Roadmap architecturale pour la création d'un système de diffusion d'alarme

Sumit Kalra, Prabhakar T.V., Saurabh Srivastava

Computer Science Department, IIT Kanpur Kanpur, 208016, India
sumitk@iitk.ac.in, tvp@iitk.ac.in, ssri@iitk.ac.in

RÉSUMÉ. *Les Systèmes de Diffusion d'Alarme (SDA) ont des spécifications complexes, exigeantes et critiques. Dans ce travail, nous visons à fournir une perspective d'architecture logicielle pour les SDA. Nous examinons à la fois les exigences fonctionnelles et de qualité pour un SDA et tentons d'identifier certains attributs de qualité spécifiques à un SDA ainsi qu'un ensemble de stratégies architecturales pour les réaliser. Nous proposons également une architecture de référence pour la conception de ces systèmes. Nous présentons de nombreux exemples pour justifier nos conclusions et les approfondir sur une étude de cas du système de prévention des collisions liées au trafic aérien.*

ABSTRACT. *Alarm Diffusion Systems (ADS) have complex, exacting and critical requirements. In this work we aim to provide a software architecture perspective towards ADS. We look at both functional and quality requirements for an ADS and also attempt to identify certain quality attributes specific to an ADS and attempted to provide a set of architectural tactics to realise them. We also propose a Reference Architecture for designing such systems. We have provided ample examples to support our inferences and take a deeper look at a case study of the Traffic Collision Avoidance System (TCAS) in aircrafts.*

MOTS-CLÉS : *architecture, système de diffusion d'alarmes.*

KEYWORDS: *architecture, alarms diffusion system.*

DOI:10.3166/ISI.21.4.11-25

1. Introduction

Les alarmes sont des processus mis en œuvre de manière assez courante dans la plupart des systèmes actuels. Elles peuvent se matérialiser de manière aussi simple qu'un avertissement via une LED sur un tableau de bord de voiture, ou aussi subtile que l'indication sur la vitesse d'un avion dangereusement basse. Les blocs de construction de tout système d'alarme peuvent, typiquement, inclure la détection/mesure de paramètres pertinents, l'application des règles/inférences pour détecter une alarme, la diffusion d'une alarme parmi les destinataires, le captage de tout retour des participants, si nécessaire, l'apprentissage/adaptation du système à partir des statistiques recueillies, etc. En particulier, tous les Systèmes de Diffusion d'Alarme (SDA) doivent diffuser les alarmes parmi un ensemble de destinataires. Ceux-ci peuvent être des humains ou des machines. Les scénarios pour un SDA peuvent donc être variables, en fonction du type de bénéficiaires, des canaux de communication, des niveaux de confidentialité appliqués, de la gravité des alarmes, etc.

Dans la plupart des cas, la fonctionnalité de diffusion du système est gérée *via* un logiciel spécialisé. Ces logiciels sont conçus afin de prendre en charge les exigences et les contraintes mentionnées ci-dessus. Un aspect essentiel de la conception de tout logiciel est d'articuler les exigences, à la fois explicites et implicites, et de les faire correspondre avec des composants logiciels qui forment l'ensemble du système. Un architecte de système exécute cette étape cruciale dans le processus appelé Architecture Système. Dans cet article, nous présentons en détail cette étape de développement de processus. Nous discutons des attributs de qualité (ISQS ISO, 2011) applicables à un SDA, et d'un ensemble de stratégies architecturales, et fournissons des conseils pour leur réalisation. Nous allons énumérer des scénarios plausibles pour des exigences spécifiques, à l'aide d'exemples.

Le papier est organisé comme suit. Dans la section 2, nous discutons les exigences associées à un SDA. En particulier, nous examinons les différents scénarios que le système peut avoir à manipuler. Nous définissons aussi quelques attributs de qualité qui peuvent être associés à un SDA ainsi que des stratégies architecturales communes (Bachmann *et al.*, 2003) qui peuvent être utilisées pour les réaliser. Dans la section 3, nous essayons de fournir une architecture de référence pour un SDA tout en mettant en évidence les aspects génériques du système à l'aide d'une vue « Component-and-Connector » (Len Bass, 2007 ; Taylor *et al.*, 2009). Nous présentons ensuite dans la section 4 une brève étude de cas sur un TACS, un système d'évitement de collisions pour les avions. L'objectif est d'analyser un TCAS du point de vue de la diffusion d'alarmes. Enfin, nous présentons notre conclusion à la section 5.

2. Analyse des besoins pour un SDA

La conception de chaque système est initiée par le recensement détaillé des objectifs qu'il est censé atteindre (Sommerville, Sawyer, 1997). Cela inclut les aspects fonctionnels du système comme les structures de données, algorithmes, opérations, politiques, communications, etc. L'autre ensemble d'exigences associées à la qualité d'un système implique d'en mesurer les paramètres, et d'adhérer à certaines attentes en matière de qualité, par exemple des limites supérieures sur les temps de réponse, une exigence pour que le code soit facilement maintenable, des options pour faciliter l'extension du système à l'avenir, etc. Bien que les exigences associées à des SDA, fonctionnelles ou liées à la qualité, soient, en apparence, spécifiques d'une application à l'autre, notre objectif est ici de recenser et formaliser des aspects génériques que partagent un grand nombre d'implantations.

