Operational Context-Based Design and Architecting of Autonomous Vehicles

Youssef Damak

Thèse de doctorat

NNT : 2020UPASC029

université PARIS-SACLAY

Operational Context-Based Design and Architecting of Autonomous Vehicles

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 573, Approches interdisciplinaires, fondements, applications et innovation (Interfaces)
Spécialité de doctorat : Génie Industriel
Unité de recherche : Université Paris-Saclay, CentraleSupélec, Laboratoire Génie Industriel, 91190, Gif-sur-Yvette, France
Référent : CentraleSupélec

Thèse présentée et soutenue à Gif-sur-Yvette, le 16 Juillet 2020, par

Youssef DAMAK

Composition du Jury

Bertrand ROSE Professeur, Université de Strasbourg, ICUBE UMR CNRS 7357	Président
Claudia ECKERT Professeur, The Open University	Rapporteur & Examinatrice
Eric BONJOUR Professeur, Université de Lorraine, Laboratoire ERPI-ENSGSI	Rapporteur & Examinateur
Bernard YANNOU Professeur, Université Paris-Saclay, CentraleSupélec, Laboratoire Génie Industriel	Examinateur
Marija JANKOVIC Professeur, Université Paris-Saclay, CentraleSupélec, Laboratoire Génie Industriel	Directrice de thèse
Yann LEROY Maitre de conférences, Université Paris-Saclay, CentraleSupélec, Laboratoire Génie Industriel	Co-Encadrant & Examinateur
Guillaume TREHARD Senior Manager, AKKA Technologies, Autonomous Systems	Invité

université PARIS-SACLAY | ÉCOLE DOCTORALE Interfaces: matériaux, systèmes, usages

ECOLE DOCTORALE N°573
Approches interdisciplinaires, fondements, applications et innovation (Interfaces)

Titre : Conception d'Architecture de Véhicules Autonomes basée sur le Contexte Opérationnel

Mots clés : Conception d'Architecture de Véhicules Autonomes ; Conception Basée sur le Contexte Opérationnel ; Ontologie de Contexte Opérationnel ; Propagation du Changement ; R&D Externalisé

Résumé : Les Véhicules Autonomes (VA) sont des systèmes émergents et considérés comme une pierre angulaire de la mobilité du futur. Leur conception est à l'origine de nombreux efforts de recherche universitaires et industrielles. L'industrialisation des VAs est un moyen pour les acteurs de la mobilité de renforcer le positionnement futur. Les VAs fonctionnent en interagissant avec leur contexte opérationnel (CO) et doivent être adaptés à celui-ci. L'adaptation des architectures des VAs à leur CO dès la conception devient un défi important dans la conception de VA robustes.
L'état de l'art actuel ne propose pas de méthodes de conception d'architecture de VAs basées sur le CO. Ce travail de recherche vise à soutenir les activités d'architecture des Véhicules Autonomes pour aboutir à des architectures adaptées à leurs contextes opérationnels.
Une ontologie du CO pour Véhicules Autonomes est proposée pour soutenir l'identification et la définition de scénarios dans la phase initiale de conception, suivant une approche de conception basée sur les scénarios. En utilisant cette ontologie, une méthode de conception de l'architecture logique des VAs basée sur l'OC est proposée. La prise en compte du CO dans les activités de conception d'architecture des VAs est renforcée par une deuxième méthode visant à évaluer l'impact du changement du CO sur l'architecture durant la phase de conception. Les contributions proposées sont validées par des études de cas industriels sur la conception d'architectures AV tenant en compte du CO et de son évolution.

Title: Operational Context-Based Design and Architecting of Autonomous Vehicles

Keywords: Autonomous Vehicles Architecting; Operational Context-Based Design; Operational Context Ontology; Change Propagation; Outsourced R&D

Abstract: Autonomous Vehicles (AV) are emerging systems and considered cornerstones of the future of mobility. Their design is a source of many academic and industrial research efforts. The industrialization of AV is the mean for mobility stakeholders to strengthen their future position. AVs function by interacting with their operational environment and must be fit for their Operational Context (OC). Adapting AVs architectures to their Operational Context during design becomes an important challenge in designing robust AV.
The current state of the art does not propose AV architecting methods based on the OC. This research work aims to support the architecting activities of Autonomous Vehicles to result in architectures fit for their Operational Context
An OC ontology for AV is proposed to support scenario identification and definition in the early design phase, for a scenario-based design approach. Using this ontology, a method to design AV logical architecture based on the OC is proposed. The consideration of the OC in the architecting activities of AV is strengthened with a second method aiming at assessing the impact of OC change on the AV's architecture during the design phase. The proposed contributions are validated with industrial case studies on the design of AV architectures given the OC and its evolution.

Abstract

Autonomous Vehicles (AV) are new complex systems seen as a cornerstone of future mobility. Vehicles manufacturers and mobility stakeholders are focusing on the AV development to strengthen their position in the future mobility. However, such novel complex systems generally necessitate new skills historically not developed in the automotive industry. Hence, the industry is reorganizing into complexes design processes in order to propose novel AV concepts and support feasibility studies. In this new context, mobility stakeholders often work with engineering consulting companies, suppliers, and new entrants with broad skillset and diversified industrial feedbacks for AVs Research & Development (R&D) and experimentations

Autonomous vehicles are systems interacting with their operational environment with cognitive and physical capabilities. As such, the technical choices are often made to be suitable for the Operational Context (OC) of the vehicle. However, a clear justification of technical choices in new projects is undermined by two main problematics: the difficulties for the end user to translate its needs into technical requirements due to the youth of the technology; and the absence of a formalized link between the AV architecture and its Operational Context. During the design phase, this situation often results in weak understanding of component role, a high frequency of changes, and significant time loss in change impact assessment. These elements provoke delay in deliveries and the clients dissatisfaction with the robustness and clarity of the design process.

This research aims to support the architecting activities of Autonomous Vehicles to result in architectures fit for their Operational Context. An initial experiment of requirement elicitation through requirements reuse by OC correspondence suggested the need for a more complex representation of the AV's OC. Hence, an OC ontology for AV is proposed to support the scenario identification and definition in the early design phase, for a scenario-based design approach. Using this ontology, a method to design AV logical architecture based on the OC is proposed. The consideration of the OC in the architecting activities of AV is strengthened with a second method aiming at assessing the impact of OC change on the AV's architecture during the design phase. This method is also

deployable in new projects on reference architectures by changing the reference OC. The proposed contributions of this thesis are validated with industrial case studies on the design of AV architectures given the OC and its evolution.

Résumé étendu

Un Véhicule Autonome (VA) est un véhicule sans conducteur qui hérite des capacités d'un véhicule classique tout en réalisant les actions du conducteur. Pour ce faire, il exhibe des capacités de perception, de supervision et d'action grâce à de multiple fonctionnalités tels que l'acquisition de données de capteur, l'interprétation, la décision, et le control. Les véhicules autonomes font partie d'une classe de système appelée Système Cyber-Physiques (CPS) véhiculaire qui perçoivent et utilisent leurs contextes afin de produire un comportement et des services mobilités pertinentes pour les usagers. Les VA sont considérées comme une composante essentielle du concept de villes intelligentes et un pilier de la mobilité future. Pour cela, leur développement et industrialisation sont devenuent un axe majeur des industries autour de la mobilité et de multiples communautés de recherche. Ainsi, l'investissement global pour leur développement en 2017 a été estimé à plus de 80 milliards de dollars (Kerry et Karsten, 2017).

La raison d'être d'un CPS véhiculaire peut se définir comme la réalisation de mission de mobilité dans le respect des règles de circulation, tout en assurant l'intégrité des passagers et du trafic en toute situation. La particularité et la complexité de cette raison d'être déterminent un système qui fonctionne dans un contexte opérationnel hautement dynamique et non contrôlé, tout en assurant des niveaux de sécurité élevées. Les approches classiques de conception visant à maitriser le contexte opérationnel et à assurer la sécurité de complexes systèmes Cyber-Physiques se basent sur l'analyse des Concepts d'Opération (ConOps). Cette analyse comprend l'analyse des activités et scénarios opérationnels du système ainsi que de ses modes nominaux et défaillants. Ces analyses servent à définir et spécifier les exigences du système. Cependant, les approches classiques trouvent leurs limites dans le contexte des véhicules autonomes principalement en raison de la complexité, de la dynamique et de la grande diversité des éléments du contexte opérationnel. La multiplicité et variété de rôles que jouent les éléments du contexte dans les situations rencontrées par le système représentent aussi un facteur limitant pour les approches classiques. Ainsi, leur application ne garantit pas la définition de tous les scénarios opérationnels importants, là où le risque d'en omettre peut conduire à l'incapacité du véhicule autonome à réaliser ses missions.

Les conséquences des limites des approches classiques ont pu être observés avec l'équipe Systèmes Autonomes d'AKKA Technologies, lors de la participation de l'auteur à quatre projets R&D pour le compte de leurs clients sur des systèmes de conduite autonome. Une analyse de leur R&D externalisée des Systèmes Cyber-Physique véhiculaires a mis en évidence de multiples défis directement liés à la complexité des objectifs des Véhicules Autonomes. (1) L'équipe avait des difficultés considérables à définir les contrats et accords avec leurs clients pour la réalisation des projets R&D. Cela est principalement due à une difficulté à déterminer les exigences du système en phase amant des projets sans les lier aux éléments du contexte opérationnel et à l'incapacité à identifier l'ensemble des variations importantes des scénarios opérationnels. (2) De plus, nous avions constaté une forte incertitude sur les choix techniques et d'importants retards sur les projets, tous deux dues à des changements tardifs au niveau du contexte opérationnel.

Les difficultés industrielles observées, ainsi que les limites des approches classiques pour la conception des VA nous ont amené à identifier deux objectifs de recherche : (1) pour la conception des véhicules autonomes, nous avons besoin de nouvelles méthodes pour explorer systématiquement leur domaine opérationnel afin de définir exhaustivement les exigences du système dans la phase de conception initiale et de concevoir l'architecture du système. (2) En deuxième lieu, nous avons besoin d'une méthode pour anticiper l'évolution inévitable de l'architecture des véhicules autonomes lorsque son domaine opérationnel change.

Pour répondre à ces limites, nous avons suivi une méthodologie de recherche basée sur la stratégie Eight Pathfold (Eckert et al., 2003). La méthode commence par une étude empirique des défis industriels dans la conception des véhicules autonomes. Cette étude a permis de définir des objectifs de recherche. Après avoir défini les questions de recherche par rapport aux objectifs de recherche. Nous avons mené une vaste étude de l'état de l'art et identifié une opportunité de recherche pour chaque question. Nous avons ensuite élaboré des modèles et des méthodes contribuants à répondre aux questions de recherche. Nous avons complété ces études par des cas d'application et des évaluations d'experts afin de valider les contributions proposées.

En suivant la méthodologie de recherche présentée, nous avons défini trois questions de recherche pour répondre aux deux objectifs de recherche présentés précédemment. Pour répondre au besoin d'une nouvelle méthode d'exploration du domaine opérationnel, nous avons examiné « Comment définir systématiquement des scénarios opérationnels basés sur le contexte opérationnel au début de la phase de conception ? » Pour répondre au même objectif, nous avons étudié une deuxième question « Comment concevoir et modéliser l'architecture d'un CPS véhiculaire en fonction du contexte opérationnel et des scénarios opérationnels définis ? » Après avoir contribué à la résolution de ces questions de recherche, nous nous sommes concentrés sur la nécessité d'une méthode permettant d'anticiper l'impact des changements dans le domaine opérationnel. Cet objectif nous a conduit à étudier une troisième question : « Comment évaluer l'évolution de l'architecture de CPS véhiculaire lorsque le contexte opérationnel change ? »

Nous présentons dans cette thèse trois contributions répondants au trois questions de recherche énoncées. Pour répondre à la première question de recherche « Comment définir systématiquement des scénarios opérationnels basés sur le contexte opérationnel au début de la phase de conception ? » une littérature extensive est présentée en détails dans le chapitre 3 pour définir les opportunités de contributions scientifiques. Se basant sur cette littérature, nous proposons d'étendre les approches existantes d'analyse des Concepts d'Opérations, par une analyse des éléments du contexte opérationnel, leurs dynamiques et leurs évolutions. L'analyse du contexte opérationnel supporte alors une méthode de définition systématique des scénarios opérationnels que peut rencontrer le Système Cyber-Physique véhiculaire.

Pour comprendre l'argumentaire qui nous a amené à proposer cette contribution, voici un résumé de la littérature extensive présentée au chapitre 3. Suivant la définition de Ulbrich (2015), un scénario opérationnel dans le cadre de la conception d'un CPS véhiculaire est « une séquence temporelle de scènes représentant une image de l'environnement incluant le décor et les éléments dynamiques, ainsi que les relations entre eux ». Selon une décomposition proposé par Bach (2016), les éléments suivants constituent les dits scénarios opérationnelles : un décor composé par la structure de la route et la météo, une situation composée par une scène et des manœuvres, des participants positionnés dans le décor et réalisants les manœuvres, et des évènements. Ainsi, notre proposition de représentation du

contexte opérationnel se doit de permettre l'identification des variations de scènes et situations que peut rencontrer un CPS véhiculaire.

De plus, l'étude de l'état de l'art sur les modélisations de scénarios dans le cadre de la conception des CPS véhiculaires montre qu'ils sont modélisés durant deux phases : (1) la phase de tests et validation ainsi que (2) la phase de conception et développement. Durant (1) les phases de tests et validation, les études se focalisent sur la génération des scénarios par des méthodes combinatoires et probabilistes sur des structures de routes prédéfinies. De l'autre côté, les études portants sur la modélisation des scénarios durant (2) les phases de conception et développement se focalisent sur la définition des scénarios basés sur les usages et cas d'utilisation. On constate ainsi un manque de méthode pour identifier et définir systématiquement les variations des structures de la route et les situations possibles en se basant sur le contexte opérationnel en phase amont de la conception.

Ainsi, nous avons étudié les différentes représentations du contexte opérationnel dans la littérature. Le contexte opérationnel est définit comme « toute information qui permet de caractériser la situation d'une entité (personne, espace, objet) et qui est pertinente lors de l'interaction entre l'utilisateur et le système » (Dey, 2001). La littérature montre que les formes de représentations ontologiques sont plus adéquates pour capturer la complexité des éléments du contexte d'un CPS véhiculaire, leurs relations, et permettent de les utiliser pour une identification systématique des scénarios. De plus, nous retrouvons dans la littérature une grande variété d'éléments définis dans le contexte opérationnel des CPS véhiculaire distribuée dans les 4 catégories suivantes : Environnement, Infrastructure de route, Infrastructure de trafic, et Objet. Cela dit, la littérature ne fournit pas de modélisation formelle du contexte opérationnel permettant d'étendre les analyses de Concepts d'Opérations par une définition systématique de scénario.

Ainsi, nous proposons dans le chapitre 3 une ontologie du contexte opérationnelle des Systèmes Cyber-Physiques véhiculaires. Elle introduit un niveau de modélisation cas d'utilisation qui permet de lier entre les éléments du contexte et l'analyse ConOps. Ainsi, elle est formalisée en 5 niveaux et définit les concepts et leurs relations : (1) Cas d'utilisation, (2) Environnement, (3) Infrastructure de route, (4) Infrastructure de trafic, et (5) Objet. Cette ontologie est définit pour permettre une indentification, définition et

caractérisation systématique de toutes les structures de la route, décores et scénarios que peut rencontrer un CPS véhiculaire pour un contexte opérationnelle donné.

La méthode d'identification et définition des scénarios se fait en 5 étapes, suivant les 5 niveaux de l'ontologies. La première étape se focalise sur la définition de l'espace des possibles pour les scénarios opérationnels, en spécifiant les caractéristiques des cas d'utilisation. Ensuite, la deuxième étape permet de définir les variations de l'environnement (période de la journée et météo) dans lequel peut se produire un scénario. A partir de la troisième étape, la méthode permet à équipe de conception de commencer à identifier les variations possibles des typologies de structure de route que le véhicule peut rencontrer. Cette étape assure de caractériser ces différentes structures. Au niveau de la quatrième étape, on diversifie encore plus les décors de scénarios possibles en ajoutant la couche de l'infrastructure du trafic (signalisations et marquages) à toutes les variations de structures de routes identifiées à l'étape précédente. Pour finir, la cinquième étape propose à l'équipe de conception de peupler les différents décors identifiés par des participants et d'identifier des situations opérationnelles que peut rencontrer le CPS véhiculaire.

Le première objectif de recherche est définit comme « la nécessité de nouvelles méthodes pour explorer systématiquement le domaine opérationnel des CPS véhiculaires afin de définir exhaustivement les exigences du système dans la phase de conception initiale et de concevoir l'architecture du système ». Pour y, répondre, une seconde question de recherche a été définie se focalisant sur « Comment concevoir et modéliser l'architecture d'un CPS véhiculaires en fonction du contexte opérationnel et des scénarios opérationnels définis ? ». Un seconde littérature extensive est présentée au chapitre 4 sur les méthodes de conception d'architecture des CPS véhiculaires. On y retrouve une vision de la conception des Systèmes Cyber-Physiques comme étant non limitée à la conception séparée des composants physiques et composants calculatoires, mais aussi à la conception des processus combinés qui produisent le comportement complexe du système. L'état de l'art montre qu'il n'existe pas de méthode permettant de lier la conception et modélisation des comportements des CPS véhiculaires et leurs architectures à leurs contextes opérationnels.

Ainsi, pour répondre à la seconde question de recherche, nous proposons une seconde méthode pour concevoir les architectures systèmes, se basant sur les résultats de la méthode

d'identification et définition systématique des scénarios opérationnelles basées sur le contexte. La méthode présente trois étapes commençant par les scénarios opérationnels définis par l'application de la première méthode : (1) modéliser le comportement opérationnel du CPS véhiculaire par processus opérationnels, (2) modéliser des chaines fonctionnelles réalisant les processus opérationnels, (3) et définir les composantes physiques et logiques pour la réalisation des fonctions.

Durant la première étape (1), et à partir des éléments (décors, participants, manœuvres et interactions) décrivant chaque scénario opérationnel défini et modélisé, l'équipe de conception modélise les activités opérationnelles des éléments extérieurs au système (élément de la structure routière, signalisations, marquages et participants) et les échanges avec le systèmes. Ensuite, l'équipe modélise la suite d'activité opérationnelle décrivant le comportement réactionnel du système vis-à-vis des stimuli de la situation rencontrée lors du scénario. Le processus permet de décrire les manœuvres du véhicule et ses différentes capacités pour les réaliser : perception, décision, contrôle et action.

A la deuxième étape (2), l'équipe de conception utilise l'ensemble d'activités opérationnelles résultants de la première étapes pour définir les fonctions du systèmes pour les réaliser. Ainsi, ils modélisent des chaines fonctionnelles qui représentent la réalisation des processus opérationnels. Les chaines fonctionnelles sont représentées par des séquences de fonctions de systèmes reliées par des échanges fonctionnels et des ports spécifiant les interfaces internes et externes. Ainsi, la traçabilité entre fonctions définies et les éléments du contexte opérationnel est conservée au travers des processus et des scénarios opérationnels.

Pour finir, la troisième étape (3) se focalise sur la définition des composants physiques et logiques du systèmes. L'étape précédente a permis d'identifier toutes les échanges et interfaces d'une fonction du système, pouvant participer à de multiples chaines fonctionnelles et être responsable de comportements variés. Ainsi, cela permet de définir rigoureusement le bon composant hardware ou software qui réalisera ces processus physiques et logiques décrits et identifiés par les processus opérationnels. De plus, le choix des composants est justifié par la traçabilité des composants aux éléments du contexte opérationnel qui ont donné lieu aux scénarios opérationnelles et aux comportements du véhicule.

Pour valider cette méthode de conception d'architecture de CPS véhiculaire se basant sur le contexte opérationnel, nous présentons un cas d'application à la fin des chapitres 3 et 4 d'un Véhicule Autonome sur demande en quartiers périphériques. Nous illustrons l'application de la méthode d'identification et définition systématique de scénarios opérationnels et la modélisation d'architectures basées dessus. D'après les évaluations d'experts, nous avons obtenu une plus large analyse et définition des scénarios opérationnels en phase amont de la conception et tous les composants étaient rigoureusement tracés et justifiés par le contexte opérationnel du véhicule et des situations qu'il peut produire.

L'extension des ConOps par l'analyse du contexte opérationnel et l'identification et définition systématique des scénarios opérationnels ouvre la perspective d'étude pour la semi-automatisation de cette définition systématique en intégrant les concepteurs dans la boucle. De plus, il est possible d'améliorer la méthode de conception d'architecture en menant une étude supplémentaire pour intégrer des paternes d'architecture de CPS véhiculaire. Ceci permettrai une possible semi-automatisation de la modélisation des chaines fonctionnelles qui mériterai une étude supplémentaire.

Suite au diagnostic industriel et à l'analyse du contexte de la conception des CPS véhiculaires, nous avions constaté une forte incertitude sur les décisions techniques et d'importants retards sur les projets, tous deux dues à des changements tardifs au niveau du contexte opérationnel. Ainsi, le second objectif de recherche qui a été défini est « d'anticiper l'évolution nécessaire de l'architecture des véhicules autonomes lorsque son domaine opérationnel change ». Pour y répondre, nous avons étudié l'état de l'art des méthodes de propagation de changements techniques au sein des systèmes, dont le détail est présenté au chapitre 5.

La littérature sur la propagation des changements techniques est riche et de multiples méthodes basées sur les matrices et les réseaux ont été proposées. Les premières études se sont concentrées sur la propagation des changements entre les composants et leurs paramètres, comme la méthode CPM (Clarkson et al., 2004). Par la suite, d'autres études ont étendu la propagation des changements à d'autres éléments de l'architecture du système, à savoir les fonctions et les exigences fonctionnelles. Toutefois, les méthodes ne

prennent en compte que les sources de changement provenant des éléments internes à l'architecture du système. En outre, les méthodes fonctionnent en supposant que l'architecture du système reste stable pendant la propagation du changement. Cette hypothèse montre clairement que les méthodes actuelles de propagation du changement dans la littérature ne fournissent pas les moyens de propager le changement du contexte opérationnel sur l'architecture du système et d'anticiper son évolution.

Pour proposer une méthode de propagation des changements du contexte opérationnel sur l'architecture de CPS véhiculaire, nous proposons d'utiliser la traçabilité obtenue grâce aux deux méthodes proposées pour le premier objectif de cette thèse. De plus, nous avons étudié le spectre des méthodes de propagation et leurs applications suivant deux axes : le type de modèle (matriciel ou graphique) et le type d'approche (déterministe ou probabiliste). Ainsi, nous proposons une méthode de propagation en deux étapes : (1) une première étape déterministe propageant le changement du contexte opérationnel suivant un chemin de propagation déterministe utilisant la traçabilité entre les éléments du contexte opérationnel et les chaines fonctionnelles de l'architecture ; (2) et une seconde étape probabiliste pour propager les changements identifiés aux niveaux des fonctions sur les composants associés, en prenant en compte l'incertitude des effets.

La phase de propagation déterministe (1) commence par la caractérisation du changement se produisant au niveau du contexte opérationnel. Le changement du contexte opérationnel peut se manifester aux niveaux de ses éléments de trois façons : l'ajout, la suppression, ou l'altération des attributs d'un élément. Suivant la traçabilité entre les éléments de contexte et les chaines fonctionnelles établie avec la méthode de conception d'architecture, l'équipe de conception analyse les effets de ces trois types de changement sur les situations opérationnelles et par extension, sur les chaines fonctionnelles. Ainsi par une analyse détaillée au chapitre 5, l'équipe de conception peut identifier l'impact sur la définition et modélisation des fonctions du système au travers des changements de situations. Pour une fonction du système, cet impact peut se présenter sous cinq formes :

- L'altération de contraintes ou du flux de données définis sur un de ses échanges fonctionnels
- L'ajout ou la suppression d'un échange fonctionnel

- L'ajout ou la suppression d'un flux fonctionnel interne
- L'utilisation d'un flux fonctionnel existant dans une nouvelle chaine fonctionnelle
- Aucun impact

Une fois l'impact du changement du contexte opérationnel sur toutes les fonctions identifié, la phase probabiliste de la propagation est entamée. les experts des différents domaines concernés estiment la probabilité que le changement de définition d'une fonction nécessite un certain type de changement au niveau du composant qui la réalise. Ainsi, ils observent et estiment la probabilité qu'un capteur ou actionneur doit évoluer au niveau de ses performances ou de ses propriétés physiques. En ce qui concerne les composants software, on propose aux experts d'évaluer la probabilité qu'un composant logique subisse un changement au niveau de ses performances, une adaptation, une réduction ou une augmentation de ses services. Nous justifions au chapitre 5 le choix de ces types de changement par type de composant à cause de l'hétérogénéité des composants d'un Système Cyber-Physique véhiculaire et de la prise en compte de la propagation des changements d'un composant à un autre.

En effet, lorsqu'un composant subit un changement d'un certain type, cela peut nécessiter, avec une certaine probabilité, que les composants qui en dépendent doivent subir un certain type de changement à leur tour. Ainsi, il faut prendre en compte les propagations indirectes entre composants dans l'anticipation de l'impact du changement du contexte opérationnel sur l'architecture du système. Celles-ci sont incertaines, d'où la raison pour laquelle nous proposons une méthode probabiliste pour estimer ces changements. Ainsi, nous proposons un réseau bayésien dont les nœuds représentent les types de changements possibles pour chaque composant du système. Chaque type de changement (par composant) est représenté par quatre nœuds qui indiquent 4 niveaux de propagation. Les nœuds d'un niveau (i) sont la cible de liens provenant des nœuds du niveau inférieur (i-1). Ces liens représentent les probabilités qu'un type de changement en provoque un autre. Enfin, ces probabilités sont estimées par les experts des différents domaines.

Les détails du modèle proposé sont présentés dans le chapitre 5 de cette thèse. Le réseau bayésien proposé se base sur les idées et modèles proposés par Lee et Hong (2017). Pour le calcul des tables de probabilités conditionnelles de chaque nœud du réseau, nous utilisons

un modèle Noisy-Or qui considère que les causes de changement sont indépendantes. Le modèle calcule la probabilité globale de propagation de plusieurs changements provenant de plusieurs sources à base de la probabilité de propagation d'un changement provenant d'un seul composant.

Pour valider cette méthode de propagation de changement du contexte opérationnel sur l'architecture de CPS véhiculaire, nous reprenons à la fin du chapitre 5 le cas du Véhicule Autonome sur demande en quartiers périphériques dont l'architecture a été conçue par l'utilisation de la méthode de conception d'architecture basée sur le contexte opérationnel et présentée au chapitre 4. Nous illustrons l'application de la méthode en changeant certains éléments du contexte opérationnel initialement défini. Le cas d'application a mis en exergue l'intérêt de la phase probabiliste en observant des augmentations des probabilités pour certains changements qui auraient pu être omis. De plus, les résultats de l'anticipation des propagations correspondent bien aux attentes des experts.

La méthode de propagation proposée au chapitre 5 de cette thèse ouvre des perspectives pour la détection automatique des causes des changements nécessaires aux niveaux des composants. Cela permettrait d'appliquer les changements nécessaires plus rapidement et plus efficacement. Une étude supplémentaire permettrait aussi d'estimer l'effort et le coût du changement en avance de phase, ce qui donnerait aux décisionnaires des projets de développement de véhicule une meilleure qualité d'information pour planifier l'adaptation des véhicules au changement du contexte.

En guise de conclusion, cette thèse présente trois contributions dans le cadre de la conception d'architectures de Systèmes Cyber-Physiques véhiculaires : (1) une méthode d'identification et définition systématique des scénarios opérationnelles à partir du contexte opérationnel qui est représenté sous forme d'une ontologie ; (2) une méthode de conception d'architectures de CPS véhiculaires basée sur le contexte opérationnel ; (3) une méthode de propagation du changement de contexte opérationnel sur l'architecture de CPS véhiculaire. Ces trois méthodes apportent respectivement (1) une extension de l'analyse de ConOps par la considération du contexte opérationnel et son impact sur le comportement du système ; (2) une traçabilité et une justification des choix techniques vis-à-vis du comportement attendu du CPS véhiculaire au sein d'un contexte opérationnel défini ; (3)

et anticipation de l'évolution de l'architecture du système lors du changement du contexte opérationnel avec une estimation probabiliste des changements nécessaires.

Les résultats de ces trois travaux de recherches contribuent à l'ingénierie et la conception des systèmes complexes qui évoluent dans un contexte fortement dynamique et incontrôlé. De même la sureté de fonctionnement des systèmes peut être mieux appréhendée dans les contextes incontrôlés par l'extension de l'analyse ConOps que nous proposons. De plus, ces travaux contribuent à la R&D externalisée des Véhicules Autonomes en amenant (1) une meilleure justification des choix pour mieux satisfaire les demandes des clients ; (2) une accélération de l'adaptation au changement demandé par les clients et éviter les délais de livraison.

Cette thèse ouvre la voie à plusieurs études dans les domaines de recherche de la gestion des exigences, de la conception collaborative de véhicules autonomes, et de l'impact du véhicule autonome sur les usagers. Dans le domaine de la gestion des exigences, des études supplémentaires permettraient d'aboutir à une élicitation des exigences du système basée sur le contexte opérationnel et d'adresser les futurs standards de la conception des véhicules autonomes. De plus, l'utilisation de paternes d'architectures liés au contexte opérationnel et aux comportements adéquats peut ouvrir la voie au recyclage des exigences pour accélérer les processus d'élicitation. Les travaux présentés dans cette thèse peuvent aussi être enrichis par d'autres études dans le but de construire une plateforme de conception collaborative des véhicules autonomes. Cette plateforme pourrait intégrer les clients, les ingénieurs systèmes, les architectes, les développeurs et les testeurs et proposer une semi-automatisation de la définition des architectures basée sur le contexte opérationnel. Elle permettrait de gérer plus facilement la propagation de changement sur les architectures. Pour finir, les travaux présentés peuvent être combinés au travers d'études supplémentaires à d'autres axes de recherche tels que les approches de mobilité en tant que service et la conception durable, et ce dans le but d'une meilleure conception des architectures de véhicules autonomes.

Acknowledgement

I wish to express my gratitude to a number of people who supported the research presented in this manuscript. Firstly, I would like to thank my advisor and mentor, Marija Jankovic, for supervising this research work and trusting me with its completion. I would also like to express my gratitude to my co-supervisor, Yann Leroy, for his precious advice on how to perform the research studies and the validate my proposals, Guillaume Trehard for providing crucial industrial insight to the research, and Bernard Yannou for making this project possible. I would also like to thank Claudia Eckert and Eric Bonjour for proofreading this manuscript, sharing their deep knowledge of Engineering Design, and providing tremendous advices to improve its quality.

I would like to acknowledge Guy André Boy, Andreas Makoto-Hein, Lara Qasim, Naouress Fatfouta, and Ilia Iuskevich for sharing their knowledge and discussing the research work on several occasions. I would also like to acknowledge Sarra Fakhfakh for sharing the joy and pain of digging into this research's contributions and challenging my ideas.

I kindly acknowledge the support, inputs, and feedback received from my coworkers, from AKKA Technologies, Elie Kahale, Alan Ali, Svetlana Dicheva, Victor Talpart, Luis Roldao, and Alexander Lepoutre.

Finally, I want to thank my family for always supporting my choices and giving me the opportunity to pursue my education, leading to this PhD. I want to give a special thanks to my wonderful wife, Rayen, who has always and unconditionally supported me until the end of this adventure

Table of Content

List of tables

List of figures

List of abbreviations

AD	Autonomous Driving
AV	Autonomous Vehicle
BN	Bayesian Network
BPMN	Business Process Modeling and Notations
CAS	Context-Aware System
CPS	Cyber Physical Systems
CA-V-CPS	Context-Aware Vehicular Cyber-Physical System
CONOPS	Concept of Operations
CP-BN	Change Propagation Bayesian Network
CPM	Change Propagation Method
DAS	Driving Assistance Systems
DMM	Domain Mapping Matrix
DSM	Design Structure Matrix
EC	Engineering Change
ERTRAC	European Road Transport Research Advisory Council
FC	Functional Chain
FE	Functional Exchange
MBSE	Model-Based System Engineering
OC	Operational Context
ODD	Operational Design Domain
OWL	Web Ontology Language
R&D	Research & Development
SAE	Society of Automotive Engineers
ToC	Type of Change

Foreword

This PhD dissertation results from a collaboration between CentraleSupélec and AKKA Technologies under the CIFRE (Conventions Industrielles de Formation par la Recherche) contract n° 2016/1409 between January 2017 and June 2020. It is partially supported by the French National Association for Research & Technology (ANRT). Chapter 1 introduces the research context and methodology, followed by chapter 2 for an overview of the contributions presented in four parts. The following chapters, from three to five, are scientific papers published or submitted for publication in international journals and conferences. Finally, chapter 6 concludes the dissertation with a discussion on the limits of the overall research contributions and further required research.

The following papers included in the dissertation have been published:

- Damak, Y., Jankovic, M., Leroy, Y., Chelbi, K., 2019. A Semi-automated Requirements Reuse and Recycling Process for Autonomous Transportation Systems R&D, in: Proceedings of the Design Society: International Conference on Engineering Design. pp. 3551–3560. https://doi.org/10.1017/dsi.2019.362

The following papers included in the dissertation has been accepted for publication:

- Damak, Y., Leroy, Y., Trehard, G., Jankovic, M., 2020. Operational Context-Based Design Method of Autonomous Vehicles Architectures, in: Proceedings of the System of Systems Engineering Conference (SoSE 2020)

The following paper included in the dissertation is a working paper to be submitted for a journal publication

- Damak, Y., Leroy, Y., Trehard, G., Jankovic, M., 2020., A Context Ontology Supported Identification of Operational Scenarios for Autonomous Vehicles in Early Design Phases
- Damak, Y., Leroy, Y., Trehard, G., Jankovic, M., 2020. Operational Context Change Propagation Prediction on Autonomous Vehicles Architectures

The dissertation by paper style may produce some repetitions between the Chapter 3, 4, and 5. The chapter "References" includes the dissertation's overall references. However, references used in the papers are also added to the end of the chapter.

1 Introduction

1.1 Context

Autonomous Vehicles (AVs) are becoming an unavoidable part of future mobility and transportation systems. The belief that intelligent transportation systems is the cornerstone of future mobility is getting stronger, and AVs are considered a mandatory part for such systems to exist. In recent years, AV experimentation projects and development challenges are increasing in number and frequency and will continue to evolve. Vehicles manufacturers and mobility stakeholders are accelerating their Research & Development (R&D) to develop the winning concept and be the first for AV industrialization and exploitation. As such, it is estimated that the overall investment on their development in 2017 topped the 80 Billion Dollar (Kerry and Karsten, 2017)

1.1.1 Autonomous Vehicles and Vehicular Cyber-Physical Systems

AVs are vehicular systems that perform some or all of the driving tasks autonomously. They differ from classical vehicles in many aspects related to business, technology, design and operations. This research work focuses on design challenges related to the (1) operations, (2) technology, (3) design differences.

1. Operational Differences:

AVs differs from classical vehicles on the **operational** level by introducing various levels of automation to the driving task. The Society of Automotive Engineers (SAE) defines five automation levels for AV. Classical vehicles are considered by the SAE standard as the level 0 of automation. Up to the second level, the driver is still fully responsible for the driving tasks but handles only the main ones while the automation improves the driving safety and efficiency. The first and second levels, named "driver assistance" and "partial automation", require the driver to keep control over the vehicle. With the third level, "conditional automation", the driver is partly responsible, is not required to monitor the environment at all time and might not handle any driving task during a small time. However, they must be ready to take over at any time if the automation conditions are not

met. As for the fourth and fifth levels, the driving responsibilities are transferred to the vehicles, in specific driving mode and operational conditions for the fourth level, and every situation in the fifth level. As such, the fourth and fifth levels are respectively named: "high automation" and "full automation" (SAE, 2018). This research work focusses on the design of the higher levels.

2. Technological Differences

The higher automation levels of AVs introduce **technological** novelties onto classical and lower automation level vehicles. the AV performs the automated driving tasks with high perception and cognitive capabilities, resulting in complex behavior (Wachenfeld et al., 2016). AVs need to perceive their environment with data acquisition and state estimation functions realized by sensors, such as LIDARs and cameras, and environmental data analysis software. The vehicle have to makes sense of the perceived environment to support the decision making of the vehicle. Decision and planning modules are designed and developed with complex software components to determine and plan the AV's operational behavior and maneuvers. The decisions are then transformed into command with control algorithms to be executed by the vehicle's actuations. Figure 1.1 shows the data flows between the vehicles functionalities and position them on their contribution to with regard to SAE level of automation.

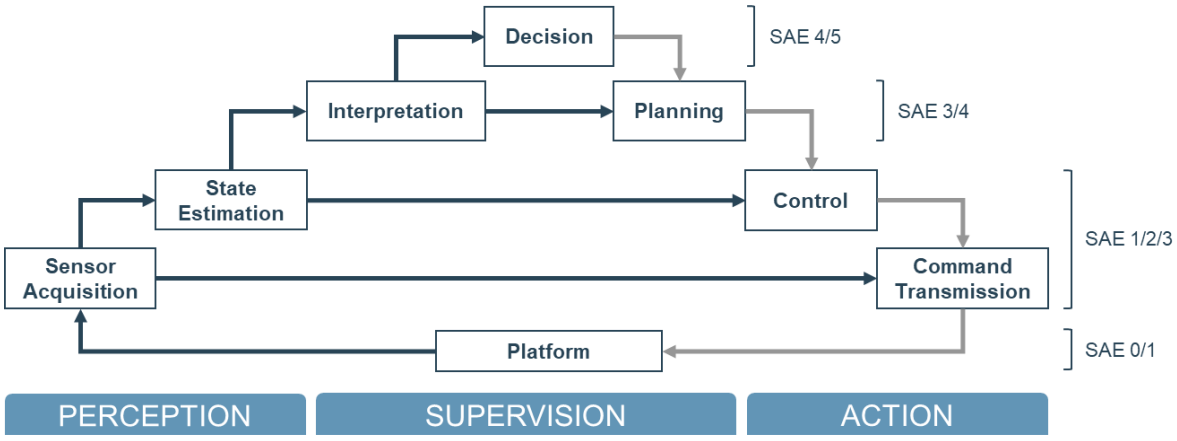


Figure 1.1: Autonomous Vehicles capabilities and automation levels

These presented AV capabilities defines them as Context-Aware Systems (CAS), as proposed by Dey (Dey, 2001): "Systems that use context to provide relevant information

and/or services to the user, where relevancy depends on the user's task". Additionally, AVs are also considered as Cyber-Physical Systems (CPS) as they integrate computational and physical capabilities to interact with their environment and act according to the state of their Operational Context (OC) (Baheti and Gill, 2011). Hence, Figure 1.2 illustrates a characterization of AVs as increasingly autonomous Context-Aware Vehicular-Cyber-Physical Systems (CA-V-CPS). Both CPS and CAS are systems highly sensitive to their Operational Context and must be designed to fit for it (Dey, 2001; Horvath, 2012).

Autonomous, Cyber-Physical and context-aware are not only limited to Autonomous Vehicles, as commonly referring to vehicles transporting peoples and goods in public road environment. They also extend to vehicles in other contexts and usages such as warehouse, ports, and airports. This research work addresses the class of systems covering the previously mentioned examples and called Vehicular Cyber-Physical Systems (Vehicular CPS).

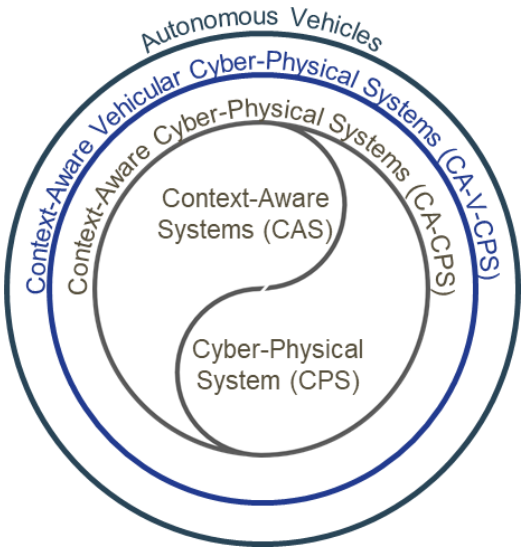


Figure 1.2: Characterization of Autonomous Vehicles

Therefore, on the **operational** and **technological** dimension levels, the purpose of a Vehicular CPS differs from the classical vehicle one. It can be defined as carrying out mobility missions while respecting the traffic rules and ensuring the integrity of the passengers and the traffic in any situation.

Furthermore, the Operational Contexts of vehicular systems is often highly varied, dynamic, and uncontrolled. For instance, the context contains objects of different types and natures such as time traffic signs, pedestrian, other vehicles, and the weather elements. Besides, its structure and composition rapidly evolve with new sceneries and objects during the vehicle's operations and there is no mean to exactly predict the new state of the Operational Context. Hence, the particularity and complexity of Vehicular CPS purpose makes it a system operating in a highly dynamic and uncontrolled Operational Context, while ensuring high levels of safety.

3. Design Differences

For all these reasons, the **design** of Vehicular CPS presents new challenges compared to classical vehicles. Due to the lack of industrial feedback and knowledge of automation technologies in the context of vehicles, the **design** approaches currently adopted for Vehicular CPS differs from classical vehicles. Mobility stakeholders address the Vehicular CPS development through multiple stages of experimentations. For instance, Google, Uber, car manufacturers (PSA, Renault, Tesla, etc.), and Operators (Transdev, RATP) experimented on Vehicular CPS in controlled tracks and cities sections with safety drivers and setups. They increase the complexity of the experimentations by integrating more complex elements to the Operational Design Domain of the vehicles. The concept of Operational Design Domain (ODD) is introduced by the SAE and defined as "the specific conditions under which a given driving automation system or feature thereof is designed to function". For example, the experimentations start with controlled tracks to evolve into simple uncontrolled tracks, and progressively integrate roundabouts and traffic lights sections.

The ODD designates the Operational Context for which the Vehicular CPS is designed to function (SAE, 2018). While the SAE proposes ODD, the term Operational Context (OC) has been used and defined in the literature for a longer period (Dey, 2001). As such, this thesis focusses on the terminology of OC, which can describe the Vehicular CPS environment at a specific time, as well as the overall ODD.

