



## **Vue Radar : déporter le gestionnaire de fenêtres sur une tablette**

Emmanuel Dubois, Ilyes Reguig, Gary Perelman, Philippe Limousin,  
Louis-Pierre Bergé

### **► To cite this version:**

Emmanuel Dubois, Ilyes Reguig, Gary Perelman, Philippe Limousin, Louis-Pierre Bergé. Vue Radar : déporter le gestionnaire de fenêtres sur une tablette. 2023. hal-04006865

**HAL Id: hal-04006865**

**<https://hal.science/hal-04006865v1>**

Preprint submitted on 27 Feb 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## **Vue Radar : déporter le gestionnaire de fenêtres sur une tablette**

### **RadarView : Unloading the Desktop Window Manager onto a Tablet**

EMMANUEL DUBOIS

Université de Toulouse, IRIT, Toulouse, France

ILYES REGUIG

CEA-Tech Occitanie, Toulouse, France

GARY PERELMAN

Université de Toulouse, IRIT, Toulouse, France

PHILIPPE LIMOUSIN

CEA-Tech Occitanie, Toulouse, France

LOUIS-PIERRE BERGE

CEA-Tech Occitanie, Toulouse, France

In an activity that involves viewing and editing a large amount of data, organising the information in the form of a dashboard is relatively common. The user must then be able to move efficiently from one area of information to another to view or edit the data in the dashboard, which sometimes exceeds even the size of the screen. Multiple window managers have been studied but their application has never been explored for information areas within a single window. The "Radar View" we propose is a miniature representation of the different information areas accessible. We study the use of the radar view according to several considerations including the location of the radar view and the layout of its content. Through a first experimental study, it appears that the results are in line with the hypothesis according to which the use of a radar view displayed on a tablet glued to the keyboard allows the user to complete the task in less time than if he/she were to use a mouse and conventional window navigation, without increasing the number of errors made and without obscuring the data being consulted and edited. These results will need to be confirmed in a further study.

**CCS CONCEPTS •Human-centered computing~Human computer interaction (HCI)~Interaction techniques**

**Additional Keywords and Phrases:** window manager, window switching, interaction technique, tablet

Au cours d'une activité impliquant la consultation et l'édition d'une grande quantité de données, organiser l'information sous la forme d'un tableau de bord est relativement commun. L'utilisateur doit alors pouvoir passer efficacement d'une zone d'information à une autre pour visualiser ou éditer les données présentes dans le tableau de bord, qui parfois dépasse même la taille de l'écran. De multiples gestionnaires de fenêtres ont été étudiés mais leur application n'a jamais été explorée pour des zones

d'information à l'intérieur d'une même fenêtre. La « Vue Radar » que nous proposons regroupe une représentation miniature des différentes zones d'informations accessibles. Nous étudions l'usage de la vue radar selon plusieurs considérations dont notamment le lieu d'affichage de la vue radar et la disposition de son contenu. Au travers d'une première étude expérimentale, il apparaît que les résultats sont compatibles avec l'hypothèse selon laquelle l'utilisation d'une vue radar affichée sur une tablette collée contre le clavier permet à l'utilisateur de réaliser la tâche en moins de temps que s'il utilisait une souris et la navigation classique dans une fenêtre, sans accroître le nombre d'erreur commise et sans occulter les données en cours de consultation et d'édition. Ces résultats nécessiteront d'être confirmés dans une étude plus approfondie.

**Mots-clés additionnels :** gestionnaire de fenêtres, changement de fenêtre, technique d'interaction, tablette

**Reference:**

IHM '23, April 3-6, Troyes, France

This is the author's draft version of the work. It is posted here for your personal use. Not for distribution. The definitive version of record was published in IHM '23: Proceedings of the 34th Conference on l'Interaction Humain-Machine.

ISBN 978-1-4503-9825-1

<https://doi.org/10.1145/3583961.3583962>

## 1 INTRODUCTION

Dans le contexte de la manipulation d'une grande quantité de données, et plus spécifiquement lors de la gestion d'une crise, la quantité d'information utile peut rapidement nécessiter un nombre conséquent de fenêtres et induit nécessairement de fréquents changements de focus pour faire évoluer les données ou pour contribuer à une prise de décision. Par exemple, pour gérer un dysfonctionnement dans une usine, il est nécessaire d'accéder au plan de l'usine, à la liste du personnel, aux données de production temps réel, aux alertes liées à des avaries, etc., chacune constituant une fenêtre distincte. De nombreux et fréquents aller-retour entre ces différentes catégories de données sont donc requis pour suivre l'évolution d'une crise et prendre note des informations, décisions ou événements nouveaux. Il est donc essentiel d'optimiser les interactions requises pour exploiter ces données et ainsi permettre un gain d'efficacité lors de la navigation / édition d'une interface de type tableau de bord en situation de suivi (gestion de crise, suivi de consommation, surveillance de trafic, ...). Pour optimiser ces situations d'interaction, plusieurs approches ont été envisagées. Le raccourci ALT+TAB de Microsoft Windows a justement pour but de faciliter le passage d'une fenêtre de travail à une autre en réduisant le temps de transition entre ces deux fenêtres. Ce raccourci a donné lieu à de nombreux travaux de recherche visant à optimiser encore davantage cette action de changement de fenêtre. Le Virtual Desktop Manager [12] en est probablement un des premiers, suivi par plusieurs autres solutions applicable en environnements de bureaux [27][28][23][4] et plus récemment sur dispositifs mobiles [18][11][5]. Pour éviter de perdre de l'espace d'affichage, ces différentes solutions de changement de fenêtres sont activées à la demande de l'utilisateur par le biais d'un mécanisme qu'il est nécessaire de connaître (raccourcis clavier, gestes spécifiques, etc.).

Néanmoins, l'utilisation répétée du mécanisme de changement de fenêtre peut s'avérer coûteux en temps et source d'erreur [28]. De plus, les tableaux de bords numériques sont communément utilisés dans de telles situations car ils fournissent une représentation concise et structurée des informations. Ces tableaux de bord résument les informations principales à monitorer et éditer ou annoter et sont constitués de plusieurs blocs d'information contigus. Rassemblés dans une même fenêtre, il est fréquent que la taille de la fenêtre dépasse celle de l'écran, induisant alors la nécessité de scroller pour accéder à ces différents blocs. Face à de telles représentations, l'utilisateur doit alors fréquemment passer non plus d'une fenêtre à une autre mais d'une zone d'information à une

autre, dans une même fenêtre, par exemple, en explorant la liste des victimes, des dernières actions de reconnaissance prise dans l'usine ou encore des communiqués de presse diffusés. Or, pour passer d'une zone à l'autre dans une même fenêtre, à notre connaissance, les gestionnaires de fenêtres présentés dans la littérature n'ont jamais été appliqués, à commencer par le raccourci ALT+TAB. Certes le navigateur Chrome offre un raccourci clavier pour passer séquentiellement et de manière circulaire d'un onglet à l'autre : mais la connaissance de ce raccourci (CTRL+TAB) est peu répandue et n'a pas fait l'objet d'étude utilisateur. Dans un formulaire, la touche TAB permet elle aussi de passer séquentiellement d'un champ au suivant : elle ne permet toutefois pas le passage d'une zone du formulaire regroupant plusieurs champs à une autre.

