
Modélisation graphique des environnements proxémiques basée sur un DSL

P. Pérez¹, P. Roose¹, M. Dalmau¹, Y. Cardinale², D. Masson³, N. Couture⁴,

1. Université de Pau et des Pays de l'Adour, LIUPPA, France

paulo.perez-daza., Philippe.Roose, marc.dalmau @iutbayonne.univ-pau.fr

2. Universidad Simón Bolívar, Venezuela

ycardinale@usb.ve

3. DEV 1.0, France

d.masson@dev1-0.com

4. Univ. Bordeaux, ESTIA INSTITUTE OF TECHNOLOGY, LaBRI, France

n.couture@estia.fr

RESUME. L'interaction proxémique est un domaine émergent dont l'objectif est d'améliorer l'expérience utilisateur. La proxémique est basée sur cinq dimensions (Distance, Identité, Localisation, Mouvement et l'Orientation que nous appelons DILMO) pouvant être utilisées pour définir des interactions entre personnes et/ou appareils numériques. Les travaux de recherche et développement actuels dans ce domaine sont essentiellement axés sur le développement d'applications proxémiques via une boîte à outils permettant d'obtenir les informations DILMO à partir de capteurs, et exploitent exclusivement du matériel spécifique. Le passage à l'échelle ne peut actuellement être réalisé en raison de la spécificité du matériel utilisé mais également en raison de l'absence d'approches de conception pour modéliser de tels environnements et les comportements proxémiques. Afin de faciliter l'intégration des capacités proxémiques dans les Interfaces Homme Machine, nous proposons une modélisation graphique basée sur un DSL permettant à des concepteurs non informaticiens d'exprimer les interactions proxémiques. Nous décrivons ici formellement notre proposition de DSL et nous l'illustrons par la mise en œuvre d'un prototype basé sur du matériel standard pour concevoir des comportements proxémiques dans différents environnements.

ABSTRACT. Proxemic interaction is an emerging area for improving HCI experiences. It describes how the five dimensions (i.e., Distance, Identity, Location, Movement, and Orientation -- DILMO) can be used to define interactions among people and digital devices. Current studies in this area are focused on developing proxemic applications with specific functionalities, based on toolkit that allow developers to obtain DILMO information from sensors. However, there exists a notable lack of general approaches able to support the whole implementation process, starting from the modeling of proxemic environments to represent general proxemics behaviors, and finalizing with the development of specific applications. In order to facilitate the integration of proxemic capabilities in Human Computer Interaction, we propose a graphical modelling based on Domain-Specific Language (DSL) that allows

designers to express proxemic interactions. We formally describe our proposed DSL and we applied it with a prototype to design proxemics behaviors in different proxemic environments.

Mots-clés : Interactions Proxémiques, , DSL, Modélisation, Conception,

KEYWORDS: Proxemic interaction, DSL, Modeling, Design.

1. Introduction

Le concept original de proxémique a été proposé par Edward T. Hall dans les années 60 (Hall, 1966). Il a présenté la façon dont les gens perçoivent, interprètent et utilisent l'espace, en particulier en ce qui concerne la distance physique entre les personnes (Evans *et al.*, 2000). Ce concept a été repris par Greenberg en 2011 dans le contexte de l'Interaction Homme-Machine. Ainsi, l'interaction proxémique est un concept utilisé principalement pour décrire comment les personnes utilisent les distances interpersonnelles pour interagir avec les appareils numériques en termes de dimensions physiques proxémiques : Distance, Identité, Localisation, Mouvement et Orientation (DILMO) (Ballendat, 2010; Greenberg, 2011).