Pour ce travail nous emettons certaines hypothèses. Premièrement, nous supposons que les étapes préliminaires nécessaires à la diffusion d'alarme, soient relativement prédictibles. Cela signifie que l'utilisation de tout matériel comme des transducteurs (si nécessaire), ainsi que la logique du logiciel nécessaire pour détecter les conditions d'alarme, peuvent être mises en œuvre sans trop de « customisation ». Cette hypothèse nous soulage des considérations liées aux déclinaisons de vastes gammes de capteurs et de modules logiciels embarqués, nous laissant plus d'espace pour nous concentrer sur les questions de diffusion au sein du système. Deuxièmement, nous supposons que le SDA est avant tout un système logiciel intensif (Hilliard, 2000). Ce qui signifie que le système attend une intervention manuelle minimale dans le processus de diffusion, le plus souvent limitée à la définition de politiques de diffusion, et que le système est capable de mettre en œuvre de façon transparente. Cette hypothèse se limite aux processus qui sont sujets à être *gérés* et non *conçus* (Prabhakar, 2009).

Ces hypothèses présentées nous pouvons maintenant formaliser des exigences d'un SDA. Nous divisons nos exigences dans les deux grandes sections, comme discuté auparavant – un pour chacun des problèmes fonctionnels et de qualité connexes.

2.1. Considérations fonctionnelles

Les aspects fonctionnels d'un système concernent la mise en œuvre des protocoles et des opérations que le système est censé remplir. Pour un SDA, ceci impliquerait de spécifier des informations telles que les moyens de communication, les priorités d'alarme, les hiérarchies de bénéficiaires, ainsi que d'autres exigences détaillant les étapes préliminaires. Ces exigences peuvent varier considérablement d'une mise en œuvre à une autre. Nous allons cependant essayer d'itérer sur certaines exigences possibles et leurs variations plausibles.

– Protocole de diffusion : un SDA peut être considéré comme un système à base de règles (Buchanan *et al*., 1984), conscient de la façon dont une alarme doit être diffusée. Il est dès lors possible que le SDA ait à choisir différentes stratégies de

diffusion, pour différentes alarmes. Nous allons brièvement donner un ensemble de classes de diffusion potentielles dans lesquelles une alarme peut se manifester :

– Tout ou rien : le protocole de diffusion pour cette classe d'alarmes est que l'alarme doit atteindre soit tous les destinataires, soit aucun. Considérons un exemple de scénario pour une telle politique. Une armée veut attaquer un poste ennemi de tous les côtés. Pour ce faire, toutes ses unités stationnées près du poste doivent attaquer en coordination les unes avec les autres. Il pourrait être catastrophique que certaines unités reçoivent ce message, et lancent l'offensive, tandis que d'autres ne l'auraient pas. Le message doit atteindre toutes les unités ou aucune. Nous appelons cette politique de diffusion une diffusion *atomique*, puisque celle-ci doit réussir pour tous les destinataires, ou aucun – tout comme le mécanisme de transaction dans une base de données.

– Exactement *m* sur *n* : le protocole de diffusion pour cette classe d'alarmes consiste à atteindre exactement m bénéficiaires, sur un total de n destinataires possibles. Les cas particuliers sont $m = 1$ ou $m = n$. Prenons un exemple de scénario dans lequel plusieurs accidents de la route ont été signalés au PU urgences d'une ville. Il suffit d'envoyer *exactement* une ambulance sur chaque théâtre d'accident. En supposant que le nombre d'ambulances disponibles soit limité, l'envoi de plus d'une ambulance à un endroit ne serait pas simplement un gaspillage de ressources, cela pourrait signifier l'indisponibilité de toute ambulance pour certaines de ces localisations. Le message pour atteindre un seul théâtre d'accident doit donc atteindre exactement une ambulance, pas moins, pas plus.

– Au plus *m* sur *n* : le protocole de diffusion pour cette classe d'alarmes est que l'alarme devrait atteindre de 0 à m destinataires sur un total de n. Prenons l'exemple d'un site de réseautage social, qui limite la portée des messages gratuits pour un utilisateur à un faible pourcentage de son réseau d'amis (cela peut résulter d'une décision pour encourager les utilisateurs à s'inscrire à des envois payants). Dans un tel cas, l'aspect critique du système est de mettre une limite supérieure au nombre de destinataires du message. Le système ne se soucie pas de la borne inférieure (qui peut être à 0).

Nous ne prétendons pas que la liste ci-dessus est exhaustive, mais la plupart des politiques de diffusion d'un SDA « réel » correspond à l'une de ces catégories.

