



Thèse de Doctorat (36 mois)

Environnement Virtuel Immersif et Interactif pour l'Architecture de Systèmes Technologiques

Mots Clés:

*Architecture Système ; Conception Préliminaire ; Réalité Virtuelle ; Visualisation ; Interaction
Model-Based Systems Engineering ; Modélisation 3D ; Analyse Fonctionnelle*

Contexte

Comme ce fut le cas pour le génie logiciel au début des années 90, l'ingénierie système connaît une crise dont les symptômes sont identiques : budgets dépassés, livraisons tardives, et un manque de maîtrise en amont du cycle de vie.

Pour affronter cette crise, une solution fut de remplacer l'approche documentaire par une approche basée sur les modèles : *Model-Based (Systems) Engineering (MBSE)*. Aujourd'hui, il est communément accepté qu'une utilisation parcimonieuse des modèles permet de mieux maîtriser l'ingénierie d'un système technique complexe. Par ailleurs, dans une démarche d'ingénierie système, c'est l'activité d'architecture (opérationnelle, fonctionnelle, logique, organique, géométrique, etc.) qui engage 70 à 80 % des coûts du cycle de vie. Malgré cette prise de conscience collective, si on compare les outils¹ de conception détaillée avec ceux de conception préliminaire, alors on constate que ces derniers sont rudimentaires vis-à-vis de la complexité technique² et organisationnelle³ des systèmes actuels.

Les insuffisances des outils utilisés pour définir et gérer l'architecture d'un système complexe sont multifactorielles. Parmi les diverses difficultés rencontrées par les ingénieurs systèmes lors de la mise en œuvre d'une démarche de MBSE, on peut citer : la prolifération des modèles, l'apprentissage de langages dédiés, le manque d'exécution des modèles, l'influence du génie logiciel, et le manque d'intelligence des environnements de modélisation.

¹ Le terme « outil » fait référence à l'ensemble des moyens technologiques de conception : logiciels, langages, modèles...

² Nombre de composants, nombre d'interfaces, multi-physique, intégration soft-hardware, propriétés émergentes, variantes, options, etc.

³ Entreprise étendue, multi-sites, internationalisation, etc.



Objectif

Doctorant au sein du laboratoire des Sciences de la Conception, l'Optimisation et la Production de Grenoble (G-SCOP UMR CNRS 5272) de Grenoble INP, vous participerez aux recherches qui visent à inventer le bureau d'études du futur.



Plateforme technologique de visualisation et interaction VISION-R au G-SCOP

Plus précisément, votre objectif sera de d'inventer, de prototyper, et d'évaluer un environnement virtuel immersif et interactif qui permettra, à un ou plusieurs architectes systèmes, de concevoir l'architecture d'un système complexe. Plutôt que de faire appel à divers logiciels spécifiques, l'environnement immersif et interactif fédérera la définition des divers points de vue (opérationnel, spécification, fonctionnel, comportemental, structural, logique, sûreté de fonctionnement, etc.) d'une architecture système dans un espace virtuel unique avant de pouvoir exporter chacun des points de vue dans des formats standardisés qui permettront aux expertises de prendre le relais pour la conception détaillée du système.

Problématiques

- **Problème 1 « La prolifération de modèles hétérogènes»**



L'utilisation intensive de modèles pour définir l'architecture d'un système complexe engendre un gros volume d'informations difficilement exploitable. En effet, chacune des vues qu'on peut avoir sur un système se matérialise par une pléthore de modèles réalisés avec des outils différents. D'un point de vue des pratiques industrielles actuelles, la définition de la vue opérationnelle et de la vue fonctionnelle d'un système technique complexe correspond à plusieurs dizaines de diagrammes modélisés au moyen de notations standardisées tels que les diagrammes SysML, Capella ou BPMN. Pour un système complexe, ces diagrammes deviennent rapidement illisibles. D'un point de vue logique⁴, ce sont majoritairement des architectures représentées par des schémas de blocs interconnectés de flux énergétiques dont la modélisation se fait avec des outils de modélisation tels que Modelica, Simscape, LMS Amesim, ou 20-sim. Malgré la richesse de ces outils, en pratique, dans les entreprises, on observe que chacune des expertises utilise un outil différent – par exemple, LMS Amesim pour l'architecture du système hydraulique et Simulink pour l'architecture du système avionique. D'un point de vue physique, l'architecture se traduira par des modèles géométriques réalisés avec des outils de conception assistée par ordinateur tels que CATIA, Creo, ou NX. La modélisation d'une architecture système se traduit alors par une profusion de modèles hétérogènes – des modèles édités avec des outils différents –, dans lesquels les informations – parfois redondantes ou contradictoires – sont éparpillées et dont le nombre et la diversité ne permettent plus une maîtrise de l'architecture. Par ailleurs, la ségrégation des modèles selon les expertises dégrade la vision holistique nécessaire à l'architecte système.