In summary, advances in **technology** permit to develop vehicles with increasing automation capabilities. These capabilities allow the Vehicular CPS to **operate** within its dynamic and complex Operational Context and behave accordingly, which was previously done by the driver. As such, the **architecting** of Vehicular CPS is not only challenging from a technological perspective, but also in considering the Operational Context. Besides, the progressive experimentation strategies for their **development** show that the system architectures evolve with respect to the increasing complexity of the Operational Context. As such, this thesis aims to address Operational Context-based architecting of Vehicular CPS.

1.1.2 The Outsourcing of Vehicular CPS Research & Development

This research work was conducted within AKKA Technologies, a technology consulting company. More specifically, it was conducted within the team addressing system automation Research and Development (R&D) projects for the company's clients: The Autonomous Systems team. The role of such team is to help clients, such as car manufacturers, developing technical solutions for their new development needs.

In the context of Vehicular CPS technology consulting companies such as AKKA Technologies provide a new range of skills relative to CPS design, such as environment sensing and robotics. These skills were not needed for the design of classical vehicles, hence historically outside of vehicle makers skillset. The technology complexity, the business risks and the field of possible are so wide that it is difficult for classic players to hold all competences and experience internally. Moreover, the diversity and multiplicity of consulting companies projects and industries provide them with technological feedback and knowledge transposable to the domain of Vehicular CPS.

During outsourced Vehicular CPS R&D and experimentation projects, clients provide vehicular platforms able to integrate new automation technologies, from the sensors, to the control software and actuation. Technology consulting companies assist their clients into designing the automation system and its architecture interfaced with the vehicular platform. However, the clients, vehicle maker and mobility stakeholder, do not directly

address technical and component specification and focus on defining the operational objective of the system.

Therefore, the challenge for the consulting company and its design team is to translate the operational objectives of the system into a technical architecture. In concreate terms, the main objective of these projects is for the vehicle to exhibit the expected reactive behavior in the right operational situation. As such, one of the success criteria of an AV experimentation is the suitability of AV design to its Operational Context.

This research work aims to assist the design team of technology consulting companies address the challenges of outsourced Vehicular CPS R&D.

1.2 Problem Statement

In outsourced AV's R&D, R&D projects often do not start with exhaustive specification and sufficient feedback on the technology. Concept of Operations (ConOps) and Scenario-based design approaches are classical approaches to address this type of projects by analyzing the system's operational activities, scenarios and modes (nominal and failures) (Fairley and Thayer, 1997; Rosson and Carroll, 2009; Sutcliffe, 2003). These operational analyses aim at specifying the system requirement based on the system's operations. ConOps with a scenario-based design approach are often adopted by Vehicular CPS design teams to define the expected behavior of the system during operational situations. The behavior is used as input to specify the functions and components of the systems in order to realize it (Höfer and Herrmann, 2017; Sippl et al., 2019)

However, classical approaches do not analyze the elements of the Operational Context and their dynamics. Due to the high dynamic, complexity and uncontrollability of the Vehicular CPS Operational Context, classical approaches do not guaranty the identification of the operational situations resulting from the various layout of the context elements. In the context of Vehicular CPS development, missing important operational scenario during the operational analysis may lead to the failure of the design in performing its mobility mission. Therefore, to address the design of Vehicular CPS, classical approach for operational

analysis need to be augmented to include an analysis of the Operational Context and its effect of the operational situations.

The limits of classical approaches have been observed by the author of this thesis during industrial observations. The author spent half of the research project time with the Autonomous Systems team of AKKA Technologies participating to Vehicular CPS R&D projects and observing the challenges of the design process. The first noted challenge was that the design team often could not define system requirements without linking them to the Operational Context elements and were often unable to identify all the important operational scenario in the early design phase. Consequently, the Autonomous Systems team could not easily define the terms of the contract with their client without agreeing on the systems requirement for the projects.

Furthermore, the author observed considerable uncertainty over the technical choices and decision. As the system architecture highly depends on the Operational Context, late change to its elements often caused important changes to the system's components and architecture. Without a precise and formal mapping between the Vehicular CPS architecture elements and the Operational Context, the design team loses considerable time (two to three days) in the identification of the impact of changes to the Operational Context on the different parts of the system, and how it propagates. As such, important project delays resulted from late changes to the Operational Context, often over 150% of the initial estimated time.

Therefore, to support the outsourced R&D of Vehicular CPS, there is a challenge of a systematic exploration of the Operational Context to specify the system requirement in early design phase and design its architecture based on the context elements. Besides, there another challenge is to anticipate the evolution of the Vehicular CPS architecture and the necessary changes when the Operational Context changes.

Previously discussed challenges underline the need for methods and tools to systematically analyze the Operational Context in order to support Vehicular CPS architecting. The current literature, detailed in later chapters, does not provide clear methods to use the Operational Context for Vehicular CPS architecting and to map it with the architecture's

elements. This research aims at providing models, methods, and tools to support the outsourced architecting activities of Vehicular CPS and linking its architecture to the Operational Context.

1.3 Research Methodology

1.3.1 Research Approach

Research studies are generally conducted following a deductive or inductive research approach depending on the research problem and context (Saunders, 2011). Deductive researches focus on the development of a theory prior to the collection of data. The theory is then tested and challenged with the examination of the hypothesis that confirms or rejects the theory. Inductive research, on the other hand, starts with the observation and analysis of collected data to develop a theory. This approach is adequate for research problems identified in the industry, as it relies on the examination and analysis of the real-word situations to contribute to the domain knowledge (Saunders, 2011).

After selecting a research approach, the researcher must design a research strategy to attain the studies objectives. This thesis falls in the domain of engineering design research, whose general goal is to understand, describe, prescribe and support the design process of industrial companies. An inductive approach is appropriate to address engineering design problems (Eckert et al., 2003). Next to the observations and problems identification, various types of research contributions are prescribed to solve the problems such as models, methods, and design tools (Blessing and Chakrabarti, 2009). Various design methodologies can be found in the literature, corresponding to the different knowledge domains and their characteristic. This thesis follows the Eightfold Path strategy, an adequate research strategies to tackle engineering design problems (Eckert et al., 2003).

1.3.2 Research Phases

The Eightfold Path strategy proposes an eight steps methodological framework to conduct research projects in engineering design. Each step output corresponds to the input of the

following step, forming a spiral. The eight phases of the methodology are as follows (Eckert et al., 2003):

- **Empirical studies of design behavior:** in this phase, the researchers conduct an empirical study with observation and interviews of designers and engineering teams to characterize the existing situation.
- **Evaluation of empirical studies:** This includes assessing the validity of the previous empirical study;
- **Development of theory:** With the results of the empirical study, the researchers build an understanding of the design practice. It can take the form of theories of design aspects or a local understanding of types of design activities.
- **Evaluation of theory:** This includes assessing the validity of the theory by comparison to existing empirical data and their grounding in the theoretical framework.
- **Development of tools and procedures:** This includes design activities of methods, tools, and procedures to support the activities of designers and engineers.
- **Evaluation of tools and procedures:** In this phase, the researchers validate the proposed tools and methodologies with iterative prototyping and testing activities.
- **Introduction of tools and procedures:** This includes the dissemination and introduction of the tool and design practice to an industrial environment and studying the consequence of this change.
- **Evaluation of dissemination:** This includes the assessment of the dissemination results validity and how they benefit the overall understanding of the design practice.

A research project does not necessarily go through all the phases of the methodology. It is more important that the researchers are aware of the underlying hypothesis they are making, and the methodology helps to frame their study (Eckert et al., 2003). As such, the research project activities should be selected to help to attain the study's objectives, as well as to ensure the validation of the results.

In this thesis, we followed the Eightfold Path on the overall research project. Figure 1.3 describes the layout of the studies conducted in this thesis. The research project started with an empirical study on the behaviors during Vehicular CPS design in outsources R&D, conducted following the action research methodology by integrating an engineering team, participating to projects, and analyzing the project documentation. The study resulted in a detailed analysis of the design process and an analysis of its issues. After the process validation in workshops with the engineers, a local understanding of autonomous vehicle design was developed. The three main contributions (chapters 3, 4, and 5) are individual studies based on the results of the previous studies and focus on the two phases of development and evaluation of models, tools, and procedures.

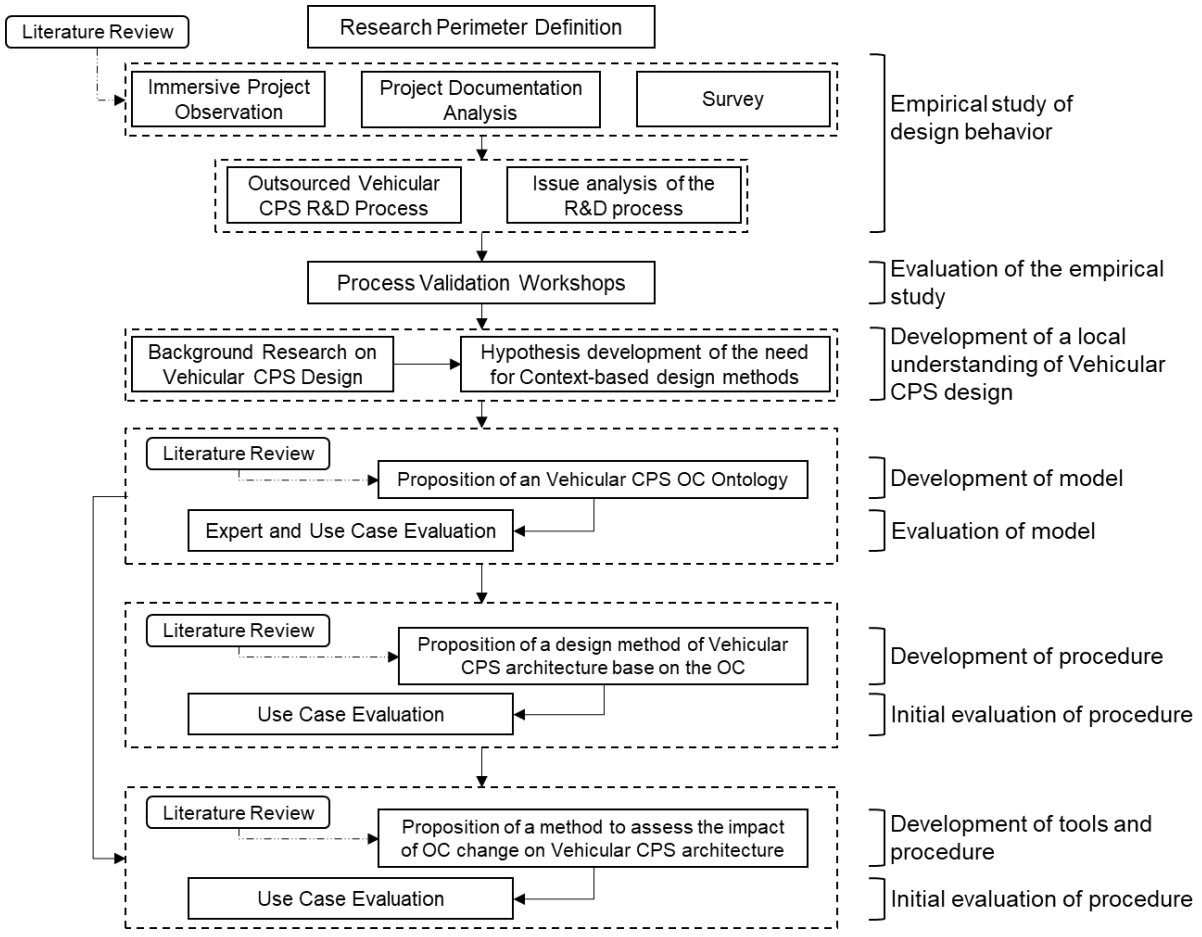


Figure 1.3: Overall research methodology based on the Eightfold Path strategy

2 Research Overview

This section introduces an overview of the research presented in this thesis. It presents the industrial background and summary of the background literature leading to the definition of this research's objectives and questions. An overview of the contributions of the thesis are presented at the end of this section.

2.1 Industrial Observations

The author of this thesis integrated the Autonomous Systems team of AKKA Technologies, a French engineering consulting company, as a system design engineer to conduct an industrial diagnosis of Vehicular CPS design. The author spent half of the research project within the industrial team. The team was composed of a team leader, three senior researchers in robotics and system automation, one senior researcher in applied mathematics, and twelve young engineers from multiple domains: system engineering, system automation, system and computer vision, and applied mathematics for machine learning. The team's perimeter and main job is to design automated driving systems for the AKKA's clients.

The aim of the empirical study was to understand the design process of Vehicular CPS in outsources R&D and comprehend the challenges associated with it. A particular attention was given to understand the differences between the classical vehicles and vehicular CPS, and how it impacted the outsourced design process. Following an action based research methodology, an observation protocol, illustrated in Figure 2.1, was designed with various quantitative analysis technics: data collection, direct observation, and case study analysis, semi-structured interviews, workshop, (Miles et al., 2018). As for the **data collection**, the author started with the documentation of two former projects and another two in progress. In addition to **direct observation** and **case study analysis**, he participated to the deployment of a Model-Based System Engineering (MBSE) method with the opensource software Capella from PolarSys (Roques, 2016). The method was applied on two projects on the design of platooning systems, three projects on the automation of robotized vehicles, and a study on the automation of trains.

During the first year and a half, the author conducted **semi-structured interviews** with the five senior researchers. The interviews were conducted two to three time with each person, and consisted of one hour discussion to identify the specificities and characteristics of the outsourced design process of Vehicular CPS. A second focus was to identify the new challenges they faced during the projects in comparison to design projects of classical vehicles. The author also conducted **interactive presentation of the results** and **workshops** with the seniors during the development of this research's contributions, detailed in the sections addressing the contributions. These workshops lasted an average of 2 hours with each domain expert and was aiming to validate and improve the models and method proposed.

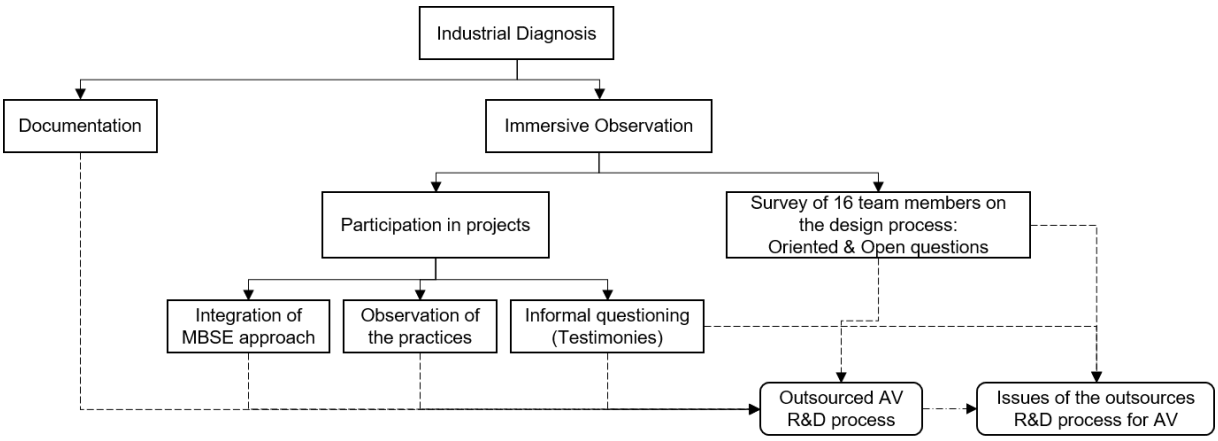


Figure 2.1: Industrial diagnosis protocol (full lines represent the composition of activities; dashed lines represents the outcomes of the activities)

The empirical study composed of the various activities presented above permitted to model the outsource design process of Vehicular CPS. Subsequently, a survey was designed to complete and fine-tune the process model. It has been answered by the 16 members of the team. The questionnaire was composed of 18 oriented and open questions, detailed in Appendix B. The analysis of the questionnaire's resulted in the design process synthesized in Figure 2.2. A detailed version of the process is proposed in Appendix B (Figure B.1 and Figure B.2). The author used the Business Process Model & Notation (BPMN) framework for the flexibility and expressiveness of its standard notations. The process describes the typical actions and exchanges between a client, the team leader, and an engineer during an outsourced R&D project on autonomous vehicles.

The first part of the process is characterized by a significant number of exchanges between the team leader and the client to improve the mutual understanding of the project's objectives. Once they agree on an initial view of the system's operations, an iterative cycle of exploration, implementation, testing and validation of technical solutions is engaged between the team leader and the engineers. The project ends with the delivery of the system and its validation by the clients.

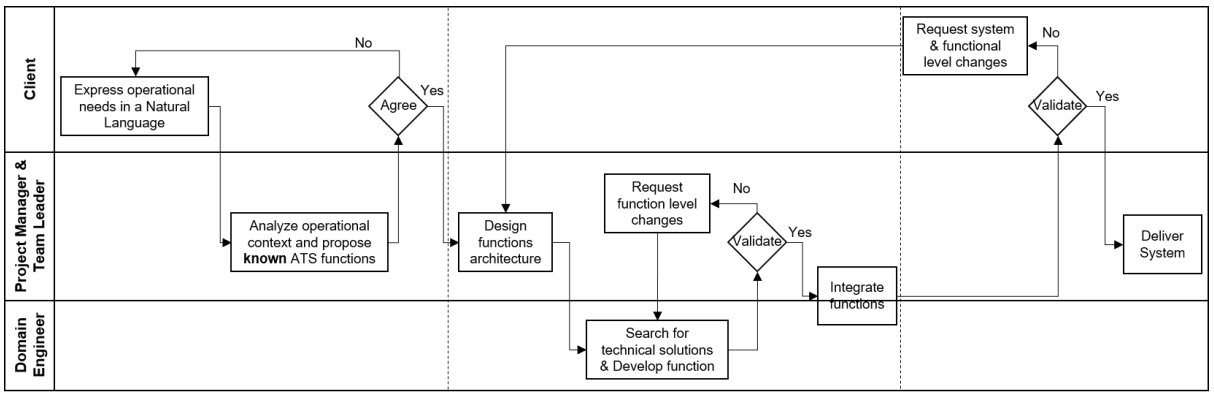


Figure 2.2: Synthesis of the design process of Autonomous Vehicles in outsourced R&D

A deep analysis of outsourced R&D process of Vehicular CPS shows that for an average of a six month project, one to two months are spend on prospecting and defining what the clients expect from the system's operational conditions and activities, prior to the start of the project. In the case of classical vehicle, the system requirements are often known and defined early by the clients, thanks to their deep knowledge and understanding of the domain and their need.

In the early phase of Vehicular CPS design projects, the system's engineers face the challenge of defining the system requirements permitting to realize the operational activities of the system. An average of a month and a half is spent on the definition of the vehicle's operational scenarios and the analysis of the system requirements. Simultaneously, the design team is pressured to start the development of the technical solutions without sufficient specifications. As a result, the design team strongly relies on their knowledge of the system's operational conditions and activities and multilevel system modeling to design the relevant computational and physical process to realize the system's operational activities. Considering the operational conditions was a necessity due to its impact on

architecture of the vehicle. Hence, the requirements and system architecture often arrive later, and multiple adjustment are needed to deliver the complete system.

In addition, during this process, the client often changes the operational conditions determine by the Operational Context of the vehicle, which have considerable impact on the technical choices and the system architecture. In the outsourced design of classical vehicles, the characteristics of the Operational Context has low effect on the system architecture, as the driver is the one responsible to deal with its dynamic. As such, this change impact is a completely new challenge faced during the design of Vehicular CPS.

The overall industrial challenges encountered during the outsourced R&D of Vehicular CPS can be summarized as follows:

- **Difficulties in defining all the important operational scenarios**: In the early design phase, due to the high complexity and dynamic of the Vehicular CPS Operational Context, the design team may sometime miss the definition of important scenario. In some cases, the missing scenarios are identified later. In worst cases, missing their definition and analysis causes system failure during tests and validation.
- **Specific difficulties in system requirements elicitation**: Without linking the requirements to the Operational Context elements, the design team often had significant difficulties in eliciting the system requirements. These difficulties result in tripling the elicitation time compared to outsourced classical vehicles projects, due the lack of formal knowledge about the link between the operational context and technological possibilities in vehicle automation.
- **Difficulties in justifying the technical choices:** Due to the low industrial feedback and the complexity to link the technical solutions to the operational activities, the design team encounter significant difficulties in justifying the decision to the clients.
- **Considerable uncertainty over technical decisions**: Due to frequent late changes to the Operational Context, the technical solutions were often subject to rework.

- **Important project delays**: Due to late changes to the Operational Context and considerable difficulties of evaluating the changes impacts on the architecture, the projects were very often delayed up to 150% of the initially defined time.

Two types of challenges have been identified important by the team's experts: challenges related to architecting activities and challenges focusing on the system requirements elicitation. They were considered primary to address in order to enhance Vehicular CPS design. As such, the research work presented in this manuscript aims at addressing the challenges of Vehicular CPS architecting base on the Operational Context. The following chapters derive the research objectives and present the contribution addressing these challenges.

As for the challenges focusing of the system requirements, they have been initially explored during this PhD with a study on the reuse and recycling of requirements based on the operational context, given in **Appendix A** (Damak et al., 2019). The study notably brought to light the importance of representing the complex operational context of Vehicular CPS and taking it into account in the design process.

2.2 Background Literature

In order to address the issues unveiled in the industrial audit, it is necessary to identify the general design challenges of Vehicular CPS and how it relates to the issues identified in the context of outsourced R&D. According to the European Road Transport Research Advisory Council (ERTRAC), the key challenges for automated driving design and deployment are divided in three types: vehicles & technology, system & services, and users & society challenges (ERTRAC, 2019). This thesis focuses on the challenges related to the design activities of the Vehicular CPS.

Vehicular CPS are novel systems differing from the classical vehicles. Their purpose is to carry out mobility missions while respecting the traffic rules and ensuring the integrity of the passengers and the traffic in any situation. As such, they are complex systems featuring a great number of heterogeneous components (software, sensors, actuators, mechanical components, etc.) and interactions realizing physical and computational processes to

produce complex operational behaviors (Baheti and Gill, 2011; Behere and Törngren, 2016; SAE, 2018). During the design of such systems, companies face the challenges of integrating new technologies and infrastructure (connectivity, perception, artificial intelligence, and big data), considering the human factor, validating the systems, ensuring the users safety and cybersecurity, proposing new mobility services, and gaining the users and societal acceptance (Dokic et al., 2015; ERTRAC, 2019).

These challenges can be further traced to the basic characteristics of Vehicular CPS. These systems exhibit adaptive operational behaviors during their interactions with their operational environment. They exhibit these behaviors through physical and computational processes realized with complex perceptive, cognitive and computational capabilities (SAE, 2018; Wachenfeld et al., 2016). Vehicular CPS are by definition Cyber-Physical Systems (CPS), but also Context-Aware Systems (CAS) (Baheti and Gill, 2011; Dey, 2001; Horvath, 2012). The architectures of Vehicular CPS are designed to perform the right behavior in the all possible traffic situation they encounter. These operational situations are defined by the layout of the Operational Context elements present at the scene, as well as their dynamic (Bach et al., 2016; Ulbrich et al., 2015). As such, Vehicular CPS architecture are highly dependent to their Operational Context (Bach et al., 2016; Behere and Törngren, 2016; Sippl et al., 2019). As such, these system architecture must be designed to be adapted to their Operational Context (Dey, 2001; Horvath, 2012).

To achieve the system's fitness to the Operational Context, multiple studies agree that the Concept of Operations (ConOps) with scenario-based design approaches provide adequate frameworks (Bagschik et al., 2018; Rosson and Carroll, 2009; Schuldt et al., 2018; Sippl et al., 2019; Ulbrich et al., 2015; Wachenfeld et al., 2016). However, these approaches do not analyze and characterize the Operational Context elements and their dynamic. The Operational Context of Vehicular CPS is composed of multiple heterogeneous, dynamic, and uncontrolled elements such as road structure, traffic signs and marking, obstacles, and other vehicles. As the operational situations occur from the complex layouts, characteristics, and dynamics of the Operational Context, classical approaches do not guaranty the identification of all important operational situations resulting from their various layout (Bagschik et al., 2018). As Vehicular CPS physical and computational processes are highly

dependent to the Operational Context, missing important operational situations during the operational analysis may lead to the failure of the design in performing its mobility mission.

Due to high dynamic, complexity and uncontrollability of the Operational Context, there is a need of extending classical ConOps approaches by a systematic analysis and characterization of the Operational Context to define the operational scenarios in the early design phase of Vehicular CPS. This research work addresses various aspects of Vehicular CPS architecting activities bases on the Operations Context. To identify the respective research gaps, an extensive literature review is proposed for each specific contribution in Chapters 3, 4, and 5.

2.3 Research Objectives and Research Questions

The aim of this research is to support Vehicular CPS architecting by considering the dependence of the system architectures to their Operational Context. The industrial diagnosis emphasized the need to support the identification of the Vehicular CPS's expected behavior during its interaction with its Operational Context and the underlying system requirements. This highlights the necessity of expending the ConOps approach by analyzing and characterizing the Operational Context for the design of Vehicular CPS

As the context of Vehicular CPS is highly complex and dynamic, there is a need for a systematic analysis of the Operational Context in order to define and cover all the important operational situations. We have already identified that exhaustively identifying the Operational Context is mandatory to design vehicles capable of exhibiting the right behaviors during all traffic situations. Hence, first research objective is:

RO1: To analyze Vehicular CPS Operational Context and systematically explore their operational domain in order to define their system architecture in the early design phase.

Achieving **RO1** requires identifying and structuring the different elements of the Operational Context. It is also necessary to understand how the context element helps describe the scenes and the overall scenarios of an Vehicular CPS. The operational scenarios

represent the medium to identify operational situations encountered by the vehicle. Hence, one can identify the related research question:

RQ1: How to systematically define operational scenarios based on the Operational Context in early design phase?

Furthermore, **RO1** requires defining an architecting method for Vehicular CPS based on the result of the systematic identification of operational scenarios and the Operational Context. This is a critical challenge as the industrial diagnoses showed a need to justify the technical solutions during Vehicular CPS design through a robust traceability between the Operational Context and the vehicle's architecture. As presented in section 2.2, the contribution must help the design team to model the computational and physical process realizing the complex behavior of the vehicles in response to its dynamic operational context. As such, a second research question was formulated to complete **RO1**:

RQ2: How to design and model Vehicular CPS architecture based on the Operational Context and the defined operational scenarios?

Another important challenge raised during the industrial observation focused on the importance of evaluating the impact of late Operation Context onto the Vehicular CPS architecture and how the architectures evolves. As such, the author defined a second research objective as follows:

RO2: To anticipate the necessary evolution of the Vehicular CPS architecture when its operational domain changes.

Achieving **RO2** requires understanding how the operational context affects the element of the Vehicular CPS architectures. It is also necessary to identify and capture how the changes of the Operational Context elements would propagate onto the operation, functional, logical, and physical levels of the architecture. Thus, the following research question:

RQ3: How to evaluate the Vehicular CPS architecture evolution when the Operational Context changes?

2.4 Identified Research Gaps

To address the three research question defined in the previous section, extensive literature reviews were conducted to identify the research gaps. These literature reviews are detailed with each specific contribution in Chapters 3, 4, and 5.

To address **RQ1**, an extensive literature review on Vehicular CPS Operational Context and operational scenario modeling presented in section 3.2 shows that there is no method to systematically identify and define operational scenario variations based on the Operational Context in the early design phase of Vehicular CPS. Besides, to the best of the author's knowledge, the literature does not cover a formal modeling and representation of Vehicular CPS Operational Context permitting to extend the ConOps with a systematic scenario definition.

A second literature review on Vehicular CPS architecting method presented in section 4.2 focus on **RQ2**. It underlines the lack of a method to specify and model the Vehicular CPS behavior and design the system architecture based on the Operational Context. Besides, the literature shows that the ConOps and the system behavior models are rarely linked to the Context elements

Finally, to address the third research question **RQ3**, an extensive literature review on engineering change propagation onto system architecture is proposed in section 5.2. Algthough there is an extensive literature on change propagation, it shows that no change propagation method permitting to assess the impact of Operational Context changes onto Vehicular CPS architecture

2.5 Résumé of Research Contributions

This section introduces the contributions of the thesis with regards to the research question. With regards to **RQ1**, chapter 1 proposes an Ontology to characterize the Operational Context of Vehicular CPS and support a systematic identification and definition method of operational scenarios in the early design phase. This ontology helps Vehicular CPS design teams to define and characterize the Operational Context for which the vehicle is designed

to operate by selecting relevant ontology elements and defining the ranges of their attributes. The identification method follows five steps corresponding to the five-level structure of the Operational Context Ontology: (0) Use case, (1) Environment, (2) Road Infrastructure, (3) Traffic infrastructure, and (4) Traffic Objects.

The ontology is designed to support reasoning on operational scenarios and operational situations encountered by the vehicle. The five levels define the layers of scenario description. The first two layers give a general context to the scenario, followed by the third and fourth layer setting the scenery. The scenario's dynamic aspect is added in the fifth layer with the definition of the actors, their actions, their positions, their interactions, and their evolution.

RQ2 is addressed with the paper on the design method of Autonomous Vehicles architectures based on its Operational Context (chapter 4). The ontology defined in chapter 3 is used to define the Operational Context and operational situations encountered in this context. Once characterized, they bring to light the behavior expected from the vehicle in reaction to the situations. This information helps the design team to model the vehicle's behavior as operational processes (see figure 2.3). The method proposes to derive Functional Chains (FCs), a sequence of functions and functional interactions, from the operational process and support the traceability between the Functional Chains and the elements of the Operational Context. Finally, logical and physical components are defined to realize the functions modeled through the Functional Chains.

With regard to **RQ3**, the journal paper, presented in chapter 5, proposes a method to assess the impact of Operational Context change propagation on Vehicular CPS architecture (Functional Chains, functions, functional interfaces, constraints, and components). Change propagation is the process where a change in one element propagates to another. It combines the direct impact of one element change on another and the combination of indirect impacts through different elements (Clarkson et al., 2004). As such, a link between the elements of the system is necessary to identify the propagation paths. The elements of the Vehicular CPS architecture model from chapter 4 are mapped to the elements of the Operational Context Ontology. Hence, they are used in chapter 5 to study Context change propagation onto Vehicular CPS architecture.

This paper proposes a method to identify and evaluate the direct impact of Context change on Vehicular architecture with a deterministic method, then estimate its indirect impact on the components with a probabilistic propagation. The change impact is characterized by Types of Changes (ToCs) required for the components to adapt. Domain experts are requested to evaluate the likelihood of direct propagation given a propagation path from an Operational Context element to a component ToC, as well as from a ToC to another. Figure 2.3 summarizes the different steps of each chapter and research objectives it helps to achieve. It details the second part of Figure 1.3 presented previously. Figure 2.3 also shows the use of some chapter's outputs as inputs for the next chapters.

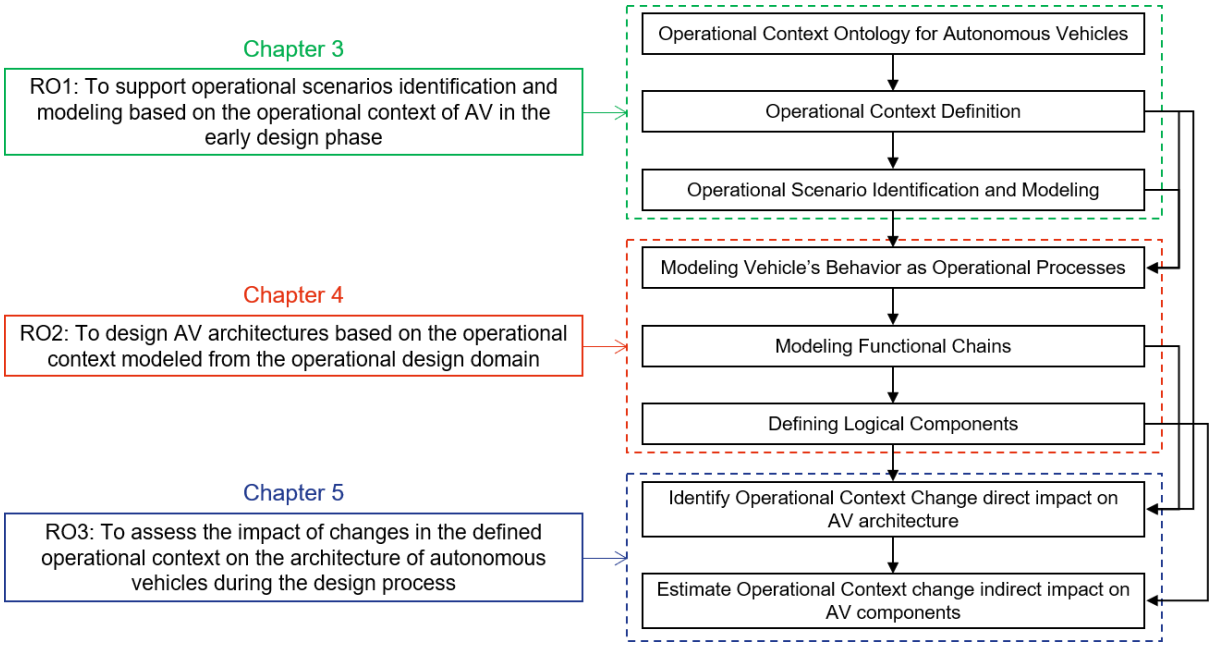


Figure 2.3: Thesis contributions layout

Case studies are presented at the end of each chapter. In chapter 3, the case study illustrates how the proposed ontology helps Autonomous Vehicles design teams identifying and modeling the various possible operational scenarios in the early design phase. The identification is based on the knowledge of a few elements from the operational design domain. Chapter 4 introduces a real case design of an Autonomous Vehicle architecture designed by AKKA's Autonomous Systems team with regards to a defined Operational Context. The proposed architecting method helped identify seven operational situations and model the appropriate operational processes, functional chains, and logical components. The same architecture is reused in the case study of chapter 5. A change is

applied to the defined Operational Context, and its propagation is assessed on the architecture's elements.

2.6 Thesis Structure

This thesis has a paper-based structure. Each of the three main papers is a chapter on its own and can be read independently. Each paper contains an introduction setting the research question, a literature review specific to the paper's contribution, the proposition, and a conclusion discussing the contribution. The thesis proceeds in 6 chapters and an appendix as follows:

- **Chapter 1** introduces the research project context and problem statement and presents the research methodology followed to solve it.
- **Chapter 2** presents the first descriptive study with an industrial audit and a global literature review on the research subject. The research objectives are then introduced followed by an overall layout of the thesis contribution.
- **Chapter 3** introduces paper #2 "A Context Ontology for Operational Scenarios Generation of Vehicular CPS" a working paper to be submitted to a journal. This chapter details a method to identify and define Vehicular CPS operational scenarios supported by an ontology for their Operational Context. The method and the ontology are endorsed with domain experts validation and use case application.
- **Chapter 4** introduces paper #3 "Operational Context-Based Design Method of Autonomous Vehicles Architectures" accepted in the System of System Engineering Conference (SoSE 2020). This chapter proposes a model-based system engineering (MBSE) method to support the design of AV's architecture based on its OC. It uses the ontology detailed in chapter 4. The applicability of the method is tested with a case study of an architecture development for a predefined OC.
- **Chapter 5** introduces paper #4 "Operational Context Change Propagation Prediction on Vehicular CPS Architecture" submitted to Computers in Industry. This chapter describes an Operational Context change impact assessment method onto Vehicular CPS architecture. The method evaluates the probability of Type of Changes (ToCs)

for the vehicle' components to adapt to the changed context. The method is validated with an industrial case study of Context change propagation.

- **Chapter 6** discusses the results and the limits of the research. It concludes with future research to improve Operational Context-based architecting activities.
- **Appendix A** introduces paper #1 "A semi-automated requirements reuse and recycling process for Autonomous Transportation Systems R&D" published in the proceedings of the International Conference on Engineering Design (ICED19). This paper is an initial study on the use of Operational Context to enhance the requirement elicitation process. The study proposes a process to reuse and recycle requirements from past projects defined for the same Context elements. This study concluded with the need for a more robust modeling of the Operational Context.
- **Appendix B** presents the details of the analysis of outsourced design of Vehicular CPS with a survey and a detailed model of the design process.

3 Paper #2. A Context Ontology Supported Identification of Operational Scenarios for Vehicular Cyber-Physical Systems in Early Design Phases

Youssef Damak, Yann Leroy, Guillaume Trehard, and Marija Jankovic

This paper is a working paper to be submitted.

Abstract. *Vehicular Cyber-Physical Systems (CPS) are emerging systems considered as pillar of the future mobility and major components of the smart city concept. Their design is source of many academic and industrial research efforts. Vehicular CPS execute physical and computational processes in response to their Operational Context (OC). As such, their system architecture must be fit for their Operational Context. While many studies agree that the Concept of Operations (ConOps) and scenarios-based design approaches are adequate for the design of Vehicular CPS, they do not address the analysis and characterization the Operational Context elements and their dynamic and do not guaranty the systematic definition of all important operational situations in the early design phase. To address this gap, this paper proposes a method to systematically identify and define operational scenarios supported by an ontology of the Vehicular CPS Operational Context. The ontology is structured in five levels of context elements: use case, environment, road infrastructure, traffic infrastructure, and traffic objects. The levels represent the different layers of operational scenarios modeling and are followed to identify the various sceneries and situations that can be encountered by the vehicles during its operations. A case study on Autonomous Vehicles on demand in the suburb illustrates the application of the method by identified a relevant set of operational scenarios from a few elements defined in vehicle's Operational Context.*

Keywords*. Scenario Identification; Operational Context Ontology, Vehicular Cyber-Physical Systems, Scenarios-based Design*

3.1 Introduction

Vehicular Cyber-Physical Systems (CPS) design is gaining more and more focus in recent years. Stakeholders acknowledge the importance of Vehicular CPS for the future mobility the smart city concept, which resulted in a substantial increase in Research & Development (R&D) for their experimentation and industrialization. However, the system's complexity and large field of possibilities prevent classic mobility stakeholders from holding all the competencies internally. Consequently, engineering consulting companies play an essential role in assisting mobility stakeholders in conducting Vehicular CPS R&D. However, outsourced R&D projects rarely start with exhaustive specifications and sufficient feedback on technical solutions. On the other hand, the operational needs and objectives are often clear from the early phases.

The success of a Vehicular CPS experimentation is achieved when the vehicle exhibits the right behavior with respect to the immediate Operational Context, in any situation. The right behavior contributes to the mobility mission of the vehicle while preserving the integrity of the passenger and traffic, which make the vehicle a safety-critical system. It is realized through the execution of physical and computational processes in response to the environment (Baheti and Gill, 2011; Behere and Törngren, 2016). As such, the success of a Vehicular CPS experimentation is partially achieved by designing a vehicle architecture fit to its operational design domain, i.e., the specific operational conditions under which it is designed to function (Horváth, 2014; SAE, 2018).

In the context of Vehicular CPS R&D, the classical solution-based design approaches have limited efficiency and do not permit address the high dependency of the vehicle's architecture to its Operational Context. More adequate design approaches addressing the analysis of complex system's operational domain and basing the architecture design on this analysis have been developed for decades in the aeronautic, aerospace, and defense industries. These approaches are the Concept of Operations (ConOps) and scenario-based design approaches, which focus on the analysis of the system's operational activities, scenarios and modes (nominal and failure modes) (Handbook, 2014; Rosson and Carroll, 2009). Multiple studies consider scenario-based design approaches adequate to address the

challenges of Vehicular CPS design (Bach et al., 2016; Geyer et al., 2014; Ulbrich et al., 2015).

However, these classical approaches do not address the analysis and characterization the Operational Context elements and their dynamic. For Vehicular CPS, the Operational Context is highly heterogeneous, dynamic, and uncontrolled. It is composed of multiple road structure, traffic signs and marks, environment condition, and dynamic and unpredictable elements. Due to the high dynamic, complexity and uncontrollability of the Vehicular CPS Operational Context, classical approaches do not guaranty the identification of all important operational situations resulting from their various layout. In the context of Vehicular CPS development, missing important operational situations during the operational analysis may lead to the failure of the design in performing its mobility mission. Therefore, to address the design of Vehicular CPS, classical approach for operational analysis need to be augmented to include an analysis of the Operational Context elements and its effect of the operational situations.

This paper proposes a method to systematically identify and define Vehicular CPS operational scenarios supported by an Operational Context ontology to support a scenario-based design approach for Vehicular CPS based on their operational design domain. The method aims to help design teams, rapidly identify a relevant set of operational scenarios to specify the Vehicular CPS behavior and requirements in early design phases.

The paper is structured as follows. Section 3.2 reviews the literature on AV context modeling and scenario identification methods. Section 3.3 presents the method to identify AV operational scenarios based on the proposition of an Operational Context ontology. A case study on Autonomous Vehicles on demand illustrates the method application in Section 3.4 before discussing future work perspective in Section 3.6.

3.2 Literature Review

3.2.1 Operational Context Definition

The notion "Context" has numerous definitions in system design or engineering domains. With the emerging of pervasive computing and context-awareness in application development, many early definitions were proposed through synonyms or through enumerating the different examples of context elements (Brown, 1995; Chen et al., 2003; Henricksen, 2003; Hull et al., 1997; Pascoe, 1998; Schilit et al., 1994; Schilit and Theimer, 1994). Dey (Dey, 2001) proposed an application-centric general definition of context as "*any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves*". This definition is widely used to define the context in later researches on context-based interactions.

Dey's definition was later extended to a less general and more operational one by Zimmerman et al. (Zimmermann et al., 2007). In their extension, they enclosed the definition in a formal part and an operational one. The formal extension introduces the five fundamental categories in contexts: individuality, activity, location, time, and relations. On the other hand, the operational extension deals with the dynamic of the context categories. The authors state in their definition that "the activity predominantly determines the relevancy of context elements in specific situations, and the location and time primarily drive the creation of relations between entities and enable the exchange of context information among entities". The other extension to Dey's definition was introduced by Baldauf et al. (Baldauf et al., 2007) in a survey on context-aware systems, and aims to distinguish context dimensions. They propose the external (physical) dimension of the context, referring to the context that can be measured by hardware sensors such as location, light, sound. The internal (logical) dimension concerns the user's aspects, such as his goals, tasks, emotional states.

