



## An Ad-Hoc Social Network Generation Approach

---

Martin Xavier Tchembe, Maurice Tchoupé Tchendji and  
Armelle Linda Matene Kakeu

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

December 10, 2019



# Une approche de génération de réseaux sociaux ad-hoc

Martin Xavier TICHEMBE<sup>\*\*</sup>, Maurice TCHOUPÉ TCHENDJI<sup>\*,\*\*</sup> et Armelle Linda MATENE KAKEU<sup>\*\*</sup>

\* Equipe FUCHSIA/LIRIMA

\*\* Département de Mathématiques et Informatique, Faculté des Sciences

Université de Dschang, BP 67, Dschang-Cameroun

ttchoupe@yahoo.fr, {martinxavierchembe, kakeulinda7}@gmail.com



**RÉSUMÉ.** L'utilisation des réseaux sociaux reste de nos jours encore confinée dans des réseaux à infrastructures comme Internet. Cependant, bien des situations (conférences, fêtes foraines, etc) peuvent nécessiter la mise en œuvre et le déploiement rapide d'une application Ad-hoc de dissémination d'informations entre participants à une activité : c'est ce type d'applications que nous appelons Réseau Social ad-hoc. Ces applications sont déployables sur des unités mobiles se déplaçant de façon libre et arbitraire, distribuées, etc. Elles partagent donc inéluctablement les mêmes caractéristiques que celles inhérentes aux réseaux mobiles ad-hoc et font par conséquent de ceux-ci de bons candidats pour leur exploitation. Dans ce papier, en faisant usage des techniques et outils issues du domaine de la programmation générative, nous proposons une démarche de production des environnements de génération de telles applications à partir de leurs spécifications dans un langage dédié. Par application de cette démarche, nous avons développé *SMGenerator*, un environnement de génération des applications mobiles de type réseaux sociaux ad-hoc déployables sur des terminaux Android. Bien plus, à partir de cette plate-forme, nous avons généré aisément l'application ConflInfo : un réseau social ad-hoc de dissémination d'informations aux participants d'une manifestation scientifique.

**ABSTRACT.** The use of social networks is still confined to infrastructure-based networks such as the Internet. However, many situations (conferences, fairs, etc.) may require the implementation and rapid deployment of an ad-hoc application for disseminating information between participants in an activity: it is this type of application that we call an Ad-hoc Social Network. These applications are deployable on mobile units moving in a free and arbitrary way, distributed, etc. They therefore inevitably share the same characteristics as those inherent in ad-hoc mobile networks and therefore make them good candidates for their deployment. In this paper, by using techniques from the field of generative programming, we propose an approach to produce environments for generating such applications from their specifications in a domain specific language. By applying this approach, we have developed *SMGenerator*, an environment for generating mobile ad-hoc social networks applications that can be deployed on Android devices. Moreover, from this platform, we easily generated the ConflInfo application: an ad-hoc social network application for disseminating information to participants of a scientific manifestation.

**MOTS-CLÉS :** Réseaux Sociaux Ad-Hoc, Réseaux Mobiles Ad-Hoc, Langages Dédiés, Programmation Générative, Dissémination d'Information, Publish/Subscribe, XML

**KEYWORDS :** Ad-Hoc Social Network, Mobile Ad-Hoc Network, Domain Specific Language, Generative Programming, Information Dissemination, Publish/Subscribe, XML



---

## 1. Introduction

De nos jours, les équipements mobiles (smartphones et tablettes) sont largement utilisés pour accéder aux réseaux sociaux. Facebook par exemple a enregistré 1 milliard et 740 millions d'utilisateurs mobiles vers la fin d'année 2016 [8]. Ces réseaux sociaux sont généralement conçus pour fonctionner dans un réseau à infrastructure comme Internet. Cependant, on peut imaginer bien des situations dans lesquelles on peut avoir besoin de créer et déployer une application de type réseau social pour répondre à un besoin spécifique. Un tel besoin peut notamment se poser si l'infrastructure est inexistante ou en panne, ou alors si elle est coûteuse, etc. De telles applications sont généralement qualifiées de *réseaux sociaux ad-hoc* (en anglais Ad-hoc Social Networks (ASN)) ou *réseaux sociaux ad-hoc mobiles* (en anglais Mobile Ad-hoc Social Networks (MASN)) et ont été au centre de plusieurs travaux de recherches [9, 10, 11]. Dans ce papier, nous nous intéressons aux réseaux sociaux ad-hoc basés sur un modèle de communication par publish/subscribe [6] dans lesquels les informations échangées dans le réseau sont regroupées par sujet: nous les appellerons *SocialMANET* par la suite).