- **Problème 2 « Des diagrammes 2D inertes influencés par le génie logiciel »**

Au-delà du problème lié à la prolifération de modèles hétérogènes, on constate que les principaux outils d'architecture système sont maladroitement dérivés du génie logiciel. Les notations standardisées graphiques (SysML, Capella, BPMN, OPM, etc.) sont appréciées car, contrairement aux documents, elles facilitent l'organisation, la gestion et le partage des données d'architecture entre les parties prenantes réalisatrices. Cependant, ces modèles, lesquels sont qualifiés de pragmatiques, sont des diagrammes 2D inertes dont le but de faciliter la communication est remis en cause par, d'une part, la nécessité d'apprendre un énième langage et, d'autre part, leur faible niveau d'expressivité. Les tentatives qui proposent de décorer les modèles pragmatiques avec des informations formelles aboutissent à des résultats

⁴ Ici, un bloc logique est une solution de conception qui est suffisamment définie pour apprécier ses propriétés comportementales et structurales – e.g. moteur DC 116 tr/min 12 V c.c 41 W, diam 8 mm – mais qui n'est pas suffisamment détaillée pour être un composant sur étagère avec un identifiant unique que l'on peut directement acheter, fabriquer ou réutiliser – e.g. moteur DC Code commande RS 834-7641.



insatisfaisants. Les modèles, surchargés et inutilement complexes, ne rendent alors plus le service de communication auxquels ils étaient originellement destinés. Par ailleurs, avec leur faible niveau d'expressivité, ces diagrammes se limitent à la modélisation de propriétés abstraites. Ils sont donc, entre autres, inadaptés à la conception d'un squelette géométrique, à la définition de l'allocation d'espace, ou à la conception préliminaire du comportement énergétique. Il faut ainsi rompre les silos entre les moyens de modélisation de propriétés abstraites et de propriétés concrètes. C'est l'influence trop importante des acteurs du génie logiciel – chercheurs, éditeurs, consultants... – qui a fait que les outils qui supportent une démarche de MBSE souffrent d'un manque de « pensée système ». Par exemple, la notation standardisée SysML, laquelle est la plus populaire en architecture système, est un profil du langage de modélisation UML dédié au génie logiciel. Ainsi, en essayant de mettre d'accord deux communautés – les ingénieurs système vs. les ingénieurs logiciel – aux intérêts pour partie convergents et pour partie antagonistes, on observe une zone de non recouvrement. En effet, les outils SysML ont hérité des spécificités de l'architecture logicielle, lesquelles diffèrent de l'architecture système. On en veut pour preuve les nombreuses entreprises – e.g. Alstom, Valéo, Thalès – qui témoignent des écueils qu'elles ont rencontré pour, *in fine*, ne conserver qu'une partie de SysML, et ce, avec une personnalisation intensive des environnements de modélisation pour satisfaire aux pratiques des architectes. Malgré ces efforts pragmatiques, les outils d'architecture système restent inadaptés pour également d'autres raisons : opérations de modélisation trop laborieuses, langage de modélisation trop riche et trop complexe, trop de libertés de modélisation, mauvaise gestion des vues multiples sur un élément, trop de modèles et d'interdépendance, etc.

Etat de l'art

Le problème de la prolifération des modèles d'architecture a déjà été étudié dans le domaine du génie logiciel. Par exemple, le problème du passage à l'échelle des modèles UML, c'est-à-dire la capacité à obtenir une vue exhaustive d'une grosse architecture logicielle, a notamment encouragé le développement des environnements virtuels immersifs dédiés à la visualisation d'un système logiciel. Ce constat partagé par d'autres informaticiens a donné naissance à des mises en œuvre diverses et variées de la réalité virtuelle pour maîtriser le gros volume de données d'une architecture (structure hiérarchique des composants, relations entre les composants, métriques de qualité du code, diagrammes UML...) et d'un code source (fonctions, classes, packages...) d'un logiciel. Dans un environnement unique et immergé dans des métaphores de visualisation originales (archipels d'îles, système solaire, ville, etc.), les informaticiens peuvent alors réaliser, seul ou en collaboration, une revue d'avancement du développement logiciel. En ingénierie système, la prolifération des modèles est un problème plus récent puisqu'il a fallu attendre l'adoption massive d'une approche centrée sur les modèles. Pour faire face à cette problématique, le Systems Engineering Research Center et le MIT ont développé un axe de



