre de commits	Mois du premier commit
Cinelerra	1 064 372	28 (meaningless)	1	Septembre 2006 (38 commits)	Juin 2003
MLT	110011	12	1	Sepetembre 2004 (53 commits)	Décembre 2003
GStreamer	435 930	205	47	Décembre 2001 (150 commits)	Janvier 2001

2.4.2 Analyse des mailing lists

Cette analyse porte sur les mailing listes officiel des différent projets :

– cinelerra : version communautaire du projet : cinelerra.org/mailinglists.php
– MLT : cinelerra.org/mailinglists.php
– GStreamer : http ://gstreamer.freedesktop.org/lists/

Logiciel	Nombre minimum par mois	Nombre maximum par mois	Nombre moyen par mois
Cinelerra	20 Juin 2011	499 Aout 2007	55
MLT	37 Mars 2009	178 May 2011	72
GStreamer	238 Janvier 2010	521 Aout 2010	387

2.4.3 Analyse des bug trackers

Cette analyse porte sur les bug trackers officiel des différent projets :

– cinelerra : version communautaire du projet : bugs.cinelerra.org/
– MLT : www.sourceforge.net/tracker/ ?group_id=96039&atid=613414
– GStreamer : produit GStreamer in https ://bugzilla.gnome.org/

Logiciel	Nombre total de bug	Nombre bug ouvert	nombre de bug fermé dans la dernière release
Cinelerra	602	269	23
MLT	142	50	8
GStreamer	11210	1363	532

On s'aperçoit que les différence en volume de bugs report, sont très importantes entre ces différents projets, en particulier entre GStreamer et les autres. Celà s'explique probablement par le fait que GStreamer est un framework multimedia avec un champs d'application beaucoup plus large. Mais ces chiffre sont intéressant afin de juger de la taille des différentes communautés.

2.4.4 Cinelerra

Tout d'abord le logiciel cinelerra est développé en interne par la société Heroine Virtual Ltd en interne en libérant leur code (code drops) de manière bi-annuel en moyenne. Une communauté c'est formé autour de ce code, et celle-ci le développe en incluant les changement d'héroïne Ltd lorsque l'entreprise release son code. Cette communauté a appelé sa version du logiciel cinelerra-cv, et la version est très proche de celle officiel. La difference majeur est que la communauté essaye d'utiliser les version upstream. L'entreprise aussi reprend la grande majorité du temps les changements effectués par la communauté au fur et à mesure.

Aussi, l'entreprise LMA, fait le support, formation du logiciel Cinelerra auprès des entreprises professionnel de l'édition vidéo. Ces deux entreprises travail en collaboration afin de répondre aux besoins des entreprises. A en croire la liste [3] de client et le type d'entreprise qui en fait partie (parmi laquelle figure Boeing, TF1, Columbia Picture...) , montre que le projet est ou a été très actif. Mais à en croire les donnés statistique précédemment mentionnés, aussi bien au niveau du code, que des communication au sein de la communauté, le projet n'est pas spécialement actif.

Il convient donc d'analyser ce projet comme un étant un projet éditer par une seul entreprise. Cette entreprise a un business modèle a part de ce que ces concurrent ont, c'est à dire qu'au lieu de faire payer ces clients pour le logiciel, elle a choisi de créer un logiciel entièrement libre et de ce rémunérer (et plus particulièrement le groupe de développer qui en fait partit) à travers des différents développement spécifique, ou de services. Afin de pouvoir se concentrer sur le code et sur les logiciels quel produit, Heroine Virtual utilise l'entreprise LMA comme

[3] Liste des client de l'entreprise LMA : http ://lmahd.com/lmahd1/index.php ?option=com_content&view=article&id=87

vitrine commercial, et ainsi, les personnes travaillant chez heroine sont très majoritairement des ingénieur développeurs d'application. L'entreprise reste dans l'anonymat, et il n'a pas été possible de communiquer directement avec l'un de ses employés.

