

# Influence des types d'organisation sur les performances d'un SAMU

Dans ce chapitre, nous étudions les données des SAMU afin d'en extraire de l'information et de la connaissance, utiles à l'établissement d'un état des lieux organisationnel objectif. Ce chapitre correspond à la partie droite du cadre (cf. figure 4.1). Nous appliquons d'abord la première étape de la méthodologie présentée dans la section 3.2.1 à propos de la récupération des données et de leur pré-traitement. Nous présentons ensuite la deuxième étape de la méthodologie présentée dans la section 3.2.2 à propos de l'interprétation des données.



Figure 4.1 – Partie de la méthodologie traitée dans ce chapitre

## 4.1 Étape 1 : Collecter et préparer les données

Grâce à nos SAMU pilotes de l'Aveyron (SAMU 12), du Tarn (SAMU 81) et de la Haute-Garonne (SAMU 31), nous avons pu avoir accès à des données de téléphonie réelles à l'échelle de l'appel. Les données ont été récoltées entre le 1<sup>er</sup> février 2018 et le 1<sup>er</sup> mars 2018 pour l'Aveyron et le Tarn. Pour la Haute-Garonne nous disposons uniquement des appels du 25 mars 2018 qui est un dimanche. Bien qu'une journée soit un échantillon d'une taille très faible, le fait que ce soit un dimanche rend cette journée significative. En effet, le dimanche est la journée où les SAMU reçoivent le plus d'appels. Il existe une quantité considérable de données de téléphonie par exemple, un appel peut générer à lui seul plusieurs dizaines de lignes de logs. Toutefois ces données ne sont pas faciles à obtenir, et encore moins à exploiter. Les SAMU n'ont pas directement la main dessus, ils ne disposent généralement pas des compétences nécessaires pour extraire et traiter les registres de téléphonie. Il a donc fallu faire appel au prestataire du logiciel de téléphonie pour qu'il fasse l'extraction et le prétraitement du registre (notamment l'anonymisation des numéros de téléphone) de manière à ce qu'il soit possible de l'exploiter dans le cadre de cette recherche.

Quoi qu'il en soit, le fait d'avoir des données de février à mars est particulièrement intéressant. En effet, ce sont des mois d'hiver durant lesquels l'épidémie saisonnière de grippe traverse la France. Durant cette période, le nombre d'appels au SAMU augmente. Ainsi ce sont des mois difficiles où les performances des SAMU sont mises à rude épreuve.

### 4.1.1 Les approches pour la découverte et la cartographie des processus

Lorsqu'on cherche à modéliser un système, on va bien souvent passer par la découverte des processus de l'organisation afin de savoir qui fait quoi, quand ? Pour découvrir les processus, cela nécessite d'obtenir de l'information à propos des activités du système. Pour faire cela, (Dumas et al., 2013) considèrent trois techniques de récupération et de création de processus :

- Les approches *Interview-based* qui réfèrent à des méthodes basées sur des entretiens auprès des experts du domaine pour leur demander comment sont effectués les processus. La difficulté étant que les processus sont bien souvent multi acteurs, ainsi pour avoir une vision globale du système il faut croiser les vues de chacun de manière à ne pas perdre d'information. L'avantage de cette technique est qu'elle permet d'avoir une vue très détaillée des processus de l'organisation. Toutefois, l'humain étant faillible par nature, les experts auront tendance à expliquer les processus dans le fonctionnement normal lorsque tout va bien ou au contraire à se focaliser sur les cas exceptionnels qui ont marqué les esprits. Ainsi, il est possible que les cas particuliers qui peuvent poser problème soient occultés.

- Les approches *workshop-based process discovery*. Cette technique consiste à former des ateliers de travail afin de rassembler tous les acteurs pour qu'ils puissent confronter leurs visions du processus et leurs places dans ce dernier. Cette méthode demande à être capable de réunir l'ensemble des rôles à une même réunion. Pour avoir une vision complète et satisfaisante du processus, plusieurs ateliers de ce type seront nécessaires.
- Les approches *Evidence-based* qui se basent sur l'étude d'éléments de descriptions tangibles d'un processus. Cela peut prendre la forme de documents écrits, d'organigrammes, de procédures. Toutefois, les documents de description édités par des humains peuvent être limités et oublier une partie du processus. C'est pourquoi il est intéressant de les croiser et de les compléter avec d'autres méthodes. Parmi les approches *Evidence-based*, on trouve aussi la découverte automatique de processus (*Process Mining*), qui elle se base sur les enregistrements faits dans le système d'information. Si les registres sont complets et que chaque événement peut être relié à un cas individuel du processus, à une activité et à un moment précis dans le temps, alors on peut reconstruire le processus dans son intégralité. Cette méthode implique d'avoir accès à ces données, ce qui n'est pas toujours évident.

**Tableau 4.1** – Forces et faiblesses de chaque méthode (d'après (Dumas et al., 2013))

Critère d'évaluation	<i>Evidence</i>	<i>Interview</i>	<i>Workshop</i>
Objectivité	fort	moyen fort	moyen fort
Richesse	moyen	fort	fort
Temps	moyen faible	moyen	fort
Retour des experts	faible	fort	fort

Chacune de ces méthodes a donc des avantages et des limites. Le tableau 4.1 résume les caractéristiques de chaque méthode selon plusieurs critères (vu par (Dumas et al., 2013)) : l'objectivité, la richesse en information, le temps passé, et le retour des experts. L'objectivité correspond à quel point on peut croire à la donnée récoltée comme étant la réalité. Il y aura plus de biais dans des données transmises par des humains que par des systèmes d'information. Par richesse, on entend la qualité et la précision de l'information que l'on va pouvoir avoir. La compréhension d'un système d'information peut être limitée là où un humain peut expliquer précisément un écart dans les données, car il sera capable de relier des données chiffrées à une situation complexe. Le temps d'une méthode à l'autre ne sera pas le même. Il sera plus long de réussir à réunir plusieurs acteurs à des ateliers que d'analyser des données, si on fait l'hypothèse que l'accès aux données est simple. Or dans notre domaine, nous l'avons vu, l'accès aux données est plutôt long et fastidieux. Selon la méthode il sera plus ou moins facile d'avoir un retour rapide des experts sur le modèle construit. Lors d'atelier ou d'entretien, le modèle de processus peut être construit de manière itérative rapidement, l'analyse des données rend l'échange avec les experts moins direct.

## 4.1.2 Sous-étape 1.a : Collecter et nettoyer les données

### Format des données récupérées

Les données ainsi récoltées, nous ont été transmises sous forme de registres d'événements (logs). Avant d'exploiter ces données, un traitement est nécessaire pour enlever les doublons, les données incomplètes et obtenir un format qui puisse être exploité. La figure 4.2 présente les deux grandes étapes appliquées et des extraits des logs à chacune des étapes. Un premier travail de pré-traitement a été fait par le prestataire de téléphonie qui nous a fourni les données. En effet, sans cela les données se présentaient sous la forme d'un fichier de texte difficilement exploitable. Le premier traitement appliqué permet d'obtenir une table de données avec différentes colonnes (cf. figure 4.3). Nous présentons ci-dessous les significations des éléments que nous avons utilisés par la suite. Afin d'illustrer le contenu de chaque colonne nous proposons un exemple dans la figure 4.4. :

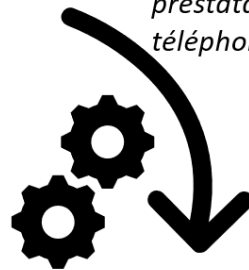
- **EventNo** : Le numéro de l'événement qui est un identifiant unique de l'événement. Dans l'exemple, nous avons 23 événements différents, chacun avec son propre identifiant.
- **EventName** : Le nom de l'événement. On constate qu'il y a deux types d'événements, ceux qui commencent par « Event », et ceux qui commencent par « Request ». Ces derniers sont des demandes d'information faites par le serveur, nous ne les utiliserons pas dans la suite, car ils ne nous apportent pas d'information pertinente. Nous reviendrons juste après sur les différents événements et leurs significations.
- **Localtime1** : Le jour de l'événement au format DD-MM-AAAA. Ici notre appel date du 31 janvier 2018.
- **LocalTime2** : L'heure de l'événement au format HH :mm :ss.ms. L'appel arrive enregistre son premier événement à 23 :43 :30 et il sera raccroché à 23 :46 :59.
- **ThisDN** : Le code du poste de travail, DN correspond à **Directory Number**. Dans notre exemple on constate 2 codes différents 25954 et 25009. Le code 25954 correspond à la file d'attente et 25009 correspond au code d'un poste de travail d'ARM.
- **OtherDN** : Le DN d'un autre poste de travail impliqué dans la gestion de l'événement courant. Cela peut par exemple être le numéro du poste du travail auquel l'appel est transféré. Typiquement sur l'événement numéro 107 **EventRouteCall** on a un **OtherDN** qui vaut 25009, cet événement nous indique que l'appel provient de la file d'attente **ThisDN** dont l'identifiant est 25954.
- **ThisQueue** : File d'attente dans laquelle est l'appel. Dans l'exemple on retrouve le code correspondant à la file d'attente soit 25954.
- **ThirdPartyDN** : Identifiant (DN) du poste de travail d'un troisième intervenant sur l'événement courant concernant l'appel.
- **ConnID** : Identifiant de l'appel, plusieurs événements vont être reliés à un même appel, afin de pouvoir regrouper ces événements l'appel à un identifiant

### Données de Téléphonie Brutes

```

@18 24 01 07 10 [5] 8.1.004.01 distributeur message EventReleased
AttributeTimeSecs 071000
AttributeTimeSecs 100802241 (18 24 01)
AttributeCustomerID "Resources"
AttributeCallMD "4375A14C269F5LP09052LPR68000026"
AttributeConnID 006f02a790029226
AttributeCallID 10009
AttributePropagatedCallType 0
AttributeCallType 0
AttributeThisDN "81111111"
@18 24 01 07 10 [5] 8.1.004.01 distributeur message EventReleased generated
@18 24 01 07 10 [5] 8.1.004.01 distributeur message EventReleased sent to [306] (00000005 StatusServer 172.18.15.118 62661)
-- call_md controlled destroyed
-- destroyed call_md_bpp 006f02a790029226@01e431a0
@18 24 01 07 10 [5] [bcp call [destroyed] [uuid 4375A14C269F5LP09052LPR68000026] [connection-id 006f02a790029226] [call-id 10009]]
-- Destroyed event Deleted
@18 24 01 07 10 [5] 8.1.004.01 distributeur message EventDelivered
AttributeTimeSecs 071000
AttributeCustomerID "Resources"
AttributeTimeSecs 100802241 (18 24 01)
AttributeAgentID 101801
AttributeThisDN 101801
    
```