2.2. Aspects qualité

Nous visons donc à construisons des systèmes en gardant certaines considérations relatives à la qualité en tête. Ces contraintes de qualité peuvent souvent ne pas être indiquées explicitement. Par exemple, un système construit pour le calcul de taxes disposera de la description détaillée des seuils, tranches, rabais, déductions autorisées, etc. comme une partie de la spécification des exigences. La vitesse à laquelle le calcul sera effectué (temps de réponse) est souvent déterminée par des facteurs humains.

Les attributs de qualité sont ces exigences qui définissent la fiabilité, « l'adéquation et la recevabilité d'un produit » (Firebrand Architect). La

communauté qui travaille sur l'architecture a identifié d'autres attributs de qualité d'un système tels que la disponibilité, l'interopérabilité, la modificabilité, la sécurité, la testabilité, la facilité d'utilisation et d'autres (ISQS ISO, 2011). Tous ces attributs de qualité sont atteints grâce à ce que nous appelons les stratégies architecturales (Bachmann, 2003). Ce stratégies sont de grandes décisions structurelles et comportementales, qui, lorsqu'appliquées à un système, permettent de mettre en œuvre certains attributs de qualité. Un compromis est souvent nécessaire entre différents plusieurs attributs de qualité : tout ne peut être réalisé dans un système en même temps. Nous allons nous intéresser à l'identification des attributs de qualité, qui sont spécifiques à un SDA. Pour chacun des attributs de qualité, nous avons spécifié les exigences en utilisant le stimulus, l'artefact, la réponse et la mesure de la réponse (Len Bass, 2007). Les attributs de qualité que nous avons identifiés pour un SDA sont les suivants :

– Traçabilité : dans certains cas, la diffusion d'une alarme n'est pas un processus simple. Prenons l'exemple de la diffusion d'un message par une page Facebook à ses « suiveurs », abonnés, amis. La tentative de mettre le message dans leurs échéanciers peut ne pas suffire. Facebook garde aussi une trace de la réception de ces messages. En particulier, il peut être intéressant à certains moments, de garder la trace des utilisateurs qui, en dehors du groupe de destinataires, ont reçu une alarme particulière (alors que d'autres dans le groupe ne l'ont pas reçue). En fait, dans certains scénarios, il peut être d'une importance capitale de le faire, dans le cas où il y a une possibilité d'alarmes « à suivre ». La traçabilité est une mesure de la dimension de la réception des alarmes diffusées, dimension évaluant dans quelle mesure un SDA se tient lui-même informé. Cela peut être utile pour ne cibler qu'un petit sous-ensemble du groupe de destinataires pour une alarme « à suivre » – uniquement ceux qui ont reçu l'alarme précédente, étant donné que pour les autres, celle-ci n'aurait pas de sens, faut d'avoir eu l'alarme initiale. L'analogie d'une alarme « à suivre » est l'envoi de notification dans Facebook, lorsque quelqu'un commente un message que vous avez aimé (la terminologie utilisée par Facebook est *abonnement à un « post »*).

Les stratégies pour mettre en œuvre la traçabilité peuvent inclure l'utilisation du pattern « Observer » (Gamma *et al.*, 1994). Une autre stratégie consiste à utiliser le pattern « Publish-Subscribe » (*idem*) où les bénéficiaires eux-mêmes souscrivent pour les alarmes « à suivre » ou extraient en continu les informations d'alarme du SDA pour vérifier si l'alarme attendue a eu lieu ou non. Dans le tableau 1, nous avons exprimé les exigences de qualité d'un système de traçabilité.

– Idempotence : certains aspects des systèmes distribués sont également applicables à un SDA. En fait, un SDA peut être visualisé sous forme d'un système distribué, si les destinataires ne sont pas locaux. Un problème auquel sont confrontés les concepteurs d'un système distribué est la retransmission de certains paquets ou messages, en raison de l'échec de la communication ou des délais d'attente. Le même scénario peut également se produire avec un SDA. Le système peut avoir à diffuser la même alarme, plusieurs fois, et pourtant, les récipiendaires doivent être

en mesure de ne traiter l'alarme *qu'une seule fois*. L'idempotence est la capacité d'un système à gérer les alarmes répétées de manière appropriée. Nous devons souligner le fait que l'idempotence exige une compréhension entre le récipiendaire et le SDA. Les SDA peuvent adopter un protocole pour de tels cas, que les récipiendaires sont tenus de respecter.

Tableau 1. Scénario général de traçabilité

Scénarios	Valeurs possibles
Stimulus	Alarme, alarme multiniveau, alarme à suivre, négation d'alarme
Artefact	Canaux de communication, stockage temporaire, processus et politiques de diffusion et les politiques
Réponse	Suivre les destinataires pour les alarmes subséquentes – Se connecter à la diffusion d'alarme – Diffuser les alarmes subséquentes uniquement aux destinataires connectés (personnes ou systèmes)
Mesure de réponse	Pourcentage de traçabilité (par exemple, 99,999 %) Proportion ou taux de certaines classes d'alarmes que le système suit avec succès
Stratégie	Le SDA suit les bénéficiaires Les bénéficiaires souscrivent ou extraient les alarmes à suivre

La stratégie la plus simple pour réaliser l'idempotence consiste à utiliser un système de « versionnage » pour les alarmes. Les alarmes suivent un système de séquençage non décroissant qui assigne un identifiant unique à chaque alarme distincte. Les détails relatifs à l'idempotence sont présentés dans le tableau 2.