On peut aussi se demander pourquoi cette entreprise a décidé de faire le développement de manière caché. En effet, on peut penser que si celle-ci faisait le développement de manière ouverte, elle pourrai bénéficier de manière plus efficace de l'aide de la communauté. Il y a deux explication possibles à cela :

– Des contrats, ou partenariats les empêches de le faire
– Les développeurs considère que travailler au sein de la communautés serai est une perte de temps puisqu'il se verrai "dans l'obligation" de collaborer de manière bien plus poussé avec ces membres

Il apparait plus probable que la première explication soit la bonne, parce qu'en général bénéficier de l'aide de la communauté est quelque chose de positif, et l'on s'aperçoit que toute les entreprise (créant de logiciel commerciaux ou non) essaye de profiter au maximum de celles-ci.

2.5 Lacunes

2.6 Solutions possibles

Chapitre 3

Conclusion

Annexes

Interview de Sophian Veri, Jeune entrepreneur dans le milieu de l'édition vidéo

1- Quel logiciel d'édition vidéo utilisez vous à l'heure actuel ? Sophian Veri : J'utilise exclusivement finalcut

2- Quel types de vidéo produisez vous ? Sophian Veri : Je produis des clips, reportages, pubs et courts métrages

3- Ce logiciel répond t il à tout vos besoin en terme de montage ? Sophian Veri : généralement oui

4- Quels défauts vous viennent à l'esprit quand vous penser à cette outil ? Sophian Veri : le montage multicamera mal géré, temps de rendu trop long car le logiciel ne fonctionne pas avec toute la mémoire vive de l'ordinateur contrairement a la suite adobe

5- Considérez vous finalcut comme étant la meilleur solution de montage, si oui, pourquoi ? Sophian Veri : Oui, sans aucun doute. Pour sa facilité d'utilisation, les codecs accepté y sont nombreux et la liste des effets est longue...

6- Quels fonctionnalité utilisez vous au quotidien (ex : Multicamera, effets, transitions, keyframes, time remapping, outils collaboratifs, proxy editing, templates...)

Sophian Veri : effet de mauvais téléviseur, time remap, cache patate (qui m'évite de passé par after.) ralenti, fondu enchainé Thibault Saunier : Qu'es que cache patate ? Sophian Veri : cache patate, c'est masque dans after

7- Quels fonctionnalités considérez vous comme indispensable, (même si vous ne les utilisé pas au quotidien) ? Thibault Saunier : Oui je vois. Et il y en a d'autre dont vous avez forcement besoin ? Sophian Veri : la modification des couleurs tout ce qui es travail de l'image contraste netteté saturation

8- Quel pourcentage des fonctionnalité du logiciel pensez vous utilisez en tout ? Sophian Veri : il y a tellement de fonctionnalités que je serai tenté de dire 25

9- Seriez vous prêts à utilisez des logiciels ayant moins de fonctionnalités, mais qui répondrais de manière plus efficace à vos besoins ? Sophian Veri : pourquoi pas a condition qu'il soi aussi intuitifs et que les effets que j'utilise soient tout aussi bien gérés

10- Le prix du logiciel est-il un critère de choix selon vous ? Sophian Veri : oui, en tant que nouvelle jeune entreprise, le prix est un critère de choix

11- Avez-vous des problèmes de stabilité (de bugs) avec finalcut ? Sophian Veri : les bugs, assez rarement

12- La dépendance vis-à-vis du créateur du logiciel que vous utilisez vous parait il être quelque chose de dangereux ? Sophian Veri : oui dans le sens ou on ne sait jamais quels transformations le logiciel subira avec la version suivante et il se peu que la version devienne moins bien et qu'elle ne me satisfasse pas de la même manière. Comme avec finalcut 10 qui a l'air d'être raté.

13- Connaissez vous certains logiciels libre d'édition vidéo ? Sophian Veri : Je ne sais pas vraiment ce que c'est.

Thibault Saunier : Merci bien d'avoir pris le temps de répondre Sophian Veri : mais de rien ...

Interview de Karim Hachemi, Monteur chez Falfyprod

1- Quel logiciel d'édition vidéo utilisez vous à l'heure actuel ? Karim Hachemi : Final cut mais surtout Adobe CS5

2- Quel types de vidéo produisez vous ? Karim Hachemi : clip, pub, institutionnel

3- Ce logiciel répond t il à tout vos besoin en terme de montage ? Karim Hachemi : Adobe Première non mais final cut oui le problème est que les rush du 5D sont mal accepter par final cut donc besoin de les convertir donc perte de temps enorme

4- Quels défauts vous viennent à l'esprit quand vous penser à cette outil ? Karim Hachemi : Final cut est un tout petit peu plus compliqué mais a est plu complet enfin c'est e que j'ai comme impression

5- Quels fonctionnalité utilisez vous au quotidien (ex : Multicamera, effets, transitions, keyframes, time remapping, outils collaboratifs, proxy editing, templates...)