Pré-traitement par le prestataire de téléphonie



### Données de Téléphonie sous forme de table (log)

EventNo	EventName	LocalTime	ThisDN	OtherDN	ThisQueue	ThirdParty DN	ConnID	Transfer ConnID	Previous ConnID	CallID	ANI	DNIS	Agent ID
15527	EventACK	2018-02-02 15:49:07.537	AgentStateDN										
15528	EventUserEvent	2018-02-02 15:49:07.537	AgentStateDN	AgentStateDN									
15529	EventTreatmentApplied	2018-02-02 15:49:07.672	25901		25901		006f02a790029226			4185	81LLL3H	25601	
15530	EventUserEvent	2018-02-02 15:49:07.672	25901		25901		006f02a790029226			4185	81LLL3H	25601	19
15531	RequestUpdateUserData	2018-02-02	25901				006f02a790029226						
15532	RequestSetCallInfo	2018-02-02					006f02a790029226						
15533	EventAttachedDataChanged	2018-02-02 15:49:07.673	25901			25901	006f02a790029226			4185	81LLL3H	25601	
15534	RequestSetCallInfo	2018-02-02					006f02a790029226						
15535	RequestDistributeEvent	2018-02-02	VQ_G_CENTRE15	V81LLL3HL			Q_G_CENTRE15			4185	81LLL3H	25601	
15536	EventACK	2018-02-02 15:49:07.673	VQ_G_CENTRE15										
15537	EventAttachedDataChanged	2018-02-02 15:49:07.673	VQ_G_CENTRE15	V81LLL3HL			Q_G_CENTRE15			4185	81LLL3H	25601	
15538	RequestUpdateUserData	2018-02-02	25901				006f02a790029226						
15539	RequestSetCallInfo	2018-02-02					006f02a790029226						
15540	EventAttachedDataChanged	2018-02-02 15:49:07.674	25901			25901	006f02a790029226			4185	81LLL3H	25601	
15541	RequestSetCallInfo	2018-02-02					006f02a790029226						
15542	RequestDistributeEvent	2018-02-02	VQ_G_CENTRE15	V81LLL3HL			Q_G_CENTRE15			4185	81LLL3H	25601	

### Données triées et nettoyées

Unnamed: 0.1	EventName	LocalTime	ThisDN	OtherDN	ConnID	DNIS
128	EventRingin	2018-01-31 23:43:30.468	25009	HV3O1V2O1	006f02a790028c34	25601.0
137	EventEstablished	2018-01-31 23:43:34.442	25009	HV3O1V2O1	006f02a790028c34	25601.0
881	EventReleased	2018-01-31 23:46:59.336	25009	HV3O1V2O1	006f02a790028c34	25601.0
142567	EventRingin	2018-02-02 07:58:24.035	25009	VTTLH202H	006f02a79002902c	25601.0
142580	EventEstablished	2018-02-02 07:58:24.573	25009	VTTLH202H	006f02a79002902c	25601.0
...	...	...	...	...	...	...
409966	EventEstablished	2018-04-02 23:53:56.719	25008	VOL8180TH	006f02a790039ab0	25601.0
409994	EventReleased	2018-04-02 23:54:52.192	25008	VOL8180TH	006f02a790039ab0	25601.0
410255	EventRingin	2018-04-02 23:57:29.018	25007	HVH8V0V0	006f02a790039ab3	25601.0

Nettoyage et traitement des données

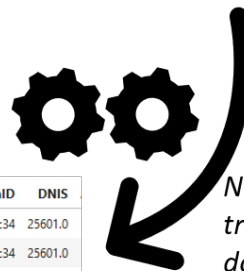


Figure 4.2 – Traitement des données pour obtenir un log exploitable.

EventNo	EventName	LocalTime	ThisDN	OtherDN	ThisQueue	ThirdParty DN	ConnID	Transfer Conn'n	Previous Conn'n	CallID	ANI	DNIS	Agent ID
15527	EventACK	2018-02-02 15:49:07.537	AgentStateDN										
15528	EventUserEvent	2018-02-02 15:49:07.537	AgentStateDN	AgentStateDN									
15529	EventTreatmentApplied	2018-02-02 15:49:07.672	25901		25901		006f02a790029226			4185	81LLL3H	25601	
15530	EventUserEvent	2018-02-02 15:49:07.672	25901		25901		006f02a790029226			4185	81LLL3H	25601	19
15531	RequestUpdateUserData	2018-02-02	25901				006f02a790029226						
15532	RequestSetCallInfo	2018-02-02					006f02a790029226						
15533	EventAttachedDataChanged	2018-02-02 15:49:07.673	25901			25901				4185	81LLL3H	25601	
15534	RequestSetCallInfo	2018-02-02					006f02a790029226						
15535	RequestDistributeEvent	2018-02-02	VQ_G_CENTRE15	V81LLL3HL	_G_CENTRE15		006f02a790029226			4185	81LLL3H	25601	
15536	EventACK	2018-02-02 15:49:07.673	VQ_G_CENTRE15										
15537	EventAttachedDataChanged	2018-02-02 15:49:07.673	VQ_G_CENTRE15	V81LLL3HL	_G_CENTRE15		006f02a790029226			4185	81LLL3H	25601	
15538	RequestUpdateUserData	2018-02-02	25901				006f02a790029226						
15539	RequestSetCallInfo	2018-02-02					006f02a790029226						
15540	EventAttachedDataChanged	2018-02-02 15:49:07.674	25901			25901				4185	81LLL3H	25601	
15541	RequestSetCallInfo	2018-02-02					006f02a790029226						
15542	RequestDistributeEvent	2018-02-02	VQ_G_CENTRE15	V81LLL3HL	_G_CENTRE15		006f02a790029226			4185	81LLL3H	25601	

Figure 4.3 – Extrait d'un log de données non traité.

unique. L'identifiant unique de notre appel est *006f02a790028c34*. C'est bien le même code qu'on retrouve pour chaque événement.

- **TransferConnID** : Identifiant de l'appel vers lequel la communication a été transférée. Lorsqu'un appel est transféré, par exemple, d'un ARM à un MR, il se peut que l'identifiant de l'appel change, car l'ARM peut mettre l'appel du patient en attente, composer un nouvel appel vers le MR puis transférer le patient sur ce nouvel appel avec le MR. Les deux appels ont alors des identifiants différents et ce champ permet de les relier. Il n'y a pas d'appels transférés dans notre exemple.
- **PreviousConnID** : Cet identifiant est utilisé dans le cas d'un transfert d'appel avec changement d'identifiant. Il correspond à l'identifiant de l'appel précédant depuis lequel le nouvel appel courant a été transféré (appel de consultation, par exemple lorsqu'un ARM appelle un MR afin de lui résumer l'appel). Dans notre exemple il n'y en a pas.
- **CallID** : Cet attribut contient l'identification de l'appel fournie par le commutateur, qui identifie un appel de manière unique. Contrairement au ConnID attribué par le serveur, le CallID est créé par le commutateur lorsque l'appel entrant arrive, ou lorsque les appels sortants de l'agent/du système sont créés. L'attribut doit être présent si le commutateur génère et distribue le paramètre correspondant au serveur (CallID est égal à zéro tant que le commutateur ne fournit pas cette information au serveur).
- **ANI** : Cela signifie *Automatic Number Identification* C'est le numéro de l'appelant. Dans notre exemple, les numéros des appelants ont été anonymisés c'est pourquoi nous avons un numéro sous la forme "HV3O1V2O1".
- **DNIS** : Cela signifie *Directory Number Identification System*, c'est le canal d'entrée utilisé par le numéro appelant, le numéro qu'il a composé pour joindre le centre d'appels. Ici le code est le 25601, qui correspond aux appels qui arrivent en ayant composé le 15.
- **AgentID** : L'identifiant de la ressource qui gère l'événement. L'appel a été géré par l'agent avec l'identifiant 25528. Si on fait le lien avec les tables de signification, on voit que cela correspond à un ARM.

Comme évoqué précédemment, dans cette table le champ **EventName** peut prendre plusieurs valeurs. De là, nous avons pu identifier les différents événements qui nous

EventNo	EventName	LocalTime	ThisDN	OtherDN	ThisQueue	ThirdPartyDN	ConnID	TransferConnID	PreviousConnID	CallID	ANI	DNIS	AgentID
103	RequestUpdateUserData	2018-01-31	25954				006f02a790028c34						
104	RequestSetCallInfo	2018-01-31					006f02a790028c34						
105	EventAttachedDataChanged	2018-01-31 23:43:30.266	25954			25954	006f02a790028c34			1430	/3O1V2C	25601	
106	RequestSetCallInfo	2018-01-31					006f02a790028c34						
107	RequestRouteCall	2018-01-31	25954	25009			006f02a790028c34						
112	EventUserEvent	2018-01-31 23:43:30.266	25954			25528	006f02a790028c34			1439			
114	EventRouteUsed	2018-01-31 23:43:30.266	25954	HV3O1V2O1	25954	25009	006f02a790028c34			1439	/3O1V2C	25601	
115	EventDiverted	2018-01-31 23:43:30.266	25954	HV3O1V2O1	25954	25009	006f02a790028c34			1439	/3O1V2C	25601	
123	RequestSetCallInfo	2018-01-31					006f02a790028c34						
124	RequestSetCallInfo	2018-01-31					006f02a790028c34						
125	RequestSetCallInfo	2018-01-31					006f02a790028c34						
126	RequestSetCallInfo	2018-01-31					006f02a790028c34						
127	RequestSetCallInfo	2018-01-31					006f02a790028c34						
128	EventCallPartyState	2018-01-31 23:43:30.467					006f02a790028c34						
129	EventRinging	2018-01-31 23:43:30.466	25009	HV3O1V2O1	25954		006f02a790028c34			1439	/3O1V2C	25601	25528
138	EventEstablished	2018-01-31 23:43:34.442	25009	HV3O1V2O1	25954		006f02a790028c34			1439	/3O1V2C	25601	25528
151	RequestUpdateUserData	2018-01-31	25954				006f02a790028c34						
152	RequestSetCallInfo	2018-01-31					006f02a790028c34						
153	EventAttachedDataChanged	2018-01-31 23:43:34.522	25009			25954	006f02a790028c34			1430	/3O1V2C	25601	
154	RequestSetCallInfo	2018-01-31					006f02a790028c34						
155	EventAttachedDataChanged	2018-01-31 23:43:34.522				25954	006f02a790028c34			1430	/3O1V2C	25601	
882	EventReleased	2018-01-31 23:46:59.336	25009	HV3O1V2O1	25954		006f02a790028c34			1439	/3O1V2C	25601	25528
883	EventReleased	2018-01-31 23:46:59.337AlcatelA4400::					006f02a790028c34			1430	/3O1V2C	25601	

Figure 4.4 – Exemple de log avant traitement, sous forme de table pour un appel.

intéressent dans un log de téléphonie afin de donner un sens métier aux événements. Les informations sont tirées du document de références du logiciel de téléphonie <sup>1</sup>.