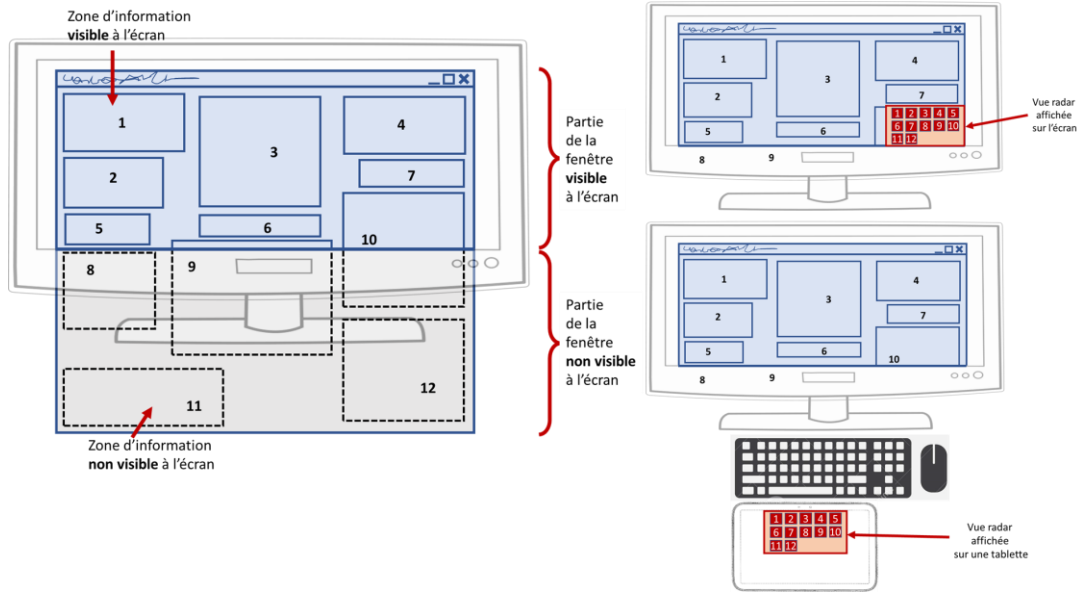


Figure 1 : Contexte d'utilisation de la « Vue Radar » (gauche) et mise en œuvre sur l'écran (haut – droite) ou sur un écran déporté (bas – droite)

Dans ces travaux, menés dans le cadre d'un projet d'informatisation de la gestion d'une crise, nous nous inspirons des gestionnaires de fenêtres proposés dans la littérature et nous proposons une technique d'interaction contribuant à faciliter et rendre plus efficiente la tâche des équipiers de crise face à un tableau de bord, i.e. naviguer entre différentes zones d'information au sein d'une même fenêtre pour les consulter et les éditer. Nous présentons donc le widget « Vue Radar », un gestionnaire de fenêtres adapté au passage d'une zone d'information à une autre à l'intérieur d'une même fenêtre affichée à l'écran. La vue radar présente une liste des zones d'informations accessibles et permet le passage rapide d'une zone à une autre, qu'elle soit visible ou non à l'écran (cf. Figure 1). Tout d'abord nous avons conçu différentes solutions basées sur quatre considérations principales : le lieu d'affichage (écran, tablette), la taille du widget, la disposition des raccourcis dans la vue radar (grille, spatiale) et l'orientation de la vue radar (horizontale, verticale). Puis nous avons comparé ces solutions avec une référence (baseline) sans ce widget et basée uniquement sur l'utilisation de la souris. Les résultats de cette première étude mettent en évidence l'intérêt prometteur de ce widget, en particulier sa version sur tablette et nous envisageons des améliorations et perspectives à ce premier prototype.

## 2 TRAVAUX EXISTANTS

### 2.1 Gestion des fenêtres en environnement de bureau

Dès les années 1986, il était admis que la taille de l'espace d'affichage des écrans était trop petite compte tenu du nombre de fenêtres manipulées simultanément par un utilisateur [12]. La première solution a alors consisté en l'utilisation de bureaux virtuels dans lesquels l'utilisateur pouvait regrouper manuellement des fenêtres. En organisant lui-même son espace de travail, le temps de passage d'une fenêtre à l'autre pouvait alors être amélioré. Ces premiers travaux ont mis en lumière deux aspects importants de la gestion des fenêtres : le groupement des fenêtres et le passage de l'une à l'autre.

La constitution des groupements contribue à la gestion des fenêtres en rapprochant entre elles les fenêtres que l'utilisateur risque d'utiliser simultanément. Au-delà de la constitution manuelle de tels groupes, plusieurs approches ont été envisagées pour constituer les groupes de fenêtres ayant un rapport entre elles. Ainsi Push-and-Pull [28] se base sur l'existence de chevauchement entre des fenêtres pour les affecter à un même groupe. TaskPosé [4] définit une mesure de « niveau de relation » basés sur le fait que des fenêtres ont fréquemment été ouvertes l'une après l'autre par exemple. WindowScape [26] et Scalable Fabric [23] regroupent quant à eux les fenêtres en fonction de l'identification de tâches de l'utilisateur. Enfin UMEA [14] et SWISH [21] adoptent des approches basées sur le contenu des titres des fenêtres et de l'utilisation temporelle des fenêtres pour former automatiquement des groupements. En termes de rendu de ces groupements, dans [25], une organisation spatiale stable des icônes représentant les fenêtres semble apporter de meilleurs résultats qu'une organisation basée sur la fréquence ou même la récence d'utilisation.

Concernant le passage d'une fenêtre à l'autre, plusieurs mécanismes ont été testés. WindowShopping [27] a notamment comparé trois techniques distinctes issues de la littérature : *Cards*, un empilement de fenêtres à l'image de la gestion proposée sur dispositifs mobiles, *Exposé*, une disposition des fenêtres en fonction de leur taille et position d'origine comme dans l'environnement Mac OS X, et *Mosaic*, une organisation en grille des fenêtres qui tient aussi compte de la taille initiale de la fenêtre. Cette dernière solution semble être plus rapide et induire moins d'erreur. L'apport GroupBar [24] est une autre alternative, offrant la possibilité de grouper manuellement des icônes représentatives des fenêtres dans des onglets de la barre des tâches de Windows. Scalable Fabric [23] s'appuie sur le principe de Focus+Contexte pour iconifier les fenêtres et créer des groupes en périphérie de l'écran d'affichage. Plus récemment, des travaux ont proposés plusieurs méthodes pour rendre perceptible à l'utilisateur des liens vers des fenêtres, applications ou zones de l'écran contenant des informations complémentaires ou en lien avec une données spécifiques [8] ; Initialement pensé comme un système de prévisualisation intégré permettant de jongler entre plusieurs fenêtres, cette approche pourrait toutefois être transformée en un gestionnaire de fenêtres.

### 2.2 Gestion fenêtres sur dispositif mobile

Avec l'accroissement des capacités de calcul des dispositifs mobiles, l'usage simultané de plusieurs applications est aussi fréquent sur ces dispositifs. D'ailleurs depuis 2016, Android permet l'affichage contigus de deux fenêtres sur une tablette ou un smartphone [1]. Pour dépasser l'utilisation de deux fenêtres et faciliter le passage de l'une à l'autre, DoThisHere [29] exploite la parole pour faire référence à une application ou un contenu hors écran. La transparence est aussi une solution envisagée pour faciliter l'utilisation combinée de multiples fenêtres : c'est

notamment le cas pour permettre l'utilisation de clavier en complément d'un éditeur de texte [16] [19] par exemple. Porous Interfaces [11] combine transparence et modalité tactile pour naviguer et parcourir les applications ouvertes : la transparence permet de voir les différentes applications tandis que la modalité tactile permet la sélection des fenêtres en fonction de la durée du toucher ou du doigt utilisé par exemple. Peek-a-view [5] exploite la coque du smartphone pour gérer le changement de fenêtres : selon l'ouverture de la coque, la fenêtre associée est affichée en plus ou moins grand à l'écran.

## 2.3 Conclusion

Les travaux portant sur la gestion de fenêtres ont donc exploré l'usage de différentes modalités et différents mécanismes de groupement : ils peuvent être manuels ou automatiques et ce, selon différents critères explicitement exprimés par l'utilisateur ou implicitement appris par le système. Dans tous les cas, ces solutions de gestion de fenêtres nécessitent une activation explicite, qu'il faut connaître, et occultent une partie de l'écran lors de leur invocation. De plus, ils ont tous été conçus et évalués dans le contexte de multiples fenêtres : aucun ne s'est intéressé à leur utilisation pour la navigation entre des zones d'informations à l'intérieur d'une même fenêtre.