Karim Hachemi : effet,transition en general je me sert que de ça

6- Quels fonctionnalités considérez vous comme indispensable, (même si vous ne les utilisé pas au quotidien) ? Le rognage de final cut sur premiere il est mal conçu et c'est très énervant.

7 Quel pourcentage des fonctionnalité du logiciel pensez vous utilisez en tout ? Karim Hachemi : Je dirais 30-35

8- Seriez vous prêts à utilisez des logiciels ayant moins de fonctionnalités, mais qui répondrais de manière plus efficace à vos besoins ? Karim Hachemi : cela dépend si il manque des trucs dont je ne me suis jamais servis je m'est égale. Le problème c'est trouver lesquels car au final mieux en avoir trop que pas assez.

9- Le prix du logiciel est-il un critère de choix selon vous ? Karim Hachemi : oui

10- Avez-vous des problèmes de stabilité (de bugs) ? Karim Hachemi : non pas trop de bug mais ce qui est ennuyeux c'est les rendu beaucoup trop long

11- Connaissez vous certains logiciels libre d'édition vidéo ? Karim Hachemi : Non, mais il faut que j'essaye

Interview de Yves Faure, responsable technique à TL7 (Télévision de Saint Étienne)

1- Quel logiciel d'édition vidéo utilisez vous à l'heure actuel ? Yves Faure : Pour le montage video, principalement Final cut pro il peu arriver dans de très rare cas que l'on utilise adobe premiere. Notre parque informatique est basé sur mac. En ce qui concerne l'habillage et l'infographie on utilise photoshop et after effect pour les effets (bien que dans de nombreux cas, on fasse les effets directement dans Final Cut).

2- Quel format de vidéo produisez vous ? Un peu de tout : Reportage, documentaire, plateau magazine, film de reportage, spot publicitaire, captation musique et théâtral.

3- Ce logiciel répond t il à tout vos besoin en terme de montage ? Oui, largement. Nous avons des besoins spécifique en terme d'organisation, archivage, gestion de sous-titrage, mais cela sort du scope du logiciel de montage.

4- Quels défauts vous viennent à l'esprit quand vous penser à cette outil ? Le gros problème qui me vient à l'esprit est le fait que Final Cut 10 soit extrêmement osé. Apple q décidé de revoir complètement l'interaction utilisateur et cela va nous faire perdre du temps (et donc de l'argent).

Il y a aussi de petits default d'ergonomie qui sont irritants.

Le fait qu'il soit aussi puissant est pour nous un default puisque cela complexifie la tache du monteur.

Son prix très élevé est aussi un problème pour notre structure (bien que bien moins chère que d'autres concurrents.

Par default, les fichier de rendu video, le cache de vignette, est stocké dans le dossier final cut pro global au système et pas avec le projet, ce qui signifie que l'on doit changer cela à chaque fois et une fois de plus c'est une grosse perte de temps.

Lorsqu'il y q des rupture de timecodes dans les fichier, le logiciel réagit mal ; et cela est source de problème régulièrement.

La gestion des formats est assez mauvaise.

Trop configurable, a tout les moment de l'édition.

Pas de sortie moniteur direct chez apple. Pour visualiser le rendu final sur les moniteur et ainsi être sur de la qualité du montage (en particulier au niveau de la lumière et des couleur,

on est obligé d'effectuer le rendu et ensuite seulement le voir sur les moniteurs dédiés. On devrai pouvoir brancher nos mac sur les moniteur et regarder en temps réel le résultat final.

5- Quels qualité fait que vous êtes satisfait de ce logiciel ?

La dernière version de Final cut permet le réetalonage automatique (de l'image et du son), cela va vraiment faciliter le travail des monteurs.

Le fait qu'il s'agit du standard actuel dans le milieu est très important pour nous. Cela nous permet de communiquer facilement avec nos confrère.