- **Event Ringing** : l'évènement annonçant la survenue d'un appel (Sonnerie de l'appel).
- **Event Established** : l'évènement exprimant qu'un appel est décroché.
- **Event Abandoned** : l'évènement exprimant qu'un l'appel est abandonné.
- **Event PartyChanged** : L'objet de téléphonie spécifié par **OtherDN** a remplacé l'objet de téléphonie spécifié par **OtherDN** dans l'évènement reçu précédemment ; ou le **Previous ConnID** de l'appel s'est vu attribuer une nouvelle valeur de **ConnID**. Cela peut être le cas quand un appel est transféré, si le **Party** est modifié c'est que l'interlocuteur change. Cet évènement a lieu notamment quand un ARM transfère l'appel à un MR ou inversement.
- **Event Network Reached** : l'appel a rejoint le réseau distant. Cela concerne les appels sortants externes, c'est-à-dire les appels qui vont être passés en dehors du réseau du centre d'appels. Par exemple lorsqu'il faut contacter une agence d'ambulances ou encore les pompiers.
- **Event Held** : l'appel est mis en attente.
- **Event Retrieved** : l'appel est repris en ligne après avoir été mis en attente.
- **Event Party Added** : un ou plusieurs participants a été ajouté à l'appel à la suite d'une conférence. Si un seul participant est ajouté (comme dans le cas d'une simple conférence téléphonique), le nouvel interlocuteur correspondant est spécifié dans OtherDN.
- **Event Party Deleted** : l'objet de téléphonie spécifié par OtherDN a été supprimé de la conférence téléphonique en question. Cela signifie qu'une personne qui était dans la conversation téléphonique a quitté l'appel. C'est le cas lorsqu'un ARM à un patient en ligne, il crée une conversation téléphonique avec le MR. Il y a donc trois personnes dans la conversation, puis l'ARM se retire de la conversation laissant le patient avec le MR, c'est à ce moment ou on aura un **Event Party Deleted** avec en **OtherDN** le code de l'ARM.
- **Event Destination Busy** : l'appel sortant sonne occupé.

1. « Genesys Events and Models Reference, System Level Guides Current - 14-09-2018 »

- **Event Released** : l'appel est raccroché et terminé.
- **Event Error** : Une erreur est arrivée dans le log.

Une première étape qui concorde avec le nettoyage et la préparation des données et d'identifier les événements de début et les événements de fin possibles. En effet, une trace pour être dite complète doit avoir un début et une fin identifiables sinon c'est une trace incomplète qui ne sera pas prise en compte.

- **Event Ringing** est un événement de départ pour les appels entrants.
- **Event Dialing** est un événement de départ pour les appels sortants.
- **Event Released** est un événement de fin pour un appel où la communication a pu être établie.
- **Event Destination Busy** est un événement de fin pour les appels sortants qui ne sont pas décrochés, car l'appelé est occupé.
- Après un **Event Established** on peut avoir une **Event Released**.

## Nettoyage et préparation des données

L'objectif de cette partie est d'effectuer un prétraitement afin d'avoir une base de données complétée qui nous permette facilement de faire des analyses. Pour ce faire, plusieurs étapes ont été suivies :

- Seulement un certaines colonnes ont été conservées (**EventNo**, **EventName**, **LocalTime**, **ThisDN**, **OtherDN**, **ConnID**, **DNIS**, **AgentID**, **ANI**). La colonne **CallID** n'a pas été conservée, car elle était redondante avec l'identifiant du **ConnID**. Les colonnes **ThisQueue** et **ThirdPartyDN** sont assez peu utilisées et ne nous apportent pas d'information que nous sommes en mesure d'exploiter. De ce fait ces colonnes ont été écartées.
- Les données sont triées par ordre chronologique.
- On nettoie les données, on supprime les lignes qui n'ont pas de **ConnID**, celles qui n'ont pas un **LocalTime** complet ainsi que les événements qui ont été identifiés comme non pertinents (**EventCallPartyState** et **EventAttachedDataChanged**) qui sont des événements que nous n'exploitons pas. Enfin on supprime les doublons. Les données brutes sont composées de 69360 appels et 9890976 événements sur les deux mois (pour le SAMU d'Albi). Après traitement des données, on obtient un jeu de données contenant 56918 appels soit 82% des données extraites initialement. Cela nous assure un nombre toujours significatif d'appels qui ne compromet pas la pertinence de notre analyse.
- On complète les données avec le jour de la semaine, cela nous permettra par la suite de faire rapidement un tri pour étudier des jours spécifiques de la semaine.
- On complète les données avec le rôle de la ressource qui effectue l'action. En effet pour chaque événement nous avons l'identifiant de l'agent qui effectue la ressource, on vient compléter cette information en renseignant si cette ressource est un ARM ou un MR, cela nous permet de donner plus de sens métier aux données.
- On transforme les données en log en les regroupant selon leur **ConnID**.



- On crée une nouvelle colonne *Type Appel* dans laquelle on va renseigner si l'appel est un **appel entrant**, un **appel interne** ou un **appel sortant**. Pour ce faire, on effectue un tri sur les appels entrants (uniquement les appels qui ne commencent par « EventRingin »). On affecte à ces derniers le type d'appel entrant. On tri les appels qui commencent par *Event Dialing*, ceux avec un événement *Event Network Reached* sont des appels sortants externes, les autres sont des appels sortants internes.

Ce premier traitement nous permet d'avoir une base de données structurée et exploitable sur laquelle il sera aisé d'appliquer d'autres algorithmes de traitement afin d'en extraire la connaissance utile à notre analyse. Cette étape est essentielle au bon déroulement des suivantes.

### 4.1.3 Sous-étape 1.b : Découvrir le processus métier

Cette étape correspond à la partie 3.2.1. Elle peut être effectuée en parallèle du traitement des données. L'objectif ici est d'obtenir de l'information sur les processus métiers. Des observations et interviews dans les centres d'appels du SAMU ont été réalisées à cette fin.

#### Méthodologie pour les entrevues

La phase d'étude terrain doit permettre de valider et préciser les enjeux métiers d'un point de vue opérationnel. De ce fait, des observations et entrevues ont été réalisées dans les SAMU partenaires : Rodez et Toulouse. Le SAMU de Carcassonne et celui de Nantes ont également fait l'objet d'une visite pour appréhender la diversité des organisations mises en place. Le planning suivant a été réalisé dans les SAMU de Rodez et de Toulouse :

- 8h pour la présentation du projet, des enjeux et du planning d'entretien.
- 8h d'observation et entretien avec un Médecin Régulateur (MR).
- 8h d'observation et entretien avec un Assistant de Régulation médicale (ARM).
- 4h d'observation et entretien avec le Gestionnaire des Moyens (GdM).
- 4h d'entretien avec le responsable du système d'information et de la téléphonie.

On peut rajouter 4h au SAMU de Nantes et 4h à celui de Carcassonne pour comprendre leur fonctionnement. Les observations sont plus succinctes et moins détaillées pour Nantes et Carcassonne. Ces visites ont tout de même permis de constater des modes de fonctionnement différents.

#### Particularité des systèmes d'information

##### Observations Toulouse

Le système de téléphonie du SAMU 31 dispose d'une interface informatisée. De cette manière les ARM peuvent décrocher, raccrocher, transférer les appels depuis l'écran de l'ordinateur. De plus, l'interface permet de visualiser les appels en attente

ainsi que de la disponibilité de chacun des acteurs (ARM et MR). Cela permet de rapidement voir qui est déjà en ligne et le nombre d'appels qu'ils ont dans leur salle d'attente personnelle. Les appels arrivent dans une salle d'attente commune et peuvent être décrochés par tous les ARM. Ces derniers choisissent les appels qu'ils souhaitent décrocher. Lorsqu'ils doivent transférer un appel à un MR, ils le choisissent de manière arbitraire. De la même manière, le MR choisit l'appel qu'il souhaite décrocher sans priorité établie par le système. De plus, il y a une visibilité sur la nature de l'appelant. Si c'est une personne dont le numéro est enregistré dans le système (médecin, ARM, SAMU de la région voisine, équipe pompier, etc.), l'identification apparaîtra en information complémentaire au numéro appelant. De même, si ce numéro a déjà appelé récemment et qu'un dossier médical a été créé, le numéro de dossier s'affiche à l'écran.

### **Observations Rodez**

Le système de téléphonie de Rodez n'est pas interfacé informatiquement pour le décrocher des appels. MR et ARM décrochent les appels « à l'ancienne » avec le combiné. Il n'y a pas de visibilité sur les appels en attente. La seule information disponible est un voyant lumineux qui clignote s'il y a plus d'un appel en attente. Il n'y a pas de visibilité sur les numéros connus. Deux téléphones doivent être gérés, celui qui reçoit les appels externes et celui qui reçoit les appels internes. Les files d'attente sont gérées avec un mode FIFO (First In, First Out), cela signifie que les premiers appels qui entrent dans la file d'attente seront les premiers appels à être décrochés.

### **Observations Carcassonne**

À Carcassonne la téléphonie est mutualisée avec celle des pompiers. Ainsi lorsqu'un appel entrant 15 n'est pas décroché par un des ARM, car ils ne sont pas disponibles, l'appel est décroché par un opérateur pompier, et inversement. Sinon le fonctionnement est le même qu'à Toulouse. À savoir, une salle d'attente commune dans laquelle les ARM viennent décrocher les appels.

### **Observations Nantes**

Le SAMU de Nantes est équipé d'un système de téléphonie similaire à celui de Toulouse avec une interface informatisée. Les appels sont décrochés automatiquement et ce n'est pas l'ARM qui décroche. Donc cela signifie que les appels sont routés automatiquement vers des opérateurs suivant des règles prédéfinies (que nous ne connaissons pas). Ce qui n'est pas le cas à Toulouse ni à Rodez.