La mise en œuvre d'un SocialMANET peut s'avérer longue et fastidieuse; bien plus, seule des personnes possédant des connaissances poussées en programmation peuvent s'y aventurer. Toutefois, si l'on observe attentivement les structures des applications qu'on peut créer de façon ad-hoc pour la dissémination ciblée d'informations entre des personnes partageant des intérêts communs et susceptibles de se retrouver très souvent par groupes (cas des participants à une conférence, à une fête foraine, à des manifestations sportives du type des jeux universitaires, etc.), on peut remarquer qu'elles constituent une famille d'applications pouvant être représentée par l'architecture logicielle générique (en 4 couches) présentée sur la figure 1. En effet, elles utilisent le même protocole de dissémination, ont des interfaces graphiques offrant les mêmes styles d'interactions et structurées à l'identique, etc. Elles ne diffèrent véritablement que relativement aux sujets de discussions en jeux (leurs intitulés et leur structuration), les images/icônes, ainsi que quelques labels utilisés spécifiquement sur l'interface de chaque application.

Si on instancie l'architecture de la figure 1 pour la mise en œuvre de deux (ou plus) applications de type réseau social ad-hoc, on constate (voir sec. 2.2) que l'ensemble des couches peut être partitionné en deux sous-ensembles : celui constitué des couches dites *communes* (couches Service, Application et Communication) dont l'implémentation est réutilisée pour tous les SocialMANET et celui constitué des couches dites *spécifiques* (couche IHM + une partie du stockage) dont l'implémentation diffère d'un SocialMANET à l'autre. Le présent papier a pour objectif de proposer et d'expérimenter une démarche permettant la génération des SocialMANET. Les *couches spécifiques* sont spécialement utilisées dans la méthodologie adoptée à cet effet; elle se décline en trois étapes: 1) description d'une démarche générique de production des environnements de génération de SocialMANET; 2) instanciation de la démarche proposée pour le développement de *SMGenerator*: un environnement de génération de SocialMANET pour les plate-formes Android; 3) expérimentation de *SMGenerator* pour la génération de ConfInfo: un SocialMANET de dissémination d'information aux participants à une conférence scientifique.

Au cœur de notre approche, se trouve l'utilisation intensive des techniques et outils du domaine de la programmation générative, permettant non seulement de générer du code à partir des spécifications grammaticales, mais aussi, de les intégrer de façon à obtenir un produit logiciel fiable et efficient.

**Organisation du manuscrit :** la section 2 présente en guise de préliminaires quelques généralités sur les SocialMANET ainsi qu'un rappel sur les techniques de génération de code à partir des spécifications grammaticales. La section 3 présente la démarche proposée ; son instanciation pour la production de l'environnement *SMGenerator* ainsi qu'une expérimentation de celui-ci pour la génération du SocialMANET Confinfo constituent l'essence de la section 4. La section 5 est consacrée à la conclusion.

---

## 2. Préliminaires

### 2.1. Généralités sur les SocialMANET: définition et principe de fonctionnement

Un *SocialMANET* est une application ad-hoc, mobile et distribuée, permettant la dissémination d'information entre personnes intéressées par des publications relatives à un ou plusieurs sujets. Ces sujets peuvent concerner des domaines variés (sport, étude, conférences, etc.) et sont décomposables si besoin est, en sous-sujets formant une hiérarchie représentable sous forme arborescente (voir fig 3).

Pour recevoir les publications effectuées dans un sujet, l'utilisateur du SocialMANET doit au préalable s'y abonner: le modèle de communication mise en œuvre ici est celui du *publish/subscribe*<sup>1</sup> [6]. Notons qu'une souscription à un sujet appartenant à une hiérarchie entraîne implicitement un abonnement à tous les (sous-)sujets situés en dessous de lui (ses "sujets enfants") dans la hiérarchie; ceci permet non seulement de pouvoir effectuer des souscriptions raffinées mais aussi, de réduire le nombre de souscription nécessaire pour recevoir des publications effectuées dans plusieurs sujets.