Throughout the literature, different notions of context are studied for systems design and engineering. To identify different sources of uncertainties for system design, De Weck et al. (De Weck et al., 2007) distinguished between corporate, product, use, market, political

and cultural contexts. The corporate context was analyzed and used by Nadoverza and Kiritsis (Nadoveza and Kiritsis, 2014) to implement a dynamic activation of Enterprise application functionalities. Product context can be refined from system design and engineering context characterization. This context, as well as the political and cultural one, were often used to specify or constrain system requirements (Alshaikh and Boughton, 2009; Bubl and Balser, 2005; Nemoto et al., 2015). As for the user context, presented by De Weck et al. (De Weck et al., 2007), its perimeter is very large, and the literature shows a further sub-division of this context.

On one side, Chen (Chen et al., 2013) defines usage context for products as "all aspects describing the context of product use that varies under different use conditions and affects product performance and consumer preferences for the product attributes". The authors link product performance and customer choices to the usage context through statistical analysis. On the other side, researchers on technical engineering and development of systems use the term context, to refer to the Operational Context in which the systems operate (Crowley et al., 2002; Fuchs et al., 2008b; Sun et al., 2016). The following section focuses on the Operational Context modeling of Vehicular CPS.

3.2.2 Vehicular CPS Context Modeling

Most of the context models started being developed with the emergence of Context-Aware Systems (CAS), defined by Dey (Dey, 2001) as "*systems that use context to provide relevant information and/or services to the user, where relevancy depends on the user's task*". Consequently, several papers reviewed context representation and reasoning for CAS. They enumerate several types of representation (Baldauf et al., 2007; Khattak et al., 2014; Perttunen et al., 2009). The most used representations found in the literature are:

- Tuple-Based representations: Also referred to as key-value models, these representations model context elements in tuples, paired with their values.

- Logic-Based representations: Context is defined and extracted through formal expressions and rules. To manage these models, we use logic-based systems, such as first-order predicate logic.

- Ontological representations: These models formally describe the context's elements and their relationships. Their formal expressiveness enables common understanding and context sharing, as well as context-based reasoning. For these reasons, they are considered the most appropriate methods for context modeling. Cabrera et al. (Cabrera et al., 2017)

Vehicular CPS are vehicular systems that safely perform some or all the driving tasks. Vehicular CPS, like Autonomous Vehicles, have various degrees of automation. The highest degrees considered by the Society of Automotive Engineers (SAE) are "high automation" and "full automation". In both levels, the vehicle performs all the human driving tasks by the vehicle capabilities related to perception, cognition, decisions, and execution in specific driving mode and operational conditions (SAE, 2018; Wachenfeld et al., 2016). Consequently, and according to Dey's definition, Vehicular CPS are Context-Aware Systems.

Context modeling for vehicles started before autonomous driving with driving assistance and automation. For such systems, logic-based representations of context are proposed to implement context-dependent dynamic activations of functionality. Weiss et al. (Weiss et al., 2013) propose a simulation showing that a context-based dynamic activation of vehicle software functions decreases their number and percentage of activation. This result provides a means of optimizing the vehicle's energy consumption through context-based dynamic configurations. For another purpose, Sathyanarayana et al. (Sathyanarayana et al., 2011) propose a context and driver aware Active Vehicle Safety (AVS) system through processing raw sensors data with logic-based reasoning. Mathematical models such as Universal Background Models for context analysis and recognition are used in this research. Context-based AVS systems are an attempt at realizing a robust, human-centric, and intelligent active safety system.

Other researchers chose ontology representations of context for context sharing and real-time rule-based reasoning. Fuchs et al. (Fuchs et al., 2008a) introduce an Operational Context ontology for Driving Assistance Systems (DAS) for scene description. Its purpose is to be used by intelligent vehicles for context description and sharing with other intelligent vehicles. The shared context understanding brought by the ontology would permit the

establishment of co-operative systems that would improve DAS performances. The authors propose important foundations in an open-source OWL ontology for the Operational Context of assisted and autonomous driving (Fuchs et al., 2008b, 2008a). Armand et al. (Armand et al., 2014) propose a simple, lightweight OWL ontology of the vehicle's OC for real-time reasoning to determine the DAS behavior based on the context. Their ontology presents usual classes of driving context elements such as static and mobile entities, interaction parameters with the entities (i.e. is close, is following, and is to reach), and spatial information about the entities ahead of the vehicle.

Geyer et al. (Geyer et al., 2014) observe the need for a unifying terminology for Vehicular CPS use-case, scenario, and situation catalogs. They propose an ontology defining the following concepts: ego vehicle, scenery, scene, situation, scenario, driving mission, and route. Ulbrich et al. (Ulbrich et al., 2014) build a context ontology based on these concepts for autonomous driving environment modeling to enhance the vehicle's decision making. The environment is dynamically modeled through sensors data in an aggregation of multilevel directed graphs representing the ego vehicles and its environment's elements. Schult et al. (Schuldt et al., 2018) also introduce a context modeling method to efficiently and systematically generate test cases scenarios for automated driving functions in virtual environment simulations. The context model is built in 4 levels: Road network; road infrastructure; dynamic elements; and environmental conditions. Using these levels, Bagschik et al. (Bagschik et al., 2018) propose a concept of an ontology for scenes generation used in Vehicular CPS development. Their work focuses on the generation of the first scenes of operational scenarios. The scenes are generated with combinations of the ontology elements and their relations, resulting in static descriptions of scenes observed by the Vehicular CPS, coupled with the vehicle's possible maneuvers.

3.2.3 Operational scenarios identification methods for Vehicular CPS

Ulbrich et al. define the scenario of a Vehicular CPS operation as a description of "*the temporal development between several scenes in a sequence of scenes*" and introduces a scene as "*a snapshot of the environment including the scenery and dynamic elements, as well as all actors' and observers' self-representations, and the relationships among those entities*". They also suggest that the scenarios

are the realization of a use-case and can be characterized by goals and actions which are determined by the said use-case (Ulbrich et al., 2015).

In recent years, multiple research works have been done on the generation of test simulation for automated driving functionality. The main challenge of scenario generation is the infinite number of possibilities due to an infinite input domain (Schuldt et al., 2018).

Some studies concentrate on the generation of test scenario for specific maneuvers and functionalities. Rocklage et al. (Rocklage et al., 2017) generate variations of scenarios to test AV functions in predefined situations with combinatorial algorithms. The generation is based on the successive parametrization of the road geometry, the weather, and dynamic objects. The motions of dynamic objects are generated on a grid discretization of the time-space. Höfer and Hermann (Höfer and Herrmann, 2017) identify three technics for test scenario generation of static and dynamic objects (with their maneuver): manual, based on map data of real test tracks, and based on measurement real test tracks.

Other studies focus on scene generation. Bagschik et al. (Bagschik et al., 2018) propose a combinatorial algorithm on the operational context element for a mass generation of driving scenes. The output scenes can be used as opening scenes of test scenarios simulation. Jesenski et al. (Jesenski et al., 2019) propose a probabilistic approach using Bayesian networks to populate intersection sceneries with dynamic vehicles. Their approach generates traffic scenes on arbitrary road structures.

Recently, a few studies focused on scenarios definition for the development and engineering of Vehicular CPS. Bach et al. (Bach et al., 2016) propose a domain model for the specification of operational scenarios during the development of Vehicular CPS. The proposed domain model describes the different concepts constituting a scenario such as scenery, scene, road, lane, situation, participant, and maneuver. With an abstraction of temporal and spatial information, Bach et al. achieve comprehensible modeling of operational scenarios based on the succession of participants maneuver inside the scenery. However, their proposition does not indicate what relevant scenarios should be modeled for the development of a Vehicular. Sippl et al. (Sippl et al., 2019) propose a scenario-based design approach for the development of automated driving functions. They analyze customer journeys and user stories to define abstract scenarios of use cases. They propose

to define catalogs of scenarios in a textual Domain-Specific Language (DSL), modeling the context elements composing a scenario (Bock et al., 2019).

The overall literature on the Operational Context modeling for Vehicular CPS shows that context models are mainly used to improve systems performances through context-awareness, dynamic behavior, context sharing, and multi-system co-operativity. Recent works start focusing on using Operational Context modeling for test-cases and scenes generation for the system testing and validation phase. However, and to the best of our knowledge, the literature does not provide a formal modeling and representation of the Operational Context permitting to extend the Concept of Operation of safety critical vehicles. This is all the more important as Vehicular CPS function by executing physical and computational processes in response to their operational environment and their architectures highly depend the Operational Context. The literature on operational scenarios identification method for Vehicular CPS shows that there is no method using the Operational Context to systematically identify and define operational scenarios in the early design phase of Vehicular CPS.

3.3 Context Ontology-based scenarios Identification and Modeling

Early phases of the design process are characterized by the identification of the system's concept of operations, to define the operational needs and the system requirements and specifications (Sutcliffe, 2003). In the case of Vehicular CPS outsourced R&D, projects often lack precise and exhaustive specifications at early design phases, due to low industrial feedbacks and technical maturity. On the other hand, the clients and stakeholders effectively define the operational context of the exploration project and the operational domain where the Vehicular CPS is designed to function.

This paper proposes a method to support a systematic definition of operational scenarios of Vehicular CPS based on an Operational Context ontology. The scenarios may are useful in various activities, particularly in the specification of the system's architecture in later phases. The method starts with the knowledge of several Operational Context elements within which the vehicle is designed to operate. These elements are used to systematically

identify and model the operational scenario variations the vehicle may encounter during its operations (see Figure 3.1)

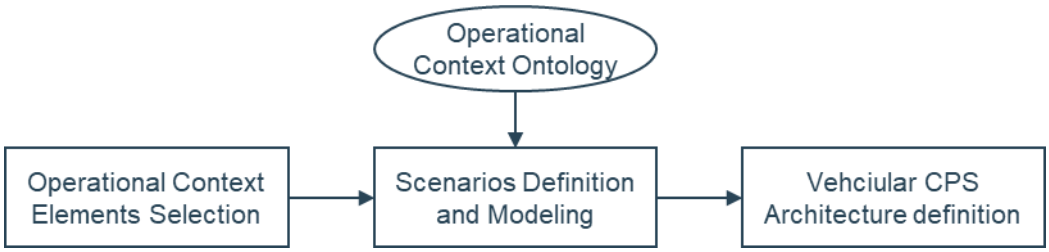


Figure 3.1: Operational context-based scenario definition for a scenario-based design approach of Vehicular CPS

3.3.1 Methodology of the ontology building

The building of the proposed Operational Context ontology has done based on empirical study and observations of the industrial design process of Vehicular CPS in addition to the ideas proposed in the literature. As for the industrial observations, the authors of this paper collaborated for this research work with an industrial team focusing on the development of Vehicular CPS from the engineering consulting company AKKA Technologies. The leading author spent half of this research project time (one year and a half) participating to R&D projects of Vehicular CPS as design systems engineer. He identified the relevant structure and concepts of the ontology through an action research methodology.

As for the ideas of the literature contributing to the development of the proposed ontology, Bach et al (Bach et al., 2016) decomposes the operational scenarios of Vehicular CPS into a scenery, situations, participants and events. According to Ulbrich et al. (Ulbrich et al., 2015), these parts of the operational scenario are modeled from various class of elements belonging to the Operational Context (the environment, the scenery elements, the dynamic element, and the actors). During a scenario, some elements are statics, while others can be dynamic. The weather and time of the day express the environment set-up of the scenario. A change in one of them defines a different operational scenario. Then, there are the elements that outline the landscape and road structure of the scenery, which are defined as the traffic infrastructure elements. Among other things, these elements introduce traffic

rules. Finally, the scenery is populated with the participants and their maneuvers to create a sequence of scenes.

Ontology building is a complex task often discussed in the scientific community. According to Poveda-Villalón et al. (Poveda-Villalón et al., 2012), the quality of an ontology can be determined with six dimensions: human understanding, logical consistency, modeling issues, language specification, real word representation, and semantic applications.

For the development of the presented ontology, we follow an evolution-based approach. We introduce a novel ontology structure, including discussed concepts from the literature and introducing new ones for the purpose of the paper (Tartir et al., 2010). To satisfy the human understanding dimension and real word representation dimension, we confronted the ontology to independent revisions of 4 Vehicular CPS professionals from the industrial partner. Their feedback helped to detect potential ambiguities and the consistency of concepts with the real world as they have observed it. Besides, the logical and modeling consistency was ensured with the implementation of the ontology on the free software Protégé with the Manchester Syntax (Horridge et al., 2006) and verified with the HermiT reasoner (Glimm et al., 2014).

3.3 Context Ontology-based scenarios Identification and Modeling

As the aim of the ontology is to extend the ConOps analysis by analyzing and characterizing the Operational Context, our industrial observations led to the addition of another class of context elements: the use case level. The use case is an abstract level that delimits the characteristics and limits of the operational scenarios to be defined in the operational analysis.

Consequently, we propose an ontology modeled in the Web Ontology Language (OWL) and structured in five levels: (0) Use-case, (1) Environment, (2) Road Infrastructure, (3) Traffic Infrastructure, and (4) Traffic Objects. The principles of Operational Context ontology proposed by Schuldt et al. (Schuldt et al., 2018) inspired the structure we propose. The structure presented in this paper proposes different layers, classes, attributes, and

relations to support the reasoning on scenarios identification in the early design phase. The order of the levels corresponds to the order of a scenario elements definition. Figure 3.2 shows the global structure of the ontology with an illustration of a roundabout.

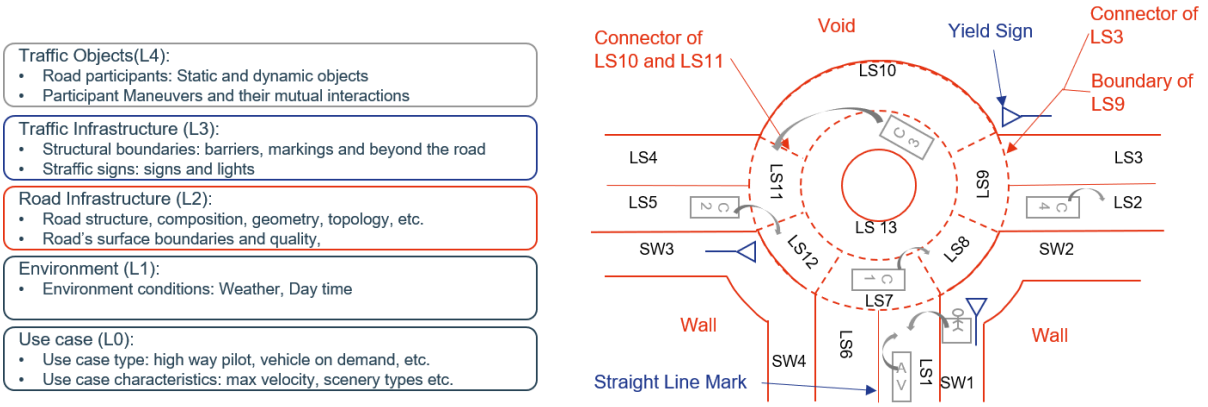


Figure 3.2: Overall structure of the Vehicular CPS OC Ontology

3.3.1 Overall steps of the systematic scenario definition

Following the levels of the ontology, the systematic identification starts with characterizing the use cases, by defining concepts such as the missions, goals and the scenery types. This step is essential to limit the scope of scenarios to be defined. The first variation to the scenarios is introduced in the second step. The ontology helps characterizing the various possible environmental conditions with the attributes of the weather and the time. The strength of this method starts from the third step, where we identify all the scenery variations. The first scenery variations comes from the different road structures encountered. As such, the design team defines simple and complex road structures and characterize their geometry, topology and quality (see Figure 3.2). At the end of the third step, the design team has identified and characterized all the relevant road structures based on the known context elements. The fourth step is when the design team adds the traffic infrastructure elements to the sceneries. In addition, this step permits to identify other situation variations. For example, the design team can derive from an intersection structure several sceneries such as: traffic light intersection, stop intersection, or yield intersection. To achieve this exploration, the context ontology proposes concepts of traffic signs and markings.

With all variations of sceneries defined, the last step focuses on the definition of the different situations encountered within these sceneries. To do so, the design team populates the sceneries with various participants and describe their maneuvers from the point of view of the vehicle under design. They describe these participant maneuvers in terms of personal maneuver such as driving up or stopping, as well as interaction maneuver such as falling back and approaching. Figure 3.3 illustrate the overall steps of the systematic scenario identification and definition method for Vehicular CPS. and the variations identified at each step.

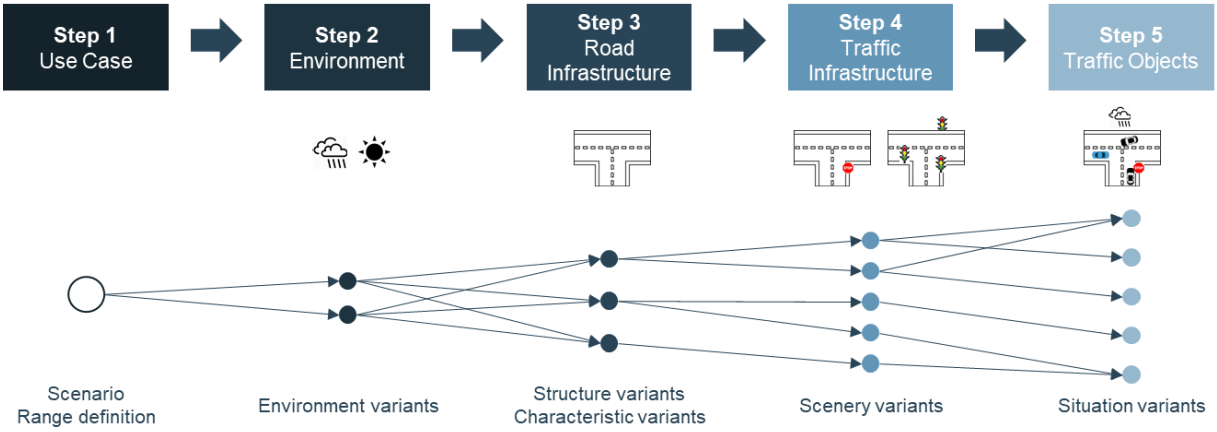


Figure 3.3: Overall steps of the systematic scenario identification and definition method for Vehicular CPS

3.3.2 Definition of the Use-case Level (0)

In the early phase of Vehicular CPS design, the design team starts with outlining the perimeter of the operational scenarios set. The first step of the process is to define the vehicle's use case and select the values of its attributes. The use case introduces restrictions on sceneries, traffic participants, and behaviors. These restrictions outline the perimeter of permitted Operational Context elements.

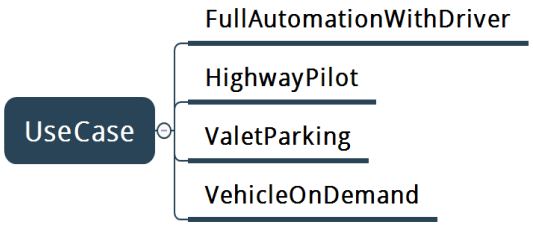


Figure 3.4: Use-case level of the Vehicular CPS Operational Context Ontology

Table 3.1: Use case's data properties

Concept	Data Property	Values
UseCase	availableHandover	driver
		electric towing
		no availability
		pilot service
		tele-operated driving
	dynamicElements	no dynamic elements
		only AV"
		only motor vehicles
		without exclusion
	externalData	AD emergency call
		Occupant state
		Remote driving input
		Traffic state
	maximumPermittedWeight	2T
		32T
		500Kg
		8T
	maximumVelocity	5 km/h
		30 km/h
		60 km/h
		120 km/h
		240 km/h
	scenery	access road
		urban arterial road
		main traffic roads
		highway
		country road
		agricultural road
		parking structure
		terrain
		special areas
	typeOfOccupant	Nothing
		Cargo
		Person
		No exclusion

Vehicular CPS use cases were defined in multiple studies. Few of them focused on what characterizes a use case. The Operational Context ontology offers the design team the four main subclasses of use cases found in the literature: highway autopilot, valet parking, full automation with a driver, and vehicle on demand (SAE, 2018; Wachenfeld et al., 2016). Other classes may be added to the ontology with the evolution of Vehicular CPS usages. In Figure 3.4, we illustrate the hierarchical decomposition of the class. Following the

characterization of Wachenfeld et al., the design team can characterize the defined "UseCase" class with seven attributes of the Enumerate type. The details of the attributes and their values are presented in Table 3.1. Each one of the four use case classes, introduces specific restrictions to the attributes. These restrictions describe the perimeter of the use case (Wachenfeld et al., 2016).

3.3.3 Definition of the Environment Level (1)

Once the use case is defined, the range of environmental conditions within which the vehicle is designed to operate must be specified. As these conditions (bright horizontal light, wet road, etc.) affect the vehicle's perception capability as well as driving conditions, they should be defined second to the use case. Consequently, the ontology's environment level introduces the classes "Weather" and "DayTime" and their attributes. As such, the design team defines from the Operational Context ontology the different weather and day times.

Table 3.2 presents the data properties and values for the two classes. During the definition of a scenario, the design team defines a time of the day and a weather. Changing these conditions would amount to consider a new scenario. Besides, the design team must be aware that some environmental attributes may affect the attributes of other Operational Context elements, such as rainy weather would result in wet lanes.

Table 3.2: Environment's data properties

Concept	Data Property	Values
DayTime	daytimeProperty	Dawn
		Morning
		Afternoon
		Dusk
		Nigh
Weather	weatherProperty	Normal
		Foggy
		Rainy
		Cloudy
		Sunny
		Snowy
		Windy

3.3.4 Definition of the Road Infrastructure Level (2)

At this point, the design team sets the various sceneries encountered by the Vehicular CPS. At the beginning of the Vehicular CPS design project, it is assumed that some Operational Context elements are known. These elements can be environmental elements, structural elements, or traffic elements. To help the design team identify the relevant scenarios based on these elements, the road infrastructure level of the ontology proposes the "RoadStructure" class and the "ComplexStructure" class to define the road structures corresponding to the structural elements of the Operational Context. These classes describe basic lane structures with the "LaneStructure", crossroads with classes "Crossroads" and complex structures composed of several lane segments such as roundabouts and intersections. The basic lane structures forming the scenery, can be defined with the classes "LaneSegment", "SideLane", "Sidewalk", and "ParkingLane". The design team can define and characterize, from these various classes, their relations, and the input Operational Context elements, the relevant sceneries for the operational scenarios in 3 main steps:

Step 1 consists of defining a set of sceneries encounterable by the vehicle. The sceneries are then modeled in terms of road structures connected with boundaries and connectors. Afterward, they are divided into positions where traffic elements and participants are positioned, as suggested by Ulbrich et al. (Ulbrich et al., 2014). For instance, if the operational design domain of the AV contains a roundabout with three entries and three exits, a corresponding scenery would be defined. As an example, the schema in Figure 3.2 illustrates the lane structures modeling such as LS1, LS7, SW2, their boundaries, and connectors.

The class "LaneBoundary" defines the lateral boundaries of a lane structure. Hence, two adjacent lane segments, such as LS2 and LS3, will share a lane boundary. Connectors, on the other hand, define the beginning and end of a lane structure. As such, two successive lane structures will share a connector, such as the lane segments LS10 and LS11. Connectors are also used to define the entering and leaving points of crossroads. With boundaries and connectors defined, we can join lane structures to form the scenery. Intersections and roundabouts lanes can be joined to basic lanes by overlapping one's connector (end), with

the boundary (side) of the other. This is illustrated in Figure 3.2 with the example of LS3 and LS9: LS9 is part of the roundabout and LS3 ends on the later.

Step 2 serves to characterize the defined lane structure with geometry, a topology, and quality with relations targeting the "RoadGeometry", "RoadTopology", and the "RoadQuality" classes, respectively. There are three possible geometries as enumerated by Schuldt et al. (Schuldt et al., 2018): straight, curve, and clothoid. While curves represent arcs with a fixed radius, the clothoid represents an arc with a variable radius. A start and end radius hence characterize Clothoids. As for the topology, the lane structure can be either flat or slope, with a varying slope degree. As for the road quality, each position would be associated with a quality, as it can vary within a lane structure. The presence of a pothole in the position can also be introduced with the class "Pothole".

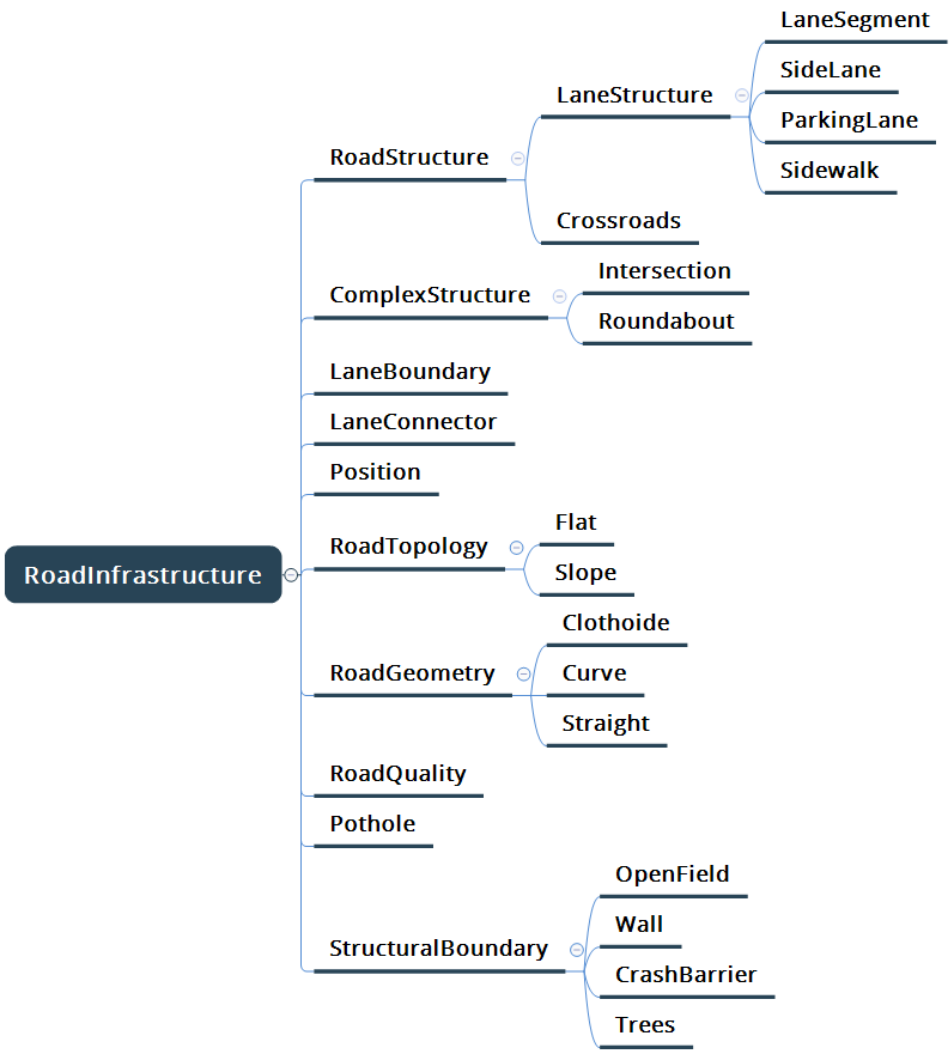


Figure 3.5: Road level of the Vehicular CPS Operational Context Ontology

Step 3 consists of describing the structural boundaries of the scenery. It introduces the boundaries beyond the considered lane structures. These boundaries are necessary for describing what can be perceived by the Vehicular CPS beyond the road structure where it operates, and how it can affect it. The Operational Context ontology proposes four types of boundaries that can impact the Vehicular CPS perception: "Wall" for structures blocking the view, "OpenField", "Tree" representing regularly alternating objects, and "CrashBarrier".

The structure of the Road Infrastructure level (2) of the OC Ontology is represented with a hierarchical decomposition of its concepts plotted in Figure 3.5. The different scenery elements described above are organized through linking their class instances with relations. For instance, we attribute a boundary to the "LaneSegment" instance LS1 by using the relation "is_right_boundary_of" from an instance of "LaneBondary" to LS1. All the relations implemented in the Ontology are displayed in Figure 3.9 of the Appendix. As for the attribute of the classes, they are detailed in Table 3.3.

Table 3.3: Road Infrastructure's data properties

Concept	Data Property	Values
LaneStructure	geographicOrientation	North
		Northeast
		East
		Southeast
		South
		Southwest
		West
		Northwest
Crossroads	nbEnteringLanes	Type: int
	nbExitingLanes	Type: int
RoadGeometry	structureWidth	Type: float
Cluthoid	startCurvature	Type: int
	endCurvature	Type: int
Curve	curveRadius	Type: int
RoadQuality	roadCondition	Normal
		Abrasion
		Icy
		Dirt
		Wet
		Snowy

3.3.5 Definition of the Traffic Infrastructure Level (3)

With the set scenery structures defined, the design team needs to add a layer of Traffic infrastructure to complete the sceneries. At this point, a scenery structure should be populated with various traffic infrastructure that would be perceived by the vehicle. Populating the structures may create multiple sceneries and operational scenarios. For instance, the structure of an intersection can be populated with a traffic light or a stop sign, which would give two different scenarios. On the other hand, the presence of a traffic element in the operational design domain of the Vehicular CPS may necessitate going back to the previous level to define a corresponding scenery. As an example, if the operational design domain contains traffic light, but does not mention intersections, an intersection scenery must be defined in the previous level solely for the traffic light situation.

The Traffic infrastructure level of the Operational Context ontology defines two main concepts: traffic signs and road markings. The "TrafficSign" Class is composed, as found in traffic regulations, by "DirectionSign", "DangerSign", "PrescriptionSign", and "TrafficLight". These signs are applied to specific lane segments. This relation is modeled through the relation "is_applied_on" from the "TrafficSign" class to "LaneSegment". The signs are also positioned inside "Position" instances of the lane structures.

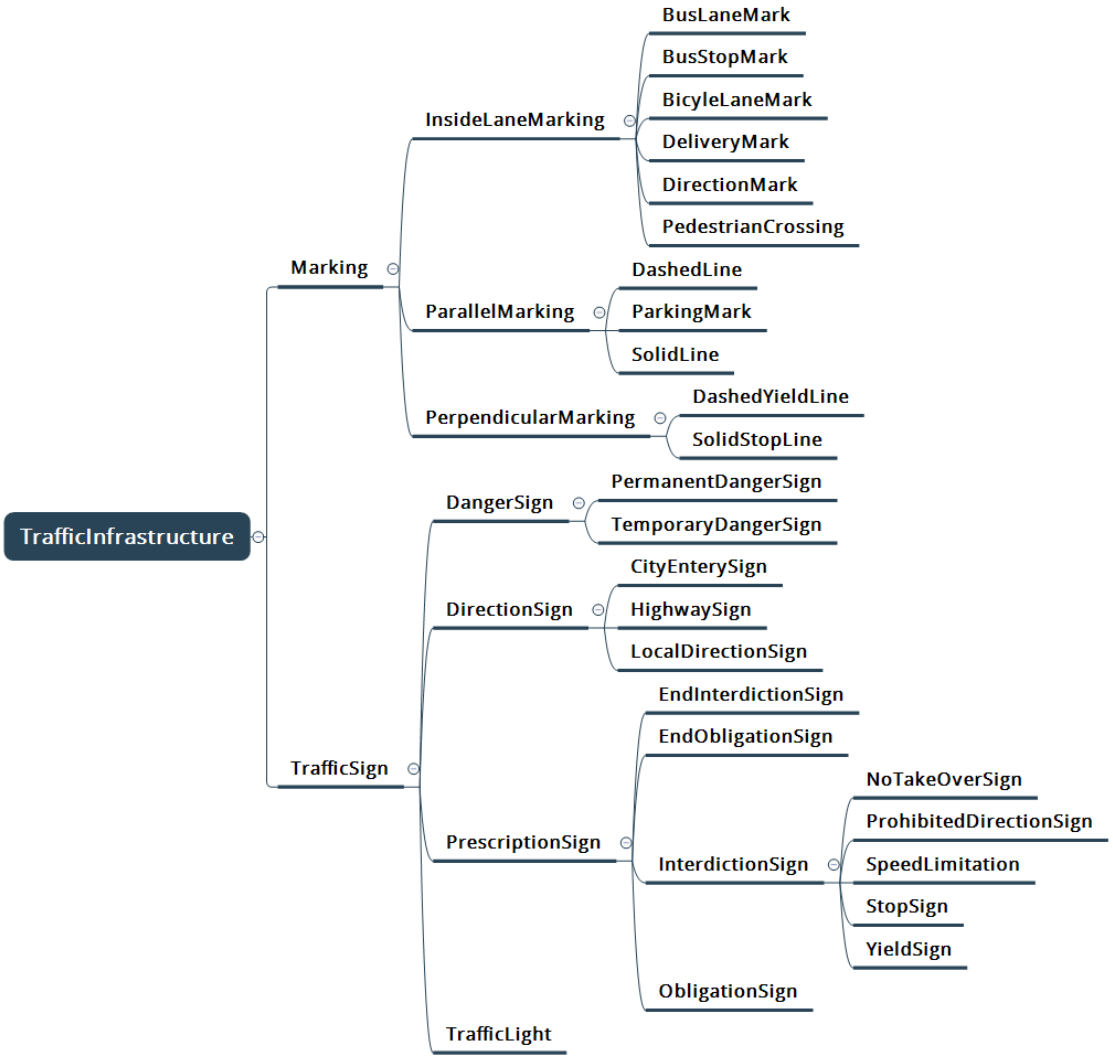


Figure 3.6: Traffic Infrastructure level of the Vehicular CPS Operational Context Ontology

Figure 3.6 illustrates the hierarchical decomposition of the Traffic Infrastructure level. It shows that the markings are of three types: inside lane, parallel, and perpendicular markings. Respectively, the classes "InsideLaneMarking", "ParallelMarking", and "PerpendicularMarking" consider the marking Markings inside lane, parallel, and perpendicular to their direction are considered. In fact, we can observe in Figure 3.9 (Appendix) that the "Position" class from Level (2) has a relation "contains_mark" with the "InsideLaneMarking" class. In contrast, the "LaneBoundary" class has "contains_parallel_marking" with "ParallelMarking", and "LaneConnector" has "contains_perpendicular_mark" with "PerpendicularBoundary".

The overall characterization of this ontology's Level (3) is given in Table 3.4. With this level defined, the set of sceneries and their traffic rules are completed

Table 3.4: Traffic Infrastructure's data properties

Concept	Data Properties	Values
Marking	markingColor	Blue
		White
		Yellow
		Red
PerpendicularMarking	perpendicularMarkWidth	Type: int
ParallelMarking	parallelMarkWidth	Type: int
DirectionMark	markDirection	Front
		Left
		Right
		Front & Left
		Front & Right
		Right & Left
TrafficSign	signOrientation	Back
		Front
SpeedLimitation	speedLimit	Type: int
TrafficLight	trafficLighState	Red
		Yellow
		Green

3.3.6 Definition of the Traffic Objects Level (4)

The set of sceneries resulted from the previous level of this identification process offers the possibility to define different operational scenarios by populating the sceneries with traffic participants and objects. At this point, the design team must define various scenarios that introduce a variety of stimuli to the Vehicular CPS from its Operational Context. These situations would be associated with responsive behaviors and help define the vehicle's system architecture.

For this purpose, the Operational Context ontology's level (4) proposes four sub-classes, as illustrated if Figure 3.7: "TrafficParticipant", "Maneuver", "VehicleRider", and "TrafficProperty". The scenario description is centered around the AV under design. Hence, the elements are modeled with respect to the AV and their interactions with it. The AV under design is represented with the class "VehicleOfInterest".

Two main steps to define operational scenarios are required at this level. **Step 1** introduces traffic participants. Two types of participants can be defined in the operational design domain: "Vehicules" and "NonVehicules". The traffic participants take a position in the previously defined instances of the "Position" class. Each type of participant has several defined maneuvers that can be executed. For non-vehicles, the ontology proposes simple maneuvers such as crossing a lane, moving on the sidewalk, or stopping. As for vehicles, Bagschik el al. define 9 maneuvers drive up, lane change, turn, turn back, safe stop, follow, approach, overtake, and fall back (Bagschik et al., 2018). we propose additional maneuvers and separate the maneuvers in personal maneuvers (drive up, safe stop, emergency stop, safe deceleration, lane keeping, lane change, turn, turn back, and park.) and interaction maneuvers (follow, approach, overtake, and fall back). Personal maneuvers describe the maneuver of vehicles without consideration of its surrounding, and each participant is given at least one personal maneuver per scene. On the other hand, as other vehicles often impact a vehicle's behavior, interaction maneuvers define maneuvers with respect to another participant or object from the traffic infrastructure level. As such, it can be observed in Figure 3.9 (Appendix) that classes "Approach", "Fallback", "Follow", and "Overtake" connected to "TrafficParticipant" and "TraffcSign". Throughout the scenario, the participants will change positions and maneuvers, which will create a succession of scenes and describe the complete scenario.

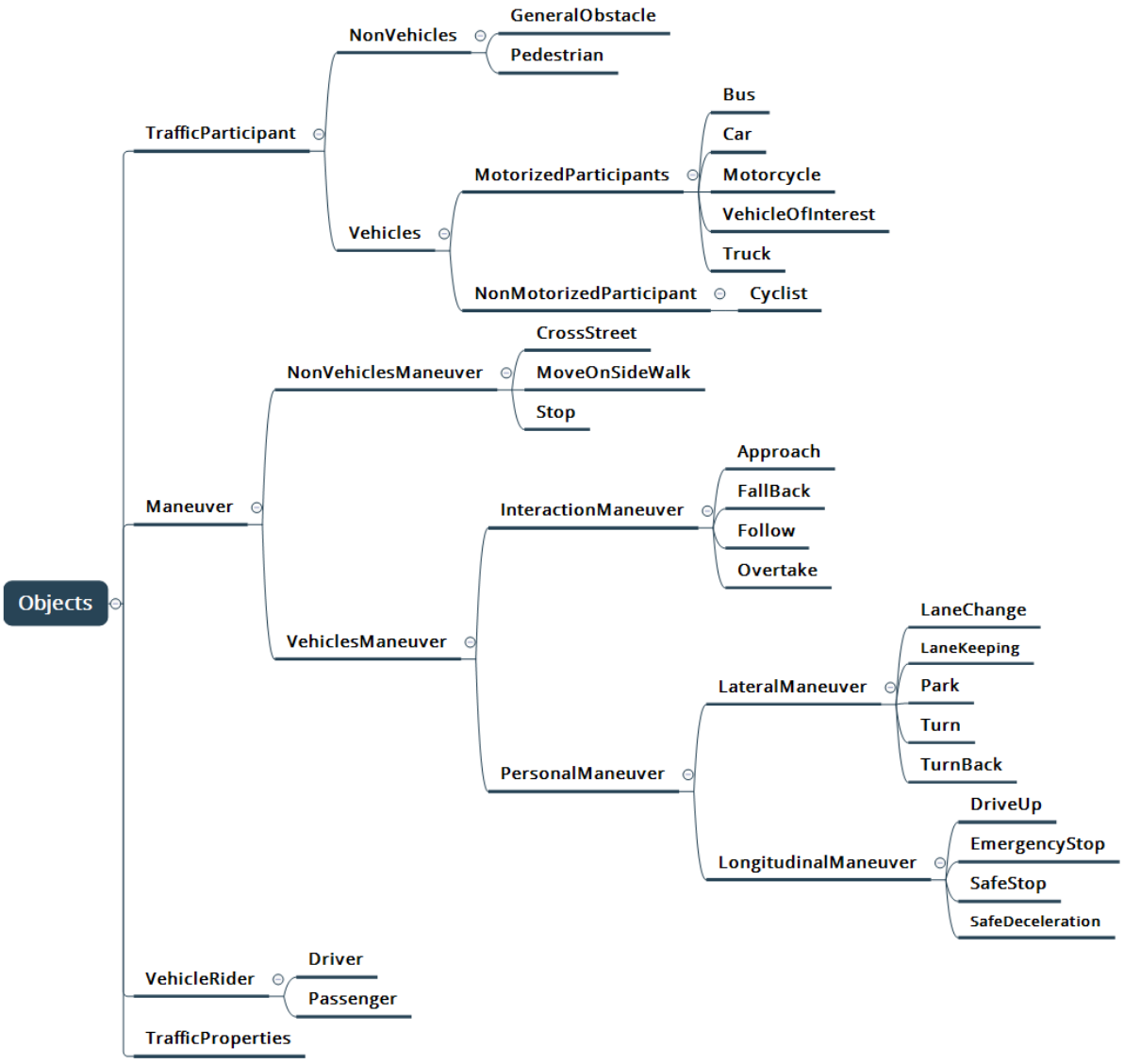


Figure 3.7: Objects level of the Vehicular CPS Operational Context Ontology

Step 2 consists of characterizing the traffic of the scenario. The "TrafficProperties" class introduces characteristics of the traffic density and flow, as detailed in Table 3.5. These parameters allow us to describe and control the situation in which the AV operates.

An additional step may be considered in some scenarios to add the Vehicular CSP riders. The Operational Context ontology proposes the "VehicleRider" class to model the vehicle's driver condition when needed, such as in the scenario of "handing over the commands to driver".

Table 3.5: Objects data properties

Concept	Data Property	Values
TrafficProperties	trafficDensity	None
		Light
		Charged
		Jam
	trafficFlow	Interrupted
		Uninterrupted
Driver	driverState	Focused
		Distracted
		Unavailable
Approach	approachingSpeed	Slow
		Fast
		Dangerous
Turn	turningDirection	Right
		Left
TrafficParticipant	participantSpeed	<15Km/h
		>15Km/h and <30Km/h
		>30Km/h and <50Km/h
		>50Km/h and <70Km/h
		>70Km/h and <90Km/h
		>90Km/h and <120Km/h
		>120Km/h

3.4 Case Study

It is commonly accepted that the number of all possible configurations for sceneries and operational scenarios exponentially increases with the number of Operational Context elements. In the early phases of a scenario-based design of Vehicular CPS, the goal is to define a humanly processable number of operational scenarios that illustrate the various Operational Context elements perceived by the vehicle and the different types of situations encountered. In order to illustrate how the proposed method helps the design team to identify such a set of operational scenarios, a case study is presented with the selection of several Operational Context elements in the early phase of an Autonomous Vehicle on demand in the suburb. The following Operational Context elements are defined as inputs:

- Use case: Autonomous Vehicle on demand in the suburb.
- Environmental condition: sunny, normal and during the day.