Tableau 2. Scénario général d'idempotence

Scénarios	Valeurs possibles
Stimulus	Alarmes répétées, alarmes identiques
Artefact	Canaux de communication, stockage temporaire, processus de diffusion
Réponse	Destruction de l'alarme identique ou transmission répétée de l'alarme
Mesure de réponse	Précision du filtrage (taux d'identification réussie des alarmes en double) Nombre de double alarmes diffusées par le système et coût de la diffusion en double
Stratégie	Gestion des versions des alarmes

– Justesse/Aptitude/Pertinence : un protocole de diffusion peut définir plus d'une sortie acceptable pour la diffusion d'une alarme particulière. Par exemple, dans le protocole « *tout ou rien* », il est acceptable de ne pas diffuser l'alarme à l'un des destinataires. Cependant, il est généralement souhaitable de parvenir à l'état « *tout* » plus souvent qu'à l'état « *rien* ». Dans un tel cas, la capacité d'un système à présenter un comportement accepté plutôt que les autres peut être une indication de la qualité du système. La pertinence est la mesure d'un système à adopter le résultat le plus souhaitable d'un protocole de diffusion par rapport aux autres états. Les exigences de qualité pour la pertinence sont spécifiées dans le tableau 3.

Tableau 3. Scénario général de la pertinence

Scénarios	Valeurs possibles
Stimulus	Alarme
Artefact	Destinataires prévus
Réponse	Parmi de multiples politiques de diffusion acceptables, certaines sont préférées à d'autres
Mesure de réponse	Le nombre de fois que la politique de diffusion préférée est choisie
Tactique	Re-transmission des alarmes Communication synchrone

3. Architecture de référence pour les SDA

Après avoir brièvement évoqué les problèmes liés à la conception d'un SDA, nous allons dans ce qui suit proposer une architecture de référence. Une architecture de référence ne cherche pas à construire un système en soi : au contraire, il s'agit de faire ressortir les éléments importants qui peuvent être assemblés pour construire une architecture spécifique, adapté à un domaine donné (Len Bass, 2007 ; Taylor *et al.*, 2009). Nous tenterons donc de proposer ici une architecture de référence pour les SDA.

Une procédure générale suivie par un SDA est représentée dans la figure 1. Le processus est essentiellement basé sur trois étapes. La détection d'une alarme, en utilisant des métadonnées d'alarme, donne lieu à sa classification (Wynne Hsu Yiming Ma, 1998) en fonction de ses caractéristiques. La deuxième étape est la diffusion de l'alarme aux groupes de destinataires prévus. Enfin, une série d'étapes post-diffusion interviennent pour traiter les cas d'alarmes « à suivre », ou d'alarmes de négation. Une architecture de référence pour un SDA est proposée en figure 2. Un SDA reçoit une alarme émanant d'étapes préliminaires, elles-mêmes pouvant mettre en œuvre des protocoles de déclenchement des alarmes : nous les appelons ici systèmes de déclenchement d'alarmes (SDéA). Le SDéA est responsable de

« capter » l'environnement, par l'intermédiaire de composants matériels ou logiciels, et de détecter les événements, qui sont des alarmes, sur la base des protocoles de déclenchement disponibles. Le SDéA ne fait pas partie du SDA, nous n'en discutons donc ici. Le SDA, est le cœur de notre système et de notre analyse. Dans le SDA, le système de détection et de dissémination de l'information (SDéDA) reçoit l'alarme du SDéA. Une fois que le SDéDA reçoit l'alarme, il utilise les politiques de diffusion disponibles pour trouver l'information pertinente à la diffusion, à partir de celles stockées localement, telles que les groupes destinataires, les politiques de confidentialité, les options de traçabilité à déployer, etc. L'alarme est ensuite lancée sur le réseau de diffusion, qui peut être construit à partir d'un ou de plusieurs dispositifs de communication, ou matériels intermédiaires. Les destinataires peuvent être différents dispositifs, humains, ou des plates-formes de médias sociaux. Un stockage local peut également être aux commandes du SDéDA, afin de conserver les informations opérationnelles telles que les données de traçabilité de certaines alarmes précédentes. Une autre propriété du SDA peut être la capacité à s'améliorer lui-même sur la période opérationnelle (nous utilisons le terme « améliorer » au sens large ici pour signifier une amélioration d'un ou plusieurs attributs de qualité). Par exemple, il peut incomber au SDA de modifier les politiques de diffusion, ou des groupes de destinataires de certaines alarmes, en fonction des commentaires que ceux-ci ont formulé, ou par détection directe de l'environnement lui-même (en particulier applicable aux cas où l'environnement est un média social ou une plate-forme comme Twitter). Le composant d'adaptation effectue quelques heuristiques sur les données disponibles et peut modifier les politiques pour les diffusions à venir.