Or des travaux menés dans le contexte de la réalité augmentée [22], bien qu'ils dépassent le contexte de nos travaux, ont établi que lors de l'utilisation combinée d'un casque semi-transparent et d'un dispositif mobile avec plusieurs fenêtres, les utilisateurs préfèrent se servir du dispositif mobile comme un support pour la gestion des fenêtres dans le casque. Dans le même esprit, Keytch [15] s'est intéressé à l'usage d'un dispositif mobile comme support à la gestion des menus en contexte de bureau. Ces deux travaux révèlent l'intérêt d'un dispositif mobile additionnel pour déporter une partie de la gestion d'un environnement fenêtré.

## 3 LE WIDGET VUE RADAR POUR LA GESTION DES FENETRES

### 3.1 Composition et fonctionnement de la vue radar

Le widget vue radar est composé de deux parties complémentaires : le support et les icônes. Le support est une zone affichée à l'écran dans laquelle sont regroupées des icônes, chaque icône représente une zone d'informations sélectionnable, qu'elle soit visible ou hors-champ de l'écran de l'utilisateur.

Sélectionner une des icônes a alors pour effet de placer la zone d'information correspondante au premier plan et de rendre actif le curseur dans cette zone. Si la zone d'information n'était initialement pas visible à l'écran, alors la fenêtre la contenant est automatiquement scrollée pour amener cette zone d'information au centre de l'écran. Pour développer ce widget, nous avons exploré quatre considérations de conception.

### 3.2 Considérations de conception

Le **lieu d'affichage** naturel consiste à dédier une partie fixe de *l'écran* pour l'affichage du support de la vue radar. Cette approche est classique et donc facile à appréhender. Elle réduit toutefois l'espace d'affichage à l'écran. Enfin, inspiré par Keytch [15] nous avons considéré un lieu d'affichage hors écran, sur une *tablette* posée entre l'utilisateur et le clavier et contre ce dernier. Il n'y a ainsi aucune occultation à l'écran et les icônes sont accessibles avec les pouces sans avoir à enlever les mains du clavier. Etant donné la taille de la tablette, son positionnement sous le clavier pourrait entraîner le déclenchement involontaire d'événements liés au contact des paumes et avant-bras sur la tablette. Néanmoins, une telle configuration a déjà été considérée dans la littérature avec le système LongPad

[10] en la combinant avec un algorithme de détection de paume, approche récemment adaptée au cas de dispositifs mobiles [20].

Le choix de l'utilisation d'une tablette plutôt que d'un smartphone comme dans les travaux cités en référence est motivé par le contexte de développement de ces travaux qui incluent déjà l'utilisation d'une tablette : au cours de la gestion de crise, l'utilisateur peut être amené à se déplacer auprès d'une autre cellule de la gestion de crise et il emporte alors sa tablette.

La **taille** du widget est guidée par des travaux antérieurs [15] qui ont montré que la plus grande partie d'un écran de smartphone de 5,8 pouces collé au clavier était atteignable avec un pouce. Nous avons donc choisi d'adopter une taille similaire pour le support de la vue radar quel que soit le support d'affichage utilisé. Affiché sur la tablette, il sera facilement accessible avec un pouce sans lever les mains du clavier. Affiché sur l'écran il aura la même capacité d'affichage que sur la tablette, facilitant ainsi la comparaison entre les deux solutions.

La **disposition des icônes** sur le support de la vue radar peut adopter une approche temporelle ou spatiale (cf. Figure 2). En effet WindowShopping [27] semble établir une préférence pour une approche temporelle alors que l'approche spatiale est identifiée comme plus intéressante par d'autres travaux [25]. Notre contexte impliquant la présence de sous-fenêtres et non plus seulement de fenêtres, il semblait particulièrement pertinent de proposer une version *grille*, représentant l'approche temporelle, et une version *spatiale* plaçant les icônes de la vue radar dans une disposition reproduisant celle des fenêtres à l'écran.



Figure 2 : Deux dispositions envisagées : grille avec 10 icônes (gauche) et spatiale avec 20 icônes (droite). Les termes utilisés pour distinguer les icônes correspondant à chaque zone sont représentatifs du contenu des zones d'information correspondantes

L'**orientation** du support de la vue radar peut soit adopter la forme utilisée par le raccourci ALT+TAB, donc une zone *horizontale*, soit préférer une forme *verticale* correspondant mieux à la forme des fenêtres affichées à l'écran et contenant des sous-fenêtre. Toutefois une étude préliminaire a permis d'écarter la solution verticale : tous les participants ont en effet clairement émis une préférence pour la version horizontale.

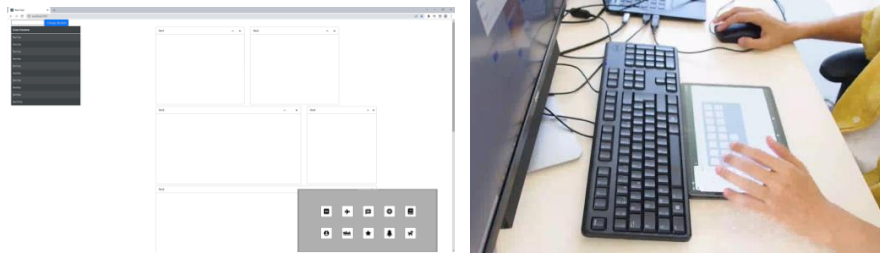


Figure 3 : Les deux lieux d’affichage de la vue radar (écran, tablette), avec une disposition grille et une orientation horizontale

### 3.3 Techniques d’interaction conçues

Compte tenu de ces considérations de conception, nous avons conçu et implémenté quatre techniques vue radar, résultant du croisement de la disposition (*grille, spatiale*) et du lieu d’affichage (*écran, tablette*).

Concernant l’affichage du support de la vue radar en lieu fixe à l’écran, nous avons choisi d’afficher le support en bas à droite de l’écran, ce qui est habituel dans l’univers du gaming et n’entraîne aucune occultation des barres de menus éventuellement placées en haut de la fenêtre (cf. Figure 3 – gauche). Cette technique est nommée *fixe*. Elle se décline en une technique Fixe-Grille et une technique Fixe-Spatiale selon la disposition des icônes adoptée. Sélectionner la zone d’information requiert donc de déplacer le curseur vers la vue radar puis de cliquer sur l’icône représentant la zone d’information recherchée.

Concernant l’affichage du support de la vue radar délocalisé sur une tablette, nous avons choisi de placer la tablette contre le clavier, comme dans [15] (cf. Figure 3 – droite). C’est la technique *mobile*. Elle se décline également en une technique Mobile-Grille et une technique Mobile-Spatiale. La vue radar est intégralement affichée sur une tablette, excluant ainsi toute occultation d’une partie de l’affichage fait sur l’écran. Pour adopter les mêmes conditions expérimentales que dans [15] et maintenir la main dominante disponible pour manipuler la souris, sélectionner la zone information requiert donc de toucher avec le pouce de la main non dominante l’icône affichée sur la tablette représentant la zone d’information cible. Etant donné la taille retenue pour le widget, et celui-ci étant affiché contre le bord supérieur de la tablette, les coins de la tablette au contact du clavier ne sont donc pas utiles à l’utilisation de la vue radar. Tout contact entre l’utilisateur et la tablette à ce niveau, en particulier ceux liés à des contacts involontaires des paumes et/ou avant-bras, pouvaient donc être ignorés dans notre implémentation du widget qui n’a donc pas nécessité la mise en œuvre d’algorithme d’identification de paume.