Dans les SocialMANET, les informations ne sont disséminées que lorsque les nœuds (terminaux mobiles) sont situés à proximité l'un de l'autre<sup>2</sup>. Toutefois, la mobilité des nœuds couplée à l'existence de *nœuds altruistes*<sup>3</sup> permettent d'assurer une dissémination à grande portée. En effet, du fait de la mobilité des nœuds, une information finira avec une probabilité assez grande à être disséminée même à ceux des abonnés se trouvant loin du lieu initial de sa publication [7].

### 2.2. Architecture générique des SocialMANET

L'architecture générique d'un nœud d'un SocialMANET (voir fig. 1) est constituée de quatre couches: les couches Communication, Application, Service et IHM (Interface Homme-Machine).

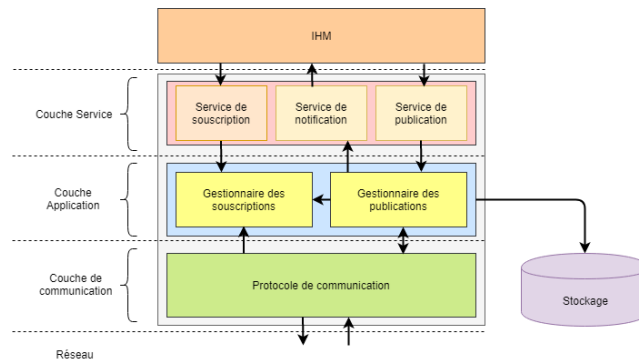
La couche de Communication assure la communication entre le nœud courant et les autres nœuds du réseau; c'est à son niveau qu'est implémenté le protocole de dissémination décrit dans [7]. A la réception d'une publication susceptible d'intéresser le nœud courant, elle la transfère au "Gestionnaire des publications" situé dans la couche Appli-

---

1. Le modèle de communication par *publish/subscribe* offre un haut niveau de découplage dans le temps et dans l'espace entre les parties en interaction. Il est constitué des éditeurs et abonnés qui interagissent autour de certains sujets. Les éditeurs peuplent le système avec des informations et les associent à des sujets par une action de publication, et le système se charge d'acheminer ces informations vers ceux ayant préalablement déclarés leurs intérêts par rapport à ces sujets (les abonnés) par une opération de souscription.

2. Dans les SocialMANET, les stations mobiles communiquent par ondes radio qu'elles émettent et captent.

3. De tels nœuds transportent des publications même sur des sujets auxquels ils ne sont pas abonnés.



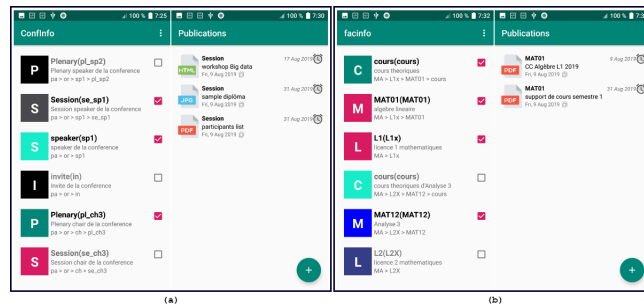
**Figure 1. Architecture générique (en couches) des SocialMANET**  
 cation. Cette dernière implémente la logique métier de l'application à travers deux composants dont les noms très évocateurs laissent présager de leurs rôles: le *Gestionnaire des publications* et le *Gestionnaire des souscriptions*. La couche Service quand à elle expose les services de Souscription et de Publication invoquables par l'utilisateur à travers l'IHM notamment pour souscrire à un sujet (Service de Souscription) ou pour publier une information (Service de Publication); elle expose aussi le Service de Notification qui peut être invoqué par le système pour notifier l'utilisateur de la réception d'une nouvelle publication. La couche IHM offre un style d'interaction permettant d'invoquer les services offerts par la couche Service. Ainsi donc, pour un SocialMANET donné, elle doit permettre d'accéder à la liste des sujets en jeu (ceux auxquels on peut souscrire) ainsi qu'à leur hiérarchie, aux fonctionnalités de souscription, de publication et de gestion (lecture, archivage, suppression, etc.) des publications.