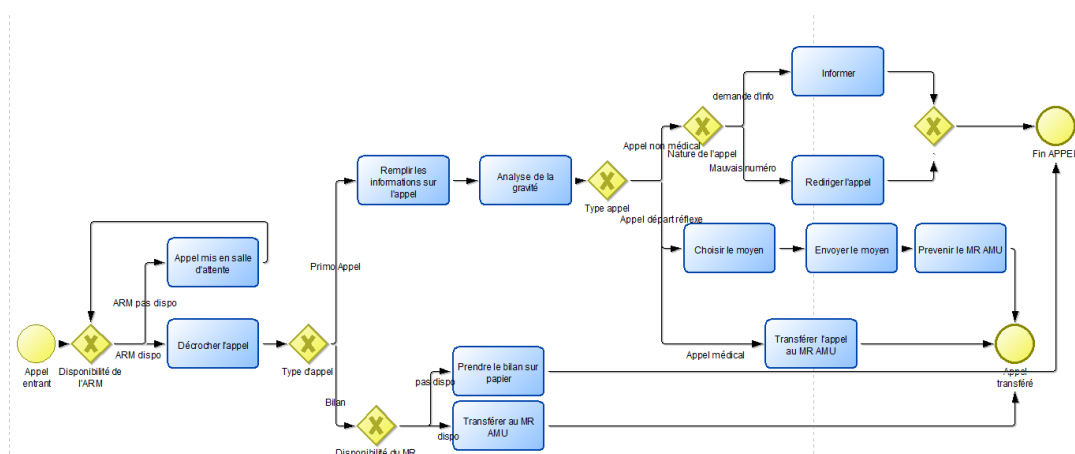
Dès cette première partie, on constate une hétérogénéité au niveau du matériel et de facto au niveau des pratiques de fonctionnement. Des méthodes de décrochés des appels différentes, automatisée à Nantes, en FIFO à Rodez, au choix à Toulouse. Une visibilité sur les appels en attente qui n'existe pas forcément. De plus les fournisseurs de téléphonie ne sont pas les mêmes d'un département à l'autre, et ils travaillent tous sur des logiciels différents.

**Tableau 4.2** – Caractéristiques des systèmes d'information dans les SAMU visités

	Toulouse	Rodez	Nantes	Albi	Carcassonne
Logiciel téléphonie	MediQ V2	CCD Alcatel +	n.d	MediQ V2	Systel
Logiciel régulation	AppliSamu	Centaure 15	AppliSamu	Centaure 15	Systel
Règles files attentes	au choix.	FIFO	n.d	n.d	n.d
Message d'accueil	oui	oui	oui	oui	oui
Décroché automatique	non	non	oui	non	n.d

## Processus de traitement des appels

Dans chacun des centres visités, lorsque cela était possible nous avons observé et modélisé les processus de réception et de traitement de l'appel. L'étude de ces processus a été effectuée dans les SAMU de Rodez et de Toulouse pour les comparer. Si l'on compare les deux processus globaux de Rodez (cf. Figure 4.5) et Toulouse (cf. Figure 4.6), on constate une première différence évidente qui est la présence d'une ressource supplémentaire dans l'organisation à Toulouse. En effet, un ARM est affecté à temps plein sur la gestion des moyens. C'est un ARM qui ne décroche pas d'appel, mais qui gère les réservations d'ambulances, l'envoi des équipes SMUR et le contact avec les pompiers pour envoyer leurs équipes.

**Figure 4.5** – Processus de la prise d'un appel d'urgence à Rodez

Si l'on regarde de plus près les processus d'évaluation de l'appel à Rodez et à Toulouse, il apparaît que les ARM de Toulouse décrochent aussi les appels de la permanence des soins afin de les transférer à un médecin généraliste en régulation, tandis qu'à Rodez une différence est faite entre les ARM de l'aide médicale urgente et ceux de la permanence des soins. Les deux sont gérés de manière indépendante. La manière de récupérer les bilans d'interventions est complètement informatisée à Toulouse, alors que l'utilisation du papier est encore très présente à Rodez. La gestion des appels en attente est différente. En effet, à Toulouse, il y a une visibilité sur les appels en attente. On sait depuis combien de temps ils attendent. Pour chaque appel, on sait aussi s'il est en lien avec un dossier médical déjà ouvert. Les ARM peuvent choisir l'appel qu'ils souhaitent décrocher. Il n'y a pas de critères officiels pour la priorisation du décroché. Intuitivement, les appels qui ont attendu le plus longtemps

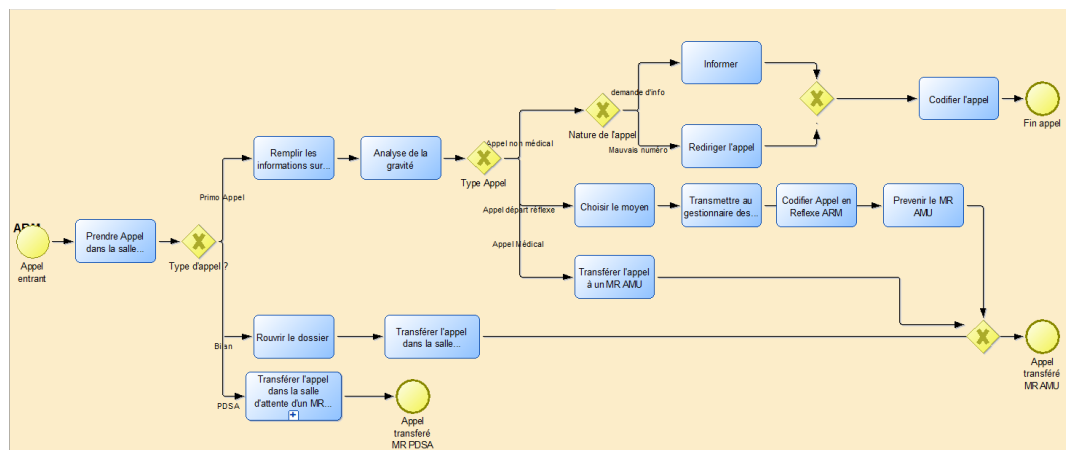


Figure 4.6 – Processus de la prise d'un appel d'urgence à Toulouse

sont décrochés en priorité, ce qui souligne l'intérêt de modéliser d'une manière fine les différents états d'un appel. Cela permet en effet de voir combien de temps un appel peut rester en attente pour définir un indicateur de temps maximum d'attente ou de nombre d'appels pris avant un certain temps. À Rodez, le logiciel de téléphonie ne permet pas la visualisation de l'attente des appels. On ne sait pas combien d'appels sont en attente d'être décrochés ni depuis combien de temps.

Bien que les grandes étapes de la régulation d'un appel soient les mêmes d'un SAMU à un autre, lorsqu'on les étudie individuellement, on constate qu'il existe des différences sensibles. En effet, en fonction des outils informatiques disponibles, la manière de décrocher l'appel ou suivant les spécialisations des ressources, les étapes et modalités associées peuvent varier.

## 4.2 Étape 2 : Interpréter et intégrer

Une fois que l'on dispose à la fois des données de téléphonie nettoyées ainsi que d'une vision précise des processus métier, il faut rassembler et lier ces informations afin d'en extraire des connaissances. Pour ce faire, nous préconisons l'utilisation d'une méthode d'exploration automatique de processus (*Process Mining*).

### Le Process Mining

Le Process Mining consiste à extraire de la connaissance d'un registre d'événements appelé *Event Log*, sauvegardé dans un système d'information. Il se place dans les méthodes *Evidence-base* pour la découverte de processus. Le *Process Mining* se base sur trois activités afin de générer de la connaissance : la découverte du processus, le contrôle de conformité et l'amélioration du processus métier (Van Der Aalst, 2015).

L'histoire du *Process Mining* commence avec les travaux de Agrawal et al. (Agrawal et al., 1998). Ils développent un algorithme afin de générer des graphes de flux à

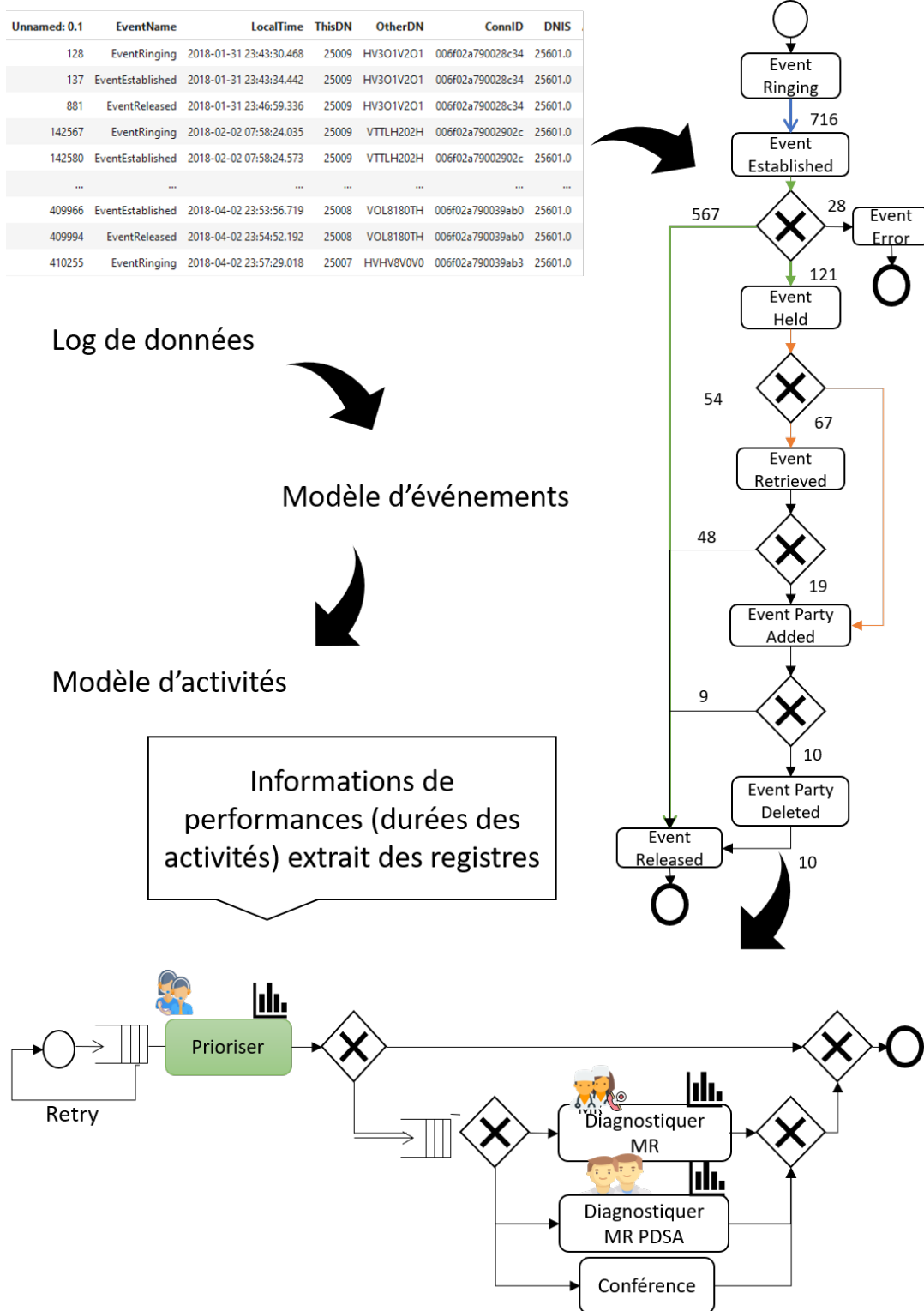


Figure 4.7 – Utilisation du Process Mining pour nos données

partir de l'exécution d'un processus enregistré sous la forme de logs de workflows. Ces travaux ont ensuite été repris et complétés par Weitjers et Van der Aalst qui ont proposé une technique de découverte et de validation de processus capable de gérer les cas particuliers et de valider le processus de flux de travail en mesurant l'écart entre le modèle et l'exécution réelle du processus.