## 4 EXPERIMENTATION

Pour évaluer l’intérêt de ce widget dans la dynamique de changement de fenêtres, nous avons mené une expérimentation comparant nos quatre techniques d’interaction vue radar à une solution de référence basée sur l’utilisation d’une souris uniquement.

### 4.1 Tâche

La tâche considérée s’inscrit dans le cadre du suivi et de la gestion d’une crise. Ce contexte motive d’une part la présence d’une tablette et d’autre part le recours à une représentation sous la forme d’un tableau de bord. De plus pour être représentative de la situation visée, la tâche considérée inclut nécessairement une imbrication de monitoring et d’annotation de données : typiquement, l’utilisateur manipule une application (présentant le ou les tableaux de bords) dans laquelle plusieurs zones d’information (les sous-fenêtres) sont regroupées et qu’il peut



consulter et/ou éditer pour suivre l'évolution de la crise. Pour refléter ce contexte d'utilisation, la tâche de notre expérimentation se décompose en 3 étapes : 1) sélectionner une zone d'information ce qui place automatiquement le focus sur le premier champ éditable de la zone, 2) simuler la saisie de texte dans ce champ en appuyant sur la touche entrée du clavier, 3) simuler la sélection d'un autre champ de la zone en cliquant sur le bouton contenu dans la zone. Nous détaillons dans la section suivante comment chaque technique d'interaction permet de réaliser ces 3 étapes.

#### **4.2 Mise en œuvre des techniques d'interaction dans la tâche**

Une solution de référence, nommée *souris*, repose sur l'utilisation de la souris uniquement. Sélectionner la zone d'information implique donc de scroller dans la fenêtre avec la souris tenue dans la main dominante et de cliquer avec le bouton de la souris sur la zone d'information cible. L'utilisateur tape ensuite sur la touche Entrée du clavier avec sa main non dominante. Enfin la sélection du bouton contenu dans la zone d'information éditée se fait de nouveau à l'aide de la souris.

Nous avons choisi cette technique comme solution de référence car à notre connaissance aucune autre technique d'interaction ne peut être utilisée en l'état pour réaliser cette tâche dans un tel contexte. En effet, les techniques existantes permettent de passer d'une fenêtre à une autre (ALT+TAB), d'un onglet d'une fenêtre à un autre (CTRL+TAB) ou d'un champ de formulaire à un autre (TAB) mais n'implémentent pas le passage d'une sous fenêtre à une autre. Adapter ces techniques reviendrait donc à proposer une nouvelle technique et nécessiterait de considérer son insertion dans un environnement interactif existant avec attention : un même raccourci clavier pourrait en effet avoir deux rôles.

Avec les deux techniques Fixe (Fixe-Spatiale et Fixe-Grille) la main dominante demeure sur la souris. Aussi, après avoir sélectionné l'icône cible dans la vue radar avec la souris tenue en main dominante (étape 1), l'utilisateur tape sur la touche Entrée du clavier avec sa main non dominante (étape 2) et la sélection du bouton dans la zone d'information se fait de nouveau à l'aide de la souris tenue en main dominante (étape 3).

Avec les deux techniques Mobile (Mobile-Spatiale et Mobile-Grille) la main non-dominante demeure à proximité de la tablette pour pouvoir agir sur la vue-radar en gardant les mains sur le clavier comme décrit dans [15]. L'appui sur la touche Entrée est alors contrôlé par la main dominante. Aussi, après avoir sélectionné l'icône cible avec son pouce de la main non dominante sur la tablette (étape 1), l'utilisateur doit taper sur la touche Entrée du clavier avec sa main dominante (étape 2) et la sélection du bouton dans la zone d'information se fait aussi à l'aide de la souris en main dominante (étape 3).

Du point de vue de l'implémentation, un composant logiciel a été développé en technologie web (ReactJS). Lors du lancement du widget, la liste des zones d'information accessibles est obtenue afin de construire la vue radar. Les communications entre le composant, les zones d'information affichées à l'écran et la tablette reposent sur des websockets.

#### **4.3 Protocole expérimental**

Au cours de l'étude les participants ont été confrontés à 5 techniques : Souris, Fixe-Grille, Fixe-Spatiale, Mobile-Grille et Mobile-Spatiale. L'ordre de passage des techniques a été contrebalancé.

Pour chacune des cinq techniques, nous avons considéré trois nombres de zones d'information accessibles différents (10,15 ou 20 cibles potentielles) ce qui impacte la hauteur de la fenêtre contenant les zones

d'informations : l'utilisateur était confronté au nombre de cibles dans l'ordre croissant. Les zones d'informations étaient toutes identiques en contenu (cf. Figure 4-A)

Pour chaque technique et chaque nombre de zones, les cibles à atteindre étaient réparties dans la fenêtre de façon à générer autant de déplacement vers le haut de la fenêtre que vers le bas. La zone à atteindre était mise en évidence par une coloration de la zone et de l'icône correspondante : cette zone est colorée en orange pour les techniques Fixe et Mobile (cf. Figure 4-C et D); pour la technique Souris, la zone est aussi colorée en orange et une flèche indiquait en outre dans quelle direction scroller pour trouver la prochaine cible évitant ainsi un biais lié à la localisation de la cible (cf. Figure 4-B).

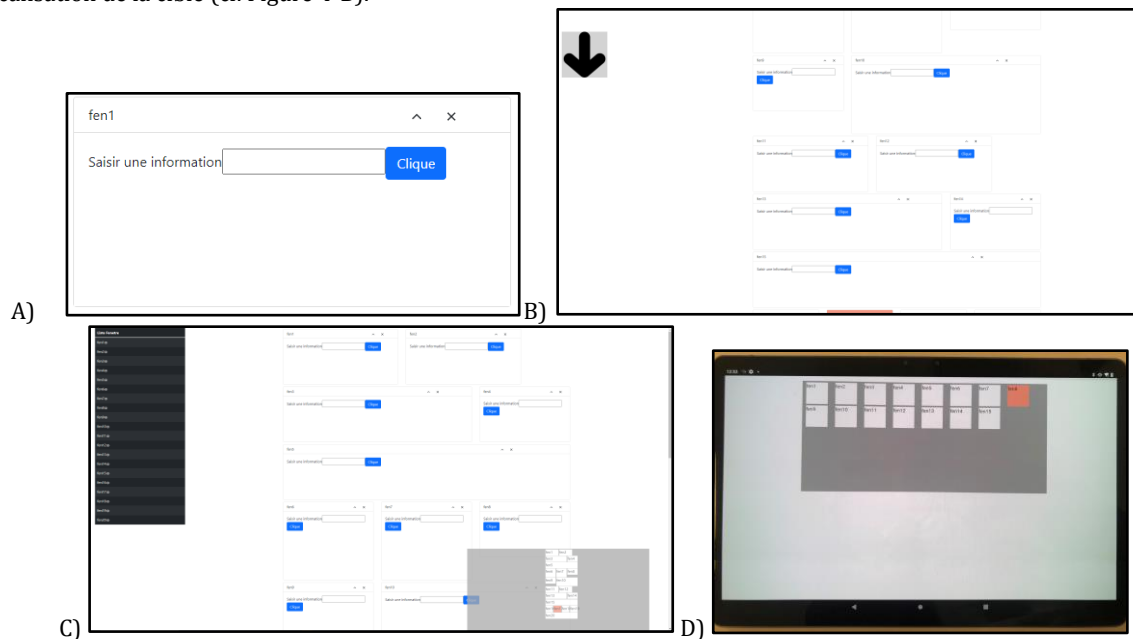


Figure 4 Illustration de l'application utilisée au cours de l'expérimentation : A) une zone d'information ; B) Combinée à la technique Souris, l'application affiche les différentes zones et une flèche indiquant le sens de Scroll requis pour atteindre la zone cible ; C) Combinée à la technique Fixe, l'application affiche les zones et la vue radar (disposition Spatiale) dans laquelle est mise en évidence la cible à atteindre ; D) combinée avec la technique Mobile, l'application affiche sur la tablette la vue radar (disposition Grille) avec mise en évidence de la cible à atteindre.