- Road and traffic elements; roundabout, traffic lights, vehicles, and potential obstacles.

With the outline of the use case and environmental conditions, and with no further information about the operational design domain, the constraints on the scenario can be specified with the attributes of the "UseCase" and "DayTime" and "Weather" classes as follows in Table 3.6:

Table 3.6: A case study constraints for operational scenarios identification

Scenario constrains	Values
Available handover	no availability
Dynamic elements	without exclusion
Maximum velocity	60 km/h
Type of occupants	Person
Sceneries	urban arterial road
	main traffic roads
Daytime property	Morning
	Afternoon
Weather property	Normal
	Sunny

Based on the elements of the operational design domain, three main road structures are directly identified for the scenarios: basic lane structure, roundabout, and a traffic light intersection. As illustrated in Figure 3.8, different characterizations of these structures types result in the identification of scenery variations early on with the road infrastructure level (2). For instance, in this case study, the basic lane type of structure may have different geometries (straight or curved), and roundabouts can contain one or more concentric lanes.

Additional variations of sceneries appear during the definition of the sceneries traffic infrastructure layer. As an example, the parallel marking of the basic lanes may be dashed or solid, which influences the Autonomous Vehicle's perception. A total of nine relevant sceneries were identified for this case study (Figure 3.8). We model the nine sceneries with the concepts of both the Road Unfractured Level (2) and Traffic Infrastructure Level (3) of OC ontology. As presented in Section 3.3, the modeling starts with the instantiation of the road structures. Lane segments are defined, surrounded by sidewalks and the structural boundaries (wall and open fields), as illustrated in the roundabout graphic of Figure 3.2. Afterward, the lane structures are divided into positions that can be filled with traffic

participants and objects (see Figure 3.8). Finally, instances of signs and marking are defined and placed in the sceneries according to the traffic regulations to outline the entities perceivable by the Autonomous Vehicle under design.

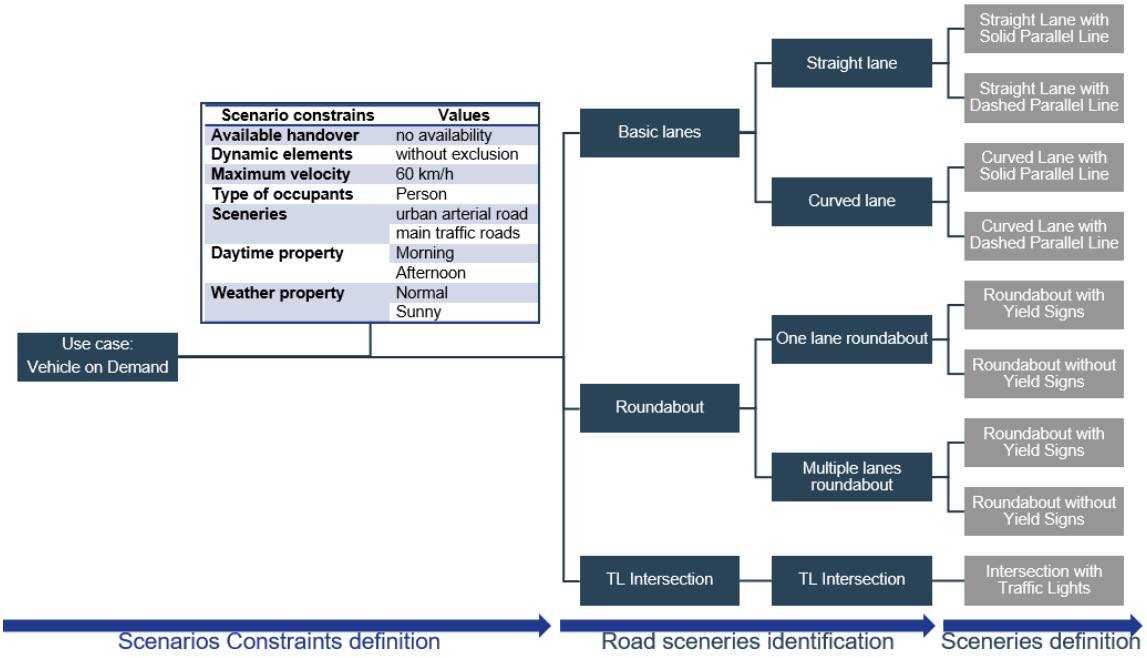


Figure 3.8: A case study of operational sceneries identification

At this point, the objective is to populate the various sceneries with the Autonomous Vehicle under design and traffic participants to create different situations. With respect to the four basic lanes sceneries, three types of situations are possible: no encounters, the detection of an obstacle on the vehicle's trajectory, and the detection of a vehicle. Using the different interaction maneuver proposed by the Operational Context ontology permits to define various scenarios for the detection of obstacles and vehicles as follows.

- A slow, fast, or dangerous approach of a vehicle or obstacle.
- Falling back from the vehicle in front.
- Following the detected vehicle with similar speed.

Similarly, the roundabout sceneries may present multiple situations that specify different reactive behaviors from the vehicle as follows:

- Approaching the roundabout.

- Entering the roundabout without obstacles.
- Encountering an obstacle at the entry of the roundabout.
- Exiting the roundabout without obstacle.
- Encountering an obstacle at the exit of the roundabout.

The traffic light scenery would also present several situations with respect to the state of the light, as well as the static obstacles in the scenery. Finally, all the combination of identified sceneries and situations are represented with sequences of scenes modeled from the Operational Context ontology. Thus, from the only knowledge of a few Context elements in the Autonomous Vehicle operational design domain and using the Operational Context ontology with the scenario identification method, we could identify, characterize and model the various types of operational scenarios that can by encounter by an Autonomous Vehicle only. From these scenarios, it is now possible to deduce the system requirements for the following phases of the design process.

3.5 Conclusion and Perspectives

Given the high dependency of Vehiclar CPS architecture to their operational design domain (SAE, 2018), we argue that there is a need to extend the Concept of Operations and scenario-based design approaches to analyze and characterize the Operational Context elements and their dynamic. This paper proposes a method to systematically identify and define operational scenarios of Vehicular CPS based on their Operational Context in the early design phase. The goal is to help design teams define all the important variation of operational scenarios to specify the vehicle's expected behavior and system architecture.

The method is supported by an Ontology representing the Vehicular CPS Operational Context. The elements of Operational Context ontology permit to characterize and model Vehicular CPS operational scenarios with five levels of abstraction: (0) use case, (1) environment, (2) road infrastructure, (3) traffic infrastructure, and (4) traffic objects. The proposed method uses the known elements of the operational design domains and follows the series of ontology levels to identify variations of sceneries progressively. These sceneries are then populated with traffic participants and objects to create various scenarios illustrating potential situations the AV may encounter.

The proposed ontology is validated with Vehicular CPS domain experts and the HermiT reasoner. Both validation means, as well as the scenario identification and modeling steps, show the following quality criteria of the Operational Context ontology: human understanding, logical consistency, modeling issues, language specification, real word representation, and semantic applications (Poveda-Villalón et al., 2012). Besides, new elements can appear in the future operational context of Vehicular CPS. As such, it may be necessary to add new concepts and remove obsolete ones in the future.

The paper opens the perspective to two main future works. First, future works should be focused on the specification and design of Vehicular CPS based on the operational scenarios and situations identified in the early phase of design. By associated the design process to the method presented in this paper, it would result in Vehicular CPS architecture fit for their Operational Context. Second, addition research should be conducted to semi-automate the scenario identification and definition process with designer-in-the-loop to profit from their implicit domain knowledge. The semi-automation could accelerate the process and improve the quality of the resulting scenario set. Learning and combinatorial algorithms could offer relevant opportunities to explore to achieve the semi-automation.

3.6 Appendix

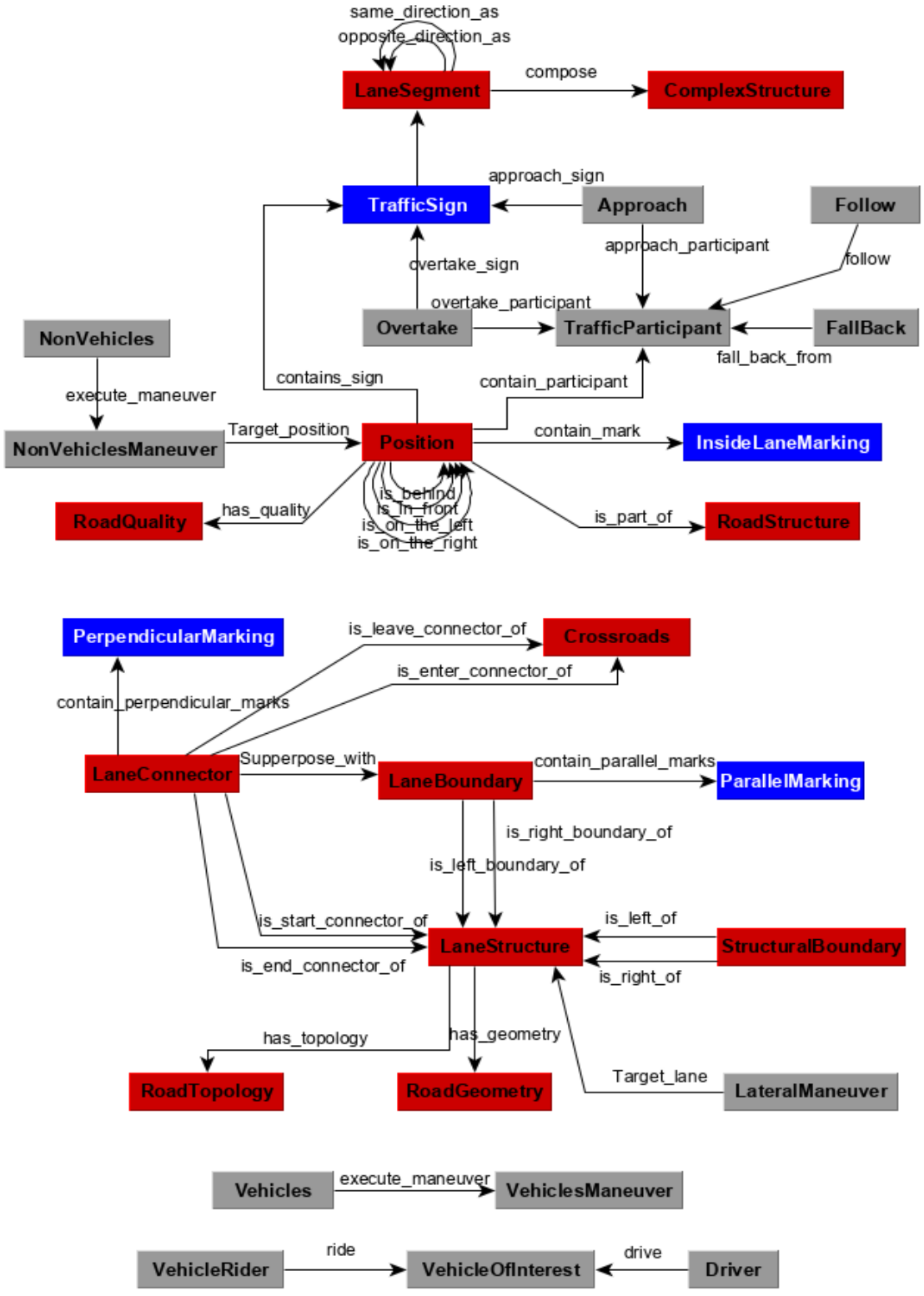


Figure 3.9: The relations between the Ontology's concepts: Level (2) concepts: Red; Level (3) concepts: Blue; Level (4): concepts: Grey

3.7 Acknowledgment

The authors would like to that the following colleagues from AKKA Technologies for their collaboration in the evaluation and validation of the proposed ontology as well as their useful comments: E. Kahale and S. Dicheva.

3.8 References

Alshaikh, Z., Boughton, C., 2009. The Context Dynamics Matrix (CDM): An Approach to Modeling Context, in: 2009 16th Asia-Pacific Software Engineering Conference. Presented at the 2009 16th Asia-Pacific Software Engineering Conference, pp. 101–108. https://doi.org/10.1109/APSEC.2009.74

Armand, A., Filliat, D., Ibañez-Guzman, J., 2014. Ontology-based context awareness for driving assistance systems, in: 2014 IEEE Intelligent Vehicles Symposium Proceedings. Presented at the 2014 IEEE Intelligent Vehicles Symposium Proceedings, pp. 227–233. https://doi.org/10.1109/IVS.2014.6856509

Bach, J., Otten, S., Sax, E., 2016. Model based scenario specification for development and test of automated driving functions, in: 2016 IEEE Intelligent Vehicles Symposium (IV). Presented at the 2016 IEEE Intelligent Vehicles Symposium (IV), pp. 1149–1155. https://doi.org/10.1109/IVS.2016.7535534

Bagschik, G., Menzel, T., Maurer, M., 2018. Ontology based Scene Creation for the Development of Automated Vehicles, in: 2018 IEEE Intelligent Vehicles Symposium (IV). Presented at the 2018 IEEE Intelligent Vehicles Symposium (IV), pp. 1813–1820. https://doi.org/10.1109/IVS.2018.8500632

Baldauf, M., Dustdar, S., Rosenberg, F., 2007. A survey on context-aware systems. International Journal of Ad Hoc and Ubiquitous Computing 2, 263. https://doi.org/10.1504/IJAHUC.2007.014070

Bock, F., Sippl, C., Heinzz, A., Lauerz, C., German, R., 2019. Advantageous Usage of Textual Domain-Specific Languages for Scenario-Driven Development of Automated Driving Functions, in: 2019 IEEE International Systems Conference (SysCon). Presented at the 2019 IEEE International Systems Conference (SysCon), pp. 1–8. https://doi.org/10.1109/SYSCON.2019.8836912

Brown, P.J., 1995. The stick-e document: a framework for creating context-aware applications. Electronic Publishing-Chichester- 8, 259–272.

Bubl, F., Balser, M., 2005. Tracing Cross-Cutting Requirements via Context-Based Constraints, in: Ninth European Conference on Software Maintenance and Reengineering. Presented at the Ninth European Conference on Software Maintenance and Reengineering, pp. 80–90. https://doi.org/10.1109/CSMR.2005.54

Cabrera, O., Franch, X., Marco, J., 2017. Ontology-based context modeling in service-oriented computing: A systematic mapping. Data & Knowledge Engineering 110, 24–53. https://doi.org/10.1016/j.datak.2017.03.008

Chen, H., Finn, T., Joshi, A., 2003. An ontology for context-aware pervasive computing environments. The Knowledge Engineering Review 18, 197–207. https://doi.org/10.1017/S0269888904000025

Chen, W., Hoyle, C., Wassenaar, H.J., 2013. A Choice Modeling Approach for Usage Context-Based Design, in: Chen, W., Hoyle, C., Wassenaar, H.J. (Eds.), Decision-Based Design: Integrating Consumer Preferences into Engineering Design. Springer London, London, pp. 255–285. https://doi.org/10.1007/978-1-4471-4036-8_10

Crowley, J.L., Coutaz, J., Rey, G., Reignier, P., 2002. Perceptual components for context aware computing. Presented at the International conference on ubiquitous computing, Springer, pp. 117–134. https://doi.org/10.1007/3-540-45809-3_9

De Weck, O., Eckert, C., Clarkson, J., 2007. A classification of uncertainty for early product and system design. Guidelines for a Decision Support Method Adapted to NPD Processes 159–160.

Dey, A.K., 2001. Understanding and Using Context. Personal and Ubiquitous Computing 5, 4–7. https://doi.org/10.1007/s007790170019

Fuchs, S., Rass, S., Kyamakya, K., 2008a. Integration of Ontological Scene Representation and Logic-Based Reasoning for Context-Aware Driver Assistance Systems. Electronic Communications of the EASST Volume 11: Contextaware Adaption Mechanisms for Pervasive and Ubiquitous Services. https://doi.org/10.14279/tuj.eceasst.11.127

Fuchs, S., Rass, S., Lamprecht, B., Kyamakya, K., 2008b. A model for ontology-based scene description for context-aware driver assistance systems. Presented at the Proceedings of the 1st international conference on Ambient media and systems, ICST (Institute for Computer Sciences, Social-Informatics and …, p. 5.

Geyer, S., Baltzer, M., Franz, B., Hakuli, S., Kauer, M., Kienle, M., Meier, S., Weißgerber, T., Bengler, K., Bruder, R., Flemisch, F., Winner, H., 2014. Concept and development of a unified ontology for generating test and use-case catalogues for

assisted and automated vehicle guidance. IET Intelligent Transport Systems 8, 183–189. https://doi.org/10.1049/iet-its.2012.0188

Glimm, B., Horrocks, I., Motik, B., Stoilos, G., Wang, Z., 2014. HermiT: An OWL 2 Reasoner. Journal of Automated Reasoning 53, 245–269. https://doi.org/10.1007/s10817-014-9305-1

Handbook, I., 2014. INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, Version 4. International Council on Systems Engineering.

Henricksen, K., 2003. A framework for context-aware pervasive computing applications. Queensland: University of Queensland.

Höfer, A., Herrmann, M., 2017. Scenario-based approach for developing ADAS and automated driving functions, in: Isermann, R. (Ed.), Fahrerassistenzsysteme 2017. Springer Fachmedien Wiesbaden, Wiesbaden, pp. 215–225.

Horridge, M., Drummond, N., Goodwin, J., Rector, A., Wang, H.H., 2006. The manchester owl syntax. Presented at the In Proc. of the 2006 OWL Experiences and Directions Workshop (OWL-ED2006), Citeseer.

Horváth, I., 2014. What the Design Theory of Social-Cyber-Physical Systems Must Describe, Explain and Predict?, in: Chakrabarti, A., Blessing, L.T.M. (Eds.), An Anthology of Theories and Models of Design: Philosophy, Approaches and Empirical Explorations. Springer London, London, pp. 99–120. https://doi.org/10.1007/978-1-4471-6338-1_5

Hull, R., Neaves, P., Bedford-Roberts, J., 1997. Towards situated computing, in: Digest of Papers. First International Symposium on Wearable Computers. Presented at the Digest of Papers. First International Symposium on Wearable Computers, pp. 146–153. https://doi.org/10.1109/ISWC.1997.629931

Jesenski, S., Stellet, J.E., Schiegg, F., Zöllner, J.M., 2019. Generation of Scenes in Intersections for the Validation of Highly Automated Driving Functions, in: 2019 IEEE Intelligent Vehicles Symposium (IV). Presented at the 2019 IEEE Intelligent Vehicles Symposium (IV), pp. 502–509. https://doi.org/10.1109/IVS.2019.8813776

Khattak, A., Akbar, N., Aazam, M., Ali, T., Khan, A., Jeon, S., Hwang, M., Lee, S., 2014. Context Representation and Fusion: Advancements and Opportunities. Sensors 14, 9628–9668. https://doi.org/10.3390/s140609628

Nadoveza, D., Kiritsis, D., 2014. Ontology-based approach for context modeling in enterprise applications. Computers in Industry 65, 1218–1231. https://doi.org/10.1016/j.compind.2014.07.007

Nemoto, Y., Uei, K., Sato, K., Shimomura, Y., 2015. A Context-based Requirements Analysis Method for PSS Design. Procedia CIRP 30, 42–47. https://doi.org/10.1016/j.procir.2015.02.095

Pascoe, J., 1998. Adding generic contextual capabilities to wearable computers, in: Digest of Papers. Second International Symposium on Wearable Computers (Cat. No.98EX215). Presented at the Digest of Papers. Second International Symposium on Wearable Computers, IEEE Comput. Soc, Pittsburgh, PA, USA, pp. 92–99. https://doi.org/10.1109/ISWC.1998.729534

Perttunen, M., Riekki, J., Lassila, O., 2009. Context representation and reasoning in pervasive computing: a review. International Journal of Multimedia and Ubiquitous Engineering 4, 1–28.

Poveda-Villalón, M., Suárez-Figueroa, M.C., Gómez-Pérez, A., 2012. Validating Ontologies with OOPS!, in: ten Teije, A., Völker, J., Handschuh, S., Stuckenschmidt, H., d'Acquin, M., Nikolov, A., Aussenac-Gilles, N., Hernandez, N. (Eds.), Knowledge Engineering and Knowledge Management. Springer Berlin Heidelberg, pp. 267–281.

Rocklage, E., Kraft, H., Karatas, A., Seewig, J., 2017. Automated scenario generation for regression testing of autonomous vehicles, in: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). Presented at the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), pp. 476–483. https://doi.org/10.1109/ITSC.2017.8317919

Rosson, M.B., Carroll, J.M., 2009. Scenario-based design, in: Human-Computer Interaction. CRC Press, pp. 161–180.

SAE, (On-Road Automated Vehicle Standards Committee), 2018. Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles (No. SAE J 3016-2018). SAE International: Warrendale, PA, USA.

Sathyanarayana, A., Boyraz, P., Hansen, J.H.L., 2011. Information fusion for robust 'context and driver aware' active vehicle safety systems. Information Fusion 12, 293–303. https://doi.org/10.1016/j.inffus.2010.06.004

Schilit, B., Adams, N., Want, R., 1994. Context-Aware Computing Applications, in: 1994 First Workshop on Mobile Computing Systems and Applications. Presented at the 1994 First Workshop on Mobile Computing Systems and Applications (WMCSA),

IEEE, Santa Cruz, California, USA, pp. 85–90. https://doi.org/10.1109/WMCSA.1994.16

Schilit, B.N., Theimer, M.M., 1994. Disseminating active map information to mobile hosts. IEEE Network 8, 22–32. https://doi.org/10.1109/65.313011

Schuldt, F., Reschka, A., Maurer, M., 2018. A Method for an Efficient, Systematic Test Case Generation for Advanced Driver Assistance Systems in Virtual Environments, in: Winner, H., Prokop, G., Maurer, M. (Eds.), Automotive Systems Engineering II. Springer International Publishing, Cham, pp. 147–175. https://doi.org/10.1007/978-3-319-61607-0_7

Sippl, C., Bock, F., Lauer, C., Heinz, A., Neumayer, T., German, R., 2019. Scenario-Based Systems Engineering: An Approach Towards Automated Driving Function Development, in: 2019 IEEE International Systems Conference (SysCon). Presented at the 2019 IEEE International Systems Conference (SysCon), IEEE, Orlando, FL, USA, pp. 1–8. https://doi.org/10.1109/SYSCON.2019.8836763

Sun, Y., Yang, G., Zhou, X., 2016. A novel ontology-based service model for cyber physical system, in: 2016 5th International Conference on Computer Science and Network Technology (ICCSNT). Presented at the 2016 5th International Conference on Computer Science and Network Technology (ICCSNT), pp. 125–131. https://doi.org/10.1109/ICCSNT.2016.8070133

Sutcliffe, A., 2003. Scenario-based requirements engineering, in: 11th IEEE International Requirements Engineering Conference, 2003. IEEE, pp. 320–329. https://doi.org/10.1109/ICRE.2003.1232776

Tartir, S., Arpinar, I.B., Sheth, A.P., 2010. Ontological Evaluation and Validation, in: Poli, R., Healy, M., Kameas, A. (Eds.), Theory and Applications of Ontology: Computer Applications. Springer Netherlands, Dordrecht, pp. 115–130. https://doi.org/10.1007/978-90-481-8847-5_5

Ulbrich, S., Menzel, T., Reschka, A., Schuldt, F., Maurer, M., 2015. Defining and Substantiating the Terms Scene, Situation, and Scenario for Automated Driving, in: 2015 IEEE 18th International Conference on Intelligent Transportation Systems. Presented at the 2015 IEEE 18th International Conference on Intelligent Transportation Systems, pp. 982–988. https://doi.org/10.1109/ITSC.2015.164

Ulbrich, S., Nothdurft, T., Maurer, M., Hecker, P., 2014. Graph-based context representation, environment modeling and information aggregation for automated driving, in: 2014 IEEE Intelligent Vehicles Symposium Proceedings. Presented at the 2014 IEEE Intelligent Vehicles Symposium Proceedings, pp. 541–547. https://doi.org/10.1109/IVS.2014.6856556

Wachenfeld, W., Winner, H., Gerdes, J. Chris, Lenz, B., Maurer, M., Beiker, S., Fraedrich, E., Winkle, T., 2016. Use Cases for Autonomous Driving, in: Maurer, M., Gerdes, J. Christian, Lenz, B., Winner, H. (Eds.), Autonomous Driving: Technical, Legal and Social Aspects. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 9–37. https://doi.org/10.1007/978-3-662-48847-8_2

Weiss, G., Grigoleit, F., Struss, P., 2013. Context modeling for dynamic configuration of automotive functions, in: 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013). Presented at the 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), pp. 839–844. https://doi.org/10.1109/ITSC.2013.6728336

Zimmermann, A., Lorenz, A., Oppermann, R., 2007. An Operational Definition of Context, in: Kokinov, B., Richardson, D.C., Roth-Berghofer, T.R., Vieu, L. (Eds.), Modeling and Using Context. Springer Berlin Heidelberg, pp. 558–571.

4 Paper #3. Operational Context-Based Design Method of Autonomous Vehicles Architectures

Youssef Damak, Yann Leroy, Guillaume Trehard, and Marija Jankovic

This paper has been accepted in the System of Systems Engineering Conference (SoSE 2020):

Abstract. *Autonomous Vehicles (AV) exhibit new characteristics rendering its architecting process challenging. They are cyber-physical context aware systems with a high sensitivity toward their operational context. In addition to the lack of industrial feedback, their architecture needs to be adapted to their Operational Context (OC). The current literature doesn't provide a clear method for designing AV architecture based on their OC. This paper proposes a four steps method following the Concept of Operation approach (ConOps) and functional chains modeling to design AV architecture based on their OC and experts knowledge. The method uses a state-of-the-art OC ontology for AV and improves the identification speed and coverage of the AV's operational needs. The method's applicability and efficiency are validated on a case study where a team engineers designed an AV architecture from the knowledge of several elements of its OC*

Keywords. *Operational Context-based design, Autonomous Vehicles, Model-based System Engineering (MBSE), Cyber-Physical Systems (CPS)*

4.1. Introduction

Autonomous Vehicles (AV) are becoming an unavoidable part of future mobility and transportation systems. The belief that intelligent transportation systems are the cornerstone of future mobility is strengthening, with AV considered as a mandatory part for such an order to exist. A significant increase in AV experimentation projects has been observed in recent years, resulting in the rise in its development challenges.

AV have been characterized as Cyber-Physical Systems (CPS) as well as Context-Aware Systems (CAS). Besides, they are vehicular systems with increasing autonomy (Damak et al., 2020a). These characteristics result in AV being very sensitive to their Operational Context (OC) (Bagschik et al., 2018). Traditional design methods adopted by vehicle makers are mainly technology and solution-based design approaches. With these approaches, there is no guaranty that the resulting AV architecture is adapted to its OC. Also, such approaches are not adapted to a design context of low industrial feedbacks such as AV design context.

On the other hand, the analysis of the operational domain of a system is a design activity often conducted to design complex systems and CPS. Concept of Operations (ConOps) and scenario-based design are the two main approaches used in the industry for operational analysis (Rosson and Carroll, 2009). However, these approaches don't provide a precise method to define the operational requirements and needs of a system based on the knowledge of its OC characteristics. As such, designing a logical architecture well adapted to the OC usually necessitate more considerable effort and many iterations for calibration in the functional analysis and detailed design phases. Hence, there is a need for a new method to design AV logical architecture adapted to their OC.

The second section of the paper reviews related work to system logical architecting and models using OC. The third section details a method for AV logical architecting based on OC. The proposed method is illustrated in the fourth section, with an AV use case. Finally, we discuss in the fifth section some limitations and future research perspectives.

4.2 Related Work

4.2.1 Systems Logical Architecting Methods

A logical architecture is a view of the system architecture with defining a collection of system functions allocated to logical components, abstraction of hardware or software components, and the specification of their interactions and interfaces (Kang and Choi, 2005; Wyatt et al., 2009). With the increasing interest in CPS, several new architecting methods adapted to their context were proposed. CPS architecting presents the challenges of capturing the further high complexity created by the heterogeneity of its component and its dynamic behavior as well as integrating software and hardware components (Dumitrache et al., 2017).

Jensen et al. (Jensen et al., 2011) propose a model-based design methodology for CPS, starting with the definition of the problems and requirements analysis. Following, they model the system's behavior towards its environment as physical processes. They derive the result into algorithms and specify associated hardware. Komoto et al. (Komoto et al., 2013) focus on adaptive systems and propose a tool-supported architecting method. Their method defines the high-level system's specification from the user requirements. Then, they identify mechanisms, sensors, and software subsystems. Following a Function, Behavior, State design approach, they refine the subsystems until they result in a physical process modeling the behavior of the system. Both architecting methods emphasize the importance of behavior identification and modeling for CPS as physical processes. However, they do not identify and link the required behavior of the system to its environment and OC.

Sippl et al. (Sippl et al., 2019) follow scenario-based design to propose an approach for the development of automated driving functions. Their approach starts with ConOps definition initiated from user stories and transformed into abstract scenarios of use cases. From the scenarios, they identify systems capability, defined as a behavior associated with a scenery. They consider the vehicle's behavior as the set of maneuvers it executes. From these couples of behavior and scenery, they derive the system's functional requirements and model its functional architecture. Although the authors link and identify the expected behavior of the vehicle from its OC, they do not model the process that realizes this behavior. Besides, they

do not propose a precise method to identify the functional requirements and architecture from the behavior knowledge.

4.2.2 AV Operational Context Modeling

A system's Operational Context (OC) is defined as the elements that characterize its situation. These elements can be individualities, activities, location, time, and relations (Dey, 2001). Context models started being developed with the increasing interest in CAS. The first studies modeling OC for vehicles emerged with low automation levels, such as driving assistance. Sathyanarayana et al. (Sathyanarayana et al., 2011) model context with logical-based representation to process sensor data for an Active Vehicle Safety (AVS) system. To capture a higher complexity level of the OC, Fuch et al. (Fuchs et al., 2008b) propose an ontology for OC of Driving Assistant Systems (DAS). Their goal is to describe vehicle scenes and sharing data between vehicles.

The previous studies focused mainly on modeling OC for situation descriptions. Later, Ulbrich et al. (Ulbrich et al., 2014) introduce the basics of a context ontology to model in more detail the environment of an AV. Their ontology models and characterizes the lanes, their boundaries and connection, the position of the vehicles, and complex lane structure such as crossroads. Building on their work, Schult et al. (Schuldt et al., 2018) propose a 4-levels structure of a context ontology to describe scenery and scenes, while Bagschik et al. (Bagschik et al., 2018) applied it to model AV scenarios. However, both studies didn't provide the details of the ontology concepts and their relations. Later, Damak et al. (Damak et al., 2020a) provide a detailed description of an OC ontology designed to describe operational scenarios for scenario-based design approaches. The scenarios modeled with instances of their ontology provide information on the AV expected behavior (maneuvers) for the situation it encounters.

4.3 A Four Step Method for AV Architecture Design

Traditional design methods of vehicles focus on technology and solution-based design approaches. These approaches are well adapted with their mastery of the vehicle's domain knowledge. However, with the increase of vehicle automation, the lack of industrial

feedback makes such approaches unsuitable for the design of AV. Also, it has been noted that AV are sensitive to their OC (Bagschik et al., 2018; Ulbrich et al., 2014). Although several studies agree that scenario-based design approaches are suited for CPS and autonomous systems, there is no guaranty that the current methods result in AV architectures adapted to its OC (Rosson and Carroll, 2009; Sippl et al., 2019).

In this section, we present a method to design AV architecture based on its OC. This method ensures the adaptability of the architecture to the characteristics of the OC and reduces the iteration in the functional analysis phase for calibration purposes. The method aims to model a logical architecture of the AV, starting from defining the OC. It requires to identify and model the systems functions and logical components realizing them. However, Analyzing and deriving functional requirements and specifications from the knowledge of the OC can be a complicated task. Directly reasoning on the AV's functions from the understanding of what elements would surround and interact with the AV may lead to over-specification or missing essential functions.

Analyzing the AV's operational behavior in response to its context is necessary for a more accurate functional analysis. ConOps is often applied to derive operational requirements for technical specifications. According to Fairley and Thayer (Fairley and Thayer, 1997), the ConOps aims at describing the system's objectives, environment, and external interfaces, features, and characteristics, as well as defining and validating operational scenarios. Following this approach, the method starts with modeling and characterizing the OC, which includes the AV use cases as well as its environment. Then the operational scenarios relative to situations encountered by the AV are defined. The responsive behavior of the AV is modeled with operational processes.

Figure 4.1 emphasizes the four steps of the method proposed in this paper. In the third step, the functional analysis phase is derived from the result of the operational analysis by modeling functional chains realizing the operational processes identified. Finally, the functions are allocated to logical components specified for their realization. The following sub-sections detail the four steps of the AV logical architecting method

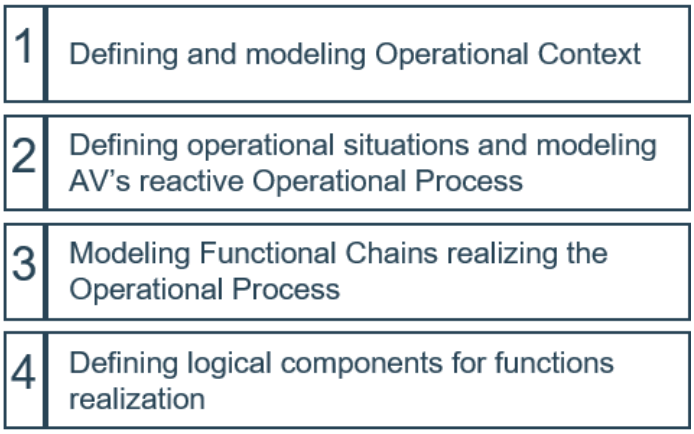


Figure 4.1: A 4 steps method to design AV architecture based on the OC

4.3.1 Operational Context Definition and Modeling

The definition of the Operational Context (OC) is equivalent to specifying an operational requirement for the Autonomous Vehicle (AV) under-design in the form of "The AV must operate within this OC". Several context models exist, as presented in section 4.2. As stated earlier, the OC model must capture and characterize the AV's environment and external interaction, but also its operational objectives. Damak et al. (Damak et al., 2020a) detail an ontology for AV's OC structured in 5 layers that exhaustively define the AV's use cases and external interactions. Their ontology is designed to model operational scenes and scenarios from an instance of the ontology: a predefined OC. It is structured in five levels, as illustrated in Figure 4.2, corresponding to the layers describing scenes: (level 0) use cases, (level 1) Environment, (level 2) Road Infrastructure, (level 3) Traffic Infrastructure; and (level 4) Traffic objects. This OC model captures the system's objective and constraining policies (levels 0, 2, and 3), the system's environment, and external interfaces (levels 1 to 4), as well as the system's features and characteristics (level 4).

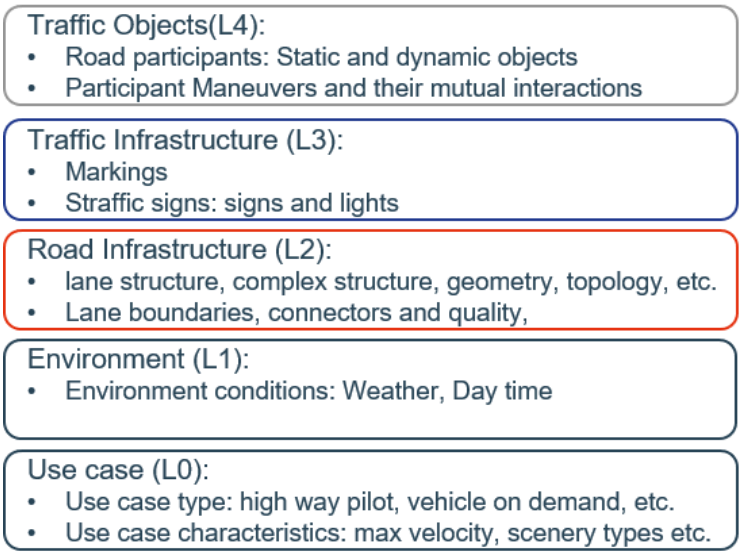


Figure 4.2: The five levels structure of the OC Ontology for AV by Damak et al. (n.d.)

In this first part of the method, designers iteratively define and model the OC with the client. The elicitation burden of the OC elements is lightened using the hierarchy and list of concepts recorded in the OC ontology. First, they identify the targeted use cases for the AV under design. The OC ontology incorporates the use cases such as highway autopilot or vehicle on demand. They also define the perimeter of environmental factors, such as the operations times of the day and the weather conditions. With the use case, designers derive the road structure where the vehicle will operate. The ontology offers the means to model the road structure with different basic lanes and complex structures such as crossroads and roundabouts. It also permits the modeling of the topology and geometry of the lane and other attributes. The next step is to identify the traffic infrastructure elements encountered by the AV: traffic markings, traffic signs, and their characteristics. Finally, other traffic participants must be considered with the last layer of the ontology. Designers define in this step the different possible maneuver for the AV under design, as well as its interaction with other participants.

4.3.2 Operational Process Definition

The second step of the method aims at modeling the reactive behavior of the AV to the OC defined in the first step. The responsive behavior is relative to encountered situations and is modeled as operational processes. The second step of the method introduces two

main steps: (1) identifying the possible situations faced by the AV based on the OC; (2) Modeling the reactive behavior of the AV to these situations as operational processes.

4.3.2.1 Operational Situations Identification

An approach using the five levels structure of the OC ontology is proposed to identify the possible situations corresponding to the previously define OC.

1. Each use case will derive a set of situations; hence, they are analyzed one by one.

2. For every use case, the designers start with the most basic driving situation, with minimal context elements. They identify the situation with the basic line structure, with no exceptional traffic elements nor any traffic participants. A subset of OC elements corresponding to this situation needs to be identified and mapped with the situation. This sub-process is done as follows:

    a. First, the concepts of the "use case", "environment" and "basic lane structure" are manually tagged and characterized.

    b. Then, using the predefined relations in the ontology, the concepts with relations to the "basic lane structure", such as "lane boundary", are tagged.

    c. The previous step is repeated recursively on the newly tagged elements until a coherent scenery is described (Damak et al., 2020a; Ulbrich et al., 2014). As an example, the designer will identify the type marking on the lane boundary in this sub-step.

    d. Once the scenery modeled with the ontology's levels 0 to 3, the maneuvers of the AV are identified from the last level. As no participant is included in the first situation, only "personal maneuvers" are tagged.

3. With the primary driving situation defined, the designers start adding untagged concepts from the predefined OC to the primary concepts subset to identify new situations. The recursive process using the relation of the concepts is applied again on the added concepts. It is preferable to add concepts separately to create distinct situations. Then it is possible to mix the concept to develop new situations if needed.

As an example, if the OC contains a "roundabout" and "motorized participants", it would result in at least three situations with a vehicle encounter situation, an empty roundabout entry situation, and a vehicle encounter in the roundabout entry situation.

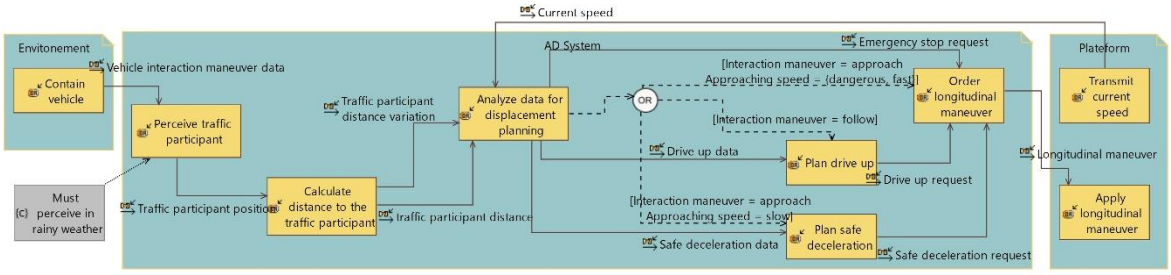


Figure 4.3: Operational process of the vehicle detected situation

As an example, a primary driving situation of the use case "vehicle on demand" would be "The AV drives up, follows a city road lane, without encountering any other road or traffic element". The tagged subset of the ontology concepts corresponding to this situation would be: {"Vehicle on demand", "Day time", "Weather", "lane segment", "sidewalk", "structural boundary: wall", "lane boundary", "lane connector", "position", "parallel marking: dashed line", "longitudinal maneuver: drive up", "lateral maneuver: keep lane"}. Several instances of these concepts may be modeled to complete a realistic scenery.

It can be noted that adding OC elements to the primary driving situation may result in more than one situation. For instance, adding the "complex structure: roundabout" concept to the primary concept subset will result in the situation: "roundabout approaching" and "Roundabout entering with no vehicle".

This iterative process on all the concepts of the predefined OC aims at covering the maximum encountered situations for a specific OC. It prevents missing critical situations. It also permits to identify situations were several OC elements are involved which can be missed with classical scenario definition methods.

4.3.2.2 Operational Process Modeling

The previous step resulted in several situations per use case. Each of these situations is associated with a subset of OC elements. An operational process representing the behavior of the AV in each situation is modeled. The focus of this step is to model operational Processes (OP) in reaction to the selected situation. It is also essential that the OP is traceable to the situation to keep traceability from the logical architecture to the OC. As such, a rational analysis of a standalone OP should permit to deduce said situation and the involved OC elements. To this end, we observed that using the semantic of the OC ontology in the OP model improves the understandability of the situation and the OC elements involved. Hence, we propose a mapping of how each OC element from the ontology should be used in the OP model in Table 4.1.

