

Chapitre 1

Conclusion

Cette thèse s’inscrit dans le contexte la visualisation et de la manipulation d’objets 3D dans un [Environnement Virtuel Collaboratif \(EVC\)](#) sur le web. Ce sujet intègre de nombreux verrous concernant différents domaines que la collaboration a tendance à réunir : gestion de la cohérence dans un environnement collaboratif massif en [pair à pair \(P2P\)](#), architecture de communication efficace pour le passage à l’échelle et la résilience, modèle de données dédiée au métier de la modélisation 3D collaborative.

1.1 Discussion

Pour répondre aux différentes questions de recherche posées en introduction de ce manuscrit, plusieurs contributions ont été proposées, implantées et soumises à des expérimentations à travers [Desprat *et al.*, 2015b, Desprat *et al.*, 2015a, Desprat *et al.*, 2016, Desprat *et al.*, 2017].

La **QR 1** concerne l’architecture réseau qu’un [Environnement Virtuel Collaboratif 3D \(EVC 3D\)](#) doit proposer pour pouvoir s’adapter à une gestion efficace, robuste et temps réel des données dans un environnement web. En effet, pour pouvoir collaborer les utilisateurs ont besoin d’une architecture de communication qui les relie de manière transparente intégrant un modèle de cohérence des données.

L’architecture de communication au sein de l’[EVC 3D](#) a été l’objet de deux contributions successives, la seconde s’inspirant des verrous issus de la première. L’architecture hybride à maillage complet se compose d’éléments simples. Seuls les navigateurs des utilisateurs participent au réseau [P2P](#). Les données correspondent à l’état des objets manipulés. Les modifications sont transférées sous la forme d’un différentiel d’état. Bien que la politique de connectivité ne permettent pas un passage à l’échelle efficace, la cohérence des données sur le réseau est garantie de manière forte et pessimiste respectivement par le protocole de transport est configuré comme fiable et ordonnés, et par le mécanisme de verrouillage afin d’éviter l’édition concurrente d’objets 3D.

Partant de ces observations, la seconde contribution liées à l’architecture de communication repose sur le paradigme événementiel. Une politique de connectivité plus lâche permet aux pairs de devenir relais des informations. La partie serveur est « descendue » au niveau de la couche [P2P](#) pour permettre un accès réparti des pairs producteurs de données (navigateurs) à la persistance des données via des pairs relais. L’ensemble des pairs partage un journal d’événements qui est maintenu à jour et cohérent par à deux niveaux de vérification : au niveau du pair, avec un système de version lié à l’ordre causal puis au niveau de la grappe, avec un système de vote lors de la détection de conflit pour accepter ou rejeter collectivement un événement. La cohérence repose sur une stratégie

optimiste avec des contraintes sur les règles métiers dont la permissivité peut être configurée. L'expérimentation 2, reposant sur cette seconde architecture, a montré que ce modèle optimiste est moins frustrant pour les utilisateurs. La couche **P2P** est enrichie de nouveaux pairs dédiés à la transmission des données qui permettent une dissémination plus stable et rapide. Selon le nombre d'utilisateur, la montée en charge peut plus facilement être absorbée.

Dans la **QR 2** sont évoqués les verrous souvent rencontrés dans la modélisation 3D et particulièrement lorsque le volet collaboratif s'invite. Les fonctionnalités d'historique et de défaire / refaire une action sont fondamentales en modélisation 3D. Pour assurer un suivi de la création sans perdre l'intention de l'utilisateur de son action à sa visualisation, la modélisation sur le paradigme événementiel est adapté par ses capacités à permettre une forte traçabilité des changements qui se produisent au sein d'un système.

La modélisation logicielle passe, dans un premier temps, par le suivi des bonnes pratiques énoncées par le **Domain Driven Design (DDD)** pour cerner les différents agrégats qui composent le domaine de la modélisation 3D collaborative. Puis, dans un second temps, deux principes sont combinés dans l'application cliente : (i) la séparation de la partie écriture de la partie lecture (**Command Query Responsibility Segregation (CQRS)**) ; (ii) la création d'événements auto-descriptifs du changement qui vient de se produire (**Event Sourcing (ES)**). Grâce à (i) impose plusieurs niveaux de vérification des données : de la saisie utilisateur à la validation des règles métier. Ainsi, le système, qui ne manipule que des événements immuables est garanti d'avoir des données valides. Côté lecture, les vues matérialisées de l'interface sont issues de projections faites à partir du flux d'événements traités par le système. Les vues sont spécifiques à l'activité, ce qui permet de présenter une interface orientée tâche plus proche des attentes métiers de l'utilisateur. La projection découple les données de la base de données des données servant une ou plusieurs vues. En effet, son rôle d'intermédiaire est fournir les événements d'intérêts sous une forme dénormalisée pour les vues (mise en cache). Le point (ii) reprend principalement les avantages de l'**ES**, i.e. enregistrer la séquence d'événements qui a amenée à l'état courant de l'application. La fonctionnalité d'historique est implicitement présente dans ce patron, même ce qui pourrait être considéré comme une « annulation » par l'utilisateur est conservé sous la forme d'un événement de compensation. L'immuabilité des événements capte toutes les manipulations métier de l'utilisateur. Par conséquent, la reconstruction des agrégats du domaine permet d'avoir un état cohérent garanti par l'aspect fonctionnel de ce patron. C'est particulièrement important pour (i) améliorer la sensibilisation des collaborateurs à l'historique des objets manipulés ; (ii) faciliter le développement d'outils de surveillance