Pour chaque technique, l'utilisateur réalisait trois essais. Un essai consistait en l'enchaînement de quatre réalisations ininterrompues de la tâche décrite précédemment : sélectionner une zone, appuyer sur la touche Entrée, cliquer sur le bouton de la zone. L'intérêt était de simuler une activité la plus écologique possible puisqu'un utilisateur en cours de gestion de crise aura à enchaîner plusieurs changements et éditions de zones d'information. Au cours d'un essai, chaque participant avait donc 4 zones distinctes à sélectionner : la distance à parcourir pour atteindre deux de ces zones cibles était telle qu'elle nécessitait un scroll avec la technique Souris, et deux scroll pour les deux autres zones. Les cibles ont été prédéfinies pour chaque valeur du nombre de zones (10,15,20). Les mêmes cibles étaient à atteindre pour chaque utilisateur, mais variaient d'un essai à l'autre et d'une technique à l'autre pour éviter les risques d'apprentissage.

Avant chaque nouvelle technique, l'utilisateur disposait d'un temps d'apprentissage pour assimiler le mode de fonctionnement. Au total, chaque utilisateur a réalisé 3 (essais) x 4 (tâches) x 5 (techniques) x 3 (nombre de zones) = 180 tâches.

#### 4.4 Données collectées et participants

Au cours de l'expérimentation nous avons comptabilisé pour chaque technique le nombre d'erreurs commise, i.e. le nombre de sélections erronées de zone d'information au cours de chaque essai (avec la souris ou le pouce sur la tablette). Pour chaque réalisation de la tâche nous avons mesuré quatre durées :

- La *durée Totale* de la tâche, qui représente le temps de réalisation de la tâche incluant ses trois étapes : sélection de la zone, appuyer sur la touche Entrée, cliquer sur le bouton contenu dans la zone.
- La *durée Focus* de la tâche, qui correspond à la durée écoulée entre le début de la tâche et le fait que le focus soit effectivement mis par le système sur la zone cible suite à une action appropriée de l'utilisateur (clic sur la fenêtre, sur l'icône ou sur la tablette). La durée Focus représente le temps mis pour la navigation entre deux zones d'information à l'intérieur de la fenêtre.
- La *durée Enter* de la tâche, qui correspond à la durée écoulée entre le moment où le focus a été effectivement mis par le système sur la zone cible et le moment où l'utilisateur a appuyé sur la touche Entrée du clavier. La durée Enter est assimilable au temps mis par l'utilisateur pour se repositionner en mode saisie de texte, compte tenu de l'étape précédente (main et technique utilisées). Nous n'avons pas jugé utile de mesurer le temps de saisie de texte à proprement parlé.
- La *durée Bouton* de la tâche, qui correspond à la durée écoulée entre le moment où l'utilisateur a appuyé sur la touche Entrée du clavier et le moment où il clique sur le bouton contenu dans la zone sélectionnée. La durée Bouton évoque le temps mis pour changer de champ à l'intérieur de la zone sélectionnée et concrètement porter son regard sur le nouveau champ. Il est représentatif de l'action consistant à modifier deux valeurs contenues dans la même zone d'information.

Un essai étant constitué de quatre enchainements de cette tâche, nous avons enregistré ces 4 temps pour chacune des 4 tâches. Par souci de clarté dans les résultats présentés ci-après, nous ne considérerons ces 4 mesures qu'au niveau de l'essai (et non par tâche). Ainsi, la *durée Totale* correspond donc à la somme des *durées Totales* de chacune des 4 tâches constituant l'essai. Il en va de même pour la *durée Focus*, la *durée Enter* et la *durée Bouton*. A l'issue des essais concernant une technique les utilisateurs ont également rempli un questionnaire d'utilisabilité SUS [9].

Etant donnée la période et les contraintes d'accès au centre de gestion de crise, nous avons pu recruter 5 participants, dont deux femmes et trois hommes, d'âge moyen 28.5 ans et tous titulaire d'un diplôme en informatique. Nous avons choisi des participants parmi les acteurs de notre projet d'informatisation de la gestion de crise, pour qui la réalisation de tels essais, incluant quatre répétitions de la tâche décrite en section 4.1, est familière. En effet même si le fait de naviguer, cliquer puis appuyer sur une touche du clavier semblent être des tâches anodines, leur enchainement prend un sens particulier dans ce domaine : réaliser cet enchainement reproduit donc une situation classique pour ces participants alors qu'elle aurait pu paraître arbitraire pour des utilisateurs de profils plus diversifiés.

#### 4.5 Analyse des données

Nous avons choisi de nous appuyer sur des techniques d'estimation avec des intervalles de confiance à 95 % (IC95) et une analyse des ratios intra-sujets, tel que recommandé par l'APA [7] et suivant les conventions récentes dans la communauté IHM (une liste non exhaustive d'études CHI et VIS sans valeur p se trouve dans [2]). Dans le cadre de cette approche, le ratio est une mesure intra-sujet qui exprime la taille de l'effet (comparaison par paire) et est calculé entre chacune des moyennes géométriques.

Tous les IC sont des intervalles de confiance à 95 % par bootstrapping BCa. Pour le lecteur plus habitué à interpréter les valeurs p, nos résultats peuvent être lus en comparant l'espacement de nos IC avec l'espacement habituel des valeurs p comme indiqué par Krzywinski et Altman [17]. Les scripts utilisés pour calculer la moyenne géométrique et les intervalles de confiance sont disponibles en ligne [6].

Toutefois, compte tenu du nombre limité de participants impliqués dans cette étude, la puissance statistique des résultats mesurés demeure limitée et les résultats devront être considérés dans le cadre défini par l'étude présentée avant d'être généralisés dans une étude plus vaste.

#### 4.6 Résultats

Les résultats observés permettent de révéler des tendances, que nous détaillons ci-après, qui permettront de comprendre les apports et limites de la vue radar pour la gestion de fenêtres dans le cadre visé par l'étude présentée et en particulier la tâche et les participants impliqués. De plus les résultats n'ont pas permis d'établir des différences en temps ou en erreur entre chacune des quatre tâches constituant un essai. Nous analysons donc les résultats par essai et non par tâche.

**Nombre d'erreur.** Les résultats montrent dans un premier temps que l'introduction de la tablette comme support à la navigation entre les différentes zones d'information d'une même fenêtre ne semble pas avoir d'effet négatif sur le nombre d'erreur induit : le nombre d'erreur de sélection est comparable pour les trois techniques Fixe, Mobile et Souris. L'analyse du ratio intra-sujet laisse toutefois envisager que les techniques Mobile limiteraient encore le nombre d'erreur par rapport à l'utilisation de la Souris (Mobile-Grille/Souris=1.42 IC[1.22;1.67] ; Mobile-Spatial/Souris = 1.41 IC[1.26;1.55]). Nous pensons que la minimisation des erreurs avec les techniques Mobiles peut être expliquée par le fait que les icônes sont en permanence visibles (à l'inverse de la technique Souris avec laquelle les zones ne sont pas toutes visibles) et que les techniques Mobiles ne nécessitent pas de mouvement avec la souris pour y accéder, mouvement perturbant le flux d'interaction avec les données affichées à l'écran. La disposition (Spatiale, Grille) ne semble pas non plus influencer les techniques Mobile (cf. Figure 5).