Table 4.1: Mapping between the OC ontology elements and the operational process elements

OC Level	OC Elements	Operational model elements
Road infrastructure	Lane structure, boundary, and connectors	External OA and interactions
	Lane topology, geometry & quality	Interface data
Traffic infrastructure	Traffic Signs, lights & markings	External & internal OA and interactions
	Marking & sign attributes (color, width)	Interface data
Traffic objects	Traffic Participant	External OA & interactions
	Personal Maneuvers	Internal OA & interactions
	Interaction Maneuver	Condition guard
	Maneuvers attributes (direction, target, speed, etc.)	Condition guard
Environment	Time and weather attributes	Constraints

An operational process is modeled with operational activities (OA) and their interactions. Forks may appear in the OP, and the interactions would have condition guards. Besides, some interactions may include exchanged data that can also be modeled in the operational

process. Finally, constraints can be added to the OA or interactions. For the sake of traceability with the functional and physical domains, we use the open-source Model-Based System Engineering (MBSE) software Capella edited by PolarSys (Bonnet et al., 2016; Roques, 2016). Capella offers the possibility to model operational processes in the operational analysis phase.

We propose a modeling framework with four entities: the environment, an operator, the Autonomous Driving (AD) system, and the driving platform. The operational process starts with OA from the environment, then describes the reactive behavior of the AV. It begins with OA referring to the situation's concepts from the levels 1 and 2 of the OC ontology. For example, these OA will refer to the presence of "lane boundaries" and "traffic lights". Also, the external interaction between the environment and the AV would refer to some concepts with exchanged data properties, such as "markings" and "lane structure". The data property information is modeled in the interface data of the interaction. OC concepts needing further processing inside the AV to produce the appropriate behavior would also be referred to in internal interaction between the AV's OA. The information about "Traffic participants", for instance, would be further processes inside the AD system to analyze their interactions with the AV. These "interaction maneuver" will result in deferent behaviors corresponding to forks in the process. As such, they are referred to in the condition guards of the fork. Finally, some constraints are added to the AD system's OA interacting with the activities of the environment

Figure 4.3 illustrates the OP of the vehicle detected situation. The use of "contain vehicle" as an activity of the environment can be noted. Other examples also demonstrate the OP modeling method, such as the external interaction "Vehicle interaction maneuver data" between the environment and the AD system, as well as "traffic participant distance', an interaction between two activities of the AD system. This example also illustrates a fork in the OP where the AV's reactive behavior depends on the data property of the "interaction maneuver" between the AV and the traffic participant. And lastly, an example of a constraint, with the symbol {C}, expresses the weather conditions under which the "perceiving traffic participant" activities must operate.

The operational process for the primary driving situation is supposed to model the scenery and the road infrastructure characteristics with the lane's structure, boundaries, and connectors, as well as structural boundaries and traffic signs. Hence, there is no need to model these elements again in other operational processes focused on specific situations unless they contribute to the process. As an example, the lane's characteristics are essential for a takeover maneuver, if it is considered by the designers as expected behavior.

4.3.3 Functional Chains Modeling

The first two parts of the AV logical architecture design method presented in this paper focus on the analysis of the operational domain of the AV under design. The next step aims at identifying the Functional Chains (FC) that would realize the OP. In this step, a transition from the operational domain to the functional one is operated. Designers work with domain engineers and experts to identify the necessary functions to realize the OA and interactions of the OP.

The starting points of the FR stays the same as the realized OP. The AD system and the driving platform's OA are replaced by functions and functional exchanges, realizing the activities. Each OP modeled in the previous step is achieved by an FC (Voirin and Tailliez, 2012).

With this method, the AV designers and engineers make sure to identify the relevant set of functions that answer the operational needs of the AV while avoiding over-specification adding unnecessary functions and constraints. The constraints and interaction data are inherited from the operational analysis phase. However, new constraints and data interfaces specific to the defined functions may emerge in the functional analysis. But traceability between the functional domain and the operational domain is guaranteed through this method.

Figure 4.4 illustrates the FC of OP "vehicle detected" described in Figure 4.3. It can be noticed that the OA "perceive traffic participant" is realized by three functions "acquire traffic participant data", "process environmental data", and "detect traffic participant". The reason behind this decomposition is that the acquisition of the environment data through sensors doesn't filter nor classify the object. Processing the data is needed to identify traffic

participants from road and traffic infrastructure. While the activity "Analyze data for displacement" can be shared by the various OP, the functions realizing it in an FC are defined to answer the reactive behavior expected by the OP. For instance, in this case of "vehicle detected", engineers set the functions "classify traffic participant", "track traffic participant", and "manage traffic participant" to fulfill the activity "Analyze data for displacement". Adequate Planification and control functions are also modeled, illustrating the same fork in the OP, depending on the traffic participant's analyzed data.

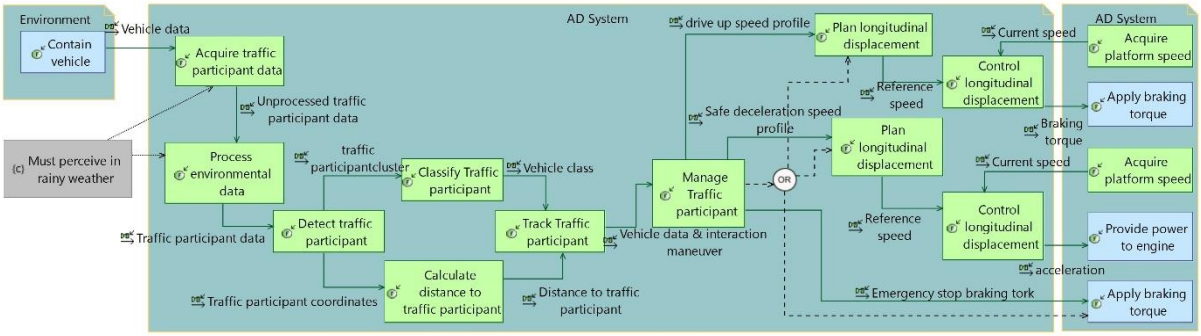


Figure 4.4: Functional chain of the vehicle detected operational process

The functional specifications of the driving platform are detailed in this step and modeled in the FC. Additional FC necessary for the correct functioning of the AV may be added by the designers and the engineer. For instance, FR for missions planning and map upload may have been ignored in the operational analysis. These FC will emerge with the study of the system functions and their specific constraints.

4.3.4 Logical architecture modeling

The final step of the proposed method consists of allocating the functions to logical components. As a Cyber-Physical Systems (CPS), AV components will be a mix of hardware and software. According to Taş et al. (Taş et al., 2016), AD systems are usually composed of five main modules: environment sensors, perception module, decision module, control module, and system management module. We also consider the storage module for maps and trajectory data. As we distinguished between the AD system and the driving platform composing this system, we separate the Environment sensors and the onboard vehicle sensors. We attribute the onboard vehicle sensors with the actuators to the driving platform.

Environment sensors, vehicle onboard sensors, and actuators are physical components of the system. Functions allocated to these components define actions such as providing, acquiring, transmitting, applying, etc. The perception, decision, control, storage, and system management modules are hardware/software components. While hardware components are attributed to each module, each function is allocated to a software component.

4.4 Case Study

To demonstrate the applicability of the proposed method, we applied it with designers and engineers from the autonomous systems team of the engineering consulting company AKKA technologies. We identified for this experiment an Operational Context (OC) for the "vehicle on-demand" use case in sunny and rainy weather. The OC contains traffic lights, roundabouts, and other vehicles and obstacles. From these elements, we used Damak et al. (Damak et al., 2020a) OC ontology to define a sub-ontology corresponding to this context. This step helped identify, characterize, and validate 26 OC elements within a single workshop.

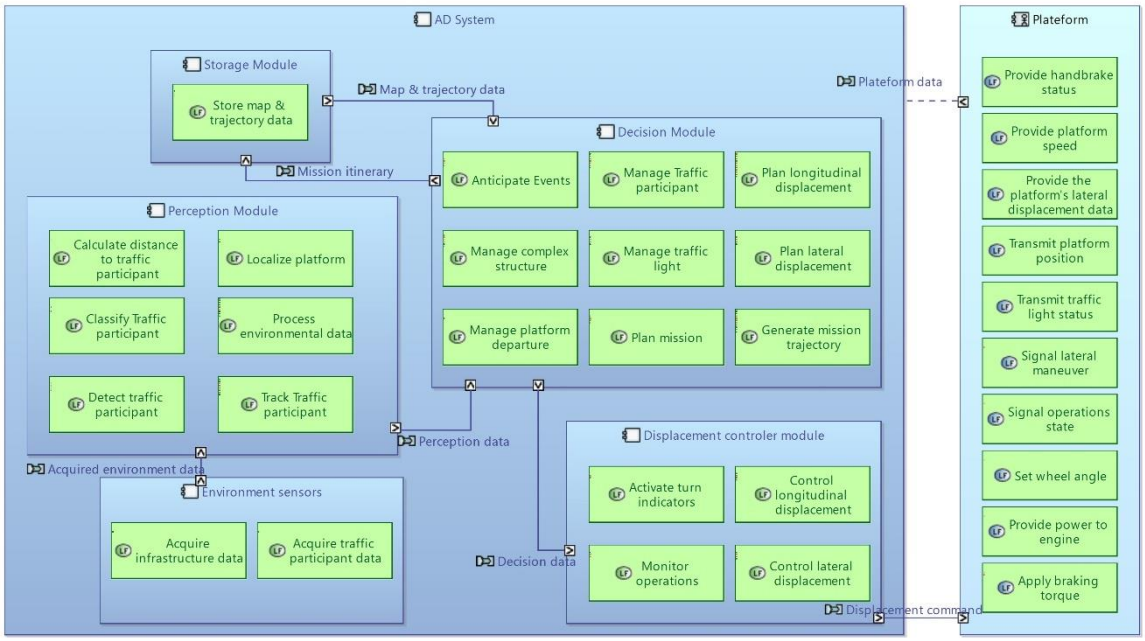


Figure 4.5: Functions and Modules allocation resulting from an application of the OC-based AV architecting method

The analysis of this OC and its main elements resulted in 7 operational situations, modeled with 7 OP as follows:

- Primary driving: The AV drives up, follows a city road lane, without encountering any other traffic element.
- Traffic light encounter: The AV drives up, follows a city road lane, and approach a traffic light. The AV manages the traffic light encounter.
- Obstacle detected: The AV drives up, follows a city road lane, and encounters an obstacle on its path. The AV manages the obstacle encounter.
- Vehicle detected: The AV drives, follows a city road lane, and encounters a vehicle on its path. The AV manages the vehicle encounter.
- Roundabout approaching: The AV drives, follows a city road lane, and approach a roundabout. The AV manages the approach.
- Roundabout entering with no vehicle: The AV arrives at a roundabout. No other vehicle or obstacle is inside the roundabout. The AV enters the roundabout.
- Roundabout entering with vehicles: The AV arrives at a roundabout. A vehicle is passing by inside the roundabout, blocking the entrance. The AV manages the vehicle encounter then enters the roundabout.

The team identified two new OP needed for the operability of the system: "Departure management", and "End of mission". A total of 9 OP were analyzed to define 9 FC. At this stage, experts noticed the improvement in the coverage of the identified operational scenarios in the operational analysis phase as well as a faster time to finalize and validate the operational analysis.

In the FC modeling step, it was decided that FC specific to the localization of the platform and the mission planning were needed to complete the functional analysis of the system. Thirty-two functions were defined to model the total of 11 FC. Figure 4.5 shows the identified functions and their allocation to the different modules described in section 4.3.4.

4.5 Conclusion

This paper proposes a four steps method to design Autonomous Vehicles (AV) logical architecture based on the Operational Context (OC). The first two steps focus on the operational analysis, defining and characterizing the OC, and identifying the operational situations. The AV reactive behavior to the situations is modeled as operational processes (OP). In the third step, Functional Chains (FC), realizing the OP, are defined with the help of domain experts knowledge. Finally, the resulting functions are allocated to logical components to obtain the logical architecture of the AV.

The application of the method resulted in reducing the operational analysis time to its third and identifying one and a half time more possible situations in early stage of design. However, the exhaustivity of the situation isn't guaranteed. Substantial domain knowledge and industrial feedback are needed for exhaustive situation generation, which isn't yet the case for AV. Besides, the number of possible situations increases exponentially with the number of OC elements. Future works may focus on the automatic generation of possible operational situations based on a specific OC.

4.6 Acknowledgment

The authors would like to thank the colleagues from AKKA Technologies for their collaboration in the application and evaluation of the proposed method.

4.7 References

Bagschik, G., Menzel, T., Maurer, M., 2018. Ontology based Scene Creation for the Development of Automated Vehicles, in: 2018 IEEE Intelligent Vehicles Symposium (IV). Presented at the 2018 IEEE Intelligent Vehicles Symposium (IV), pp. 1813–1820. https://doi.org/10.1109/IVS.2018.8500632

Bonnet, S., Voirin, J.-L., Exertier, D., Normand, V., 2016. Not (strictly) relying on SysML for MBSE: Language, tooling and development perspectives: The Arcadia/Capella rationale, in: Systems Conference (SysCon), 2016 Annual IEEE. IEEE, pp. 1–6.

Damak, Y., Leroy, Y., Trehard, G., Jankovic, M., n.d. A Context Ontology for Operational Scenarios Generation of Autonomous Vehicles. unpublished.

Dey, A.K., 2001. Understanding and Using Context. Personal and Ubiquitous Computing 5, 4–7. https://doi.org/10.1007/s007790170019

Dumitrache, I., Sacala, I.S., Moisescu, M.A., Caramihai, S.I., 2017. A Conceptual Framework for Modeling and Design of Cyber-Physical Systems. STUD INFORM CONTROL 26. https://doi.org/10.24846/v26i3y201708

Fairley, R.E., Thayer, R.H., 1997. The concept of operations: The bridge from operational requirements to technical specifications. Annals of Software Engineering 3, 417–432. https://doi.org/10.1023/A:1018985904689

Fuchs, S., Rass, S., Lamprecht, B., Kyamakya, K., 2008. A model for ontology-based scene description for context-aware driver assistance systems. Presented at the Proceedings of the 1st international conference on Ambient media and systems, ICST (Institute for Computer Sciences, Social-Informatics and …, p. 5.

Jensen, J.C., Chang, D.H., Lee, E.A., 2011. A model-based design methodology for cyber-physical systems, in: 2011 7th International Wireless Communications and Mobile Computing Conference. Presented at the 2011 7th International Wireless Communications and Mobile Computing Conference, pp. 1666–1671. https://doi.org/10.1109/IWCMC.2011.5982785

Kang, S., Choi, Y., 2005. Designing Logical Architectures of Software Systems, in: Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks (SNPD/SAWN'05). Presented at the Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks (SNPD/SAWN'05), IEEE, Towson, MD, USA, pp. 330–337. https://doi.org/10.1109/SNPD-SAWN.2005.30

Komoto, H., Hamberg, R., Tomiyama, T., 2013. Supporting the Architecting Process of Adaptive Systems, in: Model-Based Design of Adaptive Embedded Systems, Embedded Systems. Springer, New York, NY, pp. 159–188. https://doi.org/10.1007/978-1-4614-4821-1_6

Roques, P., 2016. MBSE with the ARCADIA Method and the Capella Tool, in: 8th European Congress on Embedded Real Time Software and Systems (ERTS 2016). Toulouse, France.

Rosson, M.B., Carroll, J.M., 2009. Scenario-based design, in: Human-Computer Interaction. CRC Press, pp. 161–180.

Sathyanarayana, A., Boyraz, P., Hansen, J.H.L., 2011. Information fusion for robust 'context and driver aware' active vehicle safety systems. Information Fusion 12, 293–303. https://doi.org/10.1016/j.inffus.2010.06.004

Schuldt, F., Reschka, A., Maurer, M., 2018. A Method for an Efficient, Systematic Test Case Generation for Advanced Driver Assistance Systems in Virtual Environments, in: Winner, H., Prokop, G., Maurer, M. (Eds.), Automotive Systems Engineering II. Springer International Publishing, Cham, pp. 147–175. https://doi.org/10.1007/978-3-319-61607-0_7

Sippl, C., Bock, F., Lauer, C., Heinz, A., Neumayer, T., German, R., 2019. Scenario-Based Systems Engineering: An Approach Towards Automated Driving Function Development, in: 2019 IEEE International Systems Conference (SysCon). Presented at the 2019 IEEE International Systems Conference (SysCon), IEEE, Orlando, FL, USA, pp. 1–8. https://doi.org/10.1109/SYSCON.2019.8836763

Taş, Ö.Ş., Kuhnt, F., Zöllner, J.M., Stiller, C., 2016. Functional system architectures towards fully automated driving, in: Intelligent Vehicles Symposium (IV), 2016 IEEE. IEEE, pp. 304–309.

Ulbrich, S., Nothdurft, T., Maurer, M., Hecker, P., 2014. Graph-based context representation, environment modeling and information aggregation for automated driving, in: 2014 IEEE Intelligent Vehicles Symposium Proceedings. Presented at the 2014 IEEE Intelligent Vehicles Symposium Proceedings, pp. 541–547. https://doi.org/10.1109/IVS.2014.6856556

Voirin, J.-L., Tailliez, F., 2012. Method to aid the design of a system architecture. U.S. Patent No. 8,103,490.

Wyatt, D.F., Eckert, C.M., Clarkson, P.J., 2009. Design of product architectures in incrementally developed complex products, in: DS 58-4: Proceedings of ICED 09, the 17th International Conference on Engineering Design, Vol. 4, Product and Systems Design, Palo Alto, CA, USA, 24.-27.08. 2009. pp. 167–178.

5 Paper #4. Operational Context Change Propagation Prediction on Vehicular Cyber-Physical Systems Architectures

Youssef Damak, Yann Leroy, Guillaume Trehard, and Marija Jankovic

This paper is under work to be submitted to Computers in Industry

Abstract. *Vehicular Cyber-Physical Systems (CPS) are designed to operate in a specific Operational Context (OC) and the adaptability of the vehicle's architecture to its OC is considered a major success criterion of the design. Vehicular CPS design projects are rarely started from scratch and are often based on reference architectures. As such, the reference architecture must be modified and adapted when the Operational Context changes. The current literature on engineering change propagation doesn't provide a method to identify and anticipate the impact of Operational Context changes on the Vehicular CPS architecture. This paper proposes a two steps method for Context change propagation: (1) Analyzing the direct impact of Context change with a deterministic method and (2) evaluating the probabilities of indirect change propagation with a probabilistic method on the component level. The direct impact is assessed following a propagation path based upon a model mapping between an Operational Context ontology, operational situations, and Functional Chains. The effects of Functional Chain changes on the Vehicle's components are analyzed and evaluated by domain experts with Types of Changes and associated probabilities. A Bayesian Network is proposed to calculate the probabilities of indirect change propagation between component Types of Changes. The method's applicability and efficiency are validated on a real case design of an Autonomous Vehicle architecture and how it evolves when its Operational Context changes.*

Keywords. *Operational Context change propagation; Vehicular CPS Architecture; Components Types of Changes; Bayesian Network*

5.1 Introduction

Vehicular CPS experimentations increased considerably in later years around the world. Vehicle makers compete constantly to be the first to deliver and industrialize their concept. However, Vehicular CPS design requires new skills related to robotics and automation that are not yet integrated into classical vehicle design. Engineering consulting companies are frequently sought out to help vehicle makers and transportation operators in their Research and Development (R&D) on AV. In this context, they have built considerable body of knowledge related to AV design and experimentation. They rarely start an R&D project from scratch and tend to base their work on reference architectures they previously experimented, developed, and proved efficient.

Due to this reuse of knowledge, the understanding of the system's underlying hypothesis as well as potential interdependencies is necessary. The change to parts of the design often propagates to other parts because of the increasing interdependencies between the system elements. This propagation is hard to identify and anticipate. Hence, there is a need to map out potential propagation paths with regard to new design requirements as well as potential impact identification. This is essential in early design stages where a considerable uncertainty is existing regarding customer's requirements. Therefore, an approach allowing for considering this uncertainty is needed

AV are vehicular systems operating in autonomy to offer various mobility services. The Society of Automotive Engineers (SAE) defines five levels of autonomy with the most advanced being "high automation" and "full automation". These levels correspond respectively to complete autonomy in some driving mode and a complete autonomy in all situations (SAE, 2014). To perform the driving task, AV perceive and interact with their environment with cognitive capabilities and integrated computational and physical capabilities. These characteristics are the reason why they are often considered and studied as Cyber-Physical Systems, as well as Context-Aware Systems. Several papers established the importance of AVs adaptability to their Operational Context (OC) (Bagschik et al., 2018; Damak et al., 2020a; Ulbrich et al., 2014). The authors also noted during immersive observations with an engineering consulting company, that most of the reworking undertaken on AV's components for new experimentations, is conducted to adapt a

reference architecture to the new OC characteristics. To our knowledge, no existing methodology is allowing for estimation of necessary changes to the architecture starting from the changes of the OC. Hence in order to address this gap we propose a method to assess the impact of OC change on a reference AV architecture in the early design phase. The method aims at reducing time and effort of discovering what components require adaptation or transformation for new OC, and what Type of Change (ToC) is required.

We propose the following organization of this paper. The second part of the paper reviews sources of changes and methods for engineering change impact assessment. The third section introduces AV reference architecture model used to evaluate the impact of OC change. The OC change impact assessment method is detailed in the fourth section and in section 5 AV case study evaluating the impact of the change propagation starting from the operational context is discussed. Finally, we discuss the limitation of the method and future research perspectives.

5.2 Related Work

5.2.1 Engineering Change Nature and Source

Engineering Change (*EC*) is one of the most developed scientific research in product and system development (Clarkson et al., 2004; Hamraz et al., 2012; Jarratt et al., 2011; Lee and Hong, 2017; Reddi and Moon, 2009). Jarrat et al. (Jarratt et al., 2011) define it as the process of "making alteration to a product". Authors claim that, contrary to other forms of design iterations, *EC* is operated on the system's parts or software whose design has been considered as finalized. This definition includes changes occurring before the release of the system, such as changes to prototypes.

There are several types of EC. Eckert et al.(Eckert et al., 2004) classify changes in two types: 1) *initiated* change coming from changes occurring outside the system, such as stakeholder's requirements changes, and 2) *emergent* change arising to correct issues in the system. Some examples of *Initiated* change can be changes in customer requirements, certification requirements, and innovations. Jarratt et al. (Jarratt et al., 2011) further develop the general classification of both change types. They categorize emergent changes

according to their change nature, such as change of function or error correction. Initiated changes are classified according to the different stakeholders initiating the change.

The studies on different EC propagation approaches bring to light several sources of changes. New component requirements have been considered as a primary source of structural changes undergone by system components. Clarkson et al. (Clarkson et al., 2004) and Cheng and Chu (Cheng and Chu, 2012) consider a component change as the initiator of change propagation. Reddi and Moon (Reddi and Moon, 2009) further characterize the "Type of Change (ToC)" of a component (component attributes such as material, shape, size, geometry) to estimate change propagation. Some studies refines these into system parameters (Ollinger and Stahovich, 2004; Xie and Ma, 2016; Yang and Duan, 2012). Olinger and Stahovich (Ollinger and Stahovich, 2004) and Yang and Duan (Yang and Duan, 2012) consider in their studies various types of parameters such as size, friction, stress, piston speed, maximum pressure, injection period, and spring force. Xie and Ma (Xie and Ma, 2016), on the other hand, consider feature parameters and constraints associated with these features. The notion of feature parameters is associated in this study with feature modeling and is represented with a set of variables but wasn't formally defined.

Besides components requirements, changes in functional requirement are also studied as primary sources of EC. Fei et al. (Fei et al., 2011) and Ahmad et al. (Ahmad et al., 2013) study similar approaches where the change is initiated by functional requirements. Functional requirements are mapped onto functions that are further mapped onto components. Hamraz et al. (Hamraz et al., 2012) further extend the consideration of several types of changes; they consider that the change can occur in functional requirements, components and component behavior.

Moreover, Koh et al. (Koh et al., 2012) address the system performance requirements as sources for change propagations. Morkos et al. (Morkos et al., 2014, 2012) also consider system requirement, however, they propose a method to consider change propagation but only between system requirements.

5.2.2 Engineering Change Impact Assessment

In this study we particularly focus on different elements that are considered in change propagation methods in order to review their relevance and the possibility of estimating changes propagations from the operation context modelling. EC propagation is the process where a change in one system element propagates to another. It combines the direct impact of one element change on another and the combination of indirect impacts through different elements (Clarkson et al., 2004). Clarkson et al. (Clarkson et al., 2004) have proposed the Change Prediction Method (CPM) that is considered as a significant scientific reference in the matrix-based domain. The aim of the CPM is to support the prediction of a change propagation in one system based upon the information related to its structural aspect (interfaces) as well as the importance and likelihood of the change propagating. Data gathered for the importance and likelihood of one change are provided by domain experts based upon previous projects. CPM is a probabilistic method supporting the likelihood evaluation of the direct and indirect impacts of a single component change onto other components.

Cheng and Chu (Cheng and Chu, 2012) propose a method to assess EC propagation for complex systems based on the structural connections between components. They use a weighted component network to perform typical network related analysis (such as degree analysis) in order to estimated overall direct and indirect change propagation. Edges in this network represent an aggregated information related to different flows between components i.e. the Coupling Index proposed by Martin and Ishi (Martin and Ishii, 2002). This Coupling Index aims at representing information related to component sensitivity with regard to these different flows. Flows between the components are also used by Hamraz et al. (Hamraz et al., 2013) to compute the likelihood and impact of CPM's component DSM. Using this approach, they enhance interface management controlling interface incompatibilities generated from EC propagation.

Hamraz et al. (Hamraz et al., 2012) propose a method based on CPM to assess the change propagation between functions, component behaviors, and component structures. They use the Function-Behavior-Structure (FBS) model proposed by Gero (Gero, 1990; Gero and Kannengiesser, 2014) as a basis for elements that are considered in this change

propagation method. Component behavior here is defined as properties of its structural elements, such as weight, noise, heat. The component behavior is mapped onto the structural aspect of one component, such as geometry, and material. To do so, Multi-Domain Matrices (MDM) are proposed to map relationships between these three domains. MDM are matrices that combine several types of elements, represented as a combination of Design Structure Matrices (DSM, matrices representing the relationships between the elements of one type; e.g. component onto component) and Domain Mapping Matrices (DMM, mapping elements of different types; e.g. function onto component). Koh et al. (Koh et al., 2012) extend CPM to assess the impact of change in one system component onto system level performances. Authors note that several components contribute to the satisfaction of a requirement on a system-level performance and that a component would undergo different types of changes depending on the changing requirement. As such, they introduce the notion of "change option" corresponding to a type of change applied to one component; and estimate its direct impact onto system requirements. Afterwards, they propose to evaluate the overall impact of a change option on the system requirements using the propagation likelihood between component, the correlation between the change options, and the direct impact of all the change options on the system requirements.

Reddi and Moon (Reddi and Moon, 2009) propose a network-based change propagation method addressing similar aspects to change options. Using an object-oriented database, they model EC propagation based upon component "Types of Changes (ToC)". The dependencies between components are represented with four classes: Initiator, Target, Type of Change (ToC), and Likeliness. The information of dependencies and their characterization is estimated by designers and experts in the design phase in order to be reused afterwards for the EC assessment.

Fei et al. (Fei et al., 2011) propose a multidimensional propagation method allowing an identification of propagation path from functions to components, integrating different component flows and spatial aspects of one architecture. The data for flows and spatial connections are extracted from a SysML model of the system architecture. The proposed method also allows for identification of components that are indirectly impacted using the flow data.

Ahmad et al. (Ahmad et al., 2013) propose an Information Structure Framework (ISF) that includes functional requirements, function structure, component structure, and detailed design process. This method represents also an extension of the CPM method. The change is initiated in functional requirements and propagates to functions. The allocation of functions to components is then used to identify components that are subject to change. This is the only method that we found integrating the possibility to estimate overall changes that can have multiple change sources within a probabilistic framework. Enhanced CPM probabilistic algorithm is proposed to account for this estimation. The difficulty lies in the fact that the choice of function to combine multiple likelihood effects is arbitrary within a set of conditions.

Previously discussed literature underlines the lack of matrix-based change propagation approaches allowing for a probabilistic impact assessment of multiple and simultaneous changes. This is a significant issue, as changes are often applied simultaneously. To solve this issue, Lee and Hong (Lee and Hong, 2017) propose a Bayesian Network (BN) model allowing for change propagation assessment between components where change can be initiated by one or multiple components simultaneously. The BN is built using the same structural data used in CPM, converting the direct propagation likelihood into conditional probabilities. The advantage of using the Bayesian Network is the possibility of using learning algorithms that are based upon empirical data on the likelihood of direct propagation.

Other authors explored network theories to assess EC impact. Ollinger and Stahovich (Ollinger and Stahovich, 2004) propose RedesignIT using causal network to assess the impact of changes between different system parameters. The method allows for identification of possible change plans with regard to new requirements. Yang and Duan (Yang and Duan, 2012) proposed a Parameter Linkage Network to evaluate change routing and diffusion between the system parameters. The Parameter Linkage Network represents the links between the system parameters defined by physical laws or defined by designers. Similarly, Xie and Ma (Xie and Ma, 2016) propose a network composed of two types of nodes: component feature parameters and constraints. The feature parameters are only linked to given constraints. The evaluation is done through a progressively expanded

constraint satisfaction problem (PECSP) on the association network. The EC propagation is invoked only when a constraint satisfaction problem is violated.

The different EC propagation methods and frameworks of the literature propagate the change through models linking between the observed elements. Early methods focused on linking elements with the same nature, such as components to components (Clarkson et al., 2004), or parameters to parameters (Ollinger and Stahovich, 2004). Later, multidimensional propagation are introduced using linkage such as the Function-Behavior-Structure (FBS) mapping (Hamraz et al., 2012), functions-flows-components mapping (Fei et al., 2011),or requirements-components mapping (Koh et al., 2012).

Previously discussed literature underlines that EC propagation methods and frameworks use models mapping elements that are subject to changes. Early methods focused on linking elements of the same nature, such as components to components (Clarkson et al., 2004), or system parameters to system parameters (Ollinger and Stahovich, 2004). In later work, multidimensional propagation are introduced using linkage such as the Function-Behavior-Structure (FBS) mapping (Hamraz et al., 2012), functions-flows-components mapping (Fei et al., 2011), or requirements-components mapping (Koh et al., 2012). To the best of our knowledge, no method permits change propagation assessment initiated by the changes in the Operational Context. Current design methods underline the need to assess this propagation from the Operational Context onto overall system architecture (as defined by Crawley (Crawley et al., 2004), consisting of several elements such as functions, components, interfaces, etc.) in the case of Autonomous Vehicle development. To address this gap, this research proposes a model and the methods to support this estimation.

5.3 Linking the Operational Context to Autonomous Vehicles Architecture

The definition of the Operational Context has been discussed in a previous study along with an extensive literature review on OC models (Damak et al., 2020a). An operational context ontology was proposed in order to describe scenes, situations, and scenarios for an operating AV and helps identify the vehicle's reactive behavior to its environment. It is

structured in 5 levels: (0) Use case, (1) Environment, (2) Road Infrastructure, (3) Traffic Infrastructure, and (4) Traffic Objects. The aim of this ontology is to support the design of the AV. Hence the model to link this information in the design was proposed (Figure 5.1 describes the mapping between the Operational Context and the AV reference architecture elements (Damak et al., 2020b)). Operational situations are defined on the basis of the elements of the OC ontology. Each situation is associated to a Functional Chain (FC) modeling of the internal process of the AV facing the situation. FCs involve functions, functional interactions, and constraints realized by logical components divided in three types: actuators, sensors, and software components. The change in this case is initiated in the operational context as the addition or removal of one element of the ontology, or the change in its attributes. The model represented in Figure 5.1 is used in order to identify possible propagating paths in order to support their management.

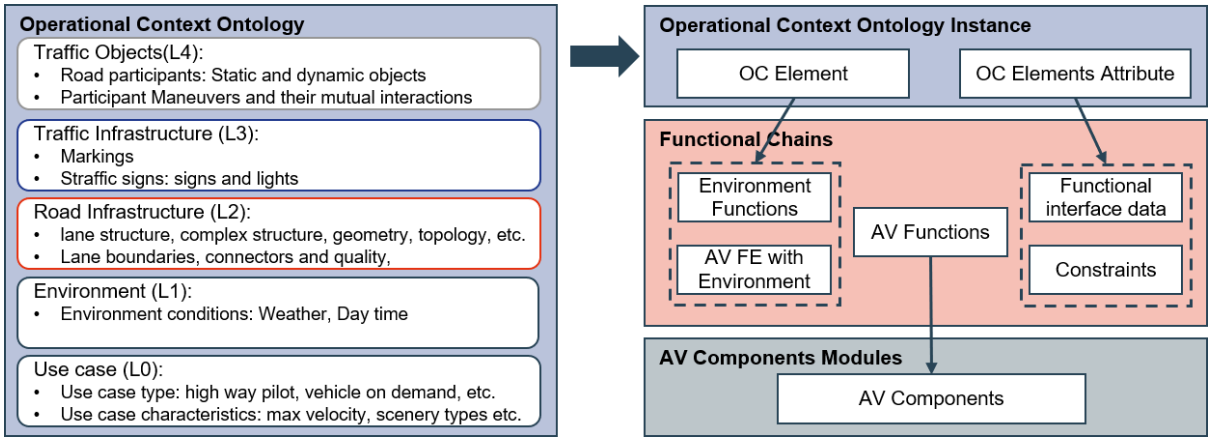


Figure 5.1: AV reference architecture model for OC change propagation

5.4 Operational Context Change Impact Assessment on Autonomous Vehicles Architecture

The objective of this research is to support the design teams in identifying possible system architecture changes with regard to the changes demanded in the Operational Context. Hence, the assessment of OC change impact on AV architecture is organized in two steps: 1) Identify and evaluate the probability of direct impacts and 2) Estimate the probabilities of indirect impacts. The step one is used to identify the probability of the initial types of changes and as an input to the second step. The second step aims at refining the evaluation

of the probability of changes onto a given component with regard to the changes that are defined in the operational context integrating the information of component interdependence.

5.4.1 Direct Impact Assessment of Operational Context Change on AV Components

The Operational Context introduces use cases with several situations handled by the vehicle. The change of an OC element changes the situations defined to be encountered by the vehicle. These changes can further be propagated (Figure 5.1) onto Functional Chains. The method to analyze direct impact is proposed in Figure 5.2: (1) Identify a changing OC element's Type of Change (ToC), (2) Trace the impact on FC associated with the changing item and analyze its effect on the involved system functions. Step (3) Evaluate the impact probabilities of function definition modification onto associated components.

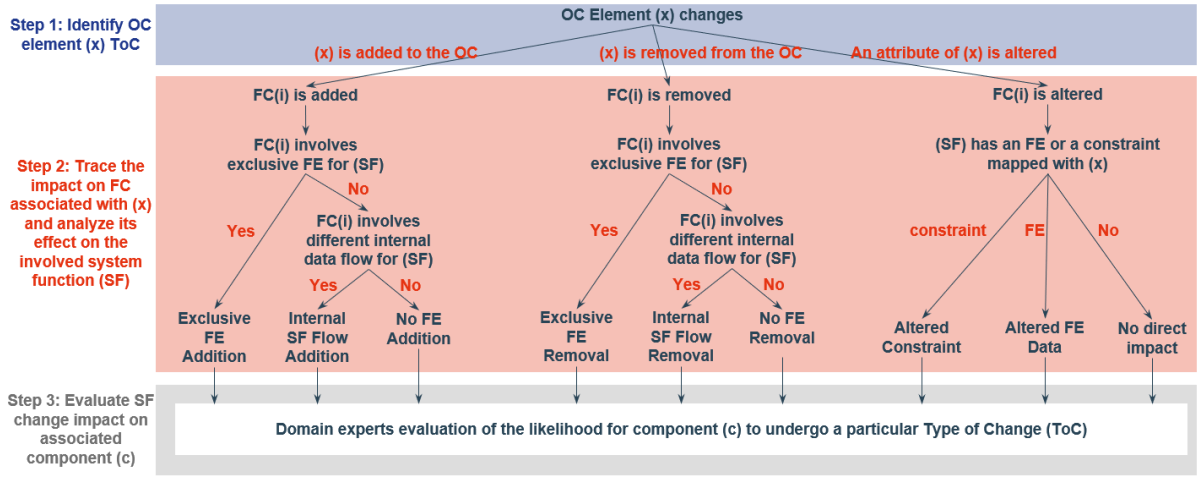


Figure 5.2: Analysis tree of the direct impact of OC element change on AV Architecture

In **Step 1**, we differentiate three ToCs for OC elements: the element's addition or removal from the OC, or an alteration of its attributes.

The **Step 2** is to identify the consequence of an OC element change onto every Functional Chain mapped to the element and to analyze its propagation onto System Function definition and modelling. Functional Chains, (modeled with Capella (Roques, 2016)) are defined as a set of functions and functional flows (they are noted in Capella as Functional

Exchanges (FE) linking output and input ports (see Figure 5.3)). FCs may also involve constraints defined on a system function or an FE (noted {c} in Capella).

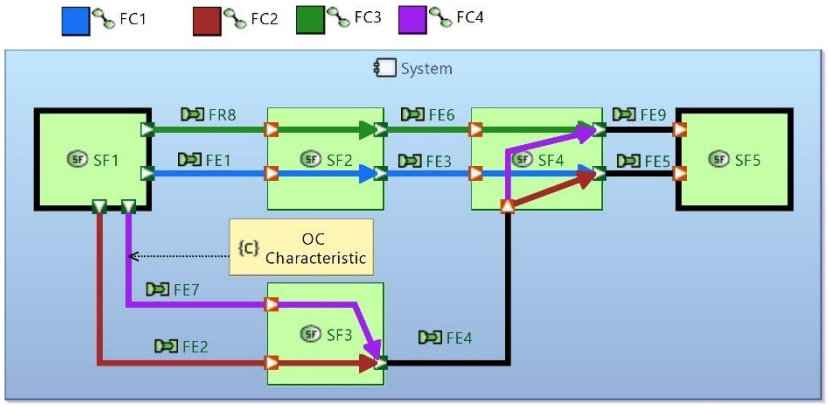


Figure 5.3: Example of Functional Chain impact on system functions

Possible impacts on FC are the following (illustrated using FC4 displayed with a purple flow in Figure 5.3):

1. An OC element is added: One or several new situations are included in the operations of the AV. Hence, an *FC* is added to integrate the definition of this new situation. The addition of the FC may impact the involved functions in three different ways:

   1.1. *Exclusive FE Addition (E.FE.A)*: The added *FC* introduces a new input or output port to the function defining a new *FE*. If one considers FC4 addition, this will add the FE7 between SF1 and SF3.

   1.2. *Internal SF Flow Addition (I.SF.F.A):* The added FC uses already defined FE with a different data flow. If one considers the addition of the FC4, there is a need to add a new internal flow to SF4 going from the FE4 output port to the FE9 input port.

   1.3. *No FE Addition* (N.FE.A): The added FC uses already defined FE and data flow.

2. An OC element is removed: One or many situations are removed from the operations of the AV. Hence, the related FC are removed. The removal of the FC may impact the involved functions in three different ways:

2.1. *Exclusive FE Removal (E.FE.R):* A removed FC induces the removal of FEs and the changes on the corresponding input and output ports. If FC4 is removed, then FE7 and the corresponding input port of the function SF3 and output port of the function SF1 need to be removed.

2.2. *Internal SF Flow Removal* (I.SF.F.R): The removed FC uses the same FE as other FCs, with a different data flow. If one considers the removal of the FC4, there is a need to remove the internal flow of SF4 going from the FE4 output port to the FE9 input port.

2.3. *No FE Removal* (N.FE.R): The removed FC uses the same FE and data flow as other FCs.

3. The attributes of an OC element are altered: As shown in the model presented in Figure 5.1, the OC element attributes define the constraints and FE data involved in Functional Chains. Hence, the change of an OC element attribute may imply the two following changes:

3.1. *Altered Constraint (A.C):* A constraint involved in an FC is altered due to the change in OC element attribute. If one considers the constraint {c} on FE7 that is defined with regard to an OC element attribute, the alteration of this attribute will modify the constraint on FE7.

3.2. *Altered FE Data (A.FE.D):* The data of an FE involved in a FC is altered due to the change in OC element attribute. If FE7 is defined given an OC element attribute, then the alteration of the attribute modifies the definition of its data.

Step 3 identifies and assesses the change from modification in FCs onto AV components. AV architecture consists of: sensors, actuators and software components. Due to their different nature and changes that can be different, we propose to distinguish Types of Changes (ToC) with regard to these component types. For sensors and actuators, we propose to consider the changes in performances and physical properties (see reference (Hamraz et al., 2012; Reddi and Moon, 2009)). As for software components, Chapin et al. (Chapin et al., 2001) categorize changes to the software component in 2 categories: 1)

software properties change and 2) business rules change. Within these two categories, authors define following types of changes: groomative, preventive, performance, adaptive, corrective, reductive, and enhancive. Groomative and preventive changes are related to improving software maintainability while corrective changes are related to errors are identified in the software. Hence, as we consider the design phase, these three types are out of the scope of OC change propagation. As such, we propose to consider only the following: performance, adaptive, reductive, and enhancive.

The identification of the possible FCs change impacts is assessed by different experts. Domain experts and engineers are also demanded to evaluate the probabilities of changes for every ToC and component type (see Table 5.1). An excerpt of this type of the evaluation is summarized in a Domain Mapping Matrix (DMM), with the resulting probabilities of changes for sensors, actuators, and software components with respect to the changes in their system functions.

The change of the operational context may affect several FCs, which in turn propagate to system function definitions. In some cases, a single system function may be impacted by multiple FCs changes. For each change in system function, experts evaluate the likelihood and Types of Changes for the impacted components. The difficulty lies in having multiple impacts onto one component Type of Change. In this case, the highest likelihood is considered in order to consider the maximal impact.