pour observer, analyser et améliorer l'application ; (iii) et organiser des audits (exemple : respect d'une procédure).

Une partie de la contribution décrivant le modèle événementiel révèle une partie de la réponse à la **QR 3**. Le fait de déporter le patron **CQRS** en totalité sur le client évite que le client soit dépendant du serveur pour valider le contenu créé. L'utilisateur a accès localement aux événements. Il peut ainsi générer de nouveaux événements sur la base de ceux qu'il possède. Les projections fonctionnent également sur ces événements stockés. Le cycle des données est donc complètement géré en interne et libère l'utilisateur d'une forte liaison avec la base de données centrale. La synchronisation des données produites hors ligne est effectuée à la reconnexion du pair au réseau. En effectuant la différence entre les agrégats locaux et distants, les répliques sont capables de se synchroniser et de mettre à jour leur Event Store.

L'autre partie de la réponse à la **QR 3** est apportée par la contribution portant sur la présentation d'une architecture de communication hybride. Le système appuie l'autonomie des utilisateurs en leur permettant de s'échanger directement des informations. Le serveur n'est plus une source d'engorgement pour l'accès à la base de données mais un ensemble de pairs reliés à une source de vérité de l'application. Les pairs liés à la base de données sont répartis dans le réseau de manière à être plus disponible et proche pour les pairs sur lesquels les utilisateurs travaillent. En cas d'arrêt temporaire de quelques uns de ces pairs, l'application peut continuer de fonctionner en collaboration sur le réseau **P2P** restant. La mise en place d'un environnement collaboratif nécessite également un mécanisme de gestion de conflits. Cette thèse présente successivement deux implantations de ce mécanisme, l'un pessimiste (verrou), l'autre optimiste (*Last Write Wins*). Considérant que le premier peut générer des effets de bord considérables dans l'environnement et amener à la frustration des utilisateurs, le second, plus permissif et accompagné de mécanismes de sensibilisation au groupe (boîtes englobantes) et de sensibilisation à l'interaction (sélection fantôme), renie ces inconvénients et laisse plus de liberté à l'utilisateur dans ses choix de sélection et plus largement d'interaction.

La **QR 4** s'intéresse à l'implémentation d'un **EVC 3D** sur le web. La question s'oriente principalement sur les choix qu'impose ce type d'environnement très contraint tant au niveau du métier de la modélisation 3D collaborative qu'au niveau des technologies et des techniques qui s'y rapportent. Les prototypes réalisés sont cross-plateformes sont réalisés à partir de technologies web issues de standards (définis par **World Wide Web Consortium (W3C)**, **Internet Engineering Task Force (IETF)** et **ECMA** pour **JavaScript (JS)**) largement supportés comme WebGL pour le rendu de l'environnement 3D et parfois encore en évolution comme WebRTC DataChannel pour la transmission de données en **P2P**.

Dans le prototype reposant sur le paradigme événementiel, l'implantation d'une [Interface Utilisateur \(IU\)](#) orientée tâches rend possible un accompagnement de l'utilisateur dans chaque actions en lui procurant une interface adaptée à la tâche en cours de réalisation. Le pendant à ce découpage de l'interface par tâches est un passage à l'échelle facilité par la production de vues matérialisées issues de projections d'événement qui peuvent être mises en cache. Une projection pouvant servir plusieurs vues, il est possible de décliner plusieurs visualisations des mêmes données. La projection liée à l'environnement 3D peut par exemple proposer plusieurs déclinaisons des mêmes données pour chaque vue qui en dépende. Cela peut se traduire par un rendu effectué par des bibliothèques 3D différentes (Three.JS, BabylonJS) ou un rendu adapté selon un profil utilisateur. Les projections sont également un facteur facilitant l'évolution de l'application simplifiant le processus d'agrégation et de filtrage des événements.