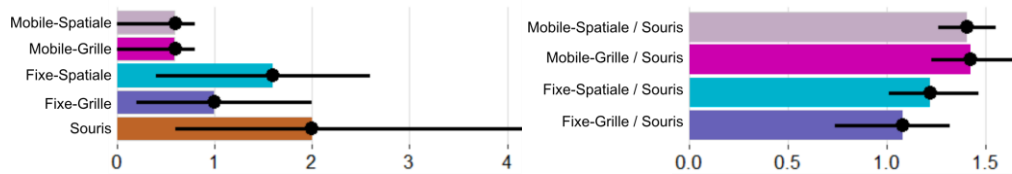


Figure 5 : Nombre moyen d'erreurs par Technique sur 180 tâches métier réalisées

**Durée totale.** Mise à part la condition impliquant 20 zones d'information, les résultats tendent à montrer que la technique Souris est plus efficiente en termes de temps de réalisation que les techniques Fixe-Grille, Mobile-Spatiale et Mobile-Grille et Mobile. Il est toutefois particulièrement intéressant de noter que les techniques Fixe et Mobile

semblent quant à elles rester stables face à l'accroissement du nombre de zones. Ce résultat est cohérent puisque le temps de scroll est nécessairement accru quand le nombre de zones d'information augmente.

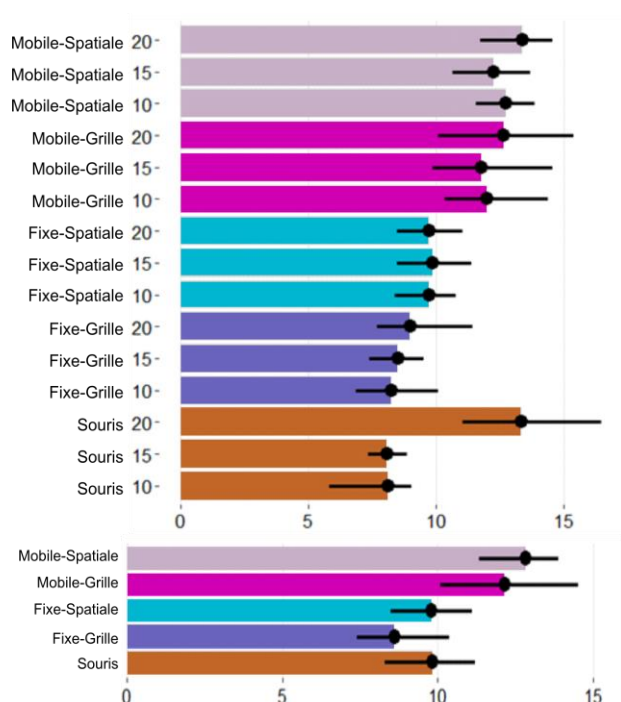


Figure 6 : Moyenne par Technique et Nombre de cible de la durée totale en s de réalisation d'un essai (Haut). Moyenne par Technique de la durée totale en s (Bas).

Globalement, sans tenir compte du nombre de zone d'information, l'analyse de la durée totale ne permet pas de mettre en évidence une différence entre les disposition Grille et Spatiale des techniques Mobile et Fixe (cf. Figure 6 – Bas). Cette observation est vraie quel que soit le nombre de zones d'information : ce résultat est intéressant car avec un plus grand nombre de zones, les icônes sont de taille plus petite dans la disposition Spatiale, ce qui augmente l'Indice de Difficulté et aurait pu accroître le temps. Cette observation nécessitera d'être confirmée dans une étude plus large.

**Durée Focus.** En considérant la durée Focus de l'essai (i.e. la somme du temps mis pour chacune des quatre tâches de l'essai pour mettre le focus sur la zone cible) les résultats indiquent que les techniques basées sur la vue radar induisent une durée Focus plus courte que la technique Souris (Souris : 6.53s ; Mobile : 5.14s (Grille) et 5.73s (Spatiale) ; Fixe : 4.09s (Grille) et 4.76s (Spatiale)). L'analyse du ratio intra-sujet permet également d'observer que les techniques Fixe et Mobile nécessitent moins de temps que la technique Souris et ce quelle que soit la disposition Souris (Mobile-Grille/Souris=0.78 IC[0.67 ;0.85] ; Mobile-Spatial/Souris = 0.89 IC[0.82 ;0.99]). Les résultats confirment donc l'hypothèse selon laquelle l'utilisation de la vue radar permet d'accroître l'efficacité de l'utilisateur pour atteindre et sélectionner une zone d'information par rapport à l'utilisation de la technique Souris.

Pour un nombre de zone d'informations accessibles égale à 20, l'utilisation de la Technique Souris induit une durée Focus nettement plus importante que pour 10 et 15 ce qui là aussi conforte l'observation faite pour la durée totale au sujet de la sensibilité de la technique Souris au nombre de zones cibles.

Notons également que, comme pour la durée totale, les résultats ne permettent pas de mettre en évidence une différence entre la disposition *Grille* et la disposition *Spatiale* des techniques Fixe et Mobile (cf. Figure 7) .

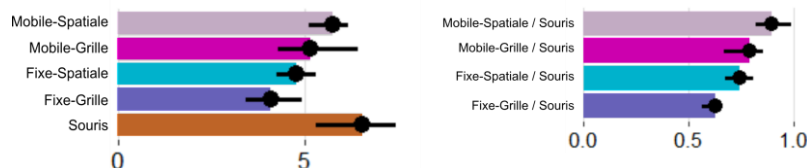


Figure 7 : Moyenne par Technique de la durée Focus en s.

**Durée Enter.** En considérant la durée Enter de l'essai (i.e. la somme du temps mis pour chacune des quatre tâches de l'essai pour appuyer sur la touche Entrée) les résultats montrent que la technique Souris minimise nettement la durée Enter (1.38s) (cf. Figure 8 – haut). Les techniques Fixe nécessitent 2.47s (Grille) et 2,75s (Spatiale) : les observations pendant l'expérimentation montrent que l'utilisateur a tendance à ramener la souris de la vue radar vers la fenêtre focus alors que seul le clavier est requis à cette étape. Enfin, les techniques Mobile nécessitent 4.36s (Grille) et 4,46s (Spatiale) : une explication possible de ce temps particulièrement long pour les techniques Mobile est la nécessité de déplacer la main dominante de la souris vers le clavier, alors que c'est la main non-dominante, non utilisée pour un autre dispositif, qui est utilisée avec les autres techniques (cf. section 4.2).

Nous observons en outre que ces résultats se maintiennent quel que soit le nombre de zones d'informations accessibles et que pour les techniques Fixes comme pour les techniques Mobile, la disposition n'affecte pas les résultats.

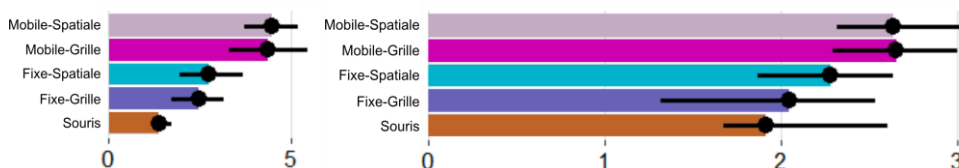


Figure 8 : Moyenne par Technique de la durée Enter en s (gauche). Moyenne par Technique de la durée Bouton en s (droite)

**Durée Bouton.** En considérant la durée Bouton de l'essai (i.e. la somme du temps mis pour chacune des quatre tâches de l'essai pour cliquer sur le bouton contenu dans la zone d'information) les résultats indiquent que les techniques Mobiles semblent nécessiter plus de temps que les autres techniques (Mobiles : 2,64s (Grille) et 2,65s (Spatiale) ; Fixe : 2,05s (Grille), 2,28s (Spatiale) ; Souris : 1,91s) Ces résultats mettent notamment en évidence que les techniques Mobiles requièrent plus de temps à cette étape : la cause possible est liée à la conception de cette technique puisque les techniques Mobile sont les seules pour lesquelles la même main est utilisée pour cette étape de la tâche (cliquer sur le bouton) et pour l'étape précédente (appuyer sur la touche Entrée du clavier) nécessitant donc un temps pour déplacer la main de la touche du clavier vers la souris.