A la lecture de la description donnée ci-dessus de l'architecture générique d'un nœud d'un SocialMANET, on se rend aisément compte que d'un SocialMANET spécifique à un autre, seules l'implémentation de la couche IHM et (dans une moindre mesure) la structuration du stockage<sup>4</sup> diffèrent. Bien plus, si l'IHM est graphique et offre un style d'interaction tactile, toutes les fenêtres des SocialMANET auront le même design (même organisation des composants graphiques) et les seules véritables différences porteront sur la nomenclature des sujets et leur hiérarchisation. La figure 2 présente deux IHM de deux SocialMANET permettant la dissémination d'information lors d'une manifestation scientifique (ConfInfo) (fig. 2(a)), et la dissémination d'informations d'ordre académique dans un établissement d'enseignement supérieur (FacInfo) (fig. 2(b)). Les hiérarchies des sujets qui y sont respectivement utilisées sont représentées sur la figure 3.

### 2.3. Génération de code à partir d'une spécification grammaticale

A la fin de la section précédente, nous avons fait le constat suivant lequel un SocialMANET spécifique ne diffère d'un autre que sur certains aspects de l'IHM se rapportant aux sujets manipulés. Nous explorons dans la présente sous-section les techniques de génération du code devant constituer l'implémentation des *couches spécifiques* de l'architecture d'un SocialMANET donné si la spécification de ses sujets est effectuée à l'aide d'un modèle grammatical.

4. La structuration du stockage en répertoires et sous-répertoires reflète la hiérarchie des sujets.

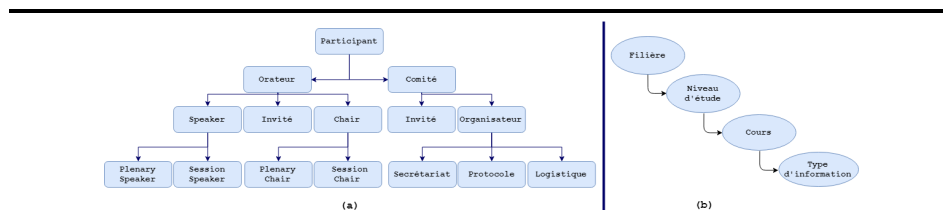


**Figure 2.** Aperçu des IHM de deux SocialMANET: ConInfo (a) et FacInfo (b)

La programmation générative est un paradigme de programmation ayant pour objet le développement des techniques permettant de générer du code source à partir des modèles afin non seulement de réduire le temps de développement des produits logiciels mais aussi, de garantir leurs corrections relativement aux modèles considérés [2]. Les modèles manipulés découlent des spécifications qui, en accord avec Jean-Claude Tarby [4] et suivant la classe du problème traitée, peuvent être graphiques (un diagramme de classe UML par exemple), fonctionnelles ou basées sur un langage dédié ou DSL (Domain Specific Language).

En tenant compte du fait qu'une des déclinaisons de notre objectif principal est de pouvoir spécifier les sujets d'un SocialMANET ainsi que leur hiérarchisation, c'est la spécification basée sur les DSL qui semble la mieux adaptée. Dans ce type de spécification, un environnement de travail dédié, permettant de manipuler directement les concepts du domaine est mis à la disposition du concepteur. Cet environnement intègre un DSL dans lequel le concepteur doit rédiger sa spécification. Bien plus, il donne accès à un ensemble d'outils (interpréteur, débogueur, éditeur spécialisé, etc.) permettant une exploitation efficace et efficiente du DSL. Par exemple, l'éditeur spécialisé doit offrir des facilités d'édition (complétion de code, coloration syntaxique, etc.) qu'on retrouve dans les éditeurs des langages généralistes traditionnels; en outre, l'interpréteur intégré doit implémenter des routines permettant de générer le code source correspondant à la spécification rédigée conformément à la syntaxe du DSL.

De nos jours, la conception et la mise en œuvre d'un DSL est relativement simple dès lors qu'on utilise les *Meta-DSL* (Xtext [3], JetBrains MPS [5] etc.) qui sont des frameworks permettant de créer des DSL; par exemple, l'environnement *SMGenerator* présenté à la section 4.1 a été créé avec Xtext.