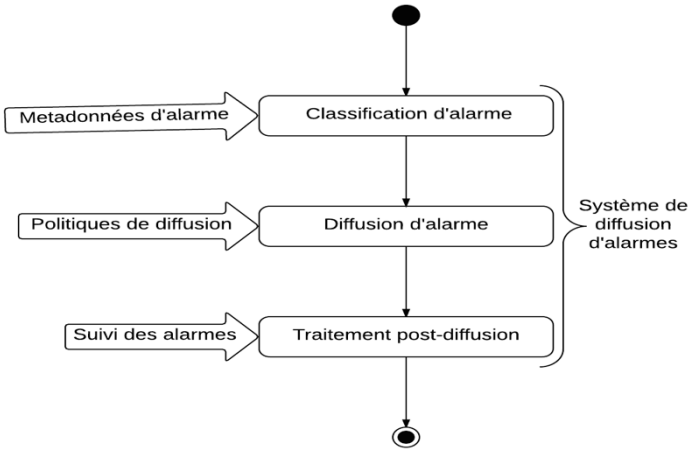

Figure 1. Le processus de diffusion des alarmes

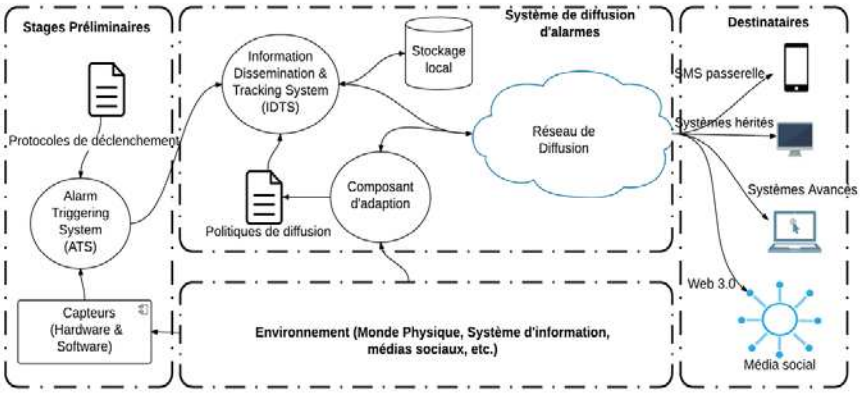

Figure 2. Architecture de référence du SDA

Un aspect important du SDA est le réseau de diffusion. Le réseau de diffusion pourrait être homogène, ou constitué d'un mélange hétérogène de différentes entités matérielles et logicielles. Puisque nous nous limitons au niveau de l'architecture de référence, nous n'abordons que certains aspects du réseau de diffusion, jouant un rôle important, dans l'architecture effective du système.

Initialisation : le processus de diffusion d'une alarme peut être déclenché de deux manières :

– SDA à base de « push » (*Push-based ADS*). Un SDA à base de « push » est un système dans lequel l'alarme est envoyée par le SDA aux destinataires. Il est de la responsabilité du SDA d'informer les bénéficiaires de la survenance d'une alarme, la plupart du temps dans une fenêtre de temps limitée après sa réception via les étapes préliminaires. Les destinataires sont donc censés être en mode écoute, pour recevoir des messages du SDA. Les SDA peuvent soit garantir la réception de l'alarme aux destinataires, soit adopter un mécanisme de livraison « best-effort » (à savoir, les SDA feront de leur mieux pour fournir l'alarme, mais ils ne le garantissent pas dans tous les cas). Un exemple d'un SDA à base de « push » peut être un système d'alerte de catastrophe naturelle utilisé par une agence gouvernementale pour avertir les citoyens d'une zone via SMS ou e-mail.

– SDA à base de « pull » (*Pull-based ADS*). Un SDA à base de « pull » est un système dans lequel l'alarme est sollicitée par les bénéficiaires du SDA, *via* l'envoi de requêtes. Le SDA reçoit des informations sur les différentes alarmes *via* les étapes préliminaires, et garde une trace de celles-ci localement, sans les diffuser à tous les destinataires. Les intéressés peuvent envoyer des requêtes périodiques aux SDA pour savoir si certaines alarmes ont été déclenchées. Les SDA répondent aux destinataires par l'affirmatif si les alarmes respectives ont été déclenchées récemment (la définition de ce qui est « récent » pouvant être différente pour différentes implémentations). Les SDA peuvent choisir d'être « bienveillants », ce

qui signifie l'envoi de réponses négatives aux destinataires dans le cas où les alarmes en question n'ont pas été déclenchées récemment, ou peuvent choisir d'ignorer simplement les requêtes, dans le cas où la réponse est négative. Un exemple de SDA à base de « pull » peut être un serveur duo App/Mobile Web, qui notifie l'utilisateur chaque fois qu'un but est marqué dans un jeu de football (le serveur Web est le SDA, les différentes instances de l'application mobile sont les destinataires).

Mixant les deux types ci-dessus, un SDA hybride peut mettre en œuvre les services à base « push » ainsi qu'à base de « pull », selon la gravité de l'alarme ou les capacités des destinataires.