**Score SUS.** Les techniques Fixe ont obtenu un score de 79, toutes dispositions confondues. Ce score est satisfaisant et correspond à une utilisabilité qualifiée de « GOOD » selon [3]. Les techniques Mobile par contre ont obtenu un score de 64 ce qui est reconnu comme étant susceptible de perturber l'utilisation de la technique du fait

de problème d'utilisabilité important. Un score de 72 est en effet une cible minimale à retenir. Ces résultats demeurent similaires quelle que soit la disposition retenue (*grille* ou *spatiale*) pour les techniques Fixes comme pour les techniques Mobile. CE score relativement faible pour les techniques Mobile est probablement lié à la nécessité de lâcher la souris pour venir appuyer sur la touche Entrée au cours de la réalisation de la tâche puisque la main dominante devait rester positionnée sur le clavier de façon à pouvoir interagir avec la tablette.

## 5 DISCUSSIONS ET PERSPECTIVES

**Limites du protocole expérimental.** Les résultats de cette étude ont établi que l'utilisation de la Technique Mobile permettait de réduire la durée Focus, i.e. le temps nécessaire pour localiser et sélectionner une zone d'information cible. Toutefois la réalisation de la tâche dans son intégralité (sélection et annotations) n'est pas optimale avec cette technique par rapport à l'utilisation de la technique Souris de référence. Néanmoins, nous avons souligné un biais dans l'expérimentation, puisque la technique Mobile est la seule nécessitant explicitement un changement de main pour la réalisation de deux étapes successives au cœur de la réalisation de la tâche (presser sur Entrée et cliquer sur le bouton). De plus le score SUS obtenu par cette technique est relativement bas. En corrigeant ce biais et en améliorant le design de la technique, nous pensons que la durée Totale de la réalisation de l'essai serait nettement améliorée pour la technique Mobile et permettrait de établir plus clairement les apports d'une telle technique dans une activité complète de monitoring et annotation d'un tableau de bord.

Par ailleurs, comme nous l'avons souligné, cette étude n'a été réalisée que par cinq participants, ce qui est bien entendu trop limité pour permettre d'établir des résultats généralisables et définitifs. Une étude complémentaire devra être menée pour corroborer ces résultats après avoir tenu compte des limitations et suggestions soulevées.

**Eviter les occultations.** Avec la technique Fixe, une partie des informations affichées à l'écran est occultée par l'utilisation de la vue radar. Il conviendrait donc d'envisager une solution permettant d'éviter cette limitation. Nous envisageons deux possibilités. La première consiste à définir un mécanisme d'activation de la vue radar, tel que la combinaison de touches ALT+TAB : ce mécanisme permettrait de déclencher l'affichage de la vue radar, dissimulée sinon. Le principal revers de cette solution réside dans l'instabilité de l'affichage à l'écran et la nécessité de mémoriser et produire ce mécanisme. Une seconde solution consiste à utiliser un affichage semi-transparent pour la vue radar. Des études ont en effet montré comment utiliser la transparence pour permettre d'utiliser deux fenêtres simultanément [13] [16].

**Dépasser la sélection.** Dans cette étude, le widget vue radar n'est utilisé que pour sélectionner une zone d'information et placer le focus sur la zone correspondante pour permettre à l'utilisateur d'annoter ou éditer des éléments de cette zone. Toutefois, des informations complémentaires pourraient être affichées sur la vue radar. En particulier, des indicateurs pourront être ajoutés aux icônes présentes dans la vue radar, pour donner une indication sur le type de contenu et faciliter encore la localisation de l'icône dans la vue radar.

De même, si une zone est modifiée par un événement extérieur (par exemple changement des prévisions météorologiques), l'icône correspondante pourrait alors être automatiquement mis en surbrillance comme illustré sur la Figure 9.



Figure 9 : Perspective d'utilisation de la vue radar avec mise en évidence des zones ayant été modifiée à l'extérieur

**Exploiter tout l'espace de la tablette.** La tablette offre un support plus grand que la taille de la zone support de la vue radar. Il semblerait donc particulièrement opportun d'exploiter la totalité de cet espace. Nous pensons en particulier que cet espace pourrait être utilisé pour faire des groupements d'icônes : un swipe avec le pouce permettrait de passer d'un ensemble d'icônes, correspondant par exemple aux zones d'information contenant des données internes à l'entreprise, à un autre regroupant les icônes associées par exemple aux zones d'informations relatives aux données externes à l'entreprise.

Nous pensons également qu'un mécanisme de « prévisualisation » pourrait être ajouté : un appui long sur une icône par exemple afficherait un aperçu rapide du contenu de la zone correspondante, avant de réellement mettre le focus sur cette zone à l'écran principal. Une étude plus approfondie avec les usagers en situation de gestion de crise permettra d'affiner la conception de cette fonctionnalité et son apport.

**Tendre vers de la gestion multi-display.** Enfin, l'analyse de données ne se passe pas toujours uniquement sur un seul dispositif d'affichage. Au contraire, plusieurs écrans peuvent être combinés à un mur de projection, une table interactive, etc. A plus long terme nous ferons évoluer ce widget vue radar pour assister l'utilisateur lors du passage d'une zone d'information à une autre dans un environnement multi-écrans.

## 6 CONCLUSION

Dans ces travaux nous avons étudié l'impact de plusieurs facteurs de conception d'une vue radar, un gestionnaire de fenêtres adapté au passage d'une zone d'information à une autre à l'intérieure d'une même fenêtre. A ces fins, nous avons conçu plusieurs versions de vue radar, et les avons comparées à une situation classique nécessitant la souris et une barre de scroll. Les résultats de cette étude révèlent que l'utilisation d'un tel widget de gestion des zones d'information sur un écran déporté, i.e. une tablette placée contre le clavier, a un effet positif sur l'efficacité et l'efficacité de l'utilisateur : d'une part l'étude menée permet notamment de mettre en avant l'apport de ces techniques sur le temps nécessaire pour passer d'une zone d'information à consulter ou annoter à une autre (Durée Focus) sans impacter le nombre d'erreurs et d'autre part cette solution évite toute occultation du contenu de l'écran, comme le ferait une des solutions existantes de la littérature et support à la gestion de fenêtres.

Ces travaux, bien que préliminaires compte tenu du nombre limité de participants à l'étude, montre le caractère prometteur de cette approche. Ils étendent également les travaux précédent, Keytch [15] qui avaient précédemment déporté les barres de menus sur un smartphone.



Enfin, ces travaux préliminaires ont donné lieu à une opérationnalisation concrète sur un logiciel de gestion de crise utilisé en situation réelle. Un tel déploiement permettra d'approfondir les bénéfices et usages possibles de la vue radar en contexte opérationnel.

## REMERCIEMENTS

Ces travaux ont été menés dans le cadre du projet Interactive<sup>Data</sup> entre le CEA Tech Occitanie et l'IRIT associé au CRCT du premier auteur. Ils ont été partiellement financés par le projet ANR INTERPLAY (Grant ANR-21-CE33-0022) et ont reçu, au travers du projet Vilagil – Aménagement Urbain, le soutien du Gouvernement Français dans le cadre du programme Territoire d'Innovation, une action du Grand Plan d'Investissement adossé à France 2030, de Toulouse Métropole et du GIS neOCampus de l'Université Toulouse III Paul Sabatier.