**Figure 3.** Hiérarchies des sujets pour les SocialMANET ConInfo (a) et FacInfo (b)

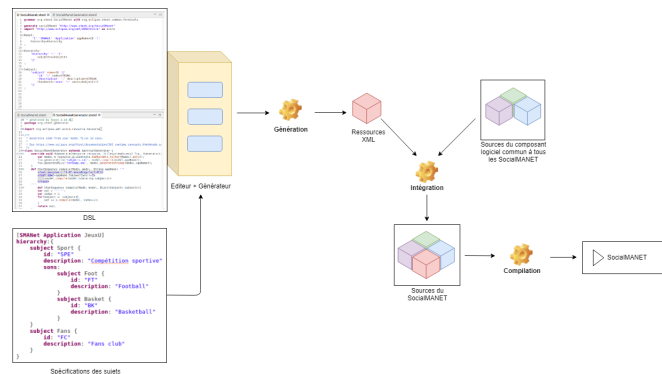


Figure 4. Vue synoptique du processus de génération d'un SocialMANET

### 3. Une approche grammaticale de génération de réseaux sociaux ad-hoc

#### 3.1. Vue synoptique de la démarche

La figure 4 donne une vue synoptique de l'approche orientée langage que nous préconisons pour la génération des SocialMANET. Elle peut être présentée comme suit: 1) créer un ensemble de composants logiciels implémentant les *couches communes* de l'architecture générique des nœuds des SocialMANET (voir fig. 1); 2) créer un DSL pour la spécification des concepts (les sujets et leur hiérarchie, le nom de l'application, etc.) utilisé pour la mise en œuvre des *couches spécifiques* de l'architecture générique des nœuds des SocialMANET; 3) fournir un outil d'intégration permettant d'assembler le résultat de l'interprétation d'un programme écrit dans le DSL créé à l'étape 2 avec les composants logiciels fournis à l'étape 1. Le résultat obtenu est compilé en une archive prête à être déployée.

Nous explicitons ci-dessous ce qui doit être fait dans les étapes 2 et 3 car, l'étape 1 consiste essentiellement en l'implémentation du protocole de dissémination présenté dans [7] et au développement des composants logiciels exposant les services de souscription, de notification et de publication (voir fig. 1). Notons dès à présent que dans la pratique (voir sec. 4), les étapes 2 et 3 seront exploitées via un environnement dédié à la spécification et à la génération des SocialMANET.

#### 3.2. Le modèle grammatical du DSL des SocialMANET et l'interpréteur associé

Comme déjà mentionné dans la section 2, les concepts pertinents utilisés pour la mise en œuvre des *couches spécifiques* d'un SocialMANET peuvent être capturés via un DSL dont une grammaire écrite sous la forme BNF (Backus-Naur Form) est donnée dans le listing 1. Cette grammaire permet au concepteur d'un SocialMANET de préciser le Nom du SocialMANET à générer à travers la catégorie syntaxique *NOM\_APPLICATION* (listing 1, ligne 4), la définition des sujets et des relations structurelles qui les lient, leurs descriptions, à travers la catégorie syntaxique *SUJET* (listing 1, ligne 3).

L'interpréteur associé au DSL créé doit émettre en sortie deux fichiers XML contenant respectivement une description hiérarchique de l'ensemble des sujets (voir fig. 5(b)) et un ensemble de labels devant être intégrés dans les composants logiciels fournis à l'étape 1.

Listing 1: Grammaire algébrique du DSL des SocialMANET

---

```
1 MODELE -> NOM_APPLICATION HIERACHIE
2 HIERACHIE -> (SUJETS)+
3 SUJET -> NOM_SUJET CODE_SUJET DESCRIPTION (SUJET)*
4 NOM_APPLICATION -> chaineDeCaractères
5 NOM_SUJET -> chaineDeCaractères
6 CODE_SUJET -> chaineDeCaractères
7 DESCRIPTION -> chaineDeCaractères
```

---

### 3.3. L'intégration

A cette étape, les deux fichiers XML générés à l'étape précédente (étape 2) constituent des ressources XML devant être intégrées aux composants logiciels fournis à l'étape 1 (voir fig. 4) afin d'obtenir le code source complet du SocialMANET spécifié via le DSL. Ce code est ensuite compilé en une archive prête à être déployée.