Les processus dans le domaine de la santé sont connus pour être fortement dynamiques, complexes et faisant intervenir de nombreux acteurs de disciplines variées (Homayounfar, 2012). C'est pourquoi le *Process Mining* est de plus en plus utilisé pour analyser les processus en utilisant les données disponibles sur les systèmes d'information des établissements de santé. Cette approche présente l'avantage de pouvoir répondre à plusieurs types de questions tels que :

- Qu'est ce qu'il s'est passé ? : Permettre d'identifier le processus et ses activités. (par exemple, quel est le traitement typique d'un appel ?)
- Pourquoi est-ce que cela est arrivé ? : Comprendre les enchaînements d'activités et leurs conséquences (par exemple, pourquoi est-ce qu'un appel a attendu avant d'être décroché ?)
- Que va-t-il se passer ? : Identifier les circonstances dans lesquelles une activité arrive dans le but de la prévoir (par exemple, est-ce que cet appel va attendre avant d'être décroché ?)

Le *Process Mining* a été utilisé dans différents domaines au sein des systèmes de santé. C'est en oncologie et en chirurgie qu'on retrouve le plus grand nombre d'études à ce propos. Dans le domaine de l'urgence, Rojas et al. identifient 3 études seulement dans leur revue de littérature (Rojas et al., 2016).

## 4.2.1 Sous-étape 2.a : Fusionner les modèles

Les données de téléphonie nous procurent des informations sur les heures d'arrivée de chaque appel et les différents événements qui en découlent. Toutefois, nous n'avons pas directement l'information sur la durée de chaque activité. C'est pourquoi il faut établir une correspondance entre les suites d'événements et les activités qu'elles représentent.

Quelques exemples d'appels et d'enchaînements d'événements sont présentés dans les figures 4.8 et 4.9 pour chaque type de trace nous avons identifié le nombre de fois où elle est retrouvée dans le log de données ainsi que les enchaînements de temps d'attente ou de communication. On constate qu'il y a des briques constitutives qui se dessinent, des enchaînements qui se répètent et qui correspondent à la même chose.

- Une première brique est l'enchaînement *Event Ringing*  $\implies$  *Event Established* qui correspond à un appel qui contacte le centre d'appels et qui réussit à le joindre, le temps qui s'écoule entre les deux événements est un temps d'attente.
- Une deuxième est l'enchaînement *Event Ringing*  $\implies$  *Event Released* qui traduit une tentative d'appel non concluante. Soit l'appelant ne rappelle pas, et cela s'arrête là, soit l'appelant essaye de rappeler et on aura de nouveau un bloc *Event Ringing*  $\implies$  *Event Released*, dans le cas d'un échec, soit un bloc *Event Ringing*  $\implies$  *Event Established* dans un cas de réussite.

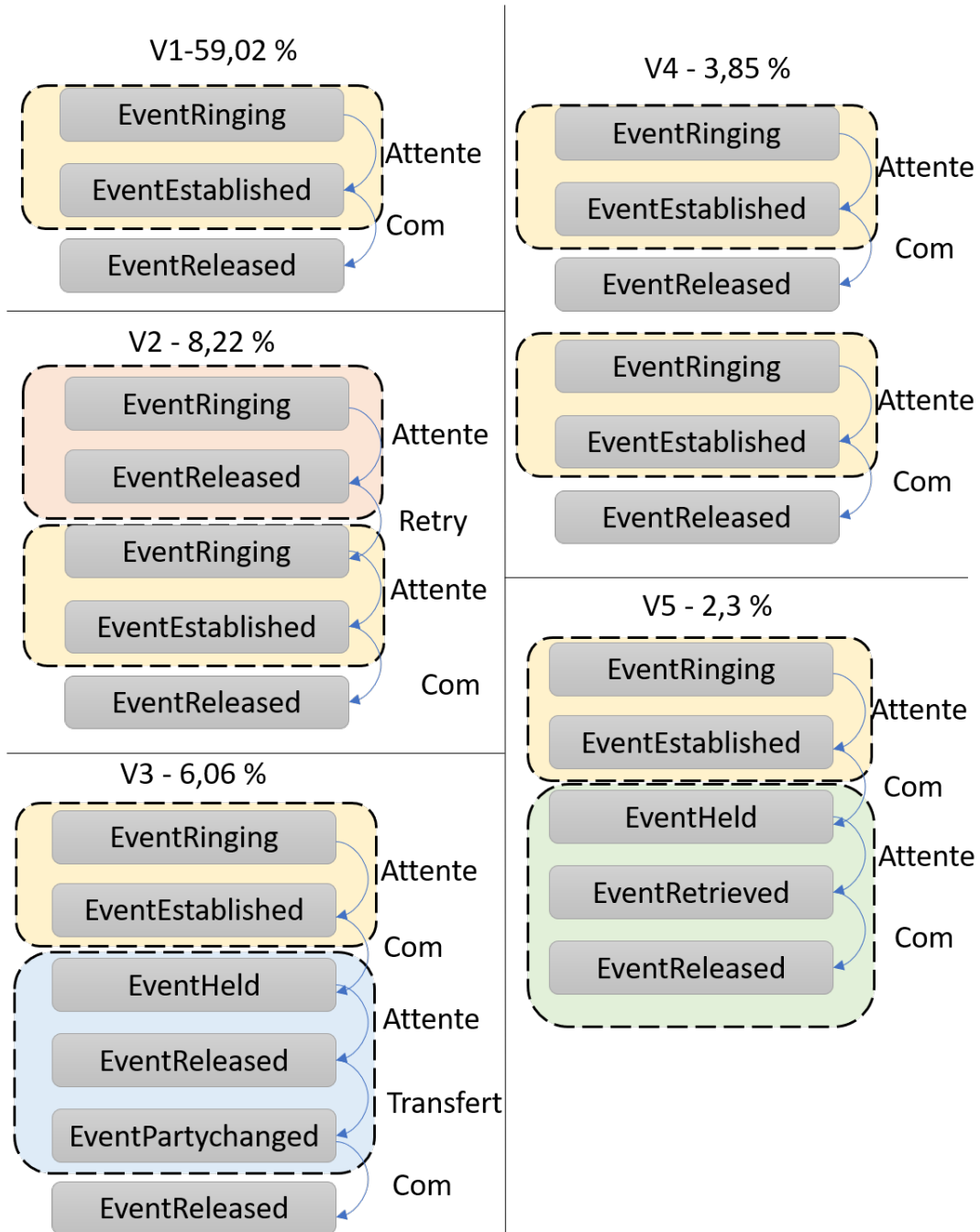


Figure 4.8 – Quelques exemples de traces (Partie 1)

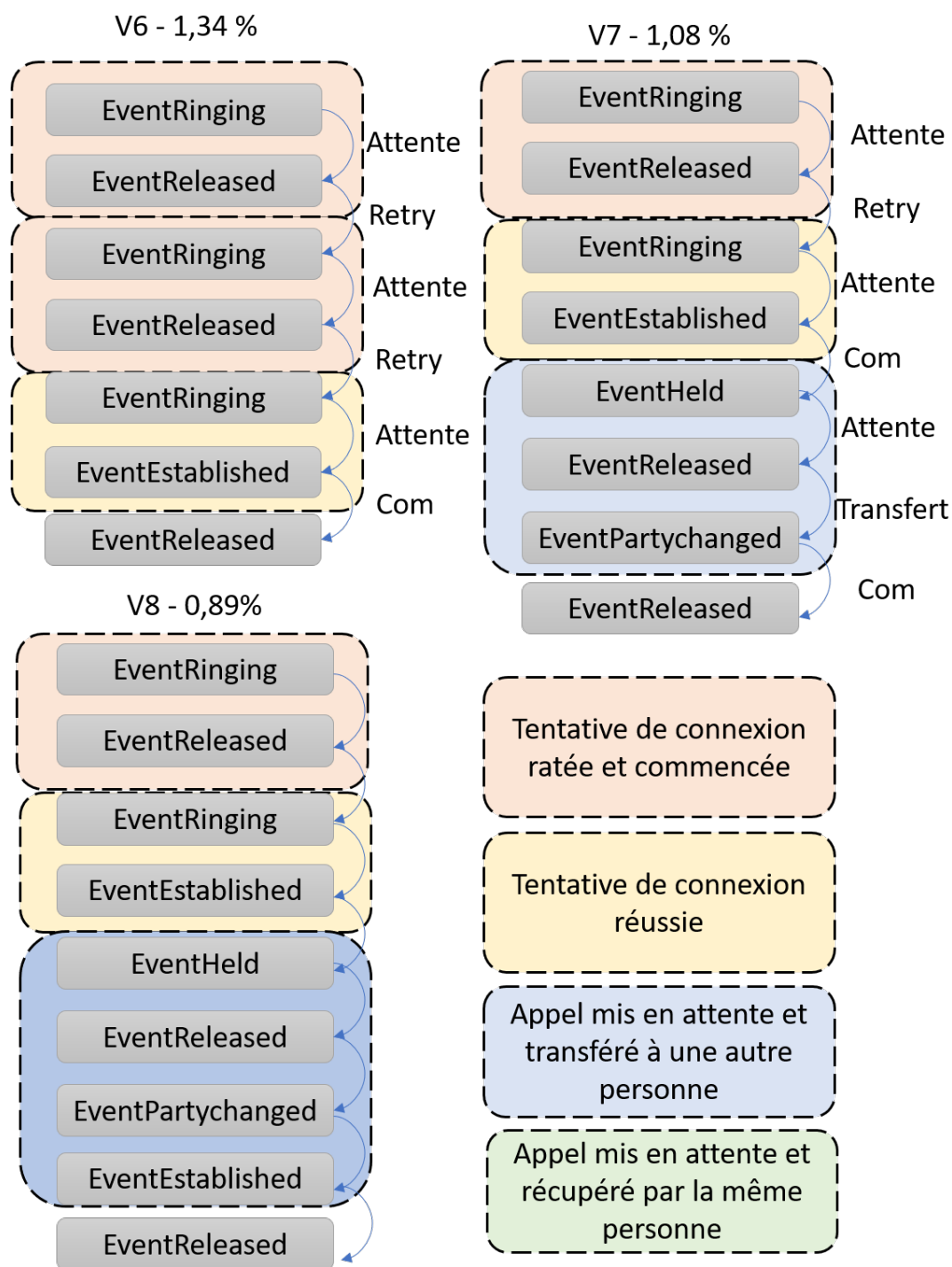


Figure 4.9 – Quelques exemples de traces (Partie 2)