Table 5.1: AV components change probability with respect to the Functional Chain's effect on the components

AV Component	**Component ToC**	**OC element ToC**							
		Addition			**Removal**			**Alteration**	
		E.*FE*.A	I.SF.F.A	N.FE.A	E.FE.R	I.SF.F.R	N.FE.A	A.C	A.FE.D
Sensor	Ph. Property	0.5	0	0	0.5	0	0	0.2	0.3
	Performance	0.5	0.3	0.2	0.5	0.3	0.2	0.5	0.2
Software	Performance	0	0.4	0.2	0	0.4	0.2	0.4	0.2
	Adaptive	0	0.4	0.2	0	0.4	0.2	0	0.4
	Reductive	0	0	0	1	0	0	0	0
	Enhancive	1	0	0	0	0	0	0	0
Actuator	Ph. Property	0	0	0	0	0	0	0	0.2
	Performance	0.7	0.5	0.1	0.7	0.5	0.1	0.5	0.2

5.4.2 Indirect Impact Assessment of Operational Context Change based upon Bayesian Network-based Propagation

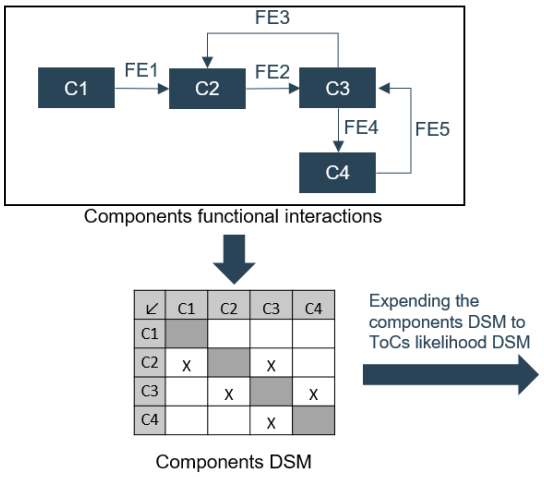
In order to integrate the interdependence between the components and possible propagation of impacts due to these interdependencies, we propose to use the data from the previous step in order to estimate indirect impacts. The major challenge in this step is to consider multiple initial sources of changes. In order to address this challenge, the literature review underlines very few methods allowing for this evaluation. In particular, Bayesian networks have been identified as an interesting approach to do so (Lee and Hong, 2017). Hence, we propose to develop a Bayesian network approach in order to estimate Operational Context indirect impacts onto Component Types of Changes.

The proposed Change Propagation Bayesian Network (CP-BN) is inspired from the work of Lee and Hong (Lee and Hong, 2017). It is based on the data on the direct change propagation likelihood from a ToC onto another to calculate indirect propagations. The generation of the CP-BN consists of the three following steps: (1) building a propagation likelihood DSM for the Component Types of Changes, (2) creating the nodes and edges of the Bayesian Network, (3) computing the nodes Conditional Probability Tables (CPT).

Step 1 consists of identifying the direct propagation probabilities between ToCs and building a propagation likelihood DSM. The AV reference architecture is used to develop a component dependency DSM based on the Functional Exchanges between components. In the case, the dependence between the two components is the Functional dependence and stems from the Functional Exchange diagram. The information on component dependence presented in Figure 5.4 is deduced from FEs and components allocated to each function. The component DSM is afterwards expanded into the ToCs likelihood DSM by adding the Types of Changes regarding the component types (cf. section 5.4.1). Here, as one can see, the component DSM is directed matrix where a change in component (j) may propagate to the component (i). Domain experts and engineers evaluate the likelihood of direct change propagation for each ToC of component (j) onto every ToC of component (i). An example of this evaluation is presented in the ToCs likelihood DSM of Figure 5.4 (right hand side). Figure 5.4 illustrates the transformation of a component DSM

into a ToCs likelihood DSM with a simplified example of 4 components: two software, one sensor, and one actuator.

The **step 2** to generate the Change Propagation BN is to create the nodes and edges of the network. Lee and Hong propose to generate nodes representing the components and their changing state (Lee and Hong, 2017). For OC change propagation, we propose to develop Bayesian Network where Boolean nodes represent Component ToCs. Figure 5.5 illustrates how to generate a CP-BN by aligning ToC nodes in four propagation steps, as four steps are considered a practical limit for EC propagation (Clarkson et al., 2004). Each ToC (j) of a component C(i) is represented with four nodes in four propagation steps as: $ToC_{j,C(i)}^{t}$, for t in {1, 2, 3, 4}. Each node $ToC_{j,C(i)}^{t}$ is a Boolean node indicating the probability that C_i undergoes the ToC (j) at the propagation step t. An edge from $ToC_{j,C(i)}^{t-1}$ to $ToC_{x,C(y)}^{t}$ is created, for t in {2, 3, 4}, if the ToCs likelihood DSM indicates a strictly positive probability of change propagating from the ToC(j) of component C(i) to the ToC(x) of component C(y).



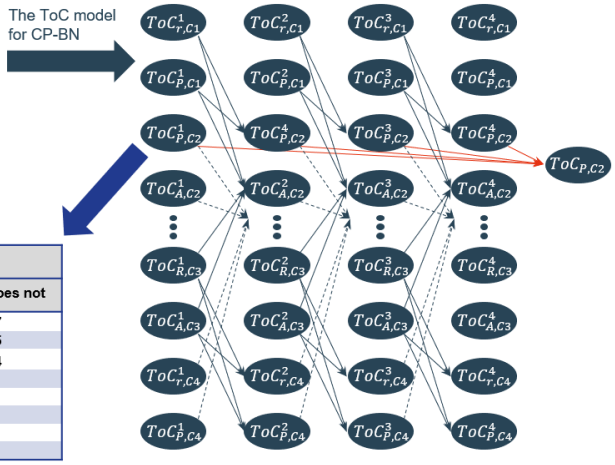
↙	C1	C2	C3	C4
C1				
C2	X		X	
C3		X		X
C4			X	

Component type	↙		C1		C2				C3				C4	
		Type of Changes	PPr	P(H)	P(S)	A(S)	R(S)	E(S)	P(S)	A(S)	R(S)	E(S)	PPr	P(H)
Sensor	C1	Physical Properties												
		Performance (H)												
Software	C2	Performance (S)	0.3	0.5					0.8	0	0	0		
		Adaptive	0.3	0.2					0	0.5	0.8	0.8		
		Reductive	0	0					0	0	0	0		
		Enhancive	0	0					0	0	0	0		
Software	C3	Performance (S)			0.8	0	0	0					0.3	0.3
		Adaptive			0	0.5	0.8	0.8					0.5	0.3
		Reductive			0	0	0	0					0	0
		Enhancive			0	0	0	0					0	0
Actuator	C4	Physical Properties							0	0	0.2	0.2		
		Performance (H)							0.2	0.2	0.5	0.5		

ToC likelihood DSM

Figure 5.4: Generation of a components DSM and it transformation into a ToCs likelihood DSM

Component type		Type of Changes	C1		C2				C3				C4	
			PPr	P(H)	P(S)	A(S)	R(S)	E(S)	P(S)	A(S)	R(S)	E(S)	PPr	P(H)
Sensor	C1	Physical Properties												
		Performance (H)												
Software	C2	Performance (S)	0.3	0.5					0.8	0	0	0		
		Adaptive	0.3	0.2					0	0.5	0.8	0.8		
		Reductive	0	0					0	0	0	0		
		Enhancive	0	0					0	0	0	0		
Software	C3	Performance (S)			0.8	0	0	0					0.3	0.3
		Adaptive			0	0.5	0.8	0.8					0.5	0.3
		Reductive			0	0	0	0					0	0
		Enhancive			0	0	0	0					0	0
Actuator	C4	Physical Properties							0	0	0.2	0.2		
		Performance (H)							0.2	0.2	0.5	0.5		



CPT			ToC(P,C2)^{t}	
ToC(PPr,C1)$^{t-1}$	ToC(P,C1)$^{t-1}$	ToC(P,C3)$^{t-1}$	Occurs	Does not
Occurs	Occurs	Occurs	1-(1-0.3)x(1-0.5)x(1-0.8)=0.93	0.07
		Doesn't	1-(1-0.3)x(1-0.5)=0.65	0.35
	Doesn't	Occurs	1-(1-0.3)x(1-0.8)=0.86	0.14
		Doesn't	0.3	0.7
Doesn't	Occurs	Occurs	1-(1-0.5)x(1-0.8)=0.9	0.1
		Doesn't	0.5	0.5
	Doesn't	Occurs	0.8	0.2
		Doesn't	0	1

Figure 5.5: Change Propagation Bayesian Network generation from Types of Changes likelihood DSM

The data from the ToCs likelihood DSM are used in **step 3** to calculate the Conditional Probability Tables of the CP-BN nodes. Assuming that different sources of a component change are independent, the CPT can be calculated with the Noisy-OR model (Lee and Hong, 2017). Hence, the calculation of the CPT for the proposed TOC Bayesian network is as follows: if a ToC_i has a direct propagation of change from components ToC_1, ToC_2, … ToC_k with a likelihood represented in the ToCs DSM as $L_{i,1}$, $L_{i,2}$, …$L_{i,k}$, then the conditional probability of ToC_i given that every parent changes is:

$$p(ToC_i^t | ToC_1^{t-1} = changes, ToC_2^{t-1} = changes, \dots, ToC_k^{t-1} = changes) = 1 - \prod_{j=1}^{k} (1 - L_{i,j}) \quad (1)$$

As edge structure is invariant through the propagation steps, the conditional probabilities of all the ToC_i^t, for t in {2, 3, 4}, with respect to their predecessors, are equals. Figure 5.5 illustrates the example of the CPT for the nodes $ToC_{P,C2}^t$, with the predecessors $ToC_{P,C1}^{t-1}$, $ToC_{r,C1}^{t-1}$, and $ToC_{P,C3}^{t-1}$. The probability of C2 undergoing a performance change (ToC_P) at step t given the ToCs of both C1 and C3 in step (t-1) is calculated with the Noisy-OR model. The probabilities that C2 does not undergo a performance change (ToC_P) at step t is the complementary, given the same conditions. In order to evaluate the overall

probability of the component (i) undergoing a ToC(j) after the change propagation, we propose to define a final node $ToC_{j,Ci}$, successor to all the $ToC_{j,Ci}^{t}$, for t in {1, 2, 3, 4}. The states of $ToC_{j,Ci}$ are directly deduced from the $ToC_{j,Ci}^{t}$ as follows: if any of the $ToC_{j,Ci}^{t}$ is true, then $ToC_{j,Ci}$ is true. As such, $ToC_{j,Ci}$ would be always true, except if all $ToC_{j,Ci}^{t}$ are false. Finally, to assess the probability that a component (i) undergoes a ToC(j) given the direct impact of OC change propagation, probability inference is used to evaluate the state of the final node $ToC_{j,Ci}$.

5.5 Case Study

In order to illustrate the proposed approach, an AV system architecture real case design has been used (Damak et al., 2020b). This project is developed by AKKA Technologies and in particular designers and engineers from the autonomous systems team. The reference architecture of this case study corresponds to an OC defined for the "Vehicle on demand" use case and containing traffic lights, roundabouts, as well as other vehicles and obstacles. Initial Functional Chains describing this case are the following:

- *FC01 - Primary driving:* The AV drives up following a city road lane, with no encounter.
- *FC02 - Traffic light encounter:* The AV drives up and approaches a traffic light. The traffic light encounter is managed by the AV.
- *FC03 - Obstacle detected:* The AV drives up, and an obstacle gets on its path. The obstacle encounter is managed by the AV.
- *FC04 - Vehicle detected:* The AV drives up, and the vehicle is detected on its path. The vehicle encounter is managed by the AV.
- *FC05 - Roundabout approaching:* The AV drives up and approaches a roundabout. The roundabout approach is managed by the AV.
- *FC06 - Roundabout entering with no vehicle:* The AV arrives at a roundabout with no vehicle or obstacle in its entrance. The AV enters the roundabout.

- *FC07 - Roundabout entering with vehicles:* The AV arrives at a roundabout with a vehicle passing by its entrance. The vehicle's encounter and roundabout entry are managed by AV.

The reference architecture consists of 40 functions organized in 11 *FC*. These functions are allocated to 20 software components for the Perception, Control, Decision, and storage modules; a set of environment sensors with cameras and Lidars; 5 vehicle's onboard sensors; and 5 actuators.

In this case, the design team needed to integrate to the Operational Context pedestrians and pedestrian crossing because the initial architecture did not integrate them, and the new testing ground presented several of them. The addition of pedestrians and pedestrian crossing to the OC introduced 2 new situations: "Pedestrian crossing approaching" and "Passing the pedestrian crossing", resulting in two additional FCs:

- *FC08 - Pedestrian crossing approaching:* The AV drives up and approaches a pedestrian crossing. The vehicle decelerates.

- *FC09 - Passing the pedestrian crossing:* If pedestrians are crossing or waiting to cross on the side of the pedestrian crossing, the AV strops, wait for them to cross, then drives up. If there are no pedestrians, the AV drives up.

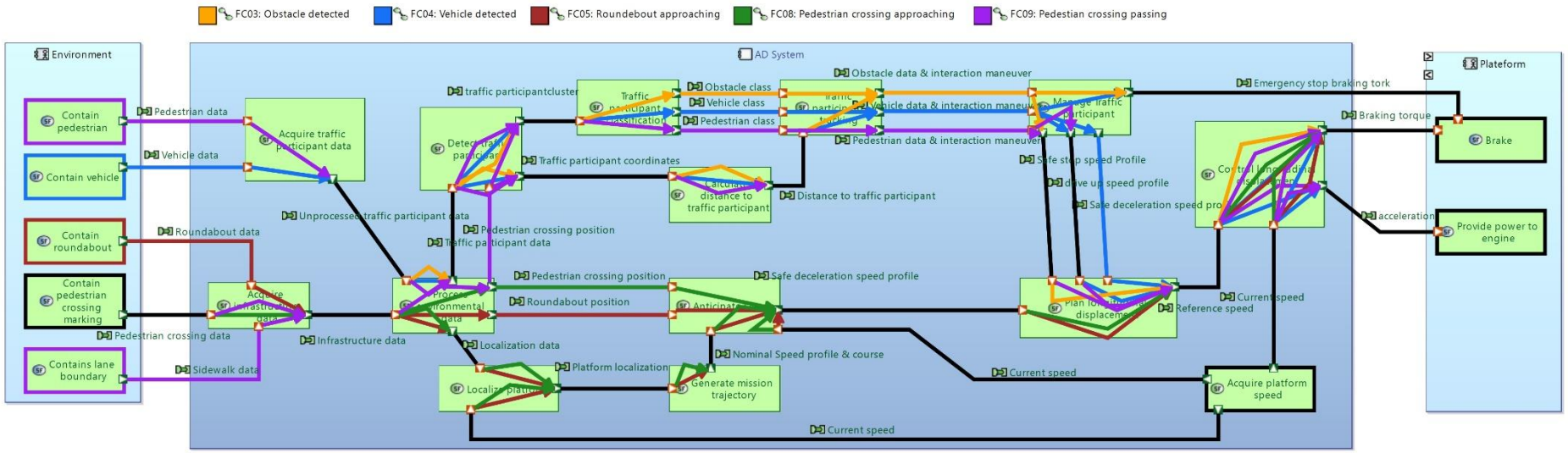


Figure 5.6: Case Study's Functional Chain Modeling: FC03, FC04, FC05, FC08, and FC09

Figure 5.6 shows the two new FCs modeled in perspective with FC03, FC04, and FC05 that share involved functions. The impact analysis of both FCs addition (cf. section 5.4.1) results in *Exclusive FE Additions* to the functions "Acquire traffic participant data", "Detect traffic participant", "Classify Traffic participant", "Track Traffic participant", "Manage Traffic participant", "Acquire infrastructure data", "Process environmental data", and "Anticipate Events". It also affects with an *Internal SF Flow Addition* the functions "Plan longitudinal displacement", and with *No FE Addition* the functions "Localize platform", "Generate mission trajectory", "Calculate distance to traffic participant", "Control longitudinal displacement", "Provide platform speed", " Apply braking torque", and "provide fuel to engine".

These changes have been analyzed by the experts in order to identify and characterize direct impacts (Table 5.2). Table 5.2 provides the likelihood and ToCs for a sample of the concerned components, which represents their initial probabilities of change in the indirect impact assessment.

Table 5.2: A sample of components ToC likelihood from OC change's direct impact

Functions	**Components**	**Type of Change**	**Likelihood**
Acquire traffic participant data	Environment sensors	Ph. Property	0.5
Acquire infrastructure data		Performance	0.5
Classify Traffic participant	Traffic participant classifier	Enhancive	1.0
Manage Traffic participant	Traffic participant manager	Enhancive	1.0
Anticipate Events	Events forecaster	Enhancive	1.0
Plan longitudinal displacement	Longitudinal planner	Performance	0.4
		Adaptive	0.4
Calculate distance to traffic participant	Distance to traffic participant Calculator	Performance	0.2
		Adaptive	0.2
Control longitudinal displacement	Longitudinal control	Performance	0.2
		Adaptive	0.2
Apply braking torque	Brake actuator	Performance	0.1

The component dependency DSM illustrated in Figure 5.7 was obtained from the functional interactions between system components. For each dependency introduced by the components DSM, domain experts evaluate the likelihood of a Type of Change propagating onto another. An extract of the resulting ToCs likelihood DSM is proposed in Figure 5.8. As an example of the experts evaluation, the components DSM shows that the "traffic light manager" (C20) is functionally dependent on "traffic light communication component" (C7) and "Environment processing component" (C8). C7 is a sensor component and may undergo physical property or performance changes. On the other hand, C8 and C20 are software components and may undergo performance, adaptive, reductive, and enhancive changes. As such, a reductive or enhancive change in C8 is very likely to propagate onto an adaptive change for C20. The probability of change has been estimated by experts to be 0.8. On the other hand, a performance change in C7 is less likely to propagate onto a performance and adaptive change for C20, estimated respectively 0.5 and 0.2. This can be justified by the fact that the improvement of a sensors performance does not often necessitate a change of the software receiving and processing the data.

			C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22	C23	C24	C25	C26	C27	C28	C29	C30	C31
Environment Sensors	Cameras & Lidars	C1																															
	Mission acquisition HMI	C2																		X													
	Platform speed sensors	C3																															
	Wheel sensors	C4																															
	handbrake sensor	C5																															
Vehicle Onboard Sensors	Platform global positioning (GPS)	C6																															
	Traffic light communication componen	C7																															
	Environment processing component	C8	X																														
	Platform localization component	C9			X	X		X		X																							
	Traffic participant Detector	C10								X																							
	Distance to participant Calculator	C11									X																						
	Traffic participant classifier	C12									x																						
Perception Module	Traffic participant tracker	C13										X	X																				
Storage Module	Map & trajectory database	C14		X												X																	
	Mission planner	C15		X						X					X																		
	Mission trajectory generator	C16								X	X				X	X																	
	Events forecaster	C17			X					X							X																
	Complex structure manager	C18			X					X				X			X																
	Platform departure manager	C19		X			X																										
	Traffic light manager	C20			X				X	X							X																
	Traffic participant manager	C21												X																			
	Lateral planner	C22															X		X														
Decision Module	Longitudinal planner	C23															X	X	X	X	X	X											
	Lateral controller	C24																						X									
	Longitudinal control	C25			X																				X								
	Operations monitor	C26																								X	X						
Control Module	Turn indicators soft	C27															X																
	Engine actuator	C28																									X						
	Wheel actuator	C29																								X							
	Brake actuator	C30																						X			X						
	Turn indicator	C31																										X					
Actuators	Operations HMI	C32																									X						

Figure 5.7: Case Study's components dependency DSM

				C5		C6		C7		C8					C24				C25			
				PhP	P	PhP	P	PhP	P	P	A	R	E		P	A	R	E	P	A	R	E
	Complex structure manager	C18	P							0.8												
			A								0.5	0.8	0.8									
			R																			
			E																			
	Platform departure manager	C19	P	0.3	0.5																	
			A	0.3	0.2																	
			R																			
			E																			
	Traffic light manager	C20	P					0.3	0.5	0.8												
			A					0.3	0.2		0.5	0.8	0.8									
			R																			
			E																			
	Wheel actuator	C29	PhP														0.2	0.2				
			P												0.2	0.2	0.5	0.5				
	Brake actuator	C30	PhP																		0.2	0.2
			P																0.2	0.2	0.5	0.5

Figure 5.8: Extract from the Types of Changes likelihood DSM

A CP-BN with 412 nodes was generated, representing 103 nodes on every propagation step. By adding the final nodes representing the overall probability of ToCs occurrence, the CP-BN contained 515 nodes (cf. section 5.4.2). We initialize the likelihood of the first step nodes with the results of the direct impact analysis (Table 5.2). Running a probability inference on the CP-BN resulted in the probabilities of occurring ToCs.

Table 5.3 shows that several software components present a probability of 1 to undergo an enhancive change, which means that the component must be enhanced to adapt to the new Operational Context and to the addition of new Functional Chains. The table also illustrates a large variation of ToC probabilities, from low ones (around 0.3) to higher ones (above 0.8). For instance, an adaptive change for the "Complex structure manager" (C18) is very likely to be needed with a probability of 0.926. On the other hand, a performance change to the "Platform speed sensors" (C3) may be needed but with a low probability (0.2).

Table 5.3: Assessment result of the impact of Pedestrian and Pedestrian Crossing addition to the OC on a reference AV architecture

Component		ToC	Likelihood
Cameras & Lidars	C1	Ph. Properties	0.500
		Performance	0.500
Platform speed sensors	C3	Performance	0.200
Environment processing component	C8	Performance	0.363
		Adaptive	0.235
		Enhancive	1.000
Platform localization component	C9	Performance	0.489
		Adaptive	0.864
Traffic participant Detector	C10	Performance	0.290

		Adaptive	0.832
		Enhancive	1.000
Distance to participant Calculator	C11	Performance	0.200
		Adaptive	0.900
Traffic participant classifier	C12	Adaptive	0.880
		Enhancive	1.000
Traffic participant tracker	C13	Performance	0.160
		Adaptive	0.935
		Enhancive	1.000
Map & trajectory database	C14	Performance	0.128
		Adaptive	0.050
Mission planner	C15	Performance	0.227
		Adaptive	0.464
Mission trajectory generator	C16	Performance	0.574
		Adaptive	0.926
Event forecaster	C17	Performance	0.532
		Adaptive	0.910
		Enhancive	1.000
Complex structure manager	C18	Performance	0.592
		Adaptive	0.989
Traffic light manager	C20	Performance	0.532
		Adaptive	0.910
Traffic participant manager	C21	Performance	0.128
		Adaptive	0.882
		Enhancive	1.000
Lateral planner	C22	Performance	0.346
		Adaptive	0.725
Longitudinal planner	C23	Performance	0.646
		Adaptive	0.998
Lateral controller	C24	Performance	0.128
		Adaptive	0.050
Longitudinal control	C25	Performance	0.573
		Adaptive	0.682
Operations monitor	C26	Performance	0.421
		Adaptive	0.204
Turn indicators soft	C27	Performance	0.268
		Adaptive	0.469
Engine actuator	C28	Performance	0.270
Brake actuator	C30	Ph. Properties	0.200
		Performance	0.693
Turn indicator	C31	Performance	0.050
Operations HMI	C32	Performance	0.050

5.6 Discussion

The presented OC change propagation method was deployed on one real Autonomous Vehicle industrial project. However, the method needs to be further tested and validated. The identified ToCs using the method were afterwards confirmed matching the real case empirical changes that occurred further in the project. However, a statistical validation is limited by the number of industrial AV architectures available. Another difficulty lies in the fact that obtaining domain expert evaluation of the direct and indirect propagation probabilities is difficult and time-consuming. First, such evaluation necessitates the domain experts familiarization with the system architecture. Second, the evaluation achieved with individual interviews, i.e. à priori and based upon expertise that can be seen as "subjective" evaluation. Some possible directions to avoid biases are the organization of group based expert evaluation allowing the discussion amongst experts themselves and possible fine tuning. In the case of possible divergent evaluations, several strategies can be considered regarding the type of the project: either the average yielding in loos of the information, or the worst-case scenario in order to be sure to capture the priorities.

The overall results of the OC change impact indicate the risks associated to the change. The interesting fact is that experts have underlined that these can be used to prioritize the engineering work but also as a proxy for an engineering rework. These are the initial discussions with the engineers, and they need to be further tested in use and eventually estimate the possibility to use further in managing these efforts. Furthermore, the ToCs DSM may assist in the identification of the ToCs causes. To use the case study's example, the matrix shows that adaptive changes for the "Complex structure manager" (C18) are required to cope with the performance and enhancive changes of the "Environment processing component" (C8). While C18 has four different inputs from four system functions (see Figure 5.7), the adaptive change source is the Functional Exchange coming from C8: "Roundabout Segment". As such, the needed rework for C18 would be where this data is used, which helps the design team to target and to estimate said rework.

5.7 Conclusion

Autonomous Vehicles are expected to be an essential component of future mobility. Important R&D efforts for AVs experimentation are being undertaken by mobility stakeholders. Multiple studies have observed that an essential factor for a successful Autonomous Vehicle (AV) design is the vehicle's fitness to its Operational Context (OC) (Bagschik et al., 2018; Damak et al., 2020a; Ulbrich et al., 2014). This paper proposes a new method for OC change impact assessment for AV reference architecture to help design teams identify and anticipate necessary changes and implicitly the rework during the design phase. The method is based upon the identification and analysis of the the direct impact of OC change, used for calculating the indirect propagation probabilities of change between the components. The propagation is based upon a model mapping between an OC Ontology for Autonomous Vehicles and a model of AV architecture (Damak et al., 2020b). This mapping is built through Functional Chains describing the functioning of the vehicles during operational situations created by a specific layout of OC elements.

The direct impact is analyzed from the changes in FC and their effects on the involved functions. This effect is then propagated onto the associated components, and a probability for required Types of Changes are estimated with respect to the component's type: software, sensors, and actuators. The results of the direct impact analysis are then propagated between the component ToCs with a Change Propagation Bayesian Network (CP-BN). The method is deployed on an industrial case study of the Operational Context change propagation onto a real industrial project of an AV reference architecture. The results show we can approximate the likelihood of impacted components with the required ToC.

The method is based on the analysis of operational situations defined by the design team based on the OC elements. This step is still a subjective and time-consuming analysis. Future works may focus on semi-automating the situation analysis and identification based on the OC to improve the change propagation method's usability.

5.8 Acknowledgment

The authors would like to thank the colleagues from AKKA Technologies for their collaboration in the application and evaluation of the proposed method as well as their useful feedbacks.

5.9 References

Ahmad, N., Wynn, D.C., Clarkson, P.J., 2013. Change impact on a product and its redesign process: a tool for knowledge capture and reuse. Research in Engineering Design 24, 219–244. https://doi.org/10.1007/s00163-012-0139-8

Bagschik, G., Menzel, T., Maurer, M., 2018. Ontology based Scene Creation for the Development of Automated Vehicles, in: 2018 IEEE Intelligent Vehicles Symposium (IV). Presented at the 2018 IEEE Intelligent Vehicles Symposium (IV), pp. 1813–1820. https://doi.org/10.1109/IVS.2018.8500632

Chapin, N., Hale, J.E., Khan, K.Md., Ramil, J.F., Tan, W.-G., 2001. Types of software evolution and software maintenance. J. Softw. Maint. Evol.: Res. Pract. 13, 3–30. https://doi.org/10.1002/smr.220

Cheng, H., Chu, X., 2012. A network-based assessment approach for change impacts on complex product. Journal of Intelligent Manufacturing 23, 1419–1431. https://doi.org/10.1007/s10845-010-0454-8

Clarkson, P.J., Simons, C., Eckert, C., 2004. Predicting Change Propagation in Complex Design. Journal of Mechanical Design 126, 788–797. https://doi.org/10.1115/1.1765117

Crawley, E., de Weck, O., Eppinger, S., Magee, C., Moses, J., Seering, W., Schindall, J., Wallace, D., Whitney, D., 2004. The influence of architecture in engineering systems. Engineering Systems Monograph 2006.

Damak, Y., Leroy, Y., Trehard, G., Jankovic, M., 2020. Operational Context-Based Design Method of Autonomous Vehicles logical Architectures. Presented at the Submitted to: System of Systems Engineering Conference (SoSE 2020).

Damak, Y., Leroy, Y., Trehard, G., Jankovic, M., n.d. A Context Ontology for Operational Scenarios Generation of Autonomous Vehicles. unpublished.

Eckert, C., Clarkson, P.J., Zanker, W., 2004. Change and customisation in complex engineering domains. Research in Engineering Design 15, 1–21. https://doi.org/10.1007/s00163-003-0031-7

Fei, G., Gao, J., Owodunni, O., Tang, X., 2011. A method for engineering design change analysis using system modelling and knowledge management techniques. International Journal of Computer Integrated Manufacturing 24, 535–551. https://doi.org/10.1080/0951192X.2011.562544

Gero, J.S., 1990. Design Prototypes: A Knowledge Representation Schema for Design. AIMag 11. https://doi.org/10.1609/aimag.v11i4.854

Gero, J.S., Kannengiesser, U., 2014. The Function-Behaviour-Structure Ontology of Design, in: Chakrabarti, A., Blessing, L.T.M. (Eds.), An Anthology of Theories and Models of Design: Philosophy, Approaches and Empirical Explorations. Springer London, London, pp. 263–283. https://doi.org/10.1007/978-1-4471-6338-1_13

Hamraz, B., Caldwell, N.H.M., John Clarkson, P., 2012. A Multidomain Engineering Change Propagation Model to Support Uncertainty Reduction and Risk Management in Design. Journal of Mechanical Design 134, 100905. https://doi.org/10.1115/1.4007397

Hamraz, B., Hisarciklilar, O., Rahmani, K., Wynn, D.C., Thomson, V., Clarkson, P.J., 2013. Change prediction using interface data. Concurrent Engineering 21, 141–154. https://doi.org/10.1177/1063293X13482473

Jarratt, T.A.W., Eckert, C.M., Caldwell, N.H.M., Clarkson, P.J., 2011. Engineering change: an overview and perspective on the literature. Research in Engineering Design 22, 103–124. https://doi.org/10.1007/s00163-010-0097-y

Koh, E.C.Y., Caldwell, N.H.M., Clarkson, P.J., 2012. A method to assess the effects of engineering change propagation. Research in Engineering Design 23, 329–351. https://doi.org/10.1007/s00163-012-0131-3

Lee, J., Hong, Y.S., 2017. Bayesian network approach to change propagation analysis. Research in Engineering Design 28, 437–455. https://doi.org/10.1007/s00163-017-0252-9

Martin, M.V., Ishii, K., 2002. Design for variety: developing standardized and modularized product platform architectures. Research in Engineering Design 13, 213–235. https://doi.org/10.1007/s00163-002-0020-2

Morkos, B., Mathieson, J., Summers, J.D., 2014. Comparative analysis of requirements change prediction models: manual, linguistic, and neural network. Research in Engineering Design 25, 139–156. https://doi.org/10.1007/s00163-014-0170-z

Morkos, B., Shankar, P., Summers, J.D., 2012. Predicting requirement change propagation, using higher order design structure matrices: an industry case study. Journal of Engineering Design 23, 905–926. https://doi.org/10.1080/09544828.2012.662273

Ollinger, G.A., Stahovich, T.F., 2004. RedesignIT—A Model-Based Tool for Managing Design Changes. Journal of Mechanical Design 126, 208–216. https://doi.org/10.1115/1.1666888

Reddi, K.R., Moon, Y.B., 2009. A framework for managing engineering change propagation. IJIL 6, 461. https://doi.org/10.1504/IJIL.2009.025060

Roques, P., 2016. MBSE with the ARCADIA Method and the Capella Tool, in: 8th European Congress on Embedded Real Time Software and Systems (ERTS 2016). Toulouse, France.

SAE, (On Road Automated Vehicle Standards Committee), 2014. Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems. Technical Report J3016_201401.

Ulbrich, S., Nothdurft, T., Maurer, M., Hecker, P., 2014. Graph-based context representation, environment modeling and information aggregation for automated driving, in: 2014 IEEE Intelligent Vehicles Symposium Proceedings. Presented at the 2014 IEEE Intelligent Vehicles Symposium Proceedings, pp. 541–547. https://doi.org/10.1109/IVS.2014.6856556

Xie, Y., Ma, Y., 2016. Well-controlled engineering change propagation via a dynamic inter-feature association map. Research in Engineering Design 27, 311–329. https://doi.org/10.1007/s00163-016-0220-9

Yang, F., Duan, G., 2012. Developing a parameter linkage-based method for searching change propagation paths. Research in Engineering Design 23, 353–372. https://doi.org/10.1007/s00163-011-0124-7

6 Conclusion and Discussion

6.1 Conclusion and Retrospective

6.1.1 Summary

This thesis aims at providing models and methods to support the design of Vehicular CPS. To identify research objectives, we conducted an empirical study on the outsourced R&D of Vehicular CPS within an engineering consulting company, AKKA Technologies, to identify and characterize the new challenges encountered during their design processes. By cross-referencing the results of the industrial audit with the literature on Vehicular CPS design, many challenges were identified due to the novelty of the subject: System integration & validation, Functional safety, Human-System Integration, Big data, Standardization, System design and architecting, New technologies integration, and Context dependency.

Finding the right research direction for the industrial partner from all the possibilities was quite challenging at the beginning of the research project. However, two pieces of evidence significantly influenced the decision: the importance for the success of a Vehicular CSP design that the vehicle's behavior is well adapted to its Operational Context; and the high dependency of their system architectures to the Operational Context in order to exhibit the right behavior during the vehicle's interaction with its context. Hence, the research objectives focus on supporting Vehicular CPS architecting by considering the dependence of the system architectures to their Operational Context.

The classical Concept of Operations approaches for system architecting focuses on the analysis of the system's operational activities, scenarios, and modes. The literature review showed a lack of method to analyze the Operational Context characteristics and dynamic for system architecting in early design phases. As such, this research objectives were as: **(RO1)** to analyze Vehicular CPS Operational Context and systematically explore their operational domain in order to define their system architecture in the early design phase; and **(RO2)** to anticipate the necessary evolution of the Vehicular CPS architecture when its operational domain changes. To achieve the research objectives, we sought to answer three research questions:

RQ1: How to systematically define operational scenarios based on the Operational Context in early design phase?

Chapter 3 introduces a method to identify and define the AV's operational scenarios in the early design phase of a scenario-based design approach. The method is based on an Operational Context ontology for Autonomous Vehicles. The ontology introduces five levels of Operational Context elements contributing successively to the identification and description of scenarios: (0) Use case, (1) Environment, (2) Road infrastructure, (3) Traffic infrastructure, and (4) Traffic objects.

The Operational Context Ontology and the systematic scenario identification method extend the Concept of Operations (ConOps) by considering the Operational Context and its importance for the vehicle's operational behavior and the system's architecture. In particular, the ontology presents several advantages:

- adaptable to different legislations,
- extendable to integrate new elements for the road and traffic infrastructure as well as new traffic participants and new maneuvers,
- usable in multiple phases of the design: Context definition, exploration, conceptualization, and validation

This research work can lead to several research perspective as follows:

- Further research can be conducted to semi-automate the identification of operational scenarios and situations. Such identification needs not only a process to identify all possible variations of situations based on a set of Operational Context elements, but also a procedure to select relevant situations to the design process. This semi-automation should include designer-in-the-loop to profit from the implicit knowledge of the design team to define all the relevant operational situations.
- Further research may be needed to extend the ontology in order to characterize the variations on the third spatial dimension. More research may also be conducted to characterize specific physical phenomena, such as light reflection, and their effects on the vehicle's behavior.

RQ2: How to design and model Vehicular CPS architecture based on the Operational Context and the defined operational scenarios?

Chapter 4 provides answers to the second research question with a design method of Autonomous Vehicles and Vehicular CPS architectures based on their Operational Context. The method starts with the Operational Context ontology to characterize the vehicle's context, identify and define operational situations, and model the Autonomous Vehicle's reactive behavior as operational processes (a sequence of internal activities in response to the system's environment). The operational processes are then analyzed to define chains of system functions and functional exchanges. Finally, logical components are modeled and allocated to the functions.

The methods to support Vehicular CPS architecting provides an uninterrupted traceability between the Operational Context elements and the vehicle's architecture elements. It also helps in the decision making of technical solutions with respect to the Operational Context and the expected situations/behavior for which the vehicle is designed. This method paves the way for further research as follows:

- Additional research may help reaching an optimal analysis of the operational behavior of the vehicle by cover enough situations while avoiding redundancy. Evaluation and optimization methods could be developed to obtain such criteria for the architecting method based on the Operational Context.
- Further research may integrate to the architecting method the consideration of new constraints and issues appearing during the detailed design of the components. This may necessitate the redefinition of some system functions and functional interactions. Iteration may be applied to the Functional Chains and Logical components definition, while preserving the traceability to the Operational Context and operational behavior of the Vehicular CPS.
- Further studies on integrating architecture patterns of Vehicular CPS may lead to semi-automate the design Functional Chains. With matching architecture patterns with Operational Context layouts, Functional Chains could be rapidly identified while integration designer-in-the-loop.

RQ3: How to evaluate the Vehicular CPS architecture evolution when the Operational Context changes?

Chapter 5 covers the third research question and proposes a method to assess the Operational Context change impact on Vehicular CPS reference architecture. One of the significant success criteria of a Vehicular CPS design is to exhibit an operational behavior suitable for its Operational Context and the situations it encounters. While the iterations on the detailed design level are out of this thesis scope, the impact of changes happening on the Operational Context level must be considered. The identification of the change impact is made following a propagation path, starting from the changes in operational situations and associated behavior, to changes in Functional Chains, through to changes in the definitions and modeling of system functions. It ends with the Types of Changes required for the components. The estimation of the direct impact on components is deterministic, following propagation paths on the architecture model. The Types of Changes occurrence is estimated by domain expert then propagated through the components with a Bayesian Network representing components Types of Changes. The second step of the change propagation is probabilistic and aims at evaluating the likelihood that a component undergoes a certain Type of Change. The overall propagation method has several advantages worth noting:

- Contrary to change propagation method in the literature, it does not consider the system architecture as stable during the change propagation. It identifies how the architecture changes and evolves to suit the new context. The change is propagated through all the elements of the architecture: expected behavior, functional chains, system functions, functional interaction, constraints, and components.
- The method not only indicate what Types of Changes are likely to be required for components, but also offers tools to identify the source of the change, to target the needed rework, and to evaluate the efforts needed.
- As the method uses a Change Propagation Bayesian Network, it is possible to refine and update the direct propagation probabilities estimated by domain experts with Bayesian statistical learning from empirical studies.

This research work on Operational Change propagation onto Vehicular CPS architecture may be extended with several research perspectives as follows:

- Further research may help integrating the analysis of the operational situations changes and their detailed impact on the Functional Chains. This step would reinforce the deterministic propagation step.
- Additional research may help evaluating the engineering rework effort and cost caused by any change to the Operational Context. Such research could be valuable to help the design team prioritize rework and development.
- Additional studies are required to add an analysis and estimation of change propagation of the automation system of the Vehicular CPS onto the mechanical parts of vehicle's platform. The propagation of sensors and actuators spatial changes onto the other components of the platform can be addressed and linked to initial change in the Operational Context.

The overall research work of the PhD thesis with its three main contribution to the design of complex systems operating in highly dynamic and uncontrolled context. With the presented model and methods, the design of safety critical systems within dynamic and uncontrolled Operational Context can be better addressed with the proposed extension of the Concept of Operations.

6.1.2 Retrospectives:

Even though the research objectives have been achieved to a certain level, several aspects could have been apprehended more efficiently. These aspects are in general linked to better use of the ontology and the types of context elements. The initial study for the use of the Operational Context in the design of Vehicular CPS (Appendix A) suggests requirements reuse and recycling from former projects based on shared Operational Context elements. The Ontology's classification of the Context elements could have increased the accuracy and efficiency of the reuse process by indicating the relevance and importance of a requirement regarding the Context elements nature. Furthermore, a similar reuse and recycling process could be considered for the architecting method (chapter 4) to exploit the knowledge from former projects.

The presence of an Operational Context element in the vehicle's operational design domain, adds possible situations to the vehicle's operation. The way each type of element changes the set of possible situations was not studied in this thesis. However, to highlight the different roles of Operational Context elements in adding potential situations to the operations of the vehicles can be observed by looking at the details of the ontology. The v ontology elements are classified to be part of the scenery, the dynamic objects, or the vehicle's operational purpose. An initial attempt at a finer classification shows that there are three categories of scenery elements: environmental elements, structural elements, and basic elements. The ways to define these elements during a design project are entirely different. Environmental elements are defined with respect to the meteorological and geographical knowledge of the operational area. Structural elements, such as intersections and roundabouts, are defined based on the area's structure and legislation. These structural elements define mandatory basic elements, such as lanes and boundary marks. Other optional basic elements, such as turning signs and marks, can be added to the structural elements and the operational design domain.

It is safe to assume that a better understanding and characterization of the role played by each type of Operational Context elements in potential operational situations, would drastically improve the efficiency of the architecting method (chapter 4) as well as the Operational Context change propagation method (chapter 5). This knowledge would help the design team to better identify and define operational situations during the architecting of AVs. It would also permit the upgrade of the mapping between the Context elements and the situation and, by extension, the vehicle's expected behavior. As such, it would benefit the analysis of the direct impact of Operational Context change of the vehicle's architecture.

6.2 Discussion

6.2.1 Assumptions

The proposed Operational Context ontology relies on the idea that **any operational scenario or situation of an Vehicular CPS happen in a particular use case and can adequately be described by the environmental setting, surrounding layout, and the behavior of the protagonists**. This assumption can be **taken** because any real case Autonomous Vehicles operates with a defined goal and encounters various infrastructure set ups and various participants with unknown goals and behaviors. This idea is also shared by multiple studies (Geyer et al., 2014; Schuldt et al., 2018; Wachenfeld et al., 2016). Following this idea, the ontology is structured in five layers separating and defining the different aspects of an operational situation: from the use case, to the road scenery, to the participant and their actions. Road sceneries generally obey the specific regulations of each country; as such, the proposed model aims at giving the flexibility, to characterize the various types of road layouts. The Operational Context ontology presents elements of the context with the different relationships between them. While it can be used to model even unrealistic operational sceneries and situations, its purpose is to describe and characterize real case sceneries and situations to help the design team identify the appropriate **behavior**, functions, constraints, components, and parameters of the system.