La réalisation de plusieurs prototypes fonctionnant sur différents paradigme pour les expérimentation ont permis de montrer les avantages et inconvénients de ceux-ci. Passer d'un paradigme orienté état à un paradigme événementiel a apporté plusieurs bénéfices liés au métier, notamment en ce qui concerne la conservation de l'intention de l'utilisateur au moment de la réalisation d'un action. En effet, les événements sont auto-descriptif et porte également le lien de causalité qui reflète cette intention. Le réseau, responsable de la dissémination de ces informations ne porte pas la responsabilité de la conservation de cette propriété, c'est à la couche applicative (aux projections) de s'en charger. Cela permet d'alléger les échanges avec des connexions [P2P](#) ayant une configuration non fiable et non ordonnée. Le protocole d'échange proposé repose sur une politique push / pull pour s'assurer de la synchronisation des différents pairs. En cas de conflits, les participants sont notifiés par le gestionnaire de conflit, ce qui leur permet de relancer la synchronisation sur une base commune en cas de problème insolvable. Le ré-ordonnancement passe par le mécanisme de visualisation flexible. A la réception des événements les événements sont stockés dans les projections. Seuls les événements en séquence (pas d'événements manquants, de « trou ») sont transmis à la vue ce qui permet de stocker en avance les futurs événements et déclenche une nouvelle demande des événements manquants. Cela permet de proposer une visualisation rapidement même si tous les événements ne sont pas reçu et de donner à l'utilisateur un retour visuel sur l'interface d'une scène se construisant par exemple, lui permettant de commencer à percevoir, s'immerger voire travailler sur la scène.

Enfin, l'expérimentation 2 a directement bénéficié de l'implantation d'un modèle orienté événements pour le prototype utilisé : l'intégration du métier a permis l'observation mi-

nutieuse du travail réalisé par les utilisateurs au sein de l'environnement. C'est la base du travail d'analyse des interactions de cette expérimentation.

Dans la **QR 5**, la volonté de trouver des métriques adaptées à l'évaluation d'un **EVC 3D** est exprimée tant concernant le réseau que la collaboration. La question se pose à la fois en termes quantitatifs et qualitatifs. Les expérimentations détaillées dans cette thèse présente plusieurs facteurs d'évaluation qualitatifs. En effet, les deux expérimentations portent sur une étude des utilisateurs dans le cas d'application de l'assemblage de modèles 3D de manière collaborative pour la visualisation et la manipulation d'objets dans un environnement web. La mise en place des protocoles d'expérimentation a permis de soumettre les utilisateurs à différentes épreuves d'assemblage : contraint par un objectif déterminé, création libre. L'évaluation a porté sur les observations de l'expérimentateur et des réponses données par les participants au questionnaire concernant différents aspects : le ressenti du participant par rapport à la latence dans l'environnement, la gestion de la cohérence, la qualité de la récupération des données en cas de déconnexion mais également sur l'apprentissage et la facilité d'utilisation de l'application. La qualité de la collaboration est également soumise au questionnaire pour savoir si la coopération améliorer l'efficacité et / ou la vitesse de réalisation de la tâche. Les contraintes liées à ce type d'expérimentations sont nombreuses : nombre de participants, taille des données 3D, gestion de la cohérence... Les métriques quantitatives n'ont pu être abordée au cours de cette thèse mais plusieurs perspectives sont évoquées dans la Section ci-après.

ref section

1.2 Perspectives

Pour obtenir des métriques quantitatives fiables, la simulation de comportements collaboratif sous forme de scénario enregistrés (grâce à l'**ES**) aura pu permettre d'obtenir des résultats sur des métriques quantitatives telles que la quantité de données transmises, une variation plus grande de la topologie du réseau. Lorsqu'on remarque que certains sont capables de réunir jusqu'à 100 personnes simultanément pour réaliser montrer le passage à l'échelle de leur système [Hu et Chen, 2017], il semble préférable pour des questions de temps, logistiques, de coûts d'évoquer les possibilités de virtualisation offertes par les plateformes réceptives au standard WebRTC.