- Lorsque les appels sont mis en attente il y a plusieurs cas possibles, soit l'appel est mis en attente et repris par la même personne plus tard alors on aura un comme dans le cas V5, *Event Held*  $\implies$  *Event Retrieved*  $\implies$  *Event Released*. Soit l'appel est transféré à une autre personne et on aura plutôt comme dans le V3, *Event Held*  $\implies$  *Event Released*  $\implies$  *Event Party Changed*, ce qui correspond à une mise en attente de l'appel, *EventReleased* suivi de *EventPartyChanged* implique que le premier interlocuteur raccroche, et que l'interlocuteur change. On notera que le fait de transférer l'appel à un interlocuteur est plus courant (6,06% des appels) que le fait de mettre en attente et de le reprendre plus tard (2,3% des appels).

Ces briques constitutives nous permettent de former des traces différentes, en effet il peut y avoir plusieurs tentatives de connexion qui échouent avant de réussir à joindre le centre d'appels, l'appel peut être mis en attente plusieurs fois. Ou encore, comme dans la trace V4 de la figure 4.8, un appel qui va être pris en charge une première fois par une ressource puis qui va rappeler et être pris en charge une deuxième fois.

Ces événements sont très précieux, car ils vont nous permettre d'identifier les différentes activités de nos processus métier. En fonction de l'enchaînement d'événements et du type de ressource qui est impliquée, ce ne sera pas la même activité. Par exemple, l'enchaînement d'un *Event Established* et d'un *Event Released* peut correspondre à un temps de communication avec l'ARM ou avec le MR, cela va dépendre du type de ressource impliquée. Si on a un enchaînement *Event Retrieved* et *Event Released* réalisé par un MR, alors c'est une activité de régulation par un MR après que l'appel ait été mis en attente. Ainsi, nous devons définir des règles métiers pour être capable de transformer nos enchaînements d'événements en activités qui ont du sens d'un point de vue métier.

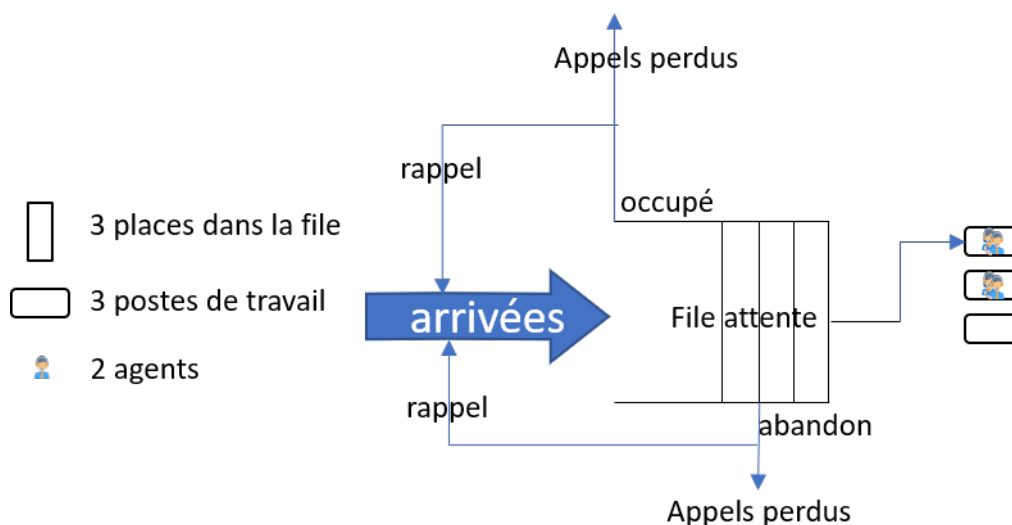
## 4.2.2 Sous-étape 2.b : Extraire les connaissances

### Les Centres d'appels peuvent être vus comme un ensemble de files d'attente.

Un centre d'appels va être constitué de  $k$  lignes de téléphones qui permettent à un appel de joindre le centre. Ces lignes vont être gérées par  $w \leq k$  postes de travail sur lesquels vont être assignés  $m \leq k$  agents (cf. figure 4.10). Lorsqu'un appel arrive, trois scénarios sont possibles :

1. L'appel est distribué directement si les agents sont disponibles.
2. L'appel est placé dans une file d'attente si les agents ne sont pas disponibles.
3. L'appel est bloqué s'il n'y a pas de lignes disponibles, toutes les lignes sont déjà occupées par un appel, la file d'attente ne peut pas accueillir plus d'appels.

Dans un tel système, les ressources sont immédiatement libérées dès qu'un appel a été servi. Il existe différents types d'arrivées (ou d'appelants), à savoir ceux qui établissent un premier contact, ceux qui recomposent en raison d'une tonalité d'occupation et ceux qui abandonnent la file d'attente en attendant. Pour caractériser ce système, la



**Figure 4.10** – Représentation d'un centre d'appels par file d'attente. (Gans et al., 2003)

notation de Kendall a été inventée (Kendall, 1953). Celle-ci décrit la file d'attente. La notation de Kendall est aujourd'hui généralement exprimée par les termes suivants 1/2/3/4/5/6, où chaque numéro représente :

1. Le processus d'arrivée des appels, c'est-à-dire la manière dont ils se répartissent dans le temps sur un créneau donné (généralement suivant une loi exponentielle, exprimé par un M).
2. La répartition du temps de service, la loi qui régit le temps de réponse de l'appel (généralement suivant une loi exponentielle, exprimé par un M).
3. Le nombre de serveurs, qui correspond au nombre de ressources disponibles pour décrocher.
4. Le nombre de troncs<sup>2</sup> (infini si l'on ne tient pas compte des troncs).
5. La taille de la population appelante (infinie si on l'ignore).
6. La modalité de gestion des files d'attente, généralement premier entré, premier sorti (FIFO).

La file d'attente la plus simple est la file M/M/1, qui est composée de l'arrivée des appels suivant une loi de Poisson, des temps de service suivant une distribution exponentielle aléatoire, et un serveur (soit une seule ressource pour décrocher). Cependant, un tel modèle est le plus souvent une simplification excessive, comme le montre le nombre des reprises possibles dans la figure 4.10 (Gans et al., 2003).

L'arrivée des appels est un grand vecteur d'aléatoire. Le processus d'arrivée des appels est généralement modélisé par une loi de Poisson (Saaty, 1961). La loi de Poisson est une loi de probabilité qui s'applique aux événements rares et indépendants les uns des autres. Ce qui est le cas pour les appels d'urgence. Si une personne passe un appel d'urgence, cela est généralement fait indépendamment de si son voisin vient d'appeler les urgences. Des cas existent néanmoins comme les attentats.

2. Cela correspond à la taille de la file d'attente et le nombre d'appels maximal qui peut être en simultané dans le centre d'appels.

## Modéliser les temps de communication

Le temps de communication ou temps de service a été défini par Mandelbaum et Zeltyn (Zeltyn et al., 2005) Comme étant « le temps qu'un agent passe à gérer un appel ». Dans les faits, cela inclut le temps de communication passé avec l'appelant, mais aussi le temps de gestion pouvant être occasionné après l'appel. Ce temps de gestion peut correspondre à la complétion du dossier du patient une fois l'appel terminé. Dans la plupart des cas, ce temps de traitement moyen est une distribution exponentielle. Ce choix de modélisation présente certaines caractéristiques qui permettent de simplifier les calculs dans certains modèles de files d'attente.

Un problème commun dans les centres d'appels est d'anticiper le nombre d'agents qu'il faut planifier à chaque moment de la journée. Ce nombre d'agents doit permettre de satisfaire une attente minimale pour l'appelant. Ces décisions découlent directement du nombre d'appels et de la manière dont ils sont repartis durant la période à planifier, donc de la prévision du nombre d'appels. Une approche assez commune pour déterminer le nombre de ressources nécessaires pour atteindre une qualité de service cible est la formule d'Erlang. (Takacs, 1969)

Créé en 1917 par Agner Erlang, la formule d'*Erlang-C* est utilisée pour déterminer le nombre d'opérateurs nécessaire pour atteindre une certaine qualité de service dans les centres d'appels. Ses avantages découlent du fait qu'elle est facile à utiliser et à coder dans un logiciel ou même dans une feuille de calcul. Elle est très utile, car très peu gourmande en termes de ressources pour le calcul des plannings. Ce qui à l'époque n'était pas négligeable.

Toutefois cette formule fonctionne sous quelques hypothèses restrictives : tous les appels sont considérés comme décrochés, ainsi on ne prend pas en compte les appels abandonnés. On considère aussi que tous les agents peuvent décrocher tous les appels, donc on ne prend pas en compte les compétences de chacun.

Pour calculer la probabilité qu'un appel soit mis en attente la formule d'Erlang C s'appuie sur plusieurs données :

- Le nombre d'agents  $N$ .
- Le temps de service moyen .
- Le nombre d'appels par intervalle de temps.
- La charge  $A$  est définie comme le temps qu'il faudra pour traiter tous les appels sur l'intervalle donné. Mettons que sur un créneau de 30 minutes on reçoive 200 appels avec un temps moyen de prise en charge de 3 minutes, alors la charge vaudra 600 appels minute. La charge s'exprime habituelle en Erlang. Un Erlang correspond au temps nécessaire pour gérer les appels durant une heure, dans notre exemple, la charge vaut alors 10 Erlangs (600 *appels minutes*/60 *minutes*).