Communication et topologie : suivant les politiques de diffusion, et la façon dont celle-ci a lieu, un autre aspect d'un SDA est de gérer les liens de communication avec les bénéficiaires. Nous allons brièvement expliquer les questions à examiner par un SDA :

– Moins de connexion ou connexion orientée. La communication entre un SDA et un destinataire pourrait mettre en œuvre moins de connexion ou une connexion orientée. Un protocole de communication avec moins de connexion fonctionne sur le mécanisme de livraison « best-effort ». Les paquets de données circulent sur un réseau et sont censés éventuellement atteindre la destination par la suite. Ils peuvent atteindre la destination hors service, ou ne pas l'atteindre du tout. Un protocole de communication avec une connexion orientée assure qu'un canal est créé entre les parties qui communiquent, et que les messages sont reçus avec des garanties. Il peut y avoir retransmission de certains paquets de données, dans le cas où un accusé de leur réception n'a pas été reçu par l'expéditeur. Un SDA informant les destinataires par e-mail ou SMS est un exemple de communication avec moins de connexion. Un SDA faisant un appel « IVR » pour inviter l'utilisateur à fournir ses informations d'identification de carte de crédit pour effectuer un achat, pourrait être considéré comme un exemple de communication avec une connexion orientée (si l'utilisateur ne capte pas, ou raccroche entre les deux, le système le détecte et peut relancer l'appel).

– Transits simples ou multiples. Un autre aspect à considérer lors de la conception d'un SDA est la distance de transit des destinataires. Un SDA peut être construit sur une base hiérarchique, par exemple, un niveau national, dont le travail pourrait être de diffuser des alarmes à un ou plusieurs SDA au niveau de l'État, qui, à son tour, peut diffuser l'alarme, ou déléguer à un SDA au niveau du district, etc. L'avantage de cette topologie est que la charge réelle de diffusion est répartie sur plusieurs systèmes et protocoles de diffusions différents, qui peuvent être utilisés à différents niveaux, selon les besoins. Un exemple d'un tel système est constitué des organismes de télédétection nationaux et régionaux. L'agence nationale a accès aux données satellitaires actuelles, qui peuvent être traitées afin de connaître des conditions alarmantes dans une ou plusieurs régions. Celles-ci peuvent ensuite être diffusées aux agences régionales respectives, qui peuvent décider de façon indépendante ce qu'il faut diffuser (ou ne pas diffuser).

– Redondance de lien. Les liens de communication sont sujets à défaillances. Des pannes de courant aux catastrophes naturelles, en passant par le vandalisme, tout peut rompre un lien entre le SDA et un destinataire. Sur la base de la gravité du système d'alarme, un SDA peut avoir à être prêt à contrer de telles situations. Les SDA peuvent avoir à établir des liens redondants avec certains ou tous les destinataires dans ces scénarios. Un exemple pourrait être l'utilisation de deux types de liens différents, fibres optiques et communication par satellite, pour la diffusion d'alarme. Dans des circonstances normales, les SDA peuvent utiliser les fibres optiques, plus rapides. Dans l'autre éventualité, les SDA peuvent basculer vers une communication satellite, qui, bien que peut-être plus lente, serait disponible avec une probabilité plus élevée.

L'architecture de référence proposée peut constituer une référence pour concevoir un SDA. Il faut noter que tous les éléments présentés ici ne sont pas applicables dans tous les scénarios. Nous avons proposé une architecture de référence en gardant à l'esprit un large éventail d'exigences fonctionnelles et de qualités plausibles pour un SDA.

4. Étude de cas : les questions liées au fonctionnement du TCAS (*Traffic Collision Avoidance System*)

Pour illustrer le processus visant à proposer les détails architecturaux les plus fins d'un SDA, nous proposons un système intéressant et très sensible utilisé par la sécurité aérienne, appelé système de trafic d'évitement des collisions (Traffic Collision Avoidance System, TCAS). Le TCAS est un système d'évitement de collision d'avions, utilisé dans des avions de ligne modernes pour éviter les possibilités de collisions aériennes. Bien que la diffusion des alarmes TCAS ne soit pas d'une grande importance, étant donné que les alarmes ne sont diffusées localement que sur un avion, l'intégration globale des TCAS dans le processus de l'aviation est importante pour toute personne intéressée par la construction d'un SDA.

Introduction au TCAS. Le débat sur la nécessité d'un TCAS a été lancé en 1950 après une collision aérienne dans le Grand Canyon (Murphy, 1990). Sans entrer dans les détails techniques, nous pouvons élaborer des TCAS comme un ensemble de composants matériels et logiciels, installés sur un plan, qui peut communiquer et négocier avec d'autres avions à proximité immédiate afin d'éviter tout risque de collision (Rich et Anderson, 1997). Dans le cas de deux plans trop proches, le TCAS peut alerter les équipages du danger potentiel, et fournir des instructions spécifiques pour modifier l'altitude des avions de telle sorte que la collision soit évitée.