Certaines études se focalisent sur le développement d'applications proxémiques dans un cadre spécifique (Ballendat, 2010 ; Brock, 2018; Brudy, 2019; Cho, 2018; Evans, 2000; Garcia, 2019; Gørnb, 2019; Pérez, 2018; Sørensen, 2013). D'autres travaux proposent des cadres et une boîte à outils plus génériques qui aident les développeurs à traiter les informations proxémiques et à mettre en œuvre les comportements proxémiques des entités (Cadenas, 2017 ; Marquardt, 2013). Le comportement est ici à prendre comme une réaction face à une action d'une entité (*cf. 3.1- définition 1*). Cependant, il manque des approches plus génériques capables de soutenir l'ensemble du processus de réalisation des applications proxémiques pour les environnements intelligents, de la phase de conception au déploiement, en passant par le développement. De plus, il existe un besoin de standardisation, et dans un premier temps, le besoin de la définition d'une approche méthodologique intuitive et compatible avec du matériel standard (ie. les smartphones du marché)

Dans ce contexte, nous proposons une formalisation et un outil graphique de modélisation des interactions proxémiques pour des non spécialistes de la proxémique. Notre proposition est un DSL (Domain Specific Language) composé de symboles et de notations formelles. Les symboles représentent tous les composants d'un environnement proxémique, c'est-à-dire les entités avec lesquelles interagir, les dimensions DILMO et les comportements proxémiques (actions associées). À partir du modèle graphique, le DSL permet à des concepteurs non forcément informaticiens de créer un schéma XML. Ce schéma XML est ensuite utilisé pour générer automatiquement des classes Java qui seront incluses dans les applications pendant la phase de développement pour la collecte et l'enregistrement des informations proxémiques. Dans cet article, nous décrivons formellement cette modélisation graphique. Pour présenter l'utilisabilité du modèle graphique, nous avons développé un prototype du DSL permettant de concevoir les comportements proxémiques dans différents environnements et de générer les schémas XML.

La théorie sur la proxémique de Hall décrit les façons dont les individus perçoivent leur espace personnel en fonction de leur distance avec les autres (Gørnbk et O'Hara, 2016). Selon la théorie proxémique de Hall les zones d'interaction sont au nombre de 4 : intime (définie comme une distance entre 0 et 50 cm), personnelle (entre 0,5 et 1 m), sociale (entre 1 et 4 m) et publique (au-delà de 4 m). Il met en évidence le rôle des relations proxémiques comme éléments de communication entre les personnes. Les concepts de proxémique sont basés sur des facteurs physiques, sociaux et culturels qui influencent et régulent les interactions interpersonnelles (Marquard et al., 2011; 2013). Afin de savoir comment ces facteurs devrait être appliqués aux interactions proxémiques pour les applications informatiques, (Greemnerg et al., 2011) a identifié cinq dimensions : Distance, Identité, Localisation, Mouvement et Orientation (DILMO) qui sont associés à des personnes ou des appareils numériques. L'utilisation des interactions proxémiques a été exploitée par certains chercheurs pour concevoir des environnements intelligents basés sur l'interaction (Lendo, 2015; Rector 2004) et pour permettre aux utilisateurs de contrôler des appareils numériques de manière plus naturelle (Ballendat, 2010 ; Brudy, 2019 ; Lendo 2015; Marquard, 2013). L'interaction proxémique se situe au niveau de l'espace vital de l'homme et de la distance physique entre un individu et un dispositif numérique classé selon une échelle taxonomique proposée par (Brudy et al., 2019), tandis que les interfaces distribuées sont classées de manière dynamique dans la même taxonomie en raison des interactions entre les différentes plateformes numériques utilisées par les différents utilisateurs. Dans cette section, nous examinons quelques études récentes et remarquées dans ce domaine.

2.1. Méthode de classification des travaux

Afin de trouver, sélectionner et analyser des études centrées sur les outils pour soutenir la conception et la mise en œuvre d'applications proxémiques, nous avons fait une étude ciblée décomposée en trois étapes principales : (i) recherche des travaux relatifs au développement d'une application proxémique (ii) sélection des articles ; et (iii) élaboration d'un tableau comparatif basé sur l'ensemble de critères suivant :

1. **Type d'application** : cet aspect se rapporte à l'analyse du champ d'application, il permet d'évaluer s'il s'agit d'un outil permettant le développement d'applications ou d'une application particulière.
2. **Dimension proxémique** : cet aspect présente comment sont utilisées les dimensions proxémiques pour concevoir les environnements proxémiques.
3. **Méthodes de conception** : cet aspect évalue les méthodes et les modèles graphiques utilisés pour représenter les comportements proxémiques et concevoir des environnements proxémiques.
4. **L'utilisabilité** : cet aspect spécifie si les travaux ont offert des solutions pour concevoir les interactions et environnements proxémiques ainsi que la disponibilité du code source.