Ce qui nous permet de calculer la formule d'Erlang-C pour obtenir la probabilité qu'un appel entrant ne soit pas décroché immédiatement, car tous les agents sont déjà en ligne :

$$X(N, A) = \frac{\frac{A^N}{N!} \frac{N}{N-A}}{\frac{A^N}{N!} \frac{N}{N-A} + \sum_{m=0}^{N-1} \frac{A^m}{m!} + \frac{A^N}{N!}} \quad (4.1)$$

Une fois la probabilité pour qu'un appel ne soit pas décroché calculée, on peut l'utiliser pour calculer la qualité de service estimée dans le centre appel :

$$QualitédeService = 1 - [X(N, A) * e^{-[(N-A)*(\frac{QoS_{cible}}{AHT})}] \quad (4.2)$$

Si la qualité de service n'atteint pas la cible alors il faut rajouter une ressource et refaire le calcul pour voir si l'ajoute de cette ressource permet d'atteindre l'objectif de qualité de service. De cette méthode, il est possible de calculer le nombre d'agents nécessaires sur chaque intervalle  $j$  durant une journée  $i$  pour chaque créneau. Pour cela il faut avoir la charge  $A_{i,j}$  et le temps moyen de communication  $S_{i,j}$  sur chacun des créneaux pour faire le calcul.

Il s'agit d'une approche simplifiée qui prédit  $N$  dans une file d'attente  $M/M/N$ . Par conséquent, comme soulignée précédemment, la formule ne tient pas compte des appels abandonnés avant d'être décrochés ou encore des appels qui sont mis en attente durant un certain temps. Dans un tel cas, le processus d'arrivée est supposé être de Poisson avec des temps d'arrivée indépendants distribués de manière exponentielle négative. Avoir des temps d'arrivée indépendants signifie que chaque appel est indépendant du précédent est distribué aléatoirement sur le créneau défini. Les temps de service sont également supposés être distribués de manière exponentielle, et les agents  $N$  sont tous statistiquement identiques. Cela signifie que l'on considère que tous les ARM sont identiques et ont les mêmes performances. Outre la formule Erlang-C, Erlang a également produit la formule de la probabilité qu'un appel soit bloqué dans une file d'attente  $M/M/N/b$ , avec un nombre  $b$  de places dans la file d'attente comme la formule Erlang-B (Brockmeyer et al., 1948).

Ces modèles d'Erlang reposent sur des hypothèses fortes qui sont largement contestées. L'hypothèse de la distribution exponentielle notamment a été remise en question ; par exemple, certaines études ont montré que la distribution log-normale permet une bien meilleure approximation des temps de session (Brown et al., 2005 ; Bolotin, 2013).

Afin de modéliser notre centre d'appels, nous avons donc besoin d'identifier et de définir le modèle de file d'attente que nous souhaitons utiliser. C'est pourquoi dans les sections suivantes nous détaillerons nos choix de modélisation pour l'arrivée des appels, la distribution des temps de communication et la manière dont on paramètre la file d'attente.

## Générer la distribution des temps de session des appels

L'objectif ici est d'avoir un modèle qui se rapproche le plus possible du comportement réel du SAMU. Comme nous disposons de 26000 appels pour établir les distributions réelles, c'est une quantité qui est statistiquement significative et qui nous permet de nous baser sur ces données pour notre modèle.

Les statistiques descriptives sur les durées d'activités sont des données importantes comme les durées minimums, moyennes, maximum, mais aussi les occurrences de chaque activité. Il est possible de calculer ces données grâce au *Process Mining*. En effet, une fois les activités identifiées, il est possible d'analyser la durée de chacune d'elle. De ce fait, que ce soit pour le temps d'attente des appels ou encore pour le temps de session des communications avec l'ARM, le MR ou le MR PDSA, il est possible de faire des statistiques sur la durée de ces appels et d'évaluer la performance actuelle des centres d'appels.

Toutefois, nous avons quand même voulu identifier la distribution théorique la plus proche de nos données réelles. Pour ce faire nous avons calculé les temps de communication de l'ARM et du MR grâce au *Process Mining*. Une fois les durées récupérées, nous construisons un histogramme pour chaque activité de communication du processus. Puis nous appliquons un test d'adéquation (*Goodness of Fit test*) (D'Agostino, 1986). C'est un algorithme de correspondance avec lequel on applique des distributions théoriques sur notre modèle réel. On cherche le meilleur jeu de paramètres pour minimiser la distance entre les données réelles et la courbe théorique.

Bien que cette procédure d'identification soit assez simple, elle soulève plusieurs questions :

- Comment choisit-on le nombre de classes de notre histogramme ? En effet, le nombre de classes aura un impact sur la forme de notre histogramme et sur le résultat de l'algorithme de correspondance.
- Comment mesure-t-on la distance entre le modèle théorique et nos données réelles pour évaluer « la meilleure distribution théorique » ?

Pour le choix du nombre de classes (et in fine de la largeur de celles-ci), il existe plusieurs règles dans la littérature scientifique. Dogan et al. détaillent les différentes méthodes et leurs usages dans leur revue de littérature sur le sujet (Dogan et al., 2010). Dans notre cas, comme nos données ne sont pas symétriquement et normalement distribuées, nous ne pouvons pas utiliser la règle de Sturge (Sturges, 1926) qui est la plus commune. Ainsi nous allons utiliser la règle de Rice<sup>3</sup> qui consiste à prendre un nombre de classes égal à deux fois la racine cubique du nombre d'observations (cf. équation 4.3).

$$k = 2\sqrt[3]{n}, \text{ où } k \text{ est le nombre de classe, } n \text{ est le nombre d'observation} \quad (4.3)$$

Pour la distance entre le modèle théorique et la distribution réelle, nous avons utilisé la SSE (*Sum Squared Error*) soit la somme des erreurs au carré. Plus celle-ci est proche de zéro et plus la distribution théorique correspond à la distribution empirique. La SSE permet d'avoir une comparaison entre les distributions théoriques. Ainsi cela permet d'identifier la distribution qui est la moins mauvaise par rapport à d'autres, mais pas nécessairement la meilleure dans l'absolu.

---

3. Online Statistics Education : A Multimedia Course of Study (<http://onlinestatbook.com/>). Project Leader : David M. Lane, Rice University

La figure 4.11 présente la distribution réelle des appels (histogramme, en bleu) ainsi que les distributions théoriques log-normale et exponentielles dont les SSE sont respectivement  $1,47e-05$  et  $2,18e-05$  pour la communication des ARM au SAMU d'Albi. Le tableau 4.3 apporte des informations complémentaires sur les statistiques descriptives de ces données. Ainsi la distribution log-normale correspond bien mieux à nos données réelles que la distribution exponentielle ce qui corrobore les résultats de Brown et al. (L. Brown et al., 2005). De ce fait notre jeu de données ne valide pas les hypothèses de la formule d'Erlang concernant la distribution exponentielle de la durée des appels. Le choix pour modéliser la durée des appels va être important surtout dans le chapitre suivant dédié au double numérique pour que notre modèle soit le plus similaire possible à la réalité.

En conclusion, nous avons décidé de travailler avec la distribution réelle des appels que ce soit pour les communications avec les ARM ou avec les MR. Toutefois, des recherches supplémentaires pourraient être effectuées pour identifier les lois de distribution théoriques les plus appropriées pour les différentes étapes de communication d'un appel d'urgence, bien que la distribution log-normale semble déjà apporter un niveau de précision satisfaisant.

### Profil d'arrivée des appels : Est-ce que l'arrivée des appels au SAMU est prévisible ?

Une des caractéristiques de l'urgence est qu'elle n'est pas planifiée. C'est pourquoi, on lui associe souvent un fort caractère stochastique, aléatoire et dépendant de la saisonnalité. Nous avons voulu vérifier ces hypothèses et voir si elles étaient valables pour nos cas d'études. Nous avons exploré les données de Rodez et Albi sur 8 semaines complètes (février et mars). La figure C.1 correspond au graphique de ce résultat pour Albi (le même graphique est disponible en annexe pour Rodez). Plusieurs constats peuvent être faits :

- On constate une différence en termes de volumétrie entre les jours de semaine et les fins de semaine. En effet, les samedi et dimanche se caractérisent par une augmentation du volume d'appels pouvant faire doubler, voir tripler le nombre d'appels entrants. Cette augmentation peut être due en partie à l'ouverture des lignes de la PDSA.

**Tableau 4.3** – Informations sur la durée de communication des ARM

<i>SAMU</i>	Albi
<i>Période</i>	Février, Mars 2018
<i>Nbre appels</i>	26084
<i>Durée moyenne appel</i>	77 secondes
<i>Durée médiane</i>	67 secondes
<i>Déviat ion standard</i>	69.66 secondes
<i>Durée maximale</i>	1215 secondes

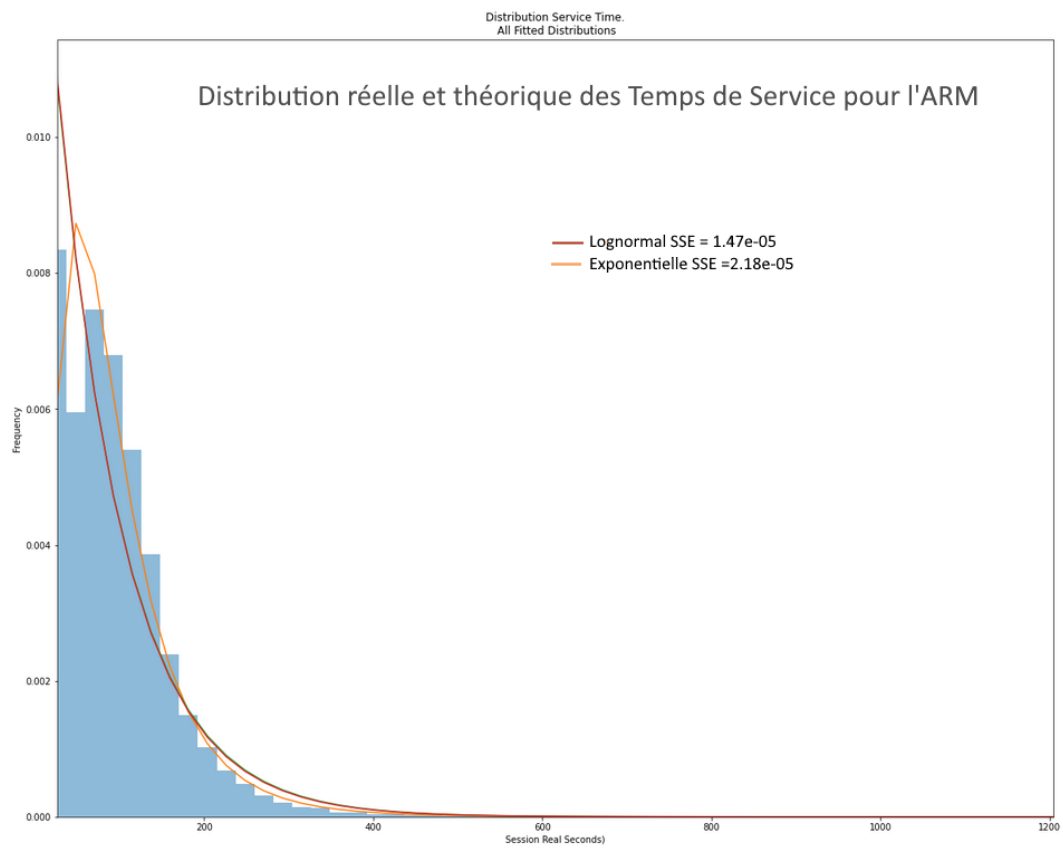


Figure 4.11 – Distribution théorique et empirique des temps de session pour l'ARM