Le fait que l'on ai les effets directement intégré dans le logiciel nous permet d'accélérer le montage dans de nombreux cas.

Dans le cadre de magazines et films publicitaire, on utilise beaucoup les animation (transformations) tel que la modification de l'échelle de l'image, le rognage. Aussi, le lissage des bords et les ombres portés de l'image nous permette régulièrement de faire de montages mieux léchés.

6- Pour vous, finalcut est la meilleur solution de montage ? Oui, c'est sure

7- Quels fonctionnalité utilisez vous au quotidien (ex : Multicamera, effets, transitions, keyframes, time remapping, outils collaboratifs, proxy editing, templates...)

Au quotidiens, nous utilisons : Transition, effet, mixage audio, montage cut, retouche de couleurs, étalonnage, colorimétrie, multi-images, Gestion de la vitesse par clip entier, pas de time remapping a proprement parlé.

8- Quels fonctionnalités considérez vous comme indispensable, (même si vous ne les utilisé pas au quotidien) ?

Je pense que le multicamera (pour la couverture d'évènement tel que les concerts) est la fonctionnalité importante même si pas utilisé au quotidien.

9- Quel pourcentage des fonctionnalité du logiciel pensez vous utilisez en tout ?

En general 10

10- Seriez vous prêts à utilisez des logiciels ayant moins de fonctionnalités, mais qui répondrais de manière plus efficace à vos besoins ?

Non, car très nous sommes très attaché à la connaissance du logiciel. Changer de logiciel voudrais dire, 40 personnes à former, changer les habitudes et cela est très complexe, en particulier pour les professionnels du montage vidéo ! Nous avons une solution qui nous conviens et qui est leader sur le marché, il faudrait une vrai évolution du marché pour que l'on pense à changer.

11- Le prix du logiciel est-il un critère de choix selon vous ?

Oui et non, c'est chère mais ca marche. Autant ce donner les moyens pour avoir un produit qui nous permet d'en gagner par la suite.

12- Avez-vous des problèmes de stabilité (de bugs) avec finalcut ?

Non, la stabilité est un point fort de Final Cut.

13- La dépendance vis-à-vis du créateur du logiciel que vous utilisez vous paraît il être quelque chose de dangereux ?

Non, on s'arrange comme on peu. La version que l'on utilise actuellement nous convient, si il faut que l'on continu avec celle-ci, on le ferra.

Index

Table des figures

Glossaire

adapter Il permet de convertir l'interface d'une classe en une autre interface que le client attend. L' Adaptateur fait fonctionner ensemble des classes qui n'auraient pas pu fonctionner sans lui, à cause d'une incompatibilité d'interfaces. Source : Wikipedia, 40

API Une interface de programmation (Application Programming Interface ou API) est une interface fournie par un programme informatique. Elle permet l'interaction des programmes les uns avec les autres, de manière analogue à une interface homme-machine, qui rend possible l'interaction entre un homme et une machine, 23

Cas d'utilisation (use case) un cas d'utilisation définit une manière d'utiliser le système et permet d'en décrire les exigences fonctionnelles., 3

codec Un codec est un procédé capable de compresser et/ou de décompresser un signal numérique. Ce procédé peut être un circuit imprimé ou un logiciel., 34

espace colorimétrique Un espace colorimétrique ou espace de couleur associe des nombres aux couleurs visibles. Compte tenu des limites de la vision humaine, ces nombres se présentent généralement sous la forme de triplets. Chaque couleur de lumière peut donc être caractérisée par un point dans un espace à trois dimensions. Lors d'une impression, pour des raisons liées à la qualité des pigments, l'espace utilisé comporte alors généralement au moins quatre dimensions. Source : Wikipedia, 35

framework Un framework est un ensemble d'outils et de composants logiciels organisés conformément à un plan d'architecture et des designs patterns (un patron de conception, motif de conception ou modèle de conception est un concept de génie logiciel destiné à résoudre les problèmes récurrents suivant le paradigme objet.), 30

git Git est un logiciel de gestion de versions décentralisée. C'est un logiciel libre créé par Linus Torvalds, le créateur du noyau Linux, et distribué sous la GNU GPL version 2., 53

gtk+ The GIMP Toolkit est un ensemble de bibliothèques logicielles, permettant de réaliser des interfaces graphiques. Cette bibliothèque a été développée originellement pour les besoins du logiciel de traitement d'images GIMP. GTK+ est maintenant utilisé dans de nombreux projets, dont les environnements de bureau GNOME, Xfce et ROX., 51