2.2. Analyse comparative

Le Tableau 1 résume l'analyse qualitative et comparative que nous décrivons ci-dessous.

La plupart des travaux portent sur la mise en œuvre d'applications pour des fonctions spécifiques. Des études récentes comme (Ballendat, 2010; Brudy 2018, Gørnb, 2018, Lendo) concernent le contrôle des appareils numériques dans des environnements intelligents pour mettre en œuvre des lecteurs de médias interactifs, de l'analyse vidéo et des applications de contrôle à distance. Des applications proxémiques ont également été utilisées pour soutenir l'interaction gestuelle et améliorer la vie quotidienne des personnes aveugles (Brock, 2018 ; Dingler 2009 ; Garcia 2019; Mentis, 2012; Mojgan, 2018; Rector, 2017). Des applications mobiles proxémiques ont été développées grâce aux nouvelles capacités des appareils mobiles qui permettent aux développeurs d'obtenir des informations proxémiques à partir des capteurs des smartphones (Brock, 2018 ; Cho, 2018; Perez, 2018; Sørensen, 2013). Certains travaux développent des applications proxémiques pour adapter les visuels sur les écrans (Dostal, 2014; Vermulen, 2015; Wolf, 2016). La manière dont les interactions proxémiques peuvent être utilisées pour les robots de service est décrite dans (Bhagya et al., 2018). Seuls quelques travaux ont développé des frameworks et des outils pour construire et concevoir des applications basées sur des interactions proxémiques (Cardenas, 2017; Gørnbk, 2016; Kim 2016; Marquard 2011). La majorité des travaux utilise tout ou partie des dimensions DILMO. Les environnements proxémiques basés sur DILMO, dans lesquels toutes les dimensions sont prises en compte, sont décrits dans (Ballendat 2010; Bhagya, 2018; Brudy, 2019; Cardenas, 2017; Dostal, 2014; Garcia, 2019; Kim 2016; Ledo, 2015; Marquardt, 2011; Mentis, 2012; Mojgan, 2018; Rector, 2017). Le travail présenté dans (Cardenas et Garcia, 2017) illustre comment les dimensions proxémiques peuvent soutenir l'interaction entre entités (personnes et objets) en tenant compte du contexte. L'étude présentée dans (Dostal et al., 2014) montre comment les personnes peuvent interagir dans différentes zones et distances autour d'un dispositif spécifique.

Les environnements proxémiques réduits aux dimensions DIMO sont pris en compte dans les études présentées dans (Brock, 2018 ; Wang 2012). Dans (Brock, 2018) est présentée une application capable d'ajuster le contenu de l'affichage d'un dispositif identifié en fonction de la distance, de la localisation, du mouvement et de l'orientation des utilisateurs. Dans le contexte des jeux, trois environnements proxémiques DLMO sont présentés dans (Gørnbk et al.,2019). Les actions des jeux sont établies en fonction de la distance, de la localisation, des mouvements et de l'orientation des appareils manipulés par les enfants. Dans (Cho et al.,2018), un environnement proxémique DIL est décrit dans le contexte d'une application qui permet la reconnaissance de matériaux grâce à une caméra thermique mobile à bas prix intégrée dans un smartphone. Cette application mesure la distance physique entre la caméra et le matériau et reconnaît une texture spécifique (l'identité). Des travaux basés sur les environnements DILM sont présentés dans (Dingler, 2015; Sørensen. 2013). Par exemple, le système de musique multi-pièces proposé dans

(Sørensen *et al.*, 2013) est une application mobile basée sur des interactions proxémiques et permet à l'utilisateur d'écouter la même playlist malgré ses déplacements grâce à l'activation de différents haut-parleurs disposés dans la maison. Les environnements proxémiques réagissant à la distance, à l'identité et à l'orientation (c'est-à-dire les environnements proxémiques DIO) sont présentés dans (Dostal, 2013; Gørnbk, 2016; Wolf, 2016). L'étude présentée dans (Gørnbk *et al.*, 2016) détaille l'utilisation de la boussole intégrée dans les appareils mobiles qui a permis d'identifier et soutenir le processus d'appariement basé sur la distance et l'orientation. Dans (Perez *et al.*, 2018), un environnement proxémique DIM est décrit dans le contexte d'une application mobile de premiers secours (FAMA). FAMA offre aux sauveteurs la possibilité d'obtenir l'identité d'une personne inconsciente en cas d'urgence lorsqu'ils se déplacent dans les zones proxémiques de la personne blessée. La distance et la localisation (DL) des personnes sont prises en compte dans (Vermeu *et al.*, 2015) pour la définition d'actions dans un affichage interactif.