- Le samedi et le dimanche n'ont pas tout à fait le même profil. On remarque que la journée du samedi présente un seul pic d'appels en milieu de journée, là où le dimanche présente deux pics d'appel un en début de matinée et un en fin d'après-midi.
- Au niveau des jours de la semaine, on constate des comportements similaires d'une semaine à l'autre. On retrouve deux pics d'appels journaliers en début de matinée et en fin de journée.
- Au sein d'une même journée on constate deux pics d'appels, un dans la matinée et un durant l'après-midi.

Si on regarde la provenance des appels par créneau horaire et par jour de la semaine sur ces mois là, on voit bien que l'augmentation du week-end est en grande partie due à l'ajout des appels de la PDSA.

Bien qu'il puisse y avoir une variabilité d'une semaine à l'autre, on retrouve néanmoins un profil similaire de l'arrivée des appels à l'échelle de la semaine. Les appels, bien que non programmés donc non prévisibles, se répartissent sur des horaires journaliers et hebdomadaires de façon assez récurrente avec peu de variabilité. Donc, aussi surprenant que cela puisse paraître, on peut prévoir avec une bonne fiabilité le nombre d'appels d'urgence sur chaque créneau horaire, chaque jour, hors considération des événements majeurs de type attentat ou pandémie par exemple. De ce fait, nous nous baserons sur les données historiques pour générer le profil des appels dans le cadre de nos simulations.

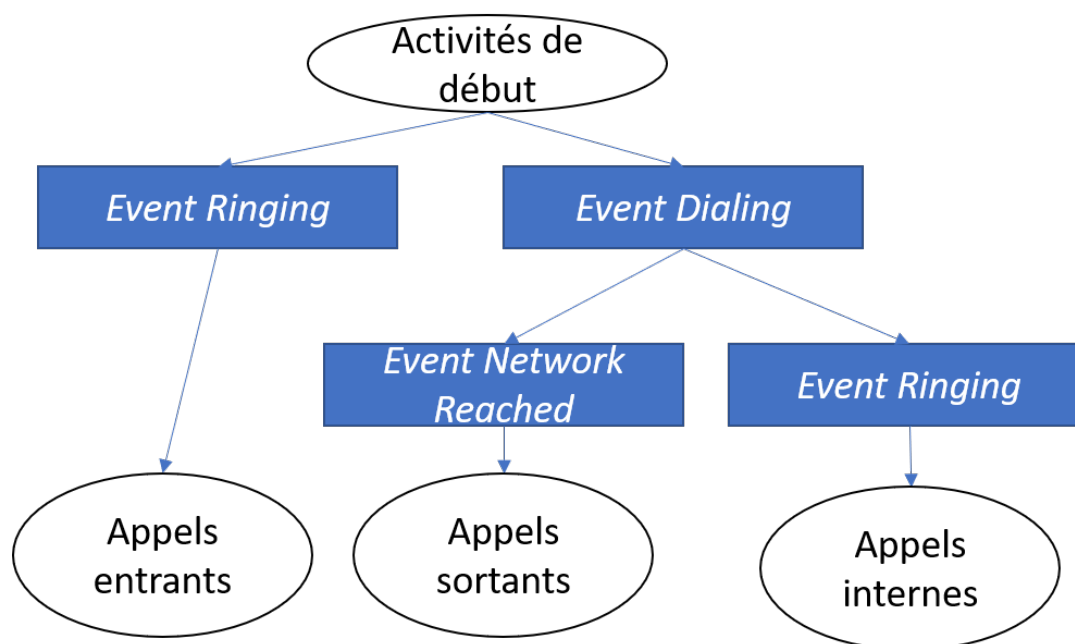
Le dimanche est une journée particulièrement critique en termes de volumétrie sur laquelle il n'y a pas forcément plus de ressources engagées pour le traitement des appels. De ce fait, nous allons cibler notre étude sur cette journée. En effet, si un centre d'appels est bien dimensionné pour tenir la charge du jour le plus critique, on peut faire l'hypothèse qu'il devrait être capable de gérer la charge des autres jours qui est moins importante.





## Analyse des flux : Appels entrants et sortants

Les centres d'appels d'urgence ont la particularité d'être des centres d'appels mixtes, cela signifie qu'ils reçoivent des appels, mais qu'ils en émettent aussi vers l'extérieur. Toutefois la gestion des appels sortants est bien souvent négligée dans la littérature scientifique (Aboueljiane et al., 2014; M. v. Buuren et al., 2017). Comme nous avons la chance de disposer des données des centres d'appels à l'échelle de l'appel, il est possible de savoir la différence entre les appels sortants et entrants en étudiant les événements du registre de données.



**Figure 4.13** – Identifier les types d'appels par leur événement initial

L'analyse du premier événement dans la gestion d'un appel nous donne une information sur le type d'appel (cf. figure 4.13). Les traces des appels ont globalement deux types d'événements de début : *Event Ringing* ou *Event Dialing*. Une trace commençant par le premier correspondra à un appel entrant, car cet événement indique qu'un téléphone est en train de sonner. *Event Dialing* nous indique qu'un appel est en train d'être composé depuis un poste interne. Cet appel peut être interne ou sortant. Un appel interne correspond à une communication entre les acteurs du centre d'appels. Par exemple, un MR qui appelle un ARM. Un appel externe implique un appel issu du centre d'appels vers l'extérieur, on le reconnaît grâce à l'événement *EventNetworkReached* qui nous indique qu'un réseau distant a été contacté. C'est par exemple le cas lorsqu'un MR contacte un hôpital pour savoir s'il y a de la place dans un service particulier.

Si on regarde une journée de dimanche et que l'on compare le nombre d'appels sortants et le nombre d'appels entrants. On constate qu'il y a en moyenne quasiment autant d'appels sortants que d'appels entrants, car ils représentent respectivement 51% et 49% du total des appels (voir tableau 4.4). Ainsi, en termes de volumétrie, il ne semble pas acceptable de faire l'impasse sur les appels sortants.

**Tableau 4.4** – Part des appels sortants et des appels entrants dans la volumétrie totale sur 8 semaines

	S 6	S 7	S 8	S 9	S 10	S 11	S 12	S 13	Moyenne
Appels entrants	46%	47%	49%	50%	48%	54%	47%	49%	<b>49%</b>
Appels sortants	54%	53%	51%	50%	52%	46%	53%	51%	<b>51%</b>

Il y a donc un nombre non négligeable d'appels qui sont émis par le centre d'appels. Il est intéressant de se demander quel est le temps passé sur ceux-ci par rapport au traitement des appels entrants. Par conséquent, nous avons regardé le temps total cumulé passé en ligne avec des appels entrants et sortant durant la journée. Les résultats sont contenus dans le tableau 4.5, on constate qu'en moyenne sur une journée il y a souvent deux fois plus de temps passé sur les appels sortants que sur les appels entrants.

Cela confirme notre intuition que les appels sortants ne peuvent pas être écartés de l'étude. La prise en compte de ces appels est une originalité nécessaire qui vient combler un manque vis-à-vis de l'état de l'art.

**Tableau 4.5** – Sur quels appels les ARM et les MR passent-ils le plus de temps ?

	S 6	S 7	S 8	S 9	S 10	S 11	S 12	S 13	Moyenne
Appels entrants	36%	35%	38%	38%	38%	34%	28%	19%	<b>33%</b>
Appels sortants	28%	65%	62%	62%	62%	66%	72%	81%	<b>62%</b>

## 4.3 Synthèse

Dans ce chapitre, nous avons appliqué la première partie de la méthodologie développée dans le chapitre 3. La principale contribution de ce chapitre consiste donc à proposer une analyse détaillée des données de téléphonie des centres d'appels d'urgence du SAMU. Nous avons pu remarquer que les profils d'arrivées des appels présentaient une saisonnalité hebdomadaire avec une augmentation du nombre d'appels durant le week-end, mais aussi une saisonnalité journalière avec deux pics d'appels dans la journée. Nous avons développé plusieurs algorithmes pour l'étude des données afin de permettre l'évaluation des temps d'attentes et des temps de sessions.

En étudiant les temps de sessions des différents types d'appels, nous avons découvert que les appels sortants occupaient les ressources pendant 62% de leur temps en moyenne. La prise en compte des appels sortants dans cette analyse constitue une véritable originalité par rapport aux travaux menés jusqu'à présent sur le sujet.

Néanmoins, afin d'aller plus loin, il serait nécessaire d'avoir une quantité de données plus importante afin de mener des expérimentations supplémentaires concernant les distributions théoriques des temps de session. En effet, il serait intéressant d'étudier

l'effet des différentes lois pour voir laquelle se rapproche le plus du comportement réel du centre d'appels. Le tableau 4.6 propose une synthèse des contributions et des limitations de ce chapitre.

**Tableau 4.6** – Contributions et Limites : Chapitre 4

	<b>Chapitre 4</b>
<b>Contributions</b>	Une analyse des données de téléphonie des centres d'appels d'urgence.
	L'inclusion des appels sortants dans notre étude.
	Proposition d'algorithmes pour l'étude des données de centres d'appels d'urgence.
<b>Limitations</b>	Un manque de données ne nous permet pas d'obtenir des distributions théoriques de temps de session.