Annexe A

Ressources pour l'implantation et les expérimentations

A.1 Description des événements par agrégat dans 3DEvent

Tableau A.1 – 3DEvent : résumé des événements par agrégats

événement	Nommage	Description
Agrégat Scène		
Scène créée	sceneCreated	Une scène a été créée
Scène supprimée	sceneRemoved	Une scène a été supprimée
Scène renommée	sceneRenamed	Une scène a été renommée
Agrégat Maillage		
Maillage ajouté	meshAdded	Un maillage a été ajouté dans la Scène à partir d'une géométrie de la bibliothèque
Maillage déposé	meshDropped	Un maillage a été déposé dans l'environnement 3D de la Scène à partir d'une géométrie de la bibliothèque
Maillage supprimé	meshRemoved	Un maillage a été supprimé de la Scène
Maillage traduit	meshTranslated	Un maillage a subi une translation dans la Scène
Maillage pivoté	meshRotated	Un maillage a subi une rotation dans la Scène
Maillage mis à l'échelle	meshScaled	Un maillage a subi une homothétie dans la Scène
Agrégat Géométrie		
Géométrie importée dans la bibliothèque	geometryImported	Une géométrie est créée à partir d'un fichier importé par un utilisateur et ajoutée à la bibliothèque de la Scène
Agrégat Utilisateur		
Utilisateur créé	userCreated	Un utilisateur a été créé dans l'application
Scène rejointe par utilisateur	userJoinedScene	Un utilisateur a rejoint une scène
Scène quittée par utilisateur	userLeftScene	Un utilisateur a quitté une scène
Nom modifié	usernameChanged	Un utilisateur a modifié son nom
Couleur modifiée	colorChanged	Un utilisateur a modifié son code couleur

A.2 Messages réseaux pour la synchronisation des Event Stores

Tableau A.2 – Type de messages lors de la synchronisation

Message	Description
STREAM_SYNC_ASK	Demande de synchronisation d'un <i>stream</i>
CHUNK	Réception d'une donnée <i>chunk</i>)
READY_ASK	Prêt pour la démarrer la demande de données de sync.
READY	Prêt pour démarrer la réception de données de sync.
ALL_EVENTS_SYNC_ASK	Demande de toutes les données typées événement
EVENTS_SYNC	Réception de données (en cours de synchronisation)
META_DATA_ASK	Demande de métadonnées
META_DATA	Réception de métadonnées
SYNC	Réception de données (en cours de synchronisation)
EVENT	Réception d'une donnée typée événement
END_SYNC	Fin de la synchronisation

parler des
tableaux

Tableau A.3 – Statut du nœud

Message	Description
ERROR	En erreur (désynchronisation)
READY	Prêt à recevoir des messages
META_DATA_ASK	En demande de métadonnées
META_DATA_RECEIVE	En réception de métadonnées
CLOSE	Déconnecté (connexion fermée)
RECEIVE_SYNC	En réception de données à synchroniser
CONNECTED	Connecté (connexion ouverte)
INIT	Initialisation
OK	Connecté et synchronisé
SEND_SYNC	En demande de synchronisation
END_SYNC	Synchronisation terminée

A.3 Experimentation 1 materials

A.3.1 Description des modèles utilisés

Tableau A.4 – Model descriptions for the experiments

Experiment	objects	size	users
Wind turbine	6	1.0 MB	2
Pick up	8	1.3 MB	4
Castle from <i>server</i>	35	1.3 MB	4
Castle from <i>peer</i>	35	1.3 MB	4

A.3.2 Questionnaire pour l'enquête utilisateur

Did you reach the goal or are you satisfied with the result ?

- Type of external communication channel used ?
- Collaboration satisfaction ?
- User interface quality : tools ? Object manipulation ? Global ?
 - Practice : difficulty or frustration ?
 - Collaborative rendering (latency, consistency, recovery) ?

A.4 Expérimentation 2

A.4.1 Description des modèles utilisés

Tableau A.5 – Modèles utilisés durant l'expérimentation

Modèle	Nombre de parties	Nombre de triangles	Taille totale
Rotor	10	62k	4Mo
Camera box	12	67k	5Mo
Car	16	170k	8Mo
Living room	16	200k	9Mo

A.4.2 Questionnaire pour l'enquête utilisateur

Seven-points scale questions from 1 (don't agree) to 7 (agree) :

- Did you enjoy this ?

- After trial, I was confident to do object manipulation in 3D virtual environment ?
- App learning : I am satisfied with the ease of use of the application
- App learning : It was easy to learn the tool
- Solo : I completed the task quickly
- Solo : I completed the task efficiently
- Collaboration : I contributed to complete the task quickly
- Collaboration : I contributed to complete the task efficiently
- In general, I am satisfied with the collaboration experience
- In general, The collaboration quality was pleasant in terms of LATENCY
- In general, The collaboration quality was pleasant in terms of CONSISTENCY
- In general, The collaboration quality was pleasant in terms of RECOVERY

Other questions :

- General efficiency is improved with the number of users ?
- General speed is improved with the number of users ?
- I would qualify this application : Non interactive/Interactive/Near real-time/Real-time
- Negative/positive/comments feedbacks