recherche intitulé « Interactive Model-Centric Systems Engineering (IMCSE) », lequel combine quatre thématiques : la fouille de données massives, la visualisation analytique, les systèmes complexes, et l'ingénierie système basée sur les modèles. Ces travaux ont démontré l'intérêt de la visualisation analytique en 2D pour explorer un gros volume de données en ingénierie système, en particulier pour l'analyse exploratoire de concepts d'architecture. La visualisation analytique, en immersion 3D stéréoscopique, de données massives – « *Immersive analytics* » –, est aujourd'hui rendue possible grâce à l'utilisation des techniques de réalité virtuelle.

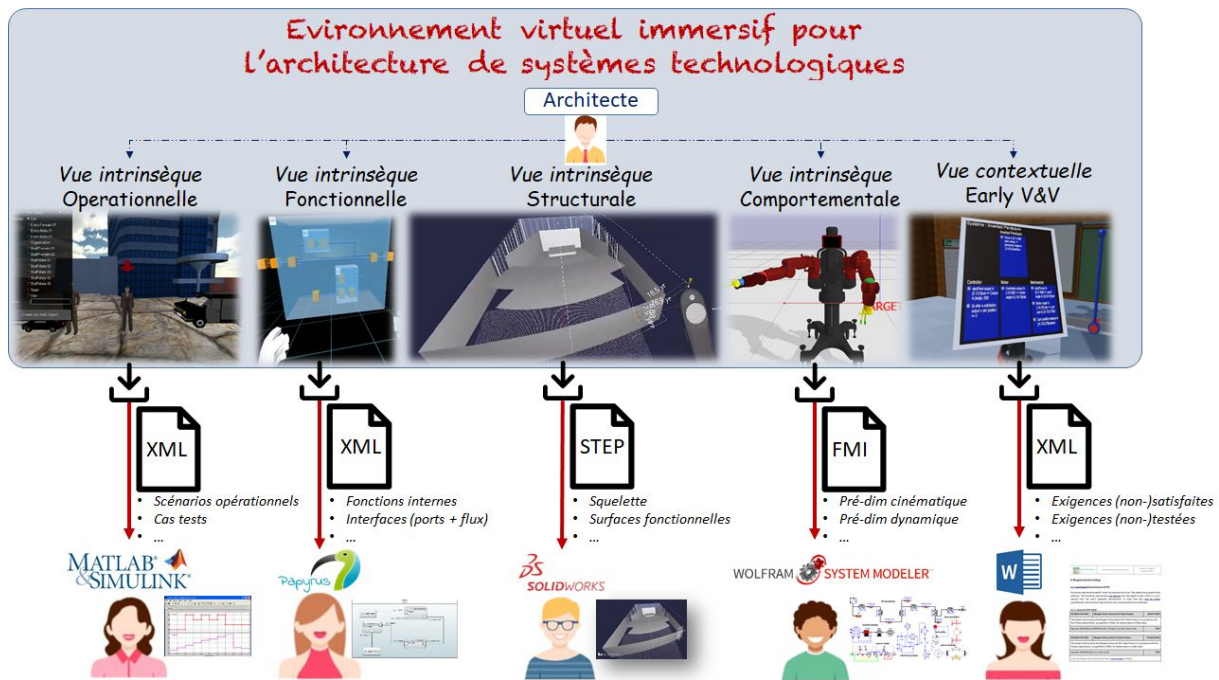
Comme alternative, de nombreux travaux ont montré que le manque d'expressivité et de formalisme des diagrammes d'architecture 2D peut être comblé par des détournements opportuns des outils de simulation. Dans un premier temps, des tentatives ont essayé d'intégrer des outils de conception détaillée au langage SysML pour permettre de modéliser des concepts mal adressés, voire absents de son périmètre, tels que des géométries ou des comportements dynamiques. Bien que faisable avec obstination, cette approche atteint vite ses limites. Cette démarche qui consiste à étendre un langage pour adresser de nouveaux objectifs de modélisation s'oppose à la démarche qui consiste à détourner des langages de ses objectifs initiaux. Par exemple, Simulink, un logiciel initialement dédié à la simulation de systèmes automatiques, est désormais l'un des outils les plus efficaces pour spécifier le comportement d'un système technologique. La spécification du comportement d'un système complexe n'est plus un ensemble ambigu d'énoncés textuels prescriptifs mais un ensemble de prédicats exécutables qui représentent graphiquement les exigences. Le langage de simulation de systèmes multi-physiques Modelica est également une excellente alternative pour supporter une spécification rigoureuse. L'exécution de ces modèles de spécifications apporte une valeur ajoutée – diminution des ambiguïtés, validation et vérification au plus tôt, génération automatique des tests, etc. – qui fait que diverses entreprises abandonnent certaines descriptions inertes, le diagramme d'exigence SysML par exemple, pour se tourner vers des descriptions plus formelles. Ce détournement pragmatique des logiciels qui permettent l'exécution des modèles a également été récemment illustré par la division avion du groupe Airbus pour la définition de l'architecture fonctionnelle d'un système. Les fonctions exécutables remplacent les graphes de flux fonctionnels tels que les diagrammes d'activité et IBD en SysML, FFBD, SADT, etc. dont l'interprétation subjective est toujours sujette à un certain relativisme. Le rôle de la simulation a également fait ses preuves en étant couplée à des scènes 3D immersives développées avec Unity3D pour supporter le développement des concepts opérationnels (ConOps). Cette dernière étude a montré que l'ajout de capacités de visualisation immersive à des simulations comportementales améliore le réalisme des scénarios opérationnels et facilite l'identification des anomalies. Pour aller plus loin dans l'émulation des propriétés comportementales et structurelles d'un système, nous avons récemment proposé un environnement immersif et interactif pour vérifier, au plus tôt, et dans des conditions réalistes, des solutions de conception. Dans un casque de réalité virtuelle, l'utilisateur stimule le système au moyen de dispositifs haptiques et s'assure, en temps réel, que le comportement multi-physique et la structure du système satisfont aux exigences. Cet environnement a l'avantage de fédérer trois vues (spécification, comportement dynamique, et géométrie) dans une seule scène. Ainsi, en opérant le système, l'utilisateur peut réaliser une validation factuelle – le système