---

## 4. Application

Cette section présente un exemple de mise œuvre (expérimentation) de la démarche proposée pour la production de *SMGenerator*. Ici également est présentée une expérimentation portant sur l'utilisation de *SMGenerator* pour générer le SocialMANET ConfInfo.

### 4.1. SMGenerator: un environnement de génération de SocialMANET

*SMGenerator* est un environnement de génération de SocialMANET déployable sur des terminaux Android. Il est distribué sous forme de plug-in Eclipse<sup>5</sup>. Après son installation, il donne accès à un éditeur spécialisé de haut niveau permettant la spécification aisée d'un nouveau SocialMANET dans une implémentation du DSL décrit à la section 3.2; l'implémentation de ce DSL a été réalisée l'aide du framework Xtext<sup>6</sup> qui offre une solution solide pour la construction des DSL textuels sous la plate-forme Eclipse.

Après avoir rédigé une spécification correcte d'un nouveau SocialMANET dans l'environnement *SMGenerator*, il suffit de l'enregistrer pour déclencher l'étape d'interprétation. Cette étape a été mise en œuvre en personnalisant l'interpréteur générique généré par Xtext lors de la création d'un nouveau DSL. Plus précisément, la personnalisation a consisté en l'écriture du code permettant de générer les ressources XML à utiliser pendant l'étape d'intégration.

Pour l'étape d'intégration, nous avons écrit un script (invocé par clic sur un bouton disponible sur l'IHM, voir fig. 6(1)) contenant un ensemble de commandes nécessaires à la compilation d'un programme Android à partir des ressources XML générées et d'autres ressources. Pour le cas d'espèce, la ressource utilisée est une implémentation pour le cas des plate-formes Android des *couches communes* (couches Service, Application et Communication) de l'architecture générique des SocialMANET (voir fig. 1); cette ressource est disponible sous la forme d'un projet Android à l'adresse <https://github.com/MartinezX21/SMGenerator>.

---

5. [www.eclipse.org](http://www.eclipse.org)

6. [www.eclipse.org/Xtext](http://www.eclipse.org/Xtext)





Figure 5. *Fichier de spécification de ConfInfo dans SMGenerator (a) et fichier des sujets généré correspondant (b)*

#### 4.2. ConfInfo : un SocialMANET généré sur SMGenerator

ConfInfo est un SocialManet conçu pour faciliter la dissémination d’information lors d’une manifestation scientifique (conférences, workshops, séminaires, etc.). Les informations à disséminer sont regroupées en sujets dont la nomenclature reflète celle des catégories dans lesquelles on peut ranger les acteurs de la manifestation: les orateurs (plenary speaker, session speaker, talker), le comité d’organisation (secrétariat, protocole, logistique), les chairs (plenary chair, session chair), les invités, etc. La hiérarchie des sujets de ConInfo est présentée sur la figure 3(a); sa spécification dans l’environnement *SMGenerator* est donnée sur la figure 5(a) et un extrait de l’une des ressources XML générées après interprétation de cette spécification est présenté sur la figure 5(b). La figure 6 présente quelques captures d’écrans résumant les étapes de génération de l’APK de ConfInfo.

Pratiquement, après génération de l’APK de ConfInfo, il est mis à la disposition<sup>7</sup> des participants d’une manifestation scientifique. Ceux-ci doivent l’installer sur leurs mobiles Android puis, souscrire aux sujets de discussions correspondants à leurs catégories. Ceci fait, tout détenteur d’une nouvelle information à disséminer doit juste la publier dans le sujet correspondant et le processus de dissémination est immédiatement déclenché suivant le protocole de dissémination décrit dans [7]: l’information est préalablement distribuée aux participants situés à proximité du publieur, puis, de proche en proche, du fait de la mobilité des participants, elle sera finalement disséminée (avec une probabilité assez grande) à ceux d’entre eux se trouvant loin du lieu initial de publication.

7. L’application peut être disponible en téléchargement sur le site web de la manifestation ou alors remise aux participants pendant leurs enregistrements à leurs arrivés sur le site de la manifestation.