En ce qui concerne les méthodes de conception, seuls trois travaux ont présenté des outils pour concevoir les interactions proxémiques. SpiderEyes (Dostal *et al.* 2014) est un système qui fournit un outil visuel qui permet aux développeurs de concevoir des applications proxémiques collaboratives. Le travail présenté dans (Kim *et al.*, 2016) illustre les interactions proxémiques dans des environnements intelligents en utilisant des modèles miniatures. Cependant, il faut du matériel spécifique pour concevoir ces environnements tels que des marqueurs de réalité augmentée (AR) et un projecteur de caméra. Dans (Marquardt *et al.*, 2011) est présenté une boîte à outils composée d'une collection de bibliothèques en C et d'une architecture à composants qui prend en compte les informations spatiales et les relations entre les objets et l'espace. Toutefois, il n'offre pas de notation graphique pour permettre aux utilisateurs finaux la modélisation des comportements proxémiques.

Tableau 1. L'évaluation comparative

Author	Type	D	I	L	M	O	Méthodes de conception	L'utilisabilité	
								Conception intuitive	Disponibilité du code source
(Ballendat <i>et al.</i> ,2010)	Application	■	■	■	■	■			
(Bhagya <i>et al.</i> ,2018)	Service robot	■	■	■	■	■			
(Brock <i>et al.</i> ,2018)	Application	■	■		■	■			
Brudy, 2019	Application	■	■	■	■	■			
(Cardenas <i>et al.</i> ,2017)	Framework	■	■	■	■	■			■
(Cho <i>et al.</i> ,2018)	Application	■	■	■					■
(Dingler <i>et al.</i> ,2015)	Application	■	■	■	■				

(Dostal et al.,2014)	Application	■	■	■	■	■	■		
(Dostal et al.,2013)	Application	■	■			■			
(Garcia et al.,2019)	Application	■	■	■	■	■			
(Gørnbk et al.,2019)	Application	■		■	■	■			
(Gørnback et al.,2016)	Application	■	■			■			
(Kim et al.,2016)	Application	■	■	■	■	■	■	■	
(Ledo et al.,2015)	Application	■	■	■	■	■			
(Marquardt et al.,2011)	Framework	■	■	■	■	■	■		■
(Mentis et al., 2012)	Application	■	■	■	■	■			
(Mojgan et al. 2018)	Application	■	■	■	■	■			
(Perez et al.,2018)	Application	■	■		■				
(Rector et al.,2017)	Application	■	■	■	■	■			
(Sørensen et al.,2013)	Application	■	■	■	■				
(Vermeu et al., 2015)	Application	■		■					
(Wang et al.,2012)	Framework	■	■		■	■			
(Wolf et al. ,2016)	Application	■	■			■			

L'utilisabilité est le critère le moins rencontré dans la grande majorité des propositions antérieures. Nous n'en avons trouvé qu'un qui offre des notations intuitives pour concevoir facilement des applications proxémiques (Kim et al, 2016). Ce cadre fournit des outils qui aident les développeurs à construire une application frontale. Cependant, il ne fournit pas de notation graphique pour permettre au concepteur de modéliser les comportements proxémiques.

3. Modélisation du comportement proxémique

Dans cette section, nous présentons une description générale de la modélisation du comportement proxémique fondée sur un DSL. Le DSL permet la représentation des environnements et des comportements proxémiques à partir de conditions de départ spécifiques des objets et entités physiques basées sur les dimensions DILMO. À partir de la modélisation graphique d'un scénario proxémique, un schéma XML est généré, afin de faciliter le développement d'applications proxémiques. La modélisation et la standardisation permettent que les concepteurs aient un langage commun pour exprimer les comportements proxémiques de chaque entité dans un environnement proxémique.