spécifié correspond-il à mes attentes ? – avec les mêmes libertés qu’en condition opérationnelle, et ce, à moindre coût et très tôt dans le cycle de vie du produit. Enfin, grâce à la co-simulation de modèles de spécification et de modèles de conception, cette expérience virtuelle permet aux concepteurs de vérifier formellement que le système satisfait aux exigences techniques.

Hypothèses

L’ensemble de ces résultats et perspectives ouvertes encouragent à faire l’hypothèse que les techniques de visualisation et d’interaction, en particulier immersives pour leur large champ de vision, permettront de mieux maîtriser le gros volume de données qui matérialise un système complexe en cours de conception préliminaire. Ainsi, nous envisageons d’inventer un environnement de conception en réalité virtuelle qui, dans un seul espace, intègre les capacités de modélisation nécessaires à la définition immersive et interactive des points de vue d’une architecture système.



Cet environnement virtuel d’architecture système doit reposer sur les fondamentaux de la « pensée système » et doit donc être dénué des influences du génie logiciel. Il devra également intégrer des capacités de simulation pour réduire les ambiguïtés de certains fondamentaux tels que les fonctions et les exigences. Enfin, des visualisations et des interactions devront permettre une manipulation aisée, sans connaissances spécifiques, de ces fondamentaux tout en permettant une « bird’s-eye-view » qui unifie des informations originellement dispersées dans les modèles.