The proposed methods to support the design of Vehicular CPS are based on the definition of system architecture as the structure of the system's components, the arrangement of the system functions, the mapping from the functional to the structural domain, and the behavior of the system (Eppinger and Browning, 2012; Wyatt et al., 2009). The method to design Vehicular CPS architecture proposed in chapter 4 is based on the idea to identify the expected reactive behavior of the vehicle for operational situations, and to design the system architecture to obtain this behavior. It is assumed that **the complex behavior of a vehicle is a chain of internal activities exchanging data and energy in reaction to stimuli from its environment**. In the context of Vehicular CPS, it is reasonable to understand the behavior as the response of the system to real time changes in its own state and its environmental conditions (Gero and Kannengiesser, 2014; Komoto et al., 2013). A way to

model the system behavior, notably for adaptive and cognitive systems, is through activity chains (Komoto et al., 2013). Based on this assumption, the method proposes to model the vehicle's reactive behavior as operational processes, a sequence of operational activities initiated from an event taking place in the vehicle's environment. The second assumption of this method is that **it is possible to design functions chains corresponding to the operational processes**. This assumption is essential as the design method uses Functional Chains to define and model the system functions and constraints, such as the functional architecture realizes the expected behavior of the vehicle.

The previous assumption is also important for the method to assess Operational Context change propagation on the Vehicular CPS architecture, as it uses the mapping between the Operational Context and the Functional Chains through the operational situations to identify the direct impact of change. The change propagation method is based on three hypotheses. The first one is that **the effect of the Functional Chainss modification on the definition of system functions is likely to require a change to the components realizing said functions**. Therefore, the method studies the different possible effects of Functional Chains change on functions to estimate the required Types of Changes to the associated components. The second hypothesis considers that **the change of an element of the system architecture directly impacts connected elements, which in turn, indirectly propagate to other elements through the connection of their neighbors**. Based on this idea, we rely on the functional interfaces between components to identify the direct dependencies of the components. Thirdly, it is considered reasonable that, in the current context of Vehicular CPS, **domain experts have sufficient knowledge to approximate the likelihood of functions changes propagation onto components as well as component change propagation through components dependencies**. Their estimation of the likelihood is mandatory to evaluate the overall impact of Operational Context change with the Bayesian Network. However, the Bayesian Network offers the perspective of improving the accuracy of their estimation with BN learning from empirical cases of Operational Context changes. Further industrial feedbacks of Vehicular CPS experimentations will provide more accurate data.

6.2.2 Method Reproducibility

The research work of this thesis addresses the design of Vehicular CPS in outsourced R&D. The studies and contributions were developed based on the design context challenges identified with the engineering consulting company AKKA Technologies. However, the resulting models and methods are not restricted to this context. As discussed in section 6.2.1, the scenario identification method, the architecting method, and the Operational Context change propagation methods are based on ideas and assumptions related solely to Autonomous Vehicles and system architecting. They are not specific to the context of AKKA technologies.

As such, it should be possible to reproduce the proposed methods in other Vehicular CPS design contexts. Other stakeholders, such as vehicle makers, suppliers, start-ups, and other engineering companies, can implement the Operational Context ontology and deploy the Vehicular CPS architecting method based on the Operational Context. The deployment of both, the Vehicular CPS architecting method and the Operational Context change propagation method, depends only on the capability of experts to transpose knowledge from other Cyber-Physical System development to Vehicular CPS design.

6.2.3 Industrial Implications

The work presented in this thesis was conducted within the engineering consulting company AKKA Technologies, specifically its Autonomous Systems Team. It aims at providing tools and methods to assist the design team in the architecting activities of Vehicular CPS.

The proposed ontology and methods were applied with the support of the team members and leaders on a real case of Autonomous Vehicles architecture. They kindly gave feedback and evaluated the various contributions. As such, the contribution's short-term implications on the industrial partner were directly observed during their test and implementation. The systems engineers and design team consider the ontology particularly helpful to encompass and define the operational design domain of the vehicle. They agree to the importance of the task in early design phases. The architecting method aligned well with the design habits of systems engineers and architects, while at the same time permitted to base the

architecture on the Operational Context. Hence, they could effectively apply it to client projects. As a result, all the team members expressed their satisfaction in gaining better justifications for the technological solutions for the Operational Context.

This model and methods of this research work contributes to the outsourced R&D of Autonomous Vehicles in three aspects. First it helped better justifying the design team's choices and satisfy their clients request. Second, it accelerates the adaptation of the system's architecture and components to change requests and avoid important projects delays. Finally, it enhances the reusability of formerly developed solutions in new projects while preserving Intellectual Property of the clients.

The Operational Context ontology and associated systems architecting methods may also have long term implications for the industrial partner. Systematic use of the methods can be considered by technology consulting companies. In this case, an implementation strategy should be designed for the systematic use within the company during Vehicular CPS design. Research & Development studies may be required to design a software to map the defined Operational Context elements with architectural patterns. It could be used to semi-automate the identification of operational situations and adequate Functional Chains. It can also automatically generate and compute the Change-Propagation Bayesian Network. Besides, a maintenance strategy would be required to update the ontology and methods with new knowledge. Finally, a generalization of the approach for the design of other Cyber-Physical Systems could be considered. The development of Operational Context ontologies could become systematic for the different systems and deployed to apply the architecting methods based on the Operational Context.

6.2.4 Future Work

The presented research work opens to three direct research avenues; (1) the improvement of the evaluation of the proposed methods, (2) the extension of the Operational Contxt ontology uses for a semi-automatic generation of operational situations Vehicular CPS design, (3) the generalization to other system of the design methods based on the OC.

An empirical study to evaluate the implementations and impact on the design process of the proposed models and methods, is needed (1). This study could focus on the observation

of some success criteria such as the method's usability, usefulness, impact on the design process, and impact on client satisfaction. The time and effort invested in understanding and using the methods, could be observed to indicate the usability of the method. Its usefulness can be studied by comparing the architecture designed with and without the method. Its impact on the design process could be studied with a qualitative analysis of the new practices and design activities, in comparison to the practices observed in the industrial audit (Appendix B). As for the client satisfaction, interviews and feedbacks could help understand the client's interest in the method for shortening the design time, reducing its cost, or improving the quality of the final product.

Further studies can continue the development of the proposed methods to support Vehicular CPS architecting. The main limitations of the design process based on the Operational Context ontology are related to the definition of operational situations. The defined set of operational situations should cover all types of situations to cover all variations of the vehicle's expected behavior. On the other hand, the behavior and situation couples are used to model the vehicle's behavior as operational processes. As such, redundancies should be avoided to keep the number of couples humanly analyzable. A study aiming at semi-automatically generating an optimal situation set could significantly improve the efficiency and usability of the proposed methods (2). Such generation processes could take the form of a combinatorial generation from Operational Context elements with redundancy elimination and evaluation of situation relevance while integration designer-in-the-loop to profit from their implicit and necessary domain knowledge. Machine learning on real driving situations and accidentology databases could also be an interesting direction to explore.

Finally, the design approach proposed in this thesis could be generalized to Cyber-Physical Systems (3). The proposed architecting method and Operational Context change propagation method are specific to Vehicular CPS only in their use of the Operational Context ontology. If an equivalent Operational Context ontology is available for another CPS, it would be possible to apply both methods. As such, a framework for the systematic development of Operational Context ontologies for CPSs could provide input to the design approach. However, the relevance and efficiency of the approach to systems other than Vehicular CPS should be studied and verified.

Furthermore, the contributions of this PhD can open the for additional research to support the design of Vehicular CPS and Autonomous Vehicles in particular. Further studies may focus on addressing requirements elicitation challenges for novel Vehicular CPS with requirement elicitation based on the Operational Context and requirement recycling using architecture patterns. Studies can also explore ways to consider future standards of Autonomous Vehicles in the requirement elicitation processes based on the Operational Context.

Additional research on the design of Vehicular CPS architectures should also focus on integrating the impact of new usages and user acceptance on the architecture and its design process. Other approaches also need to be explored in the future and integrate to the design process of Vehicular CPS. Particularly mobility as a service design and sustainable design approaches should be considered and their impact the vehicle's architecture should be studies.

Bibliography

Adam, S., Schmid, K., 2013. Effective Requirements Elicitation in Product Line Application Engineering – An Experiment, in: Requirements Engineering: Foundation for Software Quality. Springer Berlin Heidelberg, pp. 362–378. https://doi.org/10.1007/978-3-642-37422-7_26

Ahmad, N., Wynn, D.C., Clarkson, P.J., 2013. Change impact on a product and its redesign process: a tool for knowledge capture and reuse. Res. Eng. Des. 24, 219–244. https://doi.org/10.1007/s00163-012-0139-8

Alexander, I., Kiedaisch, F., 2002. Towards recyclable system requirements, in: Proceedings Ninth Annual IEEE International Conference and Workshop on the Engineering of Computer-Based Systems. pp. 9–16. https://doi.org/10.1109/ECBS.2002.999817

Alshaikh, Z., Boughton, C., 2009. The Context Dynamics Matrix (CDM): An Approach to Modeling Context, in: 2009 16th Asia-Pacific Software Engineering Conference. Presented at the 2009 16th Asia-Pacific Software Engineering Conference, pp. 101–108. https://doi.org/10.1109/APSEC.2009.74

Armand, A., Filliat, D., Ibañez-Guzman, J., 2014. Ontology-based context awareness for driving assistance systems, in: 2014 IEEE Intelligent Vehicles Symposium Proceedings. Presented at the 2014 IEEE Intelligent Vehicles Symposium Proceedings, pp. 227–233. https://doi.org/10.1109/IVS.2014.6856509

Bach, J., Otten, S., Sax, E., 2016. Model based scenario specification for development and test of automated driving functions, in: 2016 IEEE Intelligent Vehicles Symposium (IV). Presented at the 2016 IEEE Intelligent Vehicles Symposium (IV), pp. 1149–1155. https://doi.org/10.1109/IVS.2016.7535534

Bagschik, G., Menzel, T., Maurer, M., 2018. Ontology based Scene Creation for the Development of Automated Vehicles, in: 2018 IEEE Intelligent Vehicles Symposium (IV). Presented at the 2018 IEEE Intelligent Vehicles Symposium (IV), pp. 1813–1820. https://doi.org/10.1109/IVS.2018.8500632

Baheti, R., Gill, H., 2011. Cyber-physical systems. Impact Control Technol. 12, 161–166.

Baldauf, M., Dustdar, S., Rosenberg, F., 2007. A survey on context-aware systems. Int. J. Ad Hoc Ubiquitous Comput. 2, 263. https://doi.org/10.1504/IJAHUC.2007.014070

Behere, S., Törngren, M., 2016. A functional reference architecture for autonomous driving. Inf. Softw. Technol. 73, 136–150. https://doi.org/10.1016/j.infsof.2015.12.008

Berkovich, M., Leimeister, J.M., Krcmar, H., 2011. Requirements Engineering for Product Service Systems. Bus. Inf. Syst. Eng. 3, 369–380. https://doi.org/10.1007/s12599-011-0192-2

Blessing, L.T.M., Chakrabarti, A. (Eds.), 2009. DRM: A Design Reseach Methodology, in: DRM, a Design Research Methodology. Springer London, London. https://doi.org/10.1007/978-1-84882-587-1_2

Bock, F., Sippl, C., Heinzz, A., Lauerz, C., German, R., 2019. Advantageous Usage of Textual Domain-Specific Languages for Scenario-Driven Development of Automated Driving Functions, in: 2019 IEEE International Systems Conference (SysCon). Presented at the 2019 IEEE International Systems Conference (SysCon), pp. 1–8. https://doi.org/10.1109/SYSCON.2019.8836912

Bonnet, S., Voirin, J.-L., Exertier, D., Normand, V., 2016. Not (strictly) relying on SysML for MBSE: Language, tooling and development perspectives: The Arcadia/Capella rationale, in: Systems Conference (SysCon), 2016 Annual IEEE. IEEE, pp. 1–6.

Brown, P.J., 1995. The stick-e document: a framework for creating context-aware applications. Electron. Publ.-Chichester- 8, 259–272.

Bubl, F., Balser, M., 2005. Tracing Cross-Cutting Requirements via Context-Based Constraints, in: Ninth European Conference on Software Maintenance and Reengineering. Presented at the Ninth European Conference on Software Maintenance and Reengineering, pp. 80–90. https://doi.org/10.1109/CSMR.2005.54

Cabrera, O., Franch, X., Marco, J., 2017. Ontology-based context modeling in service-oriented computing: A systematic mapping. Data Knowl. Eng. 110, 24–53. https://doi.org/10.1016/j.datak.2017.03.008

Chapin, N., Hale, J.E., Khan, K.Md., Ramil, J.F., Tan, W.-G., 2001. Types of software evolution and software maintenance. J. Softw. Maint. Evol. Res. Pract. 13, 3–30. https://doi.org/10.1002/smr.220

Chen, H., Finn, T., Joshi, A., 2003. An ontology for context-aware pervasive computing environments. Knowl. Eng. Rev. 18, 197–207. https://doi.org/10.1017/S0269888904000025

Chen, W., Hoyle, C., Wassenaar, H.J., 2013. A Choice Modeling Approach for Usage Context-Based Design, in: Chen, W., Hoyle, C., Wassenaar, H.J. (Eds.), Decision-Based Design: Integrating Consumer Preferences into Engineering Design. Springer London, London, pp. 255–285. https://doi.org/10.1007/978-1-4471-4036-8_10

Cheng, H., Chu, X., 2012. A network-based assessment approach for change impacts on complex product. J. Intell. Manuf. 23, 1419–1431. https://doi.org/10.1007/s10845-010-0454-8

Clarkson, P.J., Simons, C., Eckert, C., 2004. Predicting Change Propagation in Complex Design. J. Mech. Des. 126, 788–797. https://doi.org/10.1115/1.1765117

Crawley, E., de Weck, O., Eppinger, S., Magee, C., Moses, J., Seering, W., Schindall, J., Wallace, D., Whitney, D., 2004. The influence of architecture in engineering systems. Eng. Syst. Monogr. 2006.

Crowley, J.L., Coutaz, J., Rey, G., Reignier, P., 2002. Perceptual components for context aware computing. Presented at the International conference on ubiquitous computing, Springer, pp. 117–134. https://doi.org/10.1007/3-540-45809-3_9

Curcio, K., Navarro, T., Malucelli, A., Reinehr, S., 2018. Requirements engineering: A systematic mapping study in agile software development. J. Syst. Softw. 139, 32–50. https://doi.org/10.1016/j.jss.2018.01.036

Damak, Y., Jankovic, M., Leroy, Y., Chelbi, K., 2019. A Semi-automated Requirements Reuse and Recycling Process for Autonomous Transportation Systems R&D, in: Proceedings of the Design Society: International Conference on Engineering Design. pp. 3551–3560. https://doi.org/10.1017/dsi.2019.362

Damak, Y., Jankovic, M., Leroy, Y., Yannou, B., 2018. Analysis of Safety Requirements Evolution in the Transition of Land Transportation Systems Toward Autonomy. Presented at the 15th International Design Conference, pp. 2845–2854. https://doi.org/10.21278/idc.2018.0448

Damak, Y., Leroy, Y., Trehard, G., Jankovic, M., 2020a. A Context Ontology Supported Identification of Operational Scenarios for Autonomous Vehicles in Early Design Phases. unpublished.

Damak, Y., Leroy, Y., Trehard, G., Jankovic, M., 2020b. Operational Context-Based Design Method of Autonomous Vehicles logical Architectures. Presented at the Submitted to: System of Systems Engineering Conference (SoSE 2020).

De Weck, O., Eckert, C., Clarkson, J., 2007. A classification of uncertainty for early product and system design. Guidel. Decis. Support Method Adapt. NPD Process. 159–160.

Dey, A.K., 2001. Understanding and Using Context. Pers. Ubiquitous Comput. 5, 4–7. https://doi.org/10.1007/s007790170019

Dokic, J., Müller, B., Meyer, G., 2015. European roadmap smart systems for automated driving. European Technology Platform on Smart Systems Integration (EPoSS).

Dumitrache, I., Sacala, I.S., Moisescu, M.A., Caramihai, S.I., 2017. A Conceptual Framework for Modeling and Design of Cyber-Physical Systems. Stud. Inform. Control 26. https://doi.org/10.24846/v26i3y201708

Eckert, C., Clarkson, P.J., Zanker, W., 2004. Change and customisation in complex engineering domains. Res. Eng. Des. 15, 1–21. https://doi.org/10.1007/s00163-003-0031-7

Eckert, C.M., Stacey, M., Clarkson, P., 2003. The spiral of applied research: A methodological view on integrated design research, in: Proceedings of the 14th

International Conference on Engineering Design (ICED'03). Stockholm, Sweden, pp. 19–21.

Eppinger, S.D., Browning, T.R., 2012. Design structure matrix methods and applications. MIT press.

ERTRAC, 2019. Automated Driving Roadmap.

Fairley, R.E., Thayer, R.H., 1997. The concept of operations: The bridge from operational requirements to technical specifications. Ann. Softw. Eng. 3, 417–432. https://doi.org/10.1023/A:1018985904689

Fei, G., Gao, J., Owodunni, O., Tang, X., 2011. A method for engineering design change analysis using system modelling and knowledge management techniques. Int. J. Comput. Integr. Manuf. 24, 535–551. https://doi.org/10.1080/0951192X.2011.562544

Fuchs, S., Rass, S., Kyamakya, K., 2008a. Integration of Ontological Scene Representation and Logic-Based Reasoning for Context-Aware Driver Assistance Systems. Electron. Commun. EASST Volume 11: Contextaware Adaption Mechanisms for Pervasive and Ubiquitous Services. https://doi.org/10.14279/tuj.eceasst.11.127

Fuchs, S., Rass, S., Lamprecht, B., Kyamakya, K., 2008b. A model for ontology-based scene description for context-aware driver assistance systems. Presented at the Proceedings of the 1st international conference on Ambient media and systems, ICST (Institute for Computer Sciences, Social-Informatics and …, p. 5.

Garro, A., Tundis, A., Bouskela, D., Jardin, A., Thuy, N., M. Otter, L. Buffoni, P. Fritzson, M. Sjölund, W. Schamai, H. Olsson, 2016. On formal cyber physical system properties modeling: A new temporal logic language and a Modelica-based solution, in: 2016 IEEE International Symposium on Systems Engineering (ISSE). pp. 1–8. https://doi.org/10.1109/SysEng.2016.7753137

Gero, J.S., 1990. Design Prototypes: A Knowledge Representation Schema for Design. AI Mag. 11. https://doi.org/10.1609/aimag.v11i4.854

Gero, J.S., Kannengiesser, U., 2014. The Function-Behaviour-Structure Ontology of Design, in: Chakrabarti, A., Blessing, L.T.M. (Eds.), An Anthology of Theories and Models of Design: Philosophy, Approaches and Empirical Explorations. Springer London, London, pp. 263–283. https://doi.org/10.1007/978-1-4471-6338-1_13

Geyer, S., Baltzer, M., Franz, B., Hakuli, S., Kauer, M., Kienle, M., Meier, S., Weißgerber, T., Bengler, K., Bruder, R., Flemisch, F., Winner, H., 2014. Concept and development of a unified ontology for generating test and use-case catalogues for assisted and automated vehicle guidance. IET Intell. Transp. Syst. 8, 183–189. https://doi.org/10.1049/iet-its.2012.0188

Glimm, B., Horrocks, I., Motik, B., Stoilos, G., Wang, Z., 2014. HermiT: An OWL 2 Reasoner. J. Autom. Reason. 53, 245–269. https://doi.org/10.1007/s10817-014-9305-1

Hamraz, B., Caldwell, N.H.M., John Clarkson, P., 2012. A Multidomain Engineering Change Propagation Model to Support Uncertainty Reduction and Risk Management in Design. J. Mech. Des. 134, 100905. https://doi.org/10.1115/1.4007397

Hamraz, B., Hisarciklilar, O., Rahmani, K., Wynn, D.C., Thomson, V., Clarkson, P.J., 2013. Change prediction using interface data. Concurr. Eng. 21, 141–154. https://doi.org/10.1177/1063293X13482473

Handbook, I., 2014. INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, Version 4. Int. Counc. Syst. Eng.

Henricksen, K., 2003. A framework for context-aware pervasive computing applications. Queensland: University of Queensland.

Höfer, A., Herrmann, M., 2017. Scenario-based approach for developing ADAS and automated driving functions, in: Isermann, R. (Ed.), Fahrerassistenzsysteme 2017. Springer Fachmedien Wiesbaden, Wiesbaden, pp. 215–225.

Holt, J., Perry, S.A., Brownsword, M., 2012. Model-based requirements engineering. IET.

Horridge, M., Drummond, N., Goodwin, J., Rector, A., Wang, H.H., 2006. The manchester owl syntax. Presented at the In Proc. of the 2006 OWL Experiences and Directions Workshop (OWL-ED2006), Citeseer.

Horváth, I., 2014. What the Design Theory of Social-Cyber-Physical Systems Must Describe, Explain and Predict?, in: Chakrabarti, A., Blessing, L.T.M. (Eds.), An Anthology of Theories and Models of Design: Philosophy, Approaches and Empirical Explorations. Springer London, London, pp. 99–120. https://doi.org/10.1007/978-1-4471-6338-1_5

Horvath, I., 2012. Beyond advanced mechatronics: new design challenges of Social-Cyber-Physical systems, in: Proceedings of the ACCM-Workshop on "Mechatronic Design.

Hull, R., Neaves, P., Bedford-Roberts, J., 1997. Towards situated computing, in: Digest of Papers. First International Symposium on Wearable Computers. Presented at the Digest of Papers. First International Symposium on Wearable Computers, pp. 146–153. https://doi.org/10.1109/ISWC.1997.629931

Irshad, M., Petersen, K., Poulding, S., 2018. A systematic literature review of software requirements reuse approaches. Inf. Softw. Technol. 93, 223–245. https://doi.org/10.1016/j.infsof.2017.09.009

Jarratt, T.A.W., Eckert, C.M., Caldwell, N.H.M., Clarkson, P.J., 2011. Engineering change: an overview and perspective on the literature. Res. Eng. Des. 22, 103–124. https://doi.org/10.1007/s00163-010-0097-y

Jensen, J.C., Chang, D.H., Lee, E.A., 2011. A model-based design methodology for cyber-physical systems, in: 2011 7th International Wireless Communications and Mobile Computing Conference. Presented at the 2011 7th International Wireless Communications and Mobile Computing Conference, pp. 1666–1671. https://doi.org/10.1109/IWCMC.2011.5982785

Jesenski, S., Stellet, J.E., Schiegg, F., Zöllner, J.M., 2019. Generation of Scenes in Intersections for the Validation of Highly Automated Driving Functions, in: 2019 IEEE Intelligent Vehicles Symposium (IV). Presented at the 2019 IEEE Intelligent Vehicles Symposium (IV), pp. 502–509. https://doi.org/10.1109/IVS.2019.8813776

Jiao, J. (Roger), Chen, C.-H., 2006. Customer Requirement Management in Product Development: A Review of Research Issues. Concurr. Eng. 14, 173–185. https://doi.org/10.1177/1063293X06068357

Kaiya, H., Saeki, M., 2006. Using Domain Ontology as Domain Knowledge for Requirements Elicitation, in: 14th IEEE International Requirements Engineering Conference (RE'06). pp. 189–198. https://doi.org/10.1109/RE.2006.72

Kang, S., Choi, Y., 2005. Designing Logical Architectures of Software Systems, in: Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks (SNPD/SAWN'05). Presented at the Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks (SNPD/SAWN'05), IEEE, Towson, MD, USA, pp. 330–337. https://doi.org/10.1109/SNPD-SAWN.2005.30

Kerry, C.F., Karsten, J., 2017. Gauging investment in self-driving cars. Brookings Institution, October.

Khattak, A., Akbar, N., Aazam, M., Ali, T., Khan, A., Jeon, S., Hwang, M., Lee, S., 2014. Context Representation and Fusion: Advancements and Opportunities. Sensors 14, 9628–9668. https://doi.org/10.3390/s140609628

Knethen, A. von, Paech, B., Kiedaisch, F., Houdek, F., 2002. Systematic requirements recycling through abstraction and traceability, in: Proceedings IEEE Joint International Conference on Requirements Engineering. pp. 273–281. https://doi.org/10.1109/ICRE.2002.1048538

Koh, E.C.Y., Caldwell, N.H.M., Clarkson, P.J., 2012. A method to assess the effects of engineering change propagation. Res. Eng. Des. 23, 329–351. https://doi.org/10.1007/s00163-012-0131-3

Komoto, H., Hamberg, R., Tomiyama, T., 2013. Supporting the Architecting Process of Adaptive Systems, in: Model-Based Design of Adaptive Embedded Systems, Embedded Systems. Springer, New York, NY, pp. 159–188. https://doi.org/10.1007/978-1-4614-4821-1_6

Lee, J., Hong, Y.S., 2017. Bayesian network approach to change propagation analysis. Res. Eng. Des. 28, 437–455. https://doi.org/10.1007/s00163-017-0252-9

Martin, M.V., Ishii, K., 2002. Design for variety: developing standardized and modularized product platform architectures. Res. Eng. Des. 13, 213–235. https://doi.org/10.1007/s00163-002-0020-2

Miles, M.B., Huberman, A.M., Saldana, J., 2018. Qualitative Data Analysis: A Methods Sourcebook. SAGE Publications.

Morkos, B., Mathieson, J., Summers, J.D., 2014. Comparative analysis of requirements change prediction models: manual, linguistic, and neural network. Res. Eng. Des. 25, 139–156. https://doi.org/10.1007/s00163-014-0170-z

Morkos, B., Shankar, P., Summers, J.D., 2012. Predicting requirement change propagation, using higher order design structure matrices: an industry case study. J. Eng. Des. 23, 905–926. https://doi.org/10.1080/09544828.2012.662273

Moros, B., Vicente-Chicote, C., Toval, A., 2008. REMM-Studio + : Modeling Variability to Enable Requirements Reuse, in: Conceptual Modeling - ER 2008. Springer Berlin Heidelberg, pp. 530–531. https://doi.org/10.1007%2F978-3-540-87877-3_46

Nadoveza, D., Kiritsis, D., 2014. Ontology-based approach for context modeling in enterprise applications. Spec. Issue Role Ontol. Future Web-Based Ind. Enterp. 65, 1218–1231. https://doi.org/10.1016/j.compind.2014.07.007

Nemoto, Y., Uei, K., Sato, K., Shimomura, Y., 2015. A Context-based Requirements Analysis Method for PSS Design. 7th Ind. Prod.-Serv. Syst. Conf. - PSS Ind. Transform. Sustain. Bus. 30, 42–47. https://doi.org/10.1016/j.procir.2015.02.095

Ollinger, G.A., Stahovich, T.F., 2004. RedesignIT—A Model-Based Tool for Managing Design Changes. J. Mech. Des. 126, 208–216. https://doi.org/10.1115/1.1666888

Pacheco, C.L., Garcia, I.A., Calvo-Manzano, J.A., Arcilla, M., 2015. A proposed model for reuse of software requirements in requirements catalog. J. Softw. Evol. Process 27, 1–21. https://doi.org/10.1002/smr.1698

Pascoe, J., 1998. Adding generic contextual capabilities to wearable computers, in: Digest of Papers. Second International Symposium on Wearable Computers (Cat. No.98EX215). Presented at the Digest of Papers. Second International Symposium on Wearable Computers, IEEE Comput. Soc, Pittsburgh, PA, USA, pp. 92–99. https://doi.org/10.1109/ISWC.1998.729534

Perttunen, M., Riekki, J., Lassila, O., 2009. Context representation and reasoning in pervasive computing: a review. Int. J. Multimed. Ubiquitous Eng. 4, 1–28.

Poveda-Villalón, M., Suárez-Figueroa, M.C., Gómez-Pérez, A., 2012. Validating Ontologies with OOPS!, in: ten Teije, A., Völker, J., Handschuh, S., Stuckenschmidt, H., d'Acquin, M., Nikolov, A., Aussenac-Gilles, N., Hernandez, N. (Eds.), Knowledge Engineering and Knowledge Management. Springer Berlin Heidelberg, pp. 267–281.

Reddi, K.R., Moon, Y.B., 2009. A framework for managing engineering change propagation. Int. J. Innov. Learn. 6, 461. https://doi.org/10.1504/IJIL.2009.025060

Rocklage, E., Kraft, H., Karatas, A., Seewig, J., 2017. Automated scenario generation for regression testing of autonomous vehicles, in: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). Presented at the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), pp. 476–483. https://doi.org/10.1109/ITSC.2017.8317919

Roques, P., 2016. MBSE with the ARCADIA Method and the Capella Tool, in: 8th European Congress on Embedded Real Time Software and Systems (ERTS 2016). Toulouse, France.

Rosson, M.B., Carroll, J.M., 2009. Scenario-based design, in: Human-Computer Interaction. CRC Press, pp. 161–180.

SAE, (On Road Automated Vehicle Standards Committee), 2014. Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems. Technical Report J3016_201401.

SAE, (On-Road Automated Vehicle Standards Committee), 2018. Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles (No. SAE J 3016-2018). SAE International: Warrendale, PA, USA.

Sathyanarayana, A., Boyraz, P., Hansen, J.H.L., 2011. Information fusion for robust 'context and driver aware' active vehicle safety systems. Spec. Issue Inf. Fusion Cogn. Automob. 12, 293–303. https://doi.org/10.1016/j.inffus.2010.06.004

Saunders, M.N., 2011. Research methods for business students, 5th ed. Pearson Education India.

Scherer, H., Albers, A., Bursac, N., 2017. Model Based Requirements Engineering for the Development of Modular Kits. Complex Syst. Eng. Dev. Proc. 27th CIRP Des. Conf. Cranfield Univ. UK 10th – 12th May 2017 60, 145–150. https://doi.org/10.1016/j.procir.2017.01.032

Schilit, B., Adams, N., Want, R., 1994. Context-Aware Computing Applications, in: 1994 First Workshop on Mobile Computing Systems and Applications. Presented at the 1994 First Workshop on Mobile Computing Systems and Applications (WMCSA), IEEE, Santa Cruz, California, USA, pp. 85–90. https://doi.org/10.1109/WMCSA.1994.16

Schilit, B.N., Theimer, M.M., 1994. Disseminating active map information to mobile hosts. IEEE Netw. 8, 22–32. https://doi.org/10.1109/65.313011

Schuldt, F., Reschka, A., Maurer, M., 2018. A Method for an Efficient, Systematic Test Case Generation for Advanced Driver Assistance Systems in Virtual Environments, in: Winner, H., Prokop, G., Maurer, M. (Eds.), Automotive Systems Engineering II. Springer International Publishing, Cham, pp. 147–175. https://doi.org/10.1007/978-3-319-61607-0_7

Sippl, C., Bock, F., Lauer, C., Heinz, A., Neumayer, T., German, R., 2019. Scenario-Based Systems Engineering: An Approach Towards Automated Driving Function Development, in: 2019 IEEE International Systems Conference (SysCon). Presented at the 2019 IEEE International Systems Conference (SysCon), IEEE, Orlando, FL, USA, pp. 1–8. https://doi.org/10.1109/SYSCON.2019.8836763

Sun, Y., Yang, G., Zhou, X., 2016. A novel ontology-based service model for cyber physical system, in: 2016 5th International Conference on Computer Science and Network Technology (ICCSNT). Presented at the 2016 5th International Conference on Computer Science and Network Technology (ICCSNT), pp. 125–131. https://doi.org/10.1109/ICCSNT.2016.8070133

Sutcliffe, A., 2003. Scenario-based requirements engineering, in: 11th IEEE International Requirements Engineering Conference, 2003. IEEE, pp. 320–329. https://doi.org/10.1109/ICRE.2003.1232776

Tartir, S., Arpinar, I.B., Sheth, A.P., 2010. Ontological Evaluation and Validation, in: Poli, R., Healy, M., Kameas, A. (Eds.), Theory and Applications of Ontology: Computer Applications. Springer Netherlands, Dordrecht, pp. 115–130. https://doi.org/10.1007/978-90-481-8847-5_5

Taş, Ö.Ş., Kuhnt, F., Zöllner, J.M., Stiller, C., 2016. Functional system architectures towards fully automated driving, in: Intelligent Vehicles Symposium (IV), 2016 IEEE. IEEE, pp. 304–309.

Toval, A., Nicolás, J., Moros, B., García, F., 2002. Requirements Reuse for Improving Information Systems Security: A Practitioner's Approach. Requir. Eng. 6, 205–219. https://doi.org/10.1007/PL00010360

Ulbrich, S., Menzel, T., Reschka, A., Schuldt, F., Maurer, M., 2015. Defining and Substantiating the Terms Scene, Situation, and Scenario for Automated Driving, in: 2015 IEEE 18th International Conference on Intelligent Transportation Systems. Presented at the 2015 IEEE 18th International Conference on Intelligent Transportation Systems, pp. 982–988. https://doi.org/10.1109/ITSC.2015.164

Ulbrich, S., Nothdurft, T., Maurer, M., Hecker, P., 2014. Graph-based context representation, environment modeling and information aggregation for automated driving, in: 2014 IEEE Intelligent Vehicles Symposium Proceedings. Presented at the 2014 IEEE Intelligent Vehicles Symposium Proceedings, pp. 541–547. https://doi.org/10.1109/IVS.2014.6856556

Voirin, J.-L., Tailliez, F., 2012. Method to aid the design of a system architecture. U.S. Patent No. 8,103,490.

Wachenfeld, W., Winner, H., Gerdes, J. Chris, Lenz, B., Maurer, M., Beiker, S., Fraedrich, E., Winkle, T., 2016. Use Cases for Autonomous Driving, in: Maurer, M., Gerdes, J. Christian, Lenz, B., Winner, H. (Eds.), Autonomous Driving: Technical, Legal and Social Aspects. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 9–37. https://doi.org/10.1007/978-3-662-48847-8_2

Weiss, G., Grigoleit, F., Struss, P., 2013. Context modeling for dynamic configuration of automotive functions, in: 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013). Presented at the 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), pp. 839–844. https://doi.org/10.1109/ITSC.2013.6728336

Winkler, S., von Pilgrim, J., 2010. A survey of traceability in requirements engineering and model-driven development. Softw. Syst. Model. 9, 529–565. https://doi.org/10.1007/s10270-009-0145-0

Wyatt, D.F., Eckert, C.M., Clarkson, P.J., 2009. Design of product architectures in incrementally developed complex products, in: DS 58-4: Proceedings of ICED 09, the 17th International Conference on Engineering Design, Vol. 4, Product and Systems Design, Palo Alto, CA, USA, 24.-27.08. 2009. pp. 167–178.

Xie, Y., Ma, Y., 2016. Well-controlled engineering change propagation via a dynamic inter-feature association map. Res. Eng. Des. 27, 311–329. https://doi.org/10.1007/s00163-016-0220-9

Yang, F., Duan, G., 2012. Developing a parameter linkage-based method for searching change propagation paths. Res. Eng. Des. 23, 353–372. https://doi.org/10.1007/s00163-011-0124-7

Zhang, Z., 2007. Effective requirements development-A comparison of requirements elicitation techniques, in: Software Quality Management XV: Software Quality in the Knowledge Society (SQM2007). British Computer Society, pp. 225–240.

Zimmermann, A., Lorenz, A., Oppermann, R., 2007. An Operational Definition of Context, in: Kokinov, B., Richardson, D.C., Roth-Berghofer, T.R., Vieu, L. (Eds.), Modeling and Using Context. Springer Berlin Heidelberg, pp. 558–571.

Appendix A: Paper #1. A semi-automated requirements reuse and recycling process for Autonomous Transportation Systems R&D

Youssef Damak, Marija Jankovic, Yann Leroy and Karim Chelbi

This paper has been published in Proceedings of the Design Society: International Conference on Engineering Design 2019 (ICED19) under the following reference:

Damak, Y., Jankovic, M., Leroy, Y., Chelbi, K., 2019. A Semi-automated Requirements Reuse and Recycling Process for Autonomous Transportation Systems R&D, in: Proceedings of the Design Society: International Conference on Engineering Design. pp. 3551–3560. https://doi.org/10.1017/dsi.2019.362

Abstract. *The R&D of Autonomous Transportation Systems (ATS) is hindered by the lack of industrial feedback and client's knowledge about technological possibilities. In addition, because of intellectual properties (IP) issues, technology consulting companies can't directly reuse developed functionalities with different clients. In this context, requirements reuse technics presents a good way to capitalize on their knowledge while avoiding IP issues. However, the literature review on requirements reuse processes doesn't propose methods to the application of reuse processes with little information about the system's operational context. In this paper, we present a semi-automated requirement reuse and recycle process for ATS R&D. The process helps designers' copes with the lack of inputs from the clients. Requirements candidates are retrieved from a database using Natural Language Processing and traceability propagation. It is applied on 3 use cases with inputs less than 5 concepts from the client's needs. The results validate its efficiency through number requirements retrieved and the analysis time consumption*

Keywords. *Requirements, Autonomous Transportation Systems, Early design phases, Process modelling*

A.1. Introduction

Autonomous Transportation Systems (ATS) are perceived as a pillar of future mobility by all the mobility's stakeholder. Hence, Research & Development (R&D) focus on the design and industrialization of ATS is greatly increasing. For instance, carmakers are racing to be the first to industrialize a safe Autonomous Vehicle (AV). In this effervescent context, technology consulting companies help their clients conducting their R&D for ATS. Having a large spectrum of client's industry, the consulting companies face the R&D of various contexts and needs. However, much of the explored solutions overlap, as the different ATS share many characteristics.

Up until now, there have been no industrialization or industrial experimentation of an ATS with an autonomy level 4 or above on the SAE scale. A shown in Damak et al. (2018), the transition from the level 2 and 3 to upward levels implies a structuring change on the system's requirements. Therefore, designers face a lack of industrial feedback for R&D of level 4 ATS. For this particular reason, the rich experience and expertise in technology consulting companies are highly needed to accelerate the advances in the design of ATS. However, the companies are bound with intellectual properties issues and cannot directly share and reproduce developments from a client to another. Furthermore, ATS being context sensitive, the designers have to adapt developed and known solutions and manage the context changes impact (Horváth, 2014).

Requirements reuse is a quiet standard activity in the requirements engineering process. In early design stage, it helps designers identify reusable solutions through the analysis of shared requirements and context elements. In addition, the identified differences also indicate the changes needed in known solutions. Besides, Requirements reuse allows for system design feasibility assessment without direct solution reuse. This would avoid intellectual properties issues.

This research work proposes a semi-automated process of reusing and recycling the consulting companies' knowledge about ATS through the reuse and recycling of requirements. We propose the following structure of the paper. The second part reviews the literature of requirements reuse and recycling in requirements engineering. The

proposed process is then presented in the third part. A case study in the fourth part is presented to validate its efficiency in the context of ATS design before discussing perspective in the fifth part.

A.2. Literature Review

There are many definitions of requirements from different standards of different communities such as the INCOSE, ISO, and IEEE. In this study, ATS requirements are expressed by the clients and experts of the domain in an R&D context. They express expected operational and functional properties for the ATS and are not always measurable. In the remaining of the paper, we consider the following definition of a requirement that best corresponds to the context of the study: "the definition of a property of a system that is either needed or wanted by a stakeholder" (Holt et al., 2012). The following sub-parts review the literature of requirements elicitation in requirements engineering process, more specifically, Requirements Reuse (RR) strategies and methods.

A.2.1. Requirement Engineering

starts with the elicitation of clients' needs and requirements up to the validation & verification of these requirements by the designed system. RE activities are generally defined as follows (Berkovich et al., 2011; Jiao and Chen, 2006):

- Stakeholders' requirements elicitation,
- Stakeholders' requirements analysis,
- Requirements specification,
- Requirements change management.

This paper focuses on the activities of requirements elicitation and analysis in ATS R&D. In the requirements analysis activity, Model-Based Requirements Engineering (MBRE) has become a widespread approach (Holt et al., 2012; Scherer et al., 2017). Requirements modelling consists in representing the requirements attributes, description, identifier, and possible others, and its relation to other requirements and system's functions. The principle

sub-activities of requirements analysis consist of modelling and classifying system requirements and detecting requirement conflicts.

Prior to requirements modelling and analysis, the system's requirements have to be elicited. Requirements elicitation consists of the following activities: research, discover, identify and elaborate client requirements. Zhang (2007) identified in the literature 4 types of requirements elicitation method: conversation, observation, synthetic and analytic methods.

The two first methods, conversation and observation methods have in the case of ATS R&D a low efficiency in eliciting client's requirements. In fact, the technics used in these methods such as interviews, workshops and users observations are hindered by the lack of the client's knowledge about the current technological possibilities in the design of ATS (Curcio et al., 2018). As for synthetic technics such as scenarios, storyboarding and prototyping, they are quite efficient and often used in the context of R&D (Sutcliffe, 2003). However, they are not best suited to identify reusable development and solutions, as well as making the best use of previous experiences for the design of emergent systems such as the ATS.

Lastly, analytic technics consist of documentation studies and requirement reuse technics. They are efficient for the development of systems that share similar contexts and characteristics such as modular systems and product lines (Adam and Schmid, 2013). In the context of low knowledge feedback and a need of previous experience, RR technics seem to be better suited to the efficient reuse and recycling of developed solutions.

A.2.2. Requirement Reuse

Requirements Reuse in Requirements Engineering consists in selecting requirements from defined and verified system requirements of a previous project to use them in a new project. It is used to reduce development time & cost and increase the productivity and quality of products. It is particularly useful to help stakeholder rapidly elicit a system's requirements, especially when their knowledge about the current technological possibilities is lacking (Pacheco et al., 2015; Toval et al., 2002).

RR methods and technics have been greatly explored in the software domain. The literature about software RR shows that all RR processes are based on the selection of requirements candidates from a requirement database (Irshad et al., 2018). Moros et al. (2008) introduce a method for modelling software requirements *for reuse*, then modelling new software requirements *by reuse*. The requirements for reuse are classified in catalogues while ensuring the traceability and relations between the requirements. In their cases, the catalogues are software functionalities. When such functionalities are needed in new projects, the requirements corresponding to these functionalities are selected for reuse (Moros et al., 2008).

Prior to functionality identification, stakeholders' needs are the inputs of design processes and RE processes. Other methods in the literature aim at identifying reusable requirements before the analysis and identification of system's functionalities. For instance Kaiya and Saeki (2006) map the requirement database with domain ontologies. They match concepts expressed in the stakeholders needs with concepts from the domain ontology to identify missing reusable requirements from system's requirements.