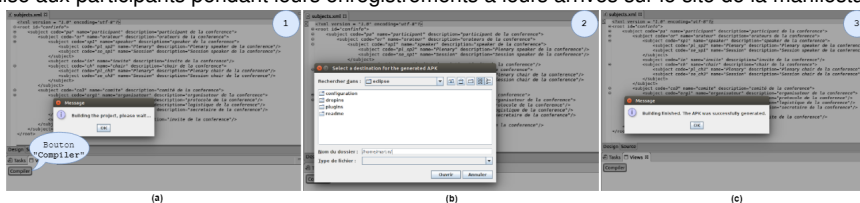


Figure 6. *Un résumé des étapes de génération de l’APK de ConFinfo*

## 5. Conclusion

Dans ce papier, nous avons présenté une approche générative de production des SocialMANET (application de type réseau social déployable sur un réseau mobile ad-hoc) qui, en plus de réduire drastiquement le temps à consacrer à leur développement (qui se résume désormais pratiquement au temps d'édition du fichier de spécification des sujets dans le DSL proposé) permet même aux personnes ne possédant pas de connaissances poussées en programmation d'en développer. La mise en œuvre suivant cette approche de l'environnement *SMGenerator* et son utilisation aisée pour la génération du SocialMANET ConfInfo nous a permis non seulement de valider l'approche, mais aussi, de confirmer son efficacité et son efficience.

Notons toutefois que le modèle grammatical proposé (voir listing 1) ne permet pas de capturer toutes les particularités qu'on peut souhaiter spécifier pour un SocialMANET donné. Un travail à court terme consisterait à enrichir ce modèle afin qu'il puisse permettre au concepteur d'un SocialMANET de préciser par exemple le logo de l'application, les thèmes de couleurs, les polices à utiliser et leurs tailles, la langue de l'application, etc.

---

## 6. Bibliographie

- [1] ANDRÉA KAPLAN , MICHAEL HAENLEIN « Users of the world, unite! The challenges and opportunities of social media », *Business Horizons*, vol. 53 (1), 2010
- [2] CARINE COURBIS, DIDIER PARIGOT, « La programmation générative pour le développement d'applications: les apports pour l'architecture », *ICSSEA*, 2003
- [3] BETTINI LORENZO « Implementing domain-specific languages with Xtext and Xtend », *Packt Publishing Ltd*, 2016
- [4] JEAN-CLAUDE TARBY, « Gestion Automatique du Dialogue Homme-Machine à partir de Spécifications Conceptuelles », *HAL*, 2007
- [5] VOELTER M. « Language and IDE Modularization and Composition with MPS », In: *Lämmel R., Saraiva J., Visser J. (eds) Generative and Transformational Techniques in Software Engineering IV. GTTSE 2011. Lecture Notes in Computer Science* vol. 7680, 2013.
- [6] PATRICK T. EUGSTER , PASCAL FELBER, RACHID GUERRAOUI, ANNE-MARIE KERMARREC « The many faces of publish/subscribe », *ACM Comput. Surv.*, vol. 35 (2), p. 114–131, 2003
- [7] MAURICE TCHOUPÉ T. , MARTIN X. TICHEMBE, INNOCENT T. NECHEU « SocialMANET: a publish/subscribe events dissemination protocol for mobile ad-hoc networks », *Journal of Computer Science*, vol. 15, p. 1237-1255, 2019.
- [8] CLEMENT J. « Facebook: number of mobile monthly active users worldwide 2009-2016 », 2018, <https://www.statista.com/statistics/277958/number-of-mobile-active-facebook-users-worldwide/> [Consulté le 5 Novembre 2019]
- [9] NAGENDER ANEJA, SAPNA GAMBHIRE « Profile-Based Ad Hoc Social Networking Using Wi-Fi Direct on the Top of Android », *Guest Editor: Subramaniam Ganesan*, vol. 8 n° 2, p. 1-8, 2018.
- [10] GAMBHIR S., ANEJA N., SILVA L. C.), « Piecewise Maximal Similarity for Ad-hoc Social Networks », *Wireless Personal Communications* vol. 97, p. 3519-3529, 2017.
- [11] WANG Y, WEI L, VASILAKOS A. V., JIN Q., « Device-to-Device based mobile social networking in proximity (MSNP) on smartphones: Framework, challenges and prototype », *Future Generation Comp. Syst.*, vol. 74, p. 241-253, 2017.