Méthode

Pour répondre au problème de la « **Prolifération des modèles** », l'architecture système a besoin de connaître une transformation analogue à celle qu'a connu l'ingénierie mécanique avec l'avènement de la maquette numérique 3D qui a remplacé les dessins de définition 2D. En effet, les diagrammes d'architecture système sont très proches des dessins de définition 2D de la mécanique d'autre fois : difficiles à tracer, difficiles à maintenir, difficiles à partager en mode collaboratif, difficiles à interpréter sans l'apprentissage d'un langage spécifique, etc. Or, en génie mécanique, le passage du 2D au 3D a facilité la définition et la gestion de la géométrie tout en permettant des revues collaboratives de la maquette numérique, lesquelles sont aujourd'hui réalisées en immersion dans un environnement unique dénommé CAVE. Par analogie, nous envisageons d'inventer un environnement de réalité virtuelle dans lequel l'architecte disposera des fonctions nécessaires à la conception des différents points de vue d'une architecture système qui, *in fine*, pourront être exportés et transférés aux expertises chargées de la conception détaillée. On se concentrera principalement sur la définition de la vue opérationnelle, de la vue fonctionnelle (interne), de la vue spécification, de la vue structurelle, et de la vue comportementale. Les vues opérationnelle, fonctionnelle et spécification de l'architecture nécessiteront l'invention d'un nouveau langage de modélisation graphique en 3D immersive. Ce langage ne nécessitera pas l'apprentissage de concepts autres que les fondamentaux de l'ingénierie système. La modélisation et la simulation des exigences s'appuiera sur des travaux antérieurs et la simulation des fonctions est déjà bien avancée dans la littérature. Ces vues dites abstraites seront exportables dans un format standardisé (e.g. ReqIF pour les exigences) ou dans un format populaire et ouvert (structure de données du plug-in SysML de Papyrus ou de Capella). La vue structurelle, elle, nécessitera une évaluation de la solution commerciale [MindDesk](#), en particulier son plug-in de modélisation géométrique Rhinoceros en réalité virtuelle. Si ce dernier répond à notre besoin, alors il pourra être intégré méthodologiquement dans notre environnement, sinon il faudra réfléchir à une solution alternative et éventuellement envisager de développer nos propres fonctionnalités. L'interface de CAO GesCAD, laquelle est basée sur la parole et les gestes, pourrait être notamment transposée et adaptée en réalité virtuelle. La vue structurelle sera exportée au format STEP pour être reprise en conception détaillée avec les modeleurs géométriques commerciaux. Enfin, la vue comportementale, elle, s'appuiera sur le [plug-in FMI](#) de Unity3D que nous mettons déjà en œuvre pour simuler le comportement dynamique de systèmes multi-physiques en immersion et/ou un moteur physique de réalité virtuelle tel que [PyBullet](#).

Pour répondre au problème des « **diagrammes 2D inertes influencés par le génie logiciel** », l'environnement virtuel d'architecture système reposera sur une ontologie holistique qui embrasse la « pensée système ». Cette ontologie sera un consensus des ontologies existantes et des normes dans lesquelles on se permettra de clarifier certains concepts ambigus comme celui de fonction dont les nuances sémantiques ont pourtant été rigoureusement analysées mais jamais intégrées. Ces fondamentaux seront manipulés au moyen de métaphores visuelles et interactives qui ne nécessiteront pas l'apprentissage d'un langage technique. Il ne sera donc pas question de transposer les représentations graphiques 2D de SysML en métaphores 3D de réalité virtuelle au risque de propager les



inconvénients largement discutés précédemment et de saborder l'intérêt de l'immersion. Des interactions avancées devront également remplacer les nombreux clics de souris aujourd'hui nécessaires à la modélisation des propriétés abstraites et à la navigation entre les boîtes de dialogue. Pour réduire les insupportables vas-et viens entre les différents diagrammes, les nouvelles métaphores visuelles permettront de disposer d'un contexte d'informations originellement dispersées dans différentes vues comme cela a été fait dans nos travaux antérieurs. Les interactions souris-clavier, elles, seront remplacées par des interactions plus naturelles. En effet, des gants de données serviront à manipuler les objets virtuels avec dextérité et le contenu des informations s'adaptera dynamiquement aux propriétés du champ de vision, par exemple, l'orientation et la distance par rapport à l'objet visualisé. Des interactions vocales seront également intégrées pour adapter le contenu des scènes et naviguer de façon transparente entre les vues d'une architecture et les liens de traçabilité verticaux (système, sous-systèmes, composants) ou horizontaux (partie prenante, fonction, exigence, structure, comportement...).

Plan de travail

1. Bibliographie

a. Travaux

- Etat de l'art des techniques de réalité virtuelle en architecture système
- Etat de l'art des outils d'architecture système et de réalité virtuelle
- Etat de l'art des critères d'ergonomie cognitive en réalité virtuelle

b. Méthodes et Techniques

- Méthode de revue systématique de la littérature
- Entretiens avec des architectes systèmes

c. Livrables

- Rapport qui synthétise les questions de recherche, l'état de l'art, les hypothèses, et une description fonctionnelle de la solution envisagée (**T0 + 6 mois**)

2. Proposition & Prototypage

a. Travaux

- Définition d'un modèle de données holistique, i.e. capable d'accueillir les divers points de vue, avec une structure de graphe