Whether the input for RR is system functionality or stakeholder need, direct reuse of requirements is not always possible. When system functionalities are reused in new contexts and integrated to new system functionalities, the description of their requirements must be adapted. This activity is called requirement recycling. It can be defined as keeping the suitable parameters in the base of the requirement description, adapting the other parameters to the new context and integrating the resulting requirement to the new system's requirements (Alexander and Kiedaisch, 2002).

Few propositions have been made for parameter recycling. Knethen et al. (2002) propose a systematic process to identify requirement's recycled parameters. For that, they use abstractions of the database requirements in the form of templates. They map these templates with conceptual model of the system. And finally, they use the differences between the former and new systems' conceptual models to deduce the change in requirements parameters. Quite similarly, Alexander and Kiedaisch (2002) map requirements with use cases. Then, in the same fashion, they use the changes is the use cases to deduce the change in requirements parameters.

On the other hand, Toval et al. (2002) recommend the inclusion of the stakeholders in the RR process for requirements recycling through parameters analysis and negotiation. The inclusion of stakeholders ensures a better consistency of new parameters elicitation for requirements to be recycled. It also serves the elicitation and integration of new requirements which is an important activity for the success of an RR process.

By contrast to software systems, ATS functionalities and capabilities implicate heterogeneous interaction and complex interfacing of ATS components. The reuse of an ATS functionality in a different ATS includes the integration of developed components with new system components. And due to the changes in the ATS physical properties and heterogeneous interactions, the reused components may necessitate important adaptation. Besides, as stated previously, RR is important in ATS R&D to overcome the lack of the client's knowledge and his inability to explicitly describing his needs and the ATS operational context. Therefore, relying on functionality reuse for RR is not suited for ATS R&D case and requirement recycling is necessary. For requirement recycling, using system's conceptual models or use cases to deduce parameters modification is also incompatible. Hence, as recommended by Toval et al. (2002), including stakeholders in the recycling seem to be better suited. However, to the best of our knowledge, no process in the literature allows efficient RR and recycling with little information about stakeholders' needs and the system's operational context.

A.3. Requirements Reuse & Recycling Process

As stated previously, ATS are context sensitive and emergent systems. Currently, designers and engineers have access to very little industrial and knowledge feedback. To implement an efficient reuse of ATS R&D knowledge the authors propose a semi-automated process for reusing and recycling ATS requirements. This process permits the identification of relevant requirements from former ATS R&D projects with little information about the new ATS's operational context. It also includes the clients in the RR process to improve his ATS domain knowledge to improve the process' accuracy.

In this part, we propose a requirements elicitation process through Requirements Reuse and Recycling (RRR) process. The process in Figure A.1 is divided into 3 main sub-

processes. First and prior to any new ATS R&D project, a database of former projects' system requirements is built. The requirements in the database are modelled *for reuse*. Second, requirement candidates are semi-automatically selected from the database then reused or recycled. Clients are integrated to the reuse and recycling activity. Third, project specific requirements are elicited with the client and integrated to the reused and recycled requirements. Finally, at the end of the whole requirements engineering process, the final ATS requirement is used to update the requirements database. Maintaining the database for future elicitation processes is one of the RRR challenges (Toval et al., 2002).

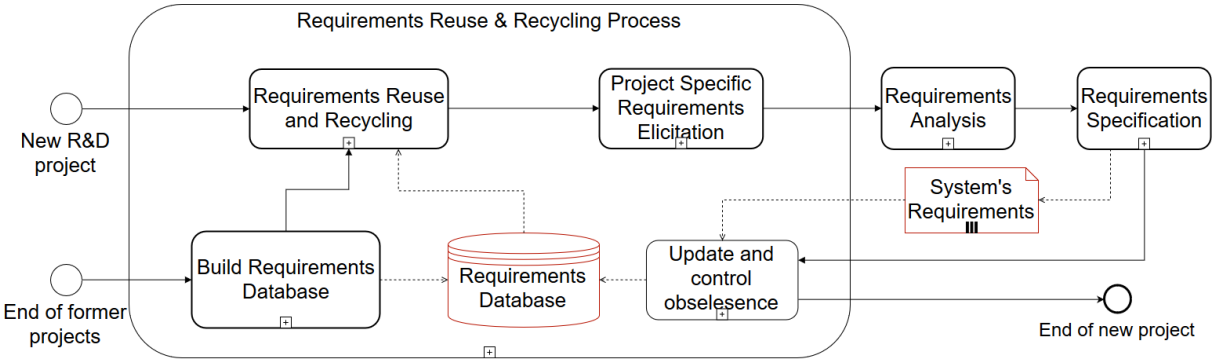


Figure A.1: Requirements Reuse and Recycling process

A.3.1. Requirement database building

To model complex systems requirements *for reuse*, we propose a modelling framework including the requirement types, the relations between requirements and the semantic structure of the requirement. To create the requirements database, we apply this modelling framework on the system requirements of previous projects using graph modelling and processing with python:

A.3.1.1. Requirement types and hierarchy:

The literature contains various types of requirements that differ between domains and uses. The most common types are Operational, Functional, Non-functional and Business Requirements (Holt et al., 2012). During our industrial observations of ATS R&D, the following 3 requirement types were recurrently expressed and used by clients, experts and other stakeholders:

- Non-Functional Requirements (NFR): an NFR express a global constraint on the system and the other types of requirements. It expresses "ilities" such as quality, safety, maintainability, etc.
- NFR example: The safety of the platooning_system must always be ensured
- Operational Requirements (OR): an OR expresses a behavioural requirement that the system must satisfy. It shouldn't indicate any functional solution to produce the intended behaviour
- OR example: The platooning_system must start if the START_command is activated from HMI
- Functional Requirements (FR): an FR expresses a function that must be fulfilled by a part of the system. Through the design process, the ATS functions are defined to satisfy ORs and FRs. More detailed FRs can be derived from these functions and their interactions.
- FR example: The communication_network must ensure the transfer of the START_command

During the design process, designers may have to decide between different solution alternatives to satisfy a system requirement. As stated previously, more detailed system requirements can be derived from each solution. This results in several alternatives of requirement sets that satisfy the system requirement. In the database, we model these requirement alternatives using decision gates. As illustrated in Figure 2, the decision gate indicates two alternatives to satisfy OR1: FR1 or FR2. During the design process, designer will have to decide between satisfying FR1 or FR2.

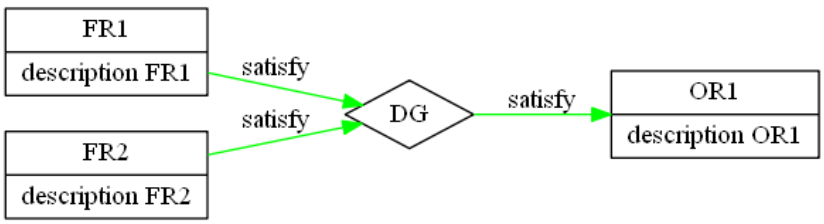


Figure A.2: Decision gates in the requirements database

A.3.1.2. Relations between requirements:

In the 3.2, we detail the identification of requirement candidates for reuse or recycling. The links between requirements is used for one of the ways to do so. According to Winkler and von Pilgrim, the definition of the traceability links depends on the analyst's interpretation (Winkler and von Pilgrim, 2010). Same as for requirements types, we identified the most relevant and used requirements links during our industrial observation of ATS R&D projects. In the database we model the following links:

- "Refine" link: the requirements R1.1 and R1.2 refine R1 when R1, being complex, is broken down into smaller and more manageable requirements R1.1 and R1.2.
- "Satisfy" link: The requirement R2 satisfies R1 when R2 is a lower level requirement that has been defined for the system to satisfy R1.
- "Evolve to" link: The requirement R1 evolve to R2 when R2 is a newer version of R1

The traceability links are also deeply connected to the hierarchical structure of a requirements system. According to these definitions, the "Refine" and "Satisfy" define the requirements hierarchy. We can also notice that the "Refine" link conserves the requirement type, while the level and type transition of requirements are modelled through the "Satisfy" link. As an example, an FR can "satisfy" an OR while an NFR can be satisfied by both ORs and FRs. This hierarchical structure verifies the key K2.

A.3.1.3. A formal semantic structure:

The use of a formal semantic structure facilitates the identification of requirements candidates with the little information obtained from the clients. It also conserves requirements consistency while recycling their parameters for database update (cf. A.3.2). In addition, it reduces the expression ambiguity of the requirement's description. To obtain the parameters of the formal semantic structure that would capture the complex aspects of ATS, we based it on a proposal for formal modelling of CPS properties (Garro et al., 2016):

- **What** is to be satisfied;
- **When** in time that is to be satisfied;

- **Which** entity in the system that is to be satisfied;
    - <source object>: the entity that satisfies the requirement
    - <Target object>: the entity that is the target of the requirement
- **How well** that is to be satisfied

The requirement must express: an entity (**Which** <source object>), that must, have to, could, or should satisfy something (**what)**. In some complex requirements, this action is applied on/to a second entity (**which** <target object >) and can be satisfied only in a certain condition (**when**) with a certain quality (**how well**). Hence, the requirement description structure is as follows:

Which <source object> <must/have to/could/should> **what**
[<on/to> **which** <target object >], [**when** and **how well**]

As an example, the following requirement illustrates the different structure's elements:

The follower_vehicle (**which** <source object>) must send a warning_message (**what**) to the fleet_management_system (**which** <target object >), if a failure is detected (**when**)

A.3.2. Requirements reuse & recycling

In part 2, Requirements candidates' identification for reusing and recycling was conducted through functionalities, use cases and domain ontologies. In The case of ATS R&D, the lack of information on the client's need and the system's operational context prevent the use of the two first. When browsing the database, the analysis of the requirement's parameters is the only way of identifying potential requirements candidates. In our proposed process, we identify requirements candidates by matching a few key concepts expressed by the stakeholders with the parameters from the formal structure of requirements' descriptions. This step is automated using Natural Language Processing (NLP) and matching concepts. The used technic is out of the scope of this paper and in the remaining of the paper, we consider that we matched key concepts from stakeholders with some of the requirements' parameters. The requirements containing these parameters are the first set of requirements candidates to be selected. From this set, we use their traceability links to other requirements in the database to retrieve more requirements candidate. Further details about this step come in the next paragraph.

The results of the literature review showed that including stakeholders in the requirement reuse and recycle process is more adapted to ATS R&D. However, if the overall candidates' selection results in a considerable number of candidates, it would be complicated and time consuming for the clients and requirements engineers to process each one of them. We propose in this process to prioritize the set of requirements candidates that are proposed to them. With these successive sets, the engineers select with the clients what requirements are relevant for the project. The first set of candidates retrieved through NLP is the most relevant set of requirements to be processed. Its requirements have direct semantic links to the clients need. We classify them as requirements of the category C1.

One by one, the C1 requirements are automatically proposed to the engineers and clients, and they manually chose to reuse, recycle or discard it. If they chose to recycle the requirements, they arbitrarily change the parameters of the requirement's semantic structure. The structure is conserved, and its parameters are updated. After processing all C1's requirements, the remaining ones are used to retrieve more candidates through their traceability link.

The next challenge is to determine which set of requirements has the highest priority to be processed. A requirement can be linked to other requirements with the "satisfy" or "refine" links. It can satisfy/refine or be satisfied/refined by other requirements. At this point, the category C1 contains the first set of reused/recycled requirements. The requirements must be realized by the system. It is then logical to analyze what are the requirements that satisfy or refine this set of requirements. Hence, the next set of candidates is composed of the requirements that satisfy or refine C1 requirements. They form the category C2. We call the process to retrieve C2 requirements *backward traces propagation*. On the other hand, it is also important to know the purpose of the reused/recycled requirements. What they satisfy or refine represent the reasons why such requirements exist in the first place. Hence, the third set of candidates is composed of the requirements that are satisfied or refined by C1 requirements. They form the category C3. We call the process to retrieve C3 requirements *frontward traces propagation*. C2 and C3 requirements are then successively processed to be either reused, recycled or discarder.

Following this reasoning, we propose a tree structure for the requirements candidates' categories. As illustrated in Figure A.3, the root of the tree is the category C1. Each node of the tree has two child nodes: the left child node represents its sub-C2 category and is filled through *backward traces propagation* on its parent nodes requirements. The right child node represents its sub-C3 category and is filled through *frontward traces propagation* on its parent nodes requirements. The generation of categories stops when no more requirements candidates are detected in the database.

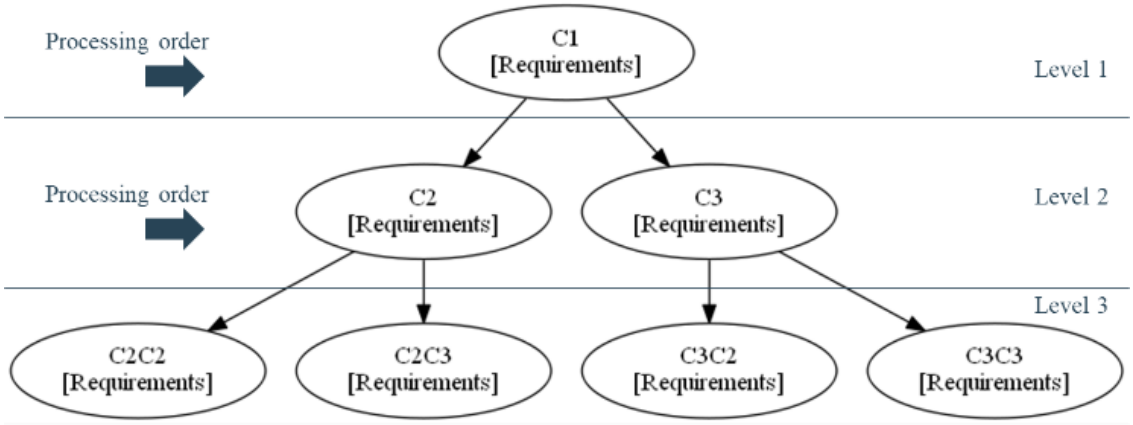


Figure A.3: Candidates Category Tree

Figure A.4 illustrate an example of the filling of the requirements category tree from a small database of 16 requirements. In his example, FR1 is detected through NLP. For the sake of the example FR8 was supposed to be discarded by the client when proposed. Through this example, we can notice several characteristics of this process. First, all the requirements linked to FR8 where not proposed to be reused or recycled. In addition, FR16 was isolated, thus couldn't be reached by the traceability propagation process. Besides, the decision gate between FR4 and FR5 was conserved to warn the requirements engineer. Later in the project, a decision must be taken to satisfy either FR4 or FR5. Finally, we can notice that once a requirement has been processed in a category, it does never appear in lower level categories, even if it has been discarded.

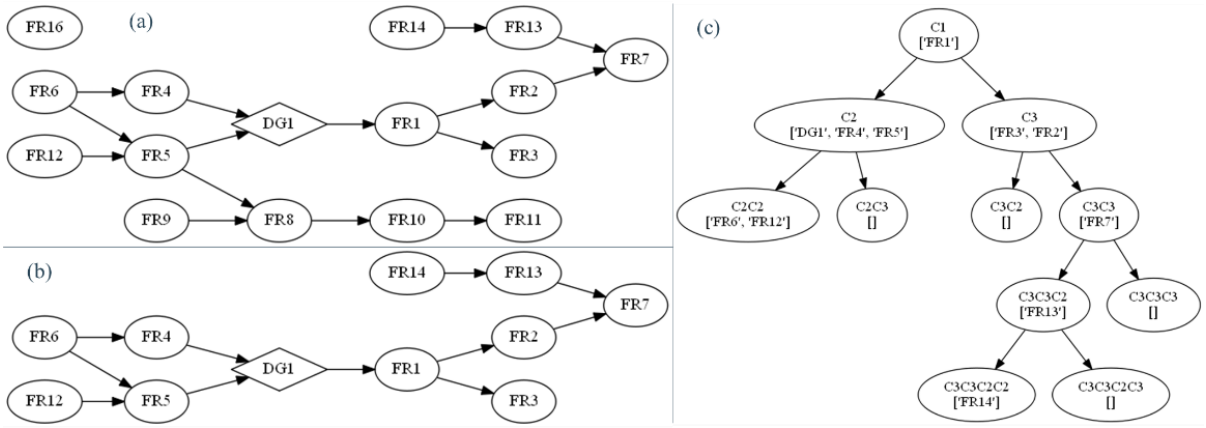


Figure A.4: Example of requirements candidates' selection: (a) Requirements database graph; (b) reused/recycled requirements graph; (c) candidates category tree

A.3.3. New requirements elicitation and requirements integration

The requirements reused and recycled in the previous step are a base for discussion with the client. In fact, during the previous step, the client's ATS domain knowledge should have increased through the analysis and negotiation of reused and recycled requirements. With a better understanding of the technological possibilities, the client is more capable to express his requirements. Hence, using reused and recycled requirements as a base, requirements engineers manually elicit with the client missing ones to complete the system's requirements. In this task, they are free to use any elicitation technic to complete and clarify the requirements.

Newly elicited requirements are integrated to the reused and recycled requirements through the traceability links analysis. The global requirements engineering process continues with the result of this sub-process. Requirements feasibility and analysis, conflict management and requirements specification are conducted until the final system's requirements are clearly defined.

A.3.4. Database update

As stated in the beginning of part 3, the requirement database maintenance for future elicitation processes is one of the RRR challenges (Toval et al., 2002). The final sub-process's aim is the update of the database. Throughout the ATS R&D project, the

requirement's system evolves drastically. The final version of the system' requirement is used to update the requirements database. For the update, the following steps are automatically applied to the database:

- Recycled requirements are added to the database with an "evolve to" link from the older version to this new version,
- New elicited requirements are added to the requirements database with their corresponding links to the recycled requirement,
- A conflicts analysis is conducted to avoid integrating conflicting requirements to the database. Each new requirement is manually checked with the requirements that satisfy the same upper level requirements. If conflict is detected between two requirements, a decision gate is included.

A.4. Case Study

To validate our process, we tested this RR process with the autonomous systems team of AKKA Technologies. We built the requirements database from a former project of platooning system design. The platooning system is an ATS composed of a lead vehicle, driven by an operator, and followed by a number of autonomous follower vehicles. The team is developing a platooning system in collaboration with other companies for the integration on a test platform and the development of communication network between the vehicles. From a part of this project, we built a database with 2 NFR, 22 OR and 76 FR and one decision gate. The database's requirement graph is illustrated in Figure A.5. The red links represent "refine" links while the green links represent "satisfy" links.

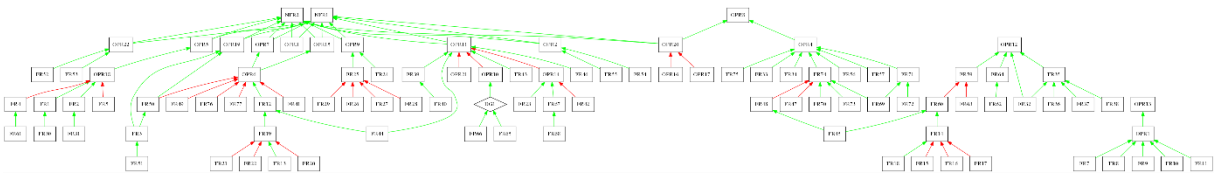


Figure A.5: Sample of the platooning system's requirements database graph

3 tests were conducted to validate our process. For each case, around an hour was taken for the reuse, recycling, and discard analysis. The results are summarized in Table A.1:

- Use case 1: A new client asks for the R&D of a new platooning system. Some aspects of the operational context were different from the former platooning project, but we used 3 key concepts expressed by the client: "follower_vehicle", "lead_vehicle", and "platoning_system". In this use case, we could retrieve all the requirements from the database. Interestingly, the category C1 contained half of the database. Even for a new project very close to the former one, we could eliminate half of the database from the initial discussion to avoid overwhelming the client.
- Use case 2: A new client asks for the R&D to automate a vehicle. In its operational context, it should be able to handle an intersection. Hence, we used the following 3 key concepts for candidates' selection: "intersection", "vehicle", and "autonomous".
- Use case 3: The use case 2 client added new information to the operational context. The vehicle must activate an emergency stop if it encounters an obstacle on its way. Hence, we added two more key concepts and used the following concepts for candidates' selection: "intersection", "vehicle", "autonomous", " obstacle", and "emergency_stop"

Table A.1: Use cases results

	Use case 1	**Use case 2**	**Use case3**
Number of C1 requirements	51	9	11
Number of C2 and children requirements	14	10	12
Number of C3 and children requirements	35	25	22
Total of reused/recycled requirements	100	44	45

The results of Table A.1 show that in an hour and with as little inputs as 3 concepts, we could reuse and recycle up to 44 requirements from a database of 100 requirements. The categorization of the candidates helped the engineers handle the important number of requirements and progressively improved their understanding of the clients' needs. However, we can also notice from the table that using 2 more concepts, in this case, didn't drastically change the number of reused/recycled requirements. Although the number didn't change, Figure A.6 shows that the categories, and the process by extension, were impacted. We conclude that more information changes the priority of the system's requirements.

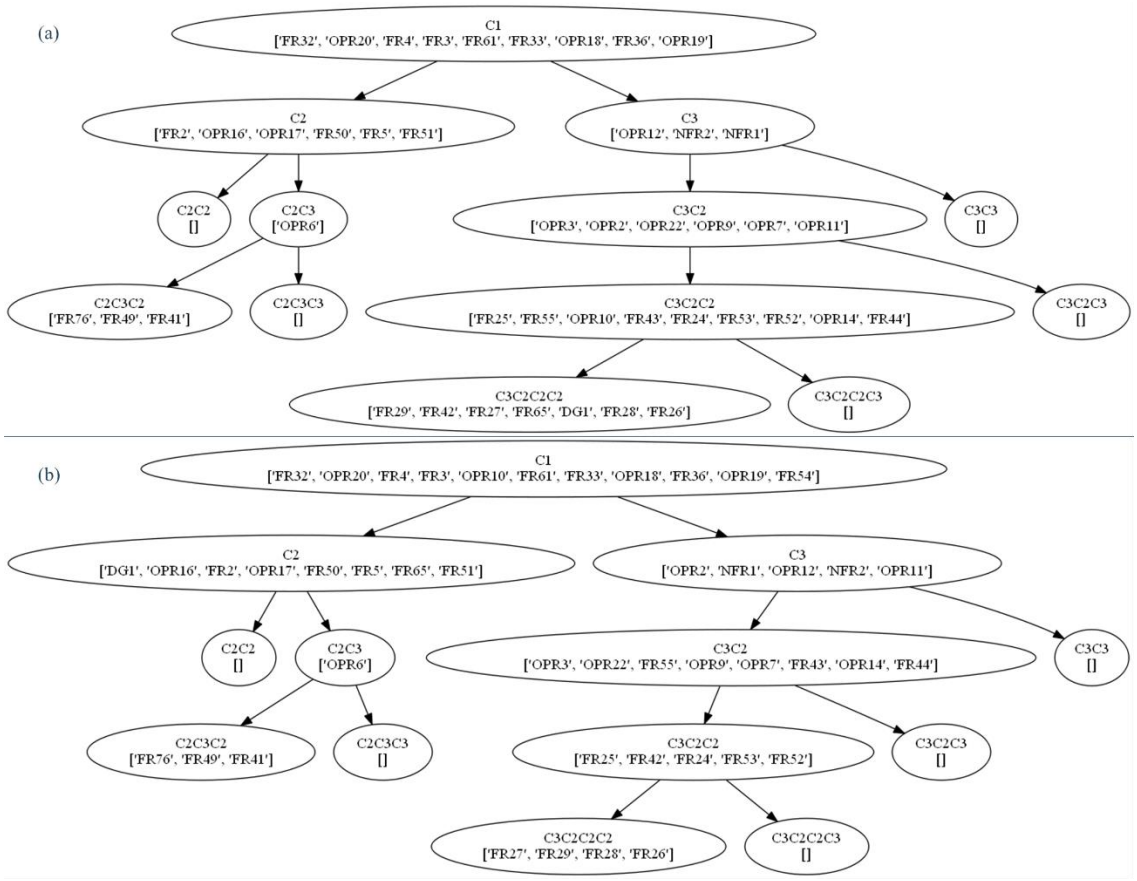


Figure A.6: Reused/recycled requirements categories: (a) use case 1; (b) use case 2

A.5. Conclusion

Requirements elicitation in ATS R&D presents many challenges. The lack of industrial feedback, and client's knowledge about technological possibilities prevent the usual elicitation methods. to accelerate this process, technology consulting companies can reuse former knowledge on ATS from different industry. However, direct reuse of functionalities and technical solutions is hinder by complex integration and intellectual property issues. Nonetheless, requirements reuse, and recycling seems to be on the best ways of reusing former knowledge while avoiding these issues. Yet, to the best of our knowledge, the literature on requirements reuse doesn't cover the lack of inputs on the new operational context as well as the structural complexity of ATS.

In this paper, we propose a requirement reusing and recycling process that deals with the mentioned challenges. Through a formal semantic modelling of the requirements and a structured database, we use NLP to identify requirements candidate to be reused. We

prioritize the candidate and classify them in categories. The prioritization allows requirements engineer to implicate the stakeholders in the reuse, recycling or discarding analysis of requirements candidates. By prioritizing the candidates, stakeholders' knowledge about the system increase progressively. Finally, the maintenance of the database is ensured through tracing the recycled requirements to the original ones. The formal semantic structure also helps conserving the database consistency while requirements parameters are recycled. We validated this approach with a comparative study on 3 use cases. The results show the efficiency in the number of reused/recycled requirements and the time of processing. With little information about the operation context, we could reuse and recycle more than 40% of the requirements database.

Although efficient, this process focusses on the recycling of requirements parameters. In our proposition, we do not consider the recycling of attributes such as requirements maturity, criticality, priority, etc. In addition, we do not control the obsolescence of requirements in the database. In future work, we should focus on improving these elements. Besides, the process should be tested and validated on a bigger scale and with a database combining several projects.

A.6 References

Adam, S., Schmid, K., 2013. Effective Requirements Elicitation in Product Line Application Engineering – An Experiment, in: Requirements Engineering: Foundation for Software Quality. Springer Berlin Heidelberg, pp. 362–378. https://doi.org/10.1007/978-3-642-37422-7_26

Alexander, I., Kiedaisch, F., 2002. Towards recyclable system requirements, in: Proceedings Ninth Annual IEEE International Conference and Workshop on the Engineering of Computer-Based Systems. pp. 9–16. https://doi.org/10.1109/ECBS.2002.999817

Berkovich, M., Leimeister, J.M., Krcmar, H., 2011. Requirements Engineering for Product Service Systems. Business & Information Systems Engineering 3, 369–380. https://doi.org/10.1007/s12599-011-0192-2

Curcio, K., Navarro, T., Malucelli, A., Reinehr, S., 2018. Requirements engineering: A systematic mapping study in agile software development. Journal of Systems and Software 139, 32–50. https://doi.org/10.1016/j.jss.2018.01.036

Damak, Y., Jankovic, M., Leroy, Y., Yannou, B., 2018. Analysis of Safety Requirements Evolution in the Transition of Land Transportation Systems Toward Autonomy. Presented at the 15th International Design Conference, pp. 2845–2854. https://doi.org/10.21278/idc.2018.0448

Garro, A., Tundis, A., Bouskela, D., Jardin, A., Thuy, N., M. Otter, L. Buffoni, P. Fritzson, M. Sjölund, W. Schamai, H. Olsson, 2016. On formal cyber physical system properties modeling: A new temporal logic language and a Modelica-based solution, in: 2016 IEEE International Symposium on Systems Engineering (ISSE). pp. 1–8. https://doi.org/10.1109/SysEng.2016.7753137

Holt, J., Perry, S.A., Brownsword, M., 2012. Model-based requirements engineering. IET.

Horváth, I., 2014. What the Design Theory of Social-Cyber-Physical Systems Must Describe, Explain and Predict?, in: Chakrabarti, A., Blessing, L.T.M. (Eds.), An Anthology of Theories and Models of Design: Philosophy, Approaches and Empirical Explorations. Springer London, London, pp. 99–120. https://doi.org/10.1007/978-1-4471-6338-1_5

Irshad, M., Petersen, K., Poulding, S., 2018. A systematic literature review of software requirements reuse approaches. Information and Software Technology 93, 223–245. https://doi.org/10.1016/j.infsof.2017.09.009

Jiao, J. (Roger), Chen, C.-H., 2006. Customer Requirement Management in Product Development: A Review of Research Issues. Concurrent Engineering 14, 173–185. https://doi.org/10.1177/1063293X06068357

Kaiya, H., Saeki, M., 2006. Using Domain Ontology as Domain Knowledge for Requirements Elicitation, in: 14th IEEE International Requirements Engineering Conference (RE'06). pp. 189–198. https://doi.org/10.1109/RE.2006.72

Knethen, A. von, Paech, B., Kiedaisch, F., Houdek, F., 2002. Systematic requirements recycling through abstraction and traceability, in: Proceedings IEEE Joint International Conference on Requirements Engineering. pp. 273–281. https://doi.org/10.1109/ICRE.2002.1048538

Moros, B., Vicente-Chicote, C., Toval, A., 2008. REMM-Studio + : Modeling Variability to Enable Requirements Reuse, in: Conceptual Modeling - ER 2008. Springer Berlin Heidelberg, pp. 530–531. https://doi.org/10.1007%2F978-3-540-87877-3_46

Pacheco, C.L., Garcia, I.A., Calvo-Manzano, J.A., Arcilla, M., 2015. A proposed model for reuse of software requirements in requirements catalog. Journal of Software: Evolution and Process 27, 1–21. https://doi.org/10.1002/smr.1698

Scherer, H., Albers, A., Bursac, N., 2017. Model Based Requirements Engineering for the Development of Modular Kits. Procedia CIRP 60, 145–150. https://doi.org/10.1016/j.procir.2017.01.032

Sutcliffe, A., 2003. Scenario-based requirements engineering, in: 11th IEEE International Requirements Engineering Conference, 2003. IEEE, pp. 320–329. https://doi.org/10.1109/ICRE.2003.1232776

Toval, A., Nicolás, J., Moros, B., García, F., 2002. Requirements Reuse for Improving Information Systems Security: A Practitioner's Approach. Requirements Engineering 6, 205–219. https://doi.org/10.1007/PL00010360

Winkler, S., von Pilgrim, J., 2010. A survey of traceability in requirements engineering and model-driven development. Software & Systems Modeling 9, 529–565. https://doi.org/10.1007/s10270-009-0145-0

Zhang, Z., 2007. Effective requirements development-A comparison of requirements elicitation techniques, in: Software Quality Management XV: Software Quality in the Knowledge Society (SQM2007). British Computer Society, pp. 225–240.

Appendix B: Details of the Analysis of the Outsourced Design Process of Autonomous Vehicles

B.1 A Survey Proposed to the Members of AKKA Technologies Autonomous Systems Team

B.1.1 General Questions:

1. What is your current position in the team ?

   …
2. How do you receive customer needs?
   - ☐ Through an oral description
   - ☐ Through a text document
   - ☐ Through an Excel sheet
   - ☐ Through presentation slides
   - ☐ Through direct discussions with the client
   - ☐ Other: …
3. Are you personally involved in eliciting (extracting information) from client's' needs?
   - ☐ Yes
   - ☐ No
4. If you answered "yes" to the previous question, can you briefly describe what tools, processes, methods and techniques you use to elicit client's needs?

   …
5. In what format do you receive the client's needs?
   - ☐ Description of operational context(s)
   - ☐ Description of operational scenarios
   - ☐ Desired capacities for the system to be developed
   - ☐ (Performance) objectives to be achieved for the system to be developed
   - ☐ Other: …
6. Do you participate in the analysis of customer needs in order to be able to think about concepts (technical solutions) that meet them?

- ☐ Yes
- ☐ No

7. If you answered "yes" to the previous question, what tools and methods do you use to analyze client's needs?
   - ☐ Operational Scenario Analysis (Sequence Diagram)
   - ☐ Operational Scenario Analysis (Text descriptions)
   - ☐ Operational Scenario Analysis (Story telling)
   - ☐ Use cases analysis (Use cases diagram)
   - ☐ Use case analysis (Mindmap)
   - ☐ Analysis of operational contexts (Mindmap)
   - ☐ Analysis of operational contexts (Matrices)
   - ☐ Analysis of the desired capacities for the system to be developed
   - ☐ Analysis of performance objectives (Matrices)
   - ☐ Analysis of operating modes (State diagram)
   - ☐ Operating mode analysis (Matrix)
   - ☐ Other: …
8. Do you validate with the clients the analysis you made of their needs?
   - ☐ you participate in the validation
   - ☐ you do not participate in the validation
   - ☐ There is no validation
9. Are the results of the customer needs analysis documented?
   - ☐ Yes
   - ☐ No
   - ☐ Other: …
10. What means do you use to take into account client's needs analysis during the project?
    - ☐ Documents
    - ☐ System models
    - ☐ Illustrations (Story telling)
    - ☐ Another person who knows them well
    - ☐ You have no way to go back on the analysis results...
    - ☐ You don't need to go back to the results of analysis
    - ☐ Other: …

11. Have you ever witnessed changes in the needs analyzed during a project?
    - ☐ Often (Go to question 12)
    - ☐ Rarely (Go to question 12)
    - ☐ Never (Skip to question 17)

B.1.2 Management of Changes in Client's Needs:

Based on your responses, you witnessed changes in the needs analyzed during a project. The following questions deal with the management of these changes

12. Are you able to identify the risks of future changes in customer needs? *
    - ☐ Often
    - ☐ Rarely
    - ☐ Never
13. If you did not answer "never" to the previous question on identification, can you describe the method or tool you use to do so?

    …
14. In analyzing a change in a client needs during a project, can you assess the following changes? (select the ones you can assess)
    - ☐ The technical feasibility of change
    - ☐ What needs to be changed in the current solution to achieve this change
    - ☐ Efforts to be made to bring about this change
    - ☐ The resources to be deployed to make this change happen
    - ☐ The time needed to make this change
    - ☐ The new priorities of the different project requirements
    - ☐ The consequence on the project schedule
    - ☐ The impact on the cost of the project
15. How do you rate in general the assessment of the elements of the previous question?
    - ☐ Very precise
    - ☐ Specifies
    - ☐ Inaccurate
    - ☐ Very imprecise
    - ☐ Not available

16. Can you describe the method or tool you use to analyze the changes and estimate the previous question, "What is the method or tool you use to estimate the change?

…

B.1.3 Difficulties in Meeting Client's Needs

To end this survey, a few questions about the difficulties encountered during the process to meet the client's needs

17. In your view, where do you see the difficulties in meeting the needs of clients today?
    - ☐ Elicitation of needs
    - ☐ Needs analysis
    - ☐ Validation of the needs analysis with customers
    - ☐ Documentation of requirements
    - ☐ Referring to the requirements during the design process
    - ☐ Validation of a concept (a technical solution) with respect to client's needs
    - ☐ Managing changes in requirements
    - ☐ Other: …
18. Can you describe the causes of these difficulties?

…

B.2 Detailed Model of the Outsourced Design Process of Autonomous Vehicles

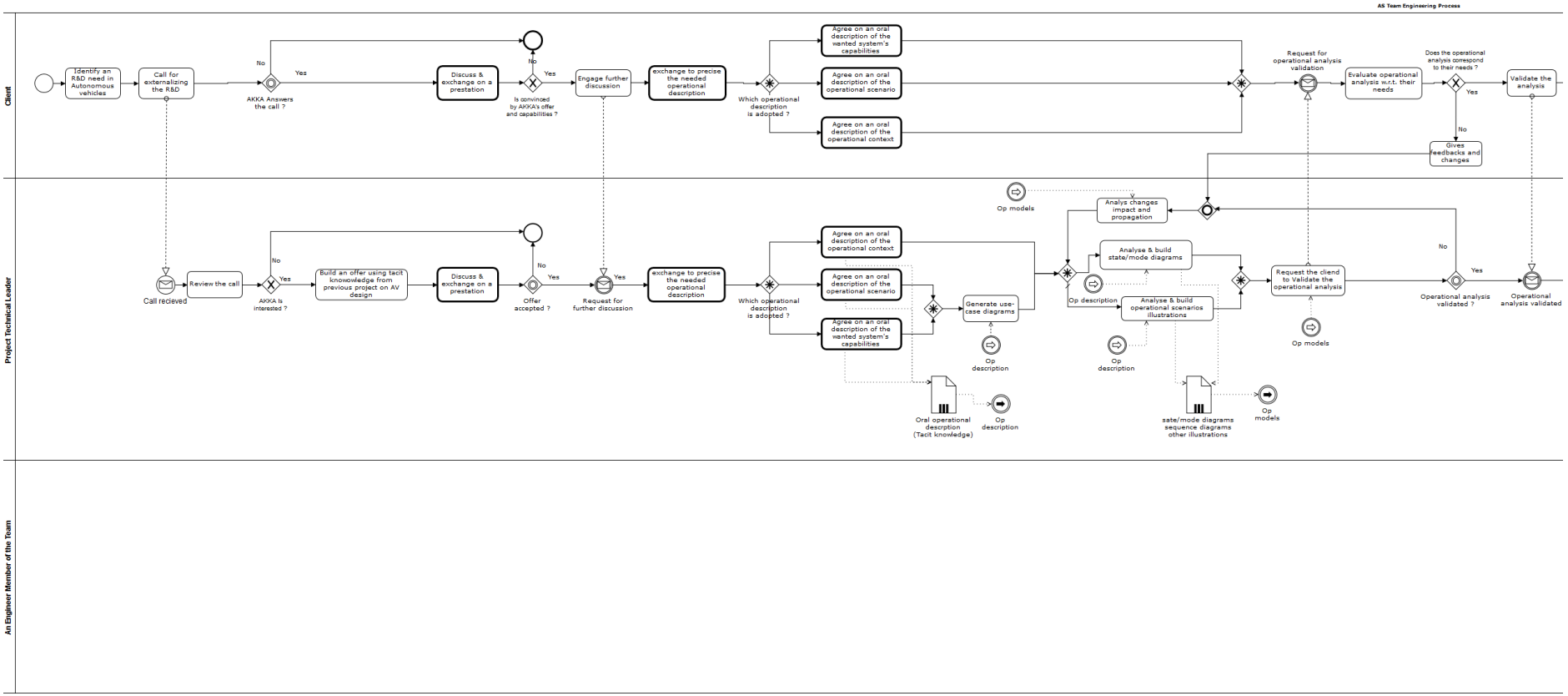


Figure B.1: Design process of Autonomous Vehicles in outsourced R&D - Part 1

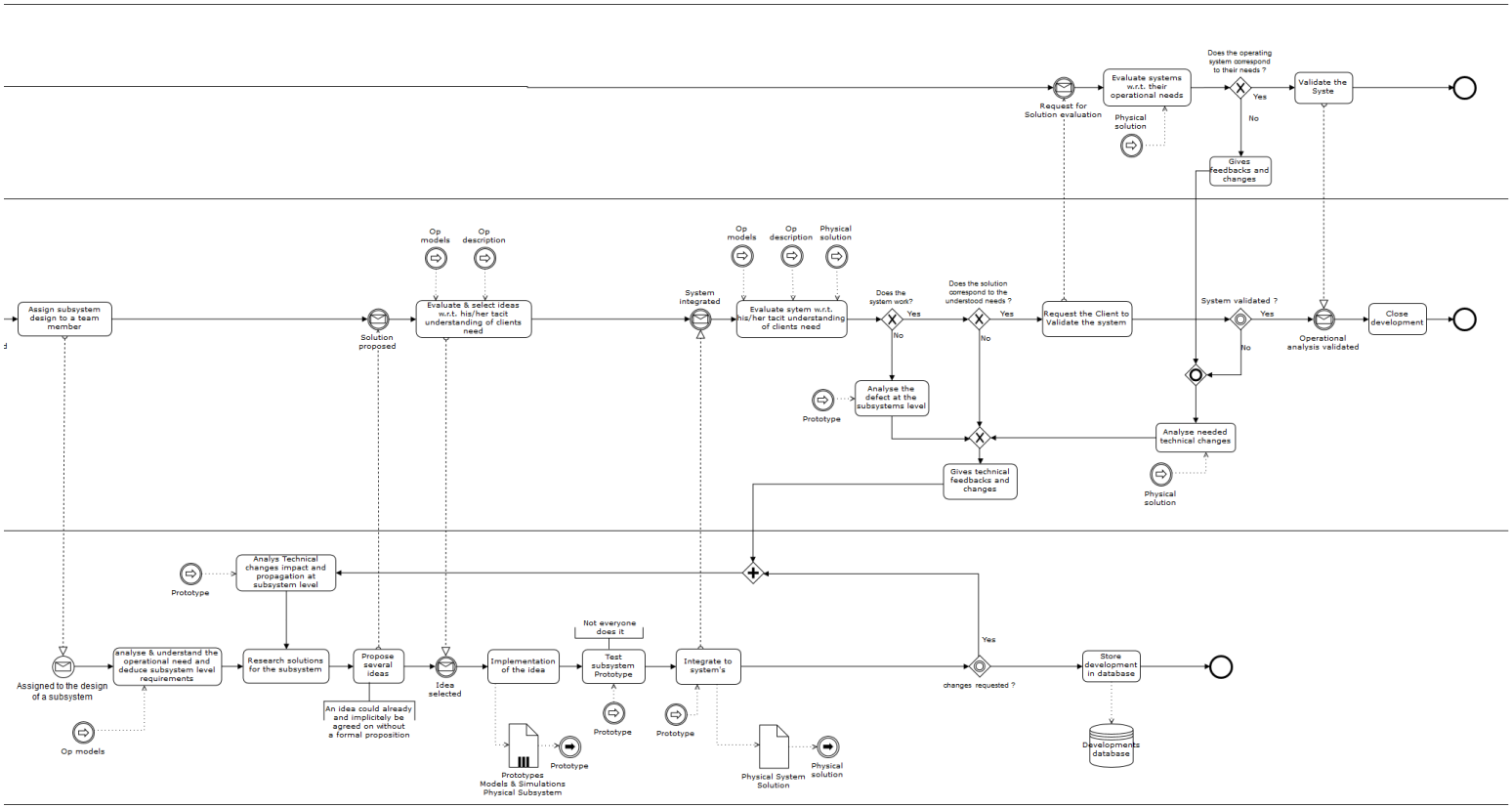


Figure B.2: Design process of Autonomous Vehicles in outsourced R&D - Part 2