## REFERENCES

- [1] Android Developer : Multi-window support. <https://developer.android.com/guide/topics/large-screens/multi-window-support>. Last retrieved 5/10/2022.
- [2] Bad Stats: Not what it Seems. Retrieved January 3, 2023 from <https://www.aviz.fr/badstats#papers>
- [3] Aaron Bangor, Philip T. Kortum, and James T. Miller. 2008. An empirical evaluation of the system usability scale. *International Journal of Human-Computer Interaction* 24, 6: 574–594. <https://doi.org/10.1080/10447310802205776>
- [4] Michael S. Bernstein, Jeff Shrager, and Terry Winograd. 2008. Taskposé: exploring fluid boundaries in an associative window visualization. In *Proceedings of the 21st annual ACM symposium on User interface software and technology (UIST '08)*. Association for Computing Machinery, New York, NY, USA, 231–234. <https://doi.org/10.1145/1449715.1449753>
- [5] Koeun Choi, Hyunjo Song, Kyle Koh, Jinwook Bok, and Jinwook Seo. 2016. Peek-a-View: Smartphone Cover Interaction for Multi-Tasking. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. Association for Computing Machinery, New York, NY, USA, 4658–4662. <https://doi.org/10.1145/2858036.2858426>
- [6] Data Analysis example with data from a real study. Retrieved May 2, 2020 from <https://aviz.fr/ci/>
- [7] Declene KE, Fogo J. *Publication Manual of the American Psychological Association*. Publication Manual of the American Psychological Association. *Occup Ther Health Care*. 2012 Jan;26(1):90-2. doi: 10.3109/07380577.2011.629024. PMID: 23899111.
- [8] Thomas Geymayer, Markus Steinberger, Alexander Lex, Marc Streit, and Dieter Schmalstieg. 2014. Show me the invisible: visualizing hidden content. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. Association for Computing Machinery, New York, NY, USA, 3705–3714. <https://doi.org/10.1145/2556288.2557032>
- [9] G. Gronier and A. Baudet, (2021). Psychometric evaluation of the F-SUS: Creation and validation of the French version of the System Usability Scale. *International Journal of Human-Computer Interaction*. <https://doi.org/10.1080/10447318.2021.1898828>
- [10] Jiseong Gu, Seongkook Heo, Jaehyun Han, Sunjun Kim, and Geehyuk Lee. 2013. LongPad : a touchpad using the entire area below the keyboard of a laptop computer. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. Association for Computing Machinery, New York, NY, USA, 1421–1430. <https://doi.org/10.1145/2470654.2466188>
- [11] Aakar Gupta, Muhammed Anwar, and Ravin Balakrishnan. 2016. Porous Interfaces for Small Screen Multitasking using Finger Identification. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. Association for Computing Machinery, New York, NY, USA, 145–156. <https://doi.org/10.1145/2984511.2984557>
- [12] D. Austin Henderson and Stuart Card. 1986. Rooms: the use of multiple virtual workspaces to reduce space contention in a window-based graphical user interface. *ACM Trans. Graph.* 5, 3 (July 1986), 211–243. <https://doi.org/10.1145/24054>.
- [13] Edward W. Ishak and Steven K. Feiner. 2004. Interacting with hidden content using content-aware free-space transparency. In *Proceedings of the 17th annual ACM symposium on User interface software and technology (UIST '04)*. Association for Computing Machinery, New York, NY, USA, 189–192. <https://doi.org/10.1145/1029632.1029666>
- [14] Victor Kaptelinin. 2003. UMEA: translating interaction histories into project contexts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*. Association for Computing Machinery, New York, NY, USA, 353–360. <https://doi.org/10.1145/642611.642673>
- [15] Elio Keddissseh, Marcos Serrano, and Emmanuel Dubois. 2021. KeyTch: Combining the Keyboard with a Touchscreen for Rapid Command Selection on Toolbars. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 191, 1–13. <https://doi.org/10.1145/3411764.3445288>
- [16] Sunjun Kim and Geehyuk Lee. 2018. Design and Evaluation of Semi-Transparent Keyboards on a Touchscreen Tablet. In *Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces (ISS '18)*. Association for Computing Machinery, New York, NY, USA, 155–166. <https://doi.org/10.1145/3279778.3279801>
- [17] Martin Krzywinski and Naomi Altman. 2013. Points of Significance: Error bars. *Nature Methods* 10, 921–922. DOI: <https://doi.org/10.1038/nmeth.2659>

- [18] Ginny, C. Kumar and K. Naik, "A Study on User Interaction with Multi-Window Screen Support Framework for Multitasking Operations on Smartphones," 2017 14th IEEE India Council International Conference (INDICON), 2017, pp. 1-6, doi: 10.1109/INDICON.2017.8487718.
- [19] Google PlayStore, Transparent Keyboard Phone. <https://play.google.com/store/apps/details?id=com.touchscapesoftware.klearkeys.phone>, last retrieved 05/10/2022.
- [20] Huy Viet Le, Thomas Kosch, Patrick Bader, Sven Mayer, and Niels Henze. 2018. PalmTouch: Using the Palm as an Additional Input Modality on Commodity Smartphones. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18). Association for Computing Machinery, New York, NY, USA, Paper 360, 1–13. <https://doi.org/10.1145/3173574.3173934>
- [21] Nuria Oliver, Greg Smith, Chintan Thakkar, and Arun C. Surendran. 2006. SWISH: semantic analysis of window titles and switching history. In Proceedings of the 11th international conference on Intelligent user interfaces (IUI '06). Association for Computing Machinery, New York, NY, USA, 194–201. <https://doi.org/10.1145/1111449.1111492>
- [22] Jie Ren, Yueting Weng, Chengchi Zhou, Chun Yu, and Yuanchun Shi. 2020. Understanding Window Management Interactions in AR Headset + Smartphone Interface. In Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems (CHI EA '20). Association for Computing Machinery, New York, NY, USA, 1–8. <https://doi.org/10.1145/3334480.3382812>
- [23] George Robertson, Eric Horvitz, Mary Czerwinski, Patrick Baudisch, Dugald Ralph Hutchings, Brian Meyers, Daniel Robbins, and Greg Smith. 2004. Scalable Fabric: flexible task management. In Proceedings of the working conference on Advanced visual interfaces (AVI '04). Association for Computing Machinery, New York, NY, USA, 85–89. <https://doi.org/10.1145/989863.989874>
- [24] Greg Smith, Patrick Baudisch, George Robertson, Mary Czerwinski, Brian Meyers, Daniel Robbins. GroupBar: The TaskBar Evolved (2003) OZCHI 2003 Conference for the Computer-Human Interaction Special Interest Group of the Human Factors Society of Australia | January 2003
- [25] Susanne Tak, Andy Cockburn, Keith Humm, David Ahlström, Carl Gutwin, and Joey Scarr. 2009. Improving Window Switching Interfaces. In Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction: Part II (INTERACT '09). Springer-Verlag, Berlin, Heidelberg, 187–200. [https://doi.org/10.1007/978-3-642-03658-3\\_25](https://doi.org/10.1007/978-3-642-03658-3_25)
- [26] Craig Tashman. 2006. WindowScape: a task oriented window manager. In Proceedings of the 19th annual ACM symposium on User interface software and technology (UIST '06). Association for Computing Machinery, New York, NY, USA, 77–80. <https://doi.org/10.1145/1166253.1166266>
- [27] Andrew Warr, Ed H. Chi, Helen Harris, Alexander Kuscher, Jenn Chen, Robert Flack, and Nicholas Jitkoff. 2016. Window Shopping: A Study of Desktop Window Switching. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16). Association for Computing Machinery, New York, NY, USA, 3335–3338. <https://doi.org/10.1145/2858036.2858526>
- [28] Quan Xu and Géry Casiez. 2010. Push-and-pull switching: window switching based on window overlapping. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10). Association for Computing Machinery, New York, NY, USA, 1335–1338. <https://doi.org/10.1145/1753326.1753526>
- [29] Jackie (Junrui) Yang, Monica S. Lam, and James A. Landay. 2020. DoThisHere: Multimodal Interaction to Improve Cross-Application Tasks on Mobile Devices. In Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology (UIST '20). Association for Computing Machinery, New York, NY, USA, 35–44. <https://doi.org/10.1145/3379337.3415841>