Le TCAS et son environnement. Bien que le TCAS semble être un moyen efficace pour éviter la collision en vol, la situation devient un peu plus compliquée lorsque nous considérons la façon dont il se situe dans le processus global de l'aviation. Un vol typique implique une série de communications entre trois entités

différentes : l'équipage de vol (généralement pilotes et co-pilotes), l'ordinateur (s) de bord et le contrôle du trafic aérien (CTA). Nous considèrerons ici le TCAS sous la rubrique des ordinateurs de bord. Trois entités peuvent, du moins en théorie, lancer le processus d'évitement d'une collision en vol, indépendamment les uns des autres. 1) Par exemple, un pilote peut voir visuellement un autre aéronef dans le voisinage, et peut décider de modifier l'altitude de l'avion pour éviter la collision. Bien que théoriquement possible, dans la pratique, le délai nécessaire pour modifier la route de l'avion, après qu'un pilote ait détecté visuellement le trafic en contexte, peut ne pas suffire. 2) Le CTA peut observer sur radar si deux avions sont sur une trajectoire de collision. Dans un tel cas, le CTA indique aux équipages de modifier leur altitude, généralement en maintenant une distance d'au moins 1 000 pieds. 3) Si les deux entités n'y parviennent pas, la dernière couche de protection est assurée par le TCAS. Le TCAS va générer des avertissements comme « *descendez* » ou « *montez* » sur les écrans des pilotes, pour éviter la collision entre les avions. Un protocole interne négocie le processus (qui *montera* et qui *descendra*) entre les deux plans, et assure qu'aucun des deux plans n'obtienne le même avertissement – monter ou descendre.

Echec du TCAS. Examinons maintenant les circonstances d'un incident tragique qui a eu lieu dans la ville allemande de Überlingen en juillet 2002, afin de voir le fonctionnement du TCAS par rapport à un SDA. Le 1er juillet 2002, deux avions de ligne sont entrés en collision en vol, sur les villes de Überlingen et Owingen en Allemagne, bien que les deux aient été équipés de TCAS (Brooker, 2008). L'enquête qui a suivi a identifié une grave lacune qui a conduit à l'accident, l'ensemble du système étant un peu ambigu sur la façon de gérer l'alarme de collision (plutôt d'évitement des collisions). Alors que les deux avions ont reçu des instructions spécifiques du TCAS pour éviter la collision, l'un d'entre eux a reçu une instruction contradictoire du CTA. La communication nécessaire entre l'équipage de l'autre avion et le CTA ayant échoué, l'autre avion a suivi les instructions données par le TCAS.

SDA pour l'évitement de collisions : nous pouvons visualiser un système anticollision comme un SDA potentiel.

– Étapes de détection d'alarme :

1) Recueillir l'information d'altitude des avions à proximité ;

2) Extrapoler leurs trajectoires ;

3) Déterminer si l'un d'entre eux peut potentiellement entrer en collision avec l'avion ou non.

– Déclenchement d'alarme :

1) Le pilote peut voir visuellement un autre avion dans ses environs, et se rendre compte qu'ils sont trop proches les uns des autres ;

2) Le CTA réalise que deux avions sont sur une trajectoire de collision, avec l'aide des radars ;

3) Le TCAS découvre qu'une trajectoire d'avion à proximité peut potentiellement provoquer une collision avec l'avion.

– Protocole :

1) Les pilotes doivent communiquer entre eux, et décider des manœuvres qui permettront d'éviter la collision (impossible dans la plupart des cas) ;

2) Le CTA devrait fournir des instructions aux avions afin de modifier leur altitude et éviter la collision. En règle générale, l'un des avions doit soit monter soit descendre, et l'autre doit faire le contraire, ou conserver son niveau de vol.

– Les deux pilotes doivent écouter les avertissements du TCAS sur leurs avions, et suivre les instructions (monter ou descendre) pour éviter la collision.

Analysons le cas d'un défaut probable dans le processus. Ecartant les manœuvres de communication pilote (qui peuvent être à la fois impraticables, aussi bien que dangereuses en tant que telles), lequel des deux autres protocoles a une préséance plus élevée ? En d'autres termes, dans le cas où un pilote reçoit des instructions à la fois du TCAS et du CTA, quelle instruction (alarme) doit être traitée par lui ? Dans le cas de la tragédie de Überlingen, l'un des pilotes a suivi les instructions du TCAS, tandis que l'autre a suivi les instructions du CTA.

Essayons maintenant de modéliser un SDA pour éviter les collisions entre avions. Par souci de simplicité, nous supposons que tous les humains impliqués dans le système se comportent selon un protocole standard prédéfini (de sorte que nous pouvons modéliser leurs actions correctement). Le système s'instancie à peu près comme à la figure 3.