Interface Utilisateur User Interface, il s'agit du terme très largement employé pour définir l'interface utilisateur, en général graphique ou GUI, 13

mpeg MPEG, sigle de Moving Picture Experts Group, est le groupe de travail SC 29/WG 11 du comité technique mixte JTC 1 de l'ISO et de la CEI pour les technologies de l'information. Ce groupe d'experts est chargé du développement de normes internationales pour la compression, la décompression, le traitement et le codage de la vidéo, de l'audio et de leur combinaison, de façon à satisfaire une large gamme d'applications. Source : Wikipedia, 35

MXF Material eXchange Format ou MXF est un conteneur utilisé par les professionnels pour les données audio et vidéo numériques. Il s'agit d'un format défini par des standards de la SMPTESMPTE. (Source : wikipedia), 14

Nvidia Nvidia Corporation est l'un des plus grands fournisseurs de processeurs graphiques, de cartes graphiques et de chipsets pour PC et consoles de jeux, 42

POSIX POSIX est le nom d'une famille de standards définis depuis 1988 par l'Institute of Electrical and Electronics Engineers et formellement désignée IEEE 1003. Ces standards ont émergé d'un projet de standardisation des API des logiciels destinés à fonctionner sur des variantes du système d'exploitation UNIX. Il s'agit de la standardisation des API des systèmes communément appelé UNIX. Source : Wikipedia, 37

QT framework orienté objet et développé en C++ par Qt Development Frameworks, filiale de Nokia. Il offre des composants d'interface graphique (widgets), d'accès aux données, de connexions réseaux, de gestion des fils d'exécution, d'analyse XML, etc. Qt est par certains aspects un framework lorsqu'on l'utilise pour concevoir des interfaces graphiques ou que l'on architecture son application en utilisant les mécanismes des signaux et slots par exemple., 43

smpte Society of Motion Picture and Television Engineers, est une association internationale, située aux É.-U., et composée d'ingénieurs. Elle développe des standards vidéos (elle en a déjà plus de 400 à son actif), qui sont utilisés par exemple par la télévision, ou le cinéma numérique (Source : http ://fr.wikipedia.org/), 11

Système de gestion de version Un logiciel de gestion de versions (ou VCS en anglais, pour Version Control System) est un logiciel qui permet de stocker un ensemble de fichiers en conservant la chronologie de toutes les modifications qui ont été effectuées dessus. Il permet notamment de retrouver les différentes versions d'un lot de fichiers connexes. Source : Wikipedia, 53

thread Il s'agit de la plus petit unité de processus qui peut être exécutée par un système d'exploitation. Ils permettent à un même "programme", processus d'exécuter du code de manière concurrente., 45

upstream Version développé par les communauté qui maintien officiellement le code source d'un logiciel., 55

VDPAU VDPAU (Video Decode and Presentation API for Unix) est une bibliothèque open source (libvdpau) et une interface de programmation conçus par NVIDIA initialement pour ses cartes graphiques GeForce 8 et ses derniers processeurs graphiques. Cette interface permet à des programmes de vidéo de décharger de la mémoire des parties du processus de décodage de vidéo et de son traitement aval vers le processeur graphique., 42

Bibliographie

[1] Ken Dancyger. *The technique of film and video editing, History, Theory, and Practice.* 2007.

[2] Mathieur Duponchelle. Gstreamer-editing-services implemented in pitivi. *http ://www.google-melange.com/gsoc/proposal/review/google/gsoc2011/mathieu_duponchelle/1001*, 2011.

[3] Frost and Sulivan. *World non linear editing market.* December 2009.

[4] Ellis Hamburger. Apple's new "final cut pro x" software is getting trashed by app store reviewers. June 2011.

[5] Edward Hervey. The result of the past few months of hacking. *http ://blogs.gnome.org/edwardrv/2009/11/30/the-result-of-the-past-few-months-of-hacking/*, 2009.

[6] Lightworks. *The lightworks open source project starts here.* 2010.