- Invention de métaphores visuelles et interactives pour modéliser chacun des points de vue d'une architecture, en particulier les vues abstraites (opérationnelle, fonctionnelle, et spécification)
- Invention de métaphores visuelles et interactives pour naviguer et/ou combiner les différents points de vue d'une architecture
- Intégration et/ou développement de capacités de modélisation géométrique en immersion (*Stage master*)
- Intégration et/ou développement de capacités de modélisation et simulation comportementale en immersion
- Développement d'un prototype logiciel de l'environnement virtuel immersif et interactif

b. Méthodes et Techniques

- Moteur graphique Unity3D et [plug-in FMI](#)
- Base de données orientée graphe [Neo4J](#)
- Modeleur géométrique en RV [MindDesk](#) (si valeur ajoutée démontrée en T1)
- Communication réseau

c. Livrables

- Rapport décrivant la démarche méthodologique, les avancées techniques, et l'implémentation de l'environnement virtuel immersif (**T0 + 24 mois**)
- Vidéos préliminaires qui illustrent les activités d'architecture dans l'environnement virtuel immersif (**T0 + 30 mois**)

3. Tests

a. Travaux

- Définition, évaluation et analyse des critères d'ergonomie de l'environnement d'architecture immersive.
- Expériences pour évaluer la valeur ajoutée de l'environnement immersif et comparer avec les approches traditionnelles dans l'industrie

b. Méthodes et Techniques

- Casque HMD
- Evaluation de tâches unitaires (e.g. conception d'un squelette géométrique) en immersion et comparaison avec une approche souris-clavier.



c. Livrables

- Rapport qui synthétise les résultats des expériences et compare avec les pratiques actuelles des architectes systèmes. **(T0 + 36 mois)**
- Vidéos finales qui illustrent les activités d'architecture dans l'environnement virtuel immersif **(T0 + 36 mois)**
- Manuscrit de thèse **(T0 + 36 mois)**

Profil du candidat

Curieux, créatif et autonome, vous êtes passionné par la recherche de nouvelles technologies qui conjuguent le numérique et l'ingénierie. Vous avez un profil d'ingénieur en ingénierie système ou réalité virtuelle, vous maîtrisez un langage de programmation et vous avez la vive volonté de développer de nouvelles compétences en sciences du numérique. Enfin, idéalement, le (la) candidat(e) possèdera des connaissances en CAO, modélisation et simulation de systèmes, MBSE (SysML, Capella, etc.), et/ou échange de données (STEP, FMI).

Compétences

- **MUST-HAVE**
 - Ingénierie système
 - Maîtrise d'un langage de programmation
- **NICE-TO-HAVE**
 - SysML ou Capella
 - CAO
 - C#
 - Unity 3D
 - XML

Cadre de travail

- **Durée : 36 mois**
- **Date : Dès que possible**
- **Contrat : CDD**
- **Lieu : plateforme Vision-R, Laboratoire G-SCOP, Grenoble INP Génie Industriel (38)**

Candidature





- Un CV + une lettre de motivation
- Les relevés de notes de votre cursus de master et/ou ingénieur
- Les avis et coordonnées de deux personnes référentes

Envoyez vos candidatures par courriel à:

- M. Romain PINQUIE (romain.pinquier@grenoble-inp.fr),

Bibliographie

- A. S. B and M. Misiak, “Visualizing software architectures in virtual reality with an island metaphor,” vol. 1, pp. 168–182, 2018.
- R. Oberhauser and C. Lecon, “Immersed in Software Structures : A Virtual Reality Approach,” no. c, pp. 181–186, 2017.
- V. Averbukh *et al.*, “Metaphors for software visualization systems based on virtual reality,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11613 LNCS, pp. 60–70, 2019.
- A. Schreiber, L. Nafeie, A. Baranowski, P. Seipel, and M. Misiak, “Visualization of Software Architectures in Virtual Reality and Augmented Reality,” 2019.
- M. A. Bone, M. R. Blackburn, D. H. Rhodes, D. N. Cohen, and J. A. Guerrero, “Transforming systems engineering through digital engineering,” 2019.
- L. Blessing, “GesCAD : An Intuitive Interface for Conceptual Architectural Design,” pp. 402–406, 2017.
- A. B. Rauzy and C. Haskins, “Foundations for model-based systems engineering and model-based safety assessment,” *Syst. Eng.*, vol. 22, no. 2, pp. 146–155, 2019.
- P. Korfiatis, T. Zigh, and M. Blackburn, “Graphical CONOPS Development to Enhance Model Based Systems Engineering Graphical CONOPS Development to Enhance Model Based Systems Engineering,” no. January, 2016.