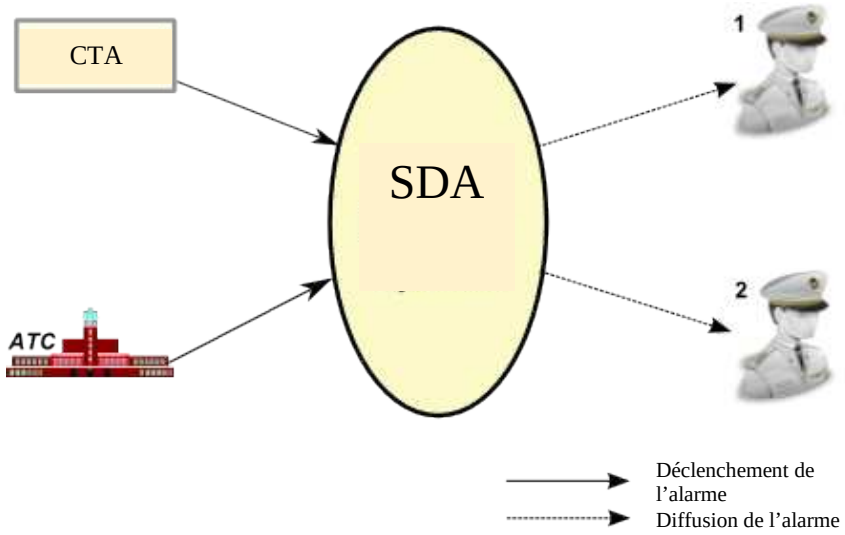

Figure 3. Envisager un SDA pour éviter les collisions aériennes

Certaines des questions évoquées pour un SDA sont applicables dans ce cas aussi. Par exemple, nous avons parlé de protocoles de diffusion dans la section 2.1. Un exemple simple du protocole « tout ou rien » peut être appliqué à des instructions (alarmes) émises par le CTA. Les instructions doivent soit joindre les deux avions, soit aucun. Il peut être catastrophique, comme dans l'accident mentionné ci-dessus, que le message atteigne l'un, et pas l'autre.

En un mot, si nous voyons les instructions nécessaires données aux pilotes des deux avions comme des alarmes et modélisons l'ensemble du système d'évitement de collision tel que présenté, nous pouvons étudier et déduire des inférences à partir de ce système, qui peuvent être appliquées à d'autres SDA. Une conclusion que nous avons tirée de notre modélisation est qu'une ambiguïté dans les protocoles de diffusion peut faire des ravages dans un SDA. Une autre conclusion est qu'un canal de communication supplémentaire entre le SDA et le destinataire peut se révéler critique dans certains scénarios, lorsque le canal principal de communication n'est pas disponible.

5. Conclusion et perspectives

Les systèmes de diffusion d'alarme sont complexes et sophistiqués. Un point de vue architectural nous permet d'aborder la complexité d'une manière systématique. La délimitation des exigences fonctionnelles et de qualité s'avère utile dans la construction de tels systèmes. Une architecture de référence suggère une mise en œuvre de base. Un SDA « pluggable », extensible et adaptable est une instance de système critique distribué pertinente : l'exigence des attributs de qualité garantit une analyse minutieuse – les interdépendances seraient intéressantes. Une étude détaillée des modèles de conception applicables à un SDA s'avèrera utile pour enrichir le potentiel de connaissances sur l'architecture pour la construction de systèmes de diffusion d'alarme.

Bibliographie

Bachmann F., Bass L., Klein M. (2003). *Deriving architectural tactics: A step toward methodical architectural design*. Technical report, DTIC Document.

Bass L. (2007) *Software architecture in practice*. Pearson Education India.

Brooker P. (2008). *The Überlingen accident: Macro-level safety lessons.* Safety Science, vol. 46, n° 10, p. 1483-1508.

Buchanan B. G, Hance Shortliffe E. *et al.* (1984). *Rule-based expert systems*, vol. 3. Addison-Wesley Reading, MA.

Firebrand Architect. *Quality attributes*. http://www.softwarearchitectures.com/qa.html.

Gamma E., Helm R., Johnson R., Vlissides J. (1994). *Design patterns: elements of reusable object-oriented software*. Pearson Education,

Hilliard R. (2000). Ieee-std-1471-2000 *recommended practice for architectural description of software-intensive systems*. IEEE, http://standards. ieee. org, 12, p. 16-20,

ISQS ISO. Iso/iec 25010". (2011). *Systems and software engineering. Systems and software Quality Requirements and Evaluation (SQuaRE).* System and software quality models.

Murphy G.K (1990). *The grand canyon midair collision: A stimulus for change*. The American journal of forensic medicine and pathology, vol. 11, n° 2, p. 102-105.

Prabhakar T.V (2009). cs654 software architecture class notes. Dept. of CSE, IIT Kanpur,

Rich R.S., Anderson M.W (1997). *Traffic alert and collision avoidance coding system*, June 3. US Patent 5,636,123.

Sommerville I., Sawyer P. (1997). *Requirements engineering: a good practice guide*. John Wiley & Sons, Inc.

Taylor R.N., Medvidovic N., Dashofy E.M (2009). *Software architecture: foundations, theory, and practice*. Wiley Publishing.

Wynne Hsu Yiming Ma B. L. (1998). *Integrating classification and association rule mining.* In Proceedings of the *4th International Conference on Knowledge Discovery and Data Mining.*