3.1. Définitions formelles pour les environnements proxémiques

Un environnement proxémique (défini précisément plus bas) est décrit à partir d'une entité cible (CPS, définie plus bas) qui réagit selon les dimensions DILMO avec toutes les autres entités. La première définition concerne donc les entités qui peuvent interagir dans un environnement proxémique

Définition 1. Entité (E), une entité, désignée par E, représente un objet d'interaction telle qu'une personne, un objet ou un dispositif, qui peut être identifié ou non, dans un espace physique.

Définition 2. Cyber Physical System (CPS), représente l'entité cible, à partir de laquelle un environnement proxémique (E_P) est défini. Toutes les zones proxémiques (Z_P) et les dimensions DILMO sont mesurées pour toutes les entités (E) par rapport au CPS.

Les définitions 3 à 6 portent sur les attributs proxémique.

Définition 3. L'identité, notée (I), représente une entité (E) clairement identifiée dans l'environnement proxémique. Par analogie avec la définition 1, il s'agit de 'cette' personne, 'ce' dispositif ou de 'cet' objet.

Définition 4. La distance, notée (D), est la mesure physique entre des entités.

Définition 5. Localisation, notée (L), représente les coordonnées cartésiennes indiquant la localisation d'une entité (E) par rapport au CPS dans chaque zone proxémique ; elle est désignée comme E.L = (x₁, y₁) ou I.L = (x₁, y₁).

Définition 6. Le mouvement, noté (M), est un vecteur représentant la magnitude et la direction d'une entité (E ou I) par rapport au CPS. Elle est désignée par E.M = (m,d) ou I.M = (m,d) ;

- Si d = 1, E ou I se déplace vers le CPS.
- Si d = 0, E ou I est statique
- Si d = -1, E ou I s'éloigne du CPS.

Définition 7. L'orientation, notée (O) représente l'angle relatif à l'alignement face à face des entités par rapport au CPS, elle est désignée comme E.O ou I.O.

L'environnement proxémique est l'espace physique dans lequel les entités peuvent interagir avec le CPS selon les variables DILMO.

Définition 8. Zone proxémique (Z_P), une zone proxémique représente une zone circulaire délimitée par une distance maximale par rapport au CPS. Il existe quatre zones proxémiques définies en fonction de cette distance maximale : Z_P_{intimité}, Z_P_{personnelle}, Z_P_{sociale}, et Z_P_{publique}.

La zone proxémique dans laquelle une entité est située est désignée comme : E.P_Z ou I.Z_P. Elle peut prendre les valeurs suivantes : tt

- $E.Z_P = Z_P_{intimité}, si \ 0 < E.D \leq MAX_ID;$
- $E.Z_P = Z_P_{personnelle}, si \ MAX_ID < E.D \leq MAX_PD;$
- $E.Z_P = Z_P_{sociale}, si \ MAX_PD < E.D \leq MAX_SD;$
- $E.Z_P = Z_P_{publique}, si \ MAX_SD < E.D \leq MAX_PubD;$

où MAX_ID, MAX_PD, MAX_SD, et MAX Pub_D représentent les distances maximales définissant respectivement les zones intimes, privées, sociales, et publiques. Il s'agit de paramètres qui doivent être fournis par les utilisateurs ou les développeurs. Par analogie, la même notation s'applique pour les entités identifiées (I) I.Z_P.

Définition 9. Environnement Proxémique, noté (E_P), représente un ensemble de capteurs et de dispositifs attachés à des entités (E) qui peuvent à leur tour interagir selon (DILMO) dans le cadre d'un scénario.

Définition 10. Comportement (B), un comportement représente le changement des mesures D, L, M et O ou du Z_P d'une entité (E ou I), à partir de son comportement initial (B₀), sachant que l'identité de I ne changera pas. Le comportement est désigné par un tuple $E.B_i = \langle E.D_i, E.L_i, E.M_i, E.O_i, E.Z_P_i \rangle$

Cette notation est produite par une transition à partir du comportement précédent E_{i-1} , de telle sorte que :

$$E.B_0 = \langle E.D_0, E.L_0, E.M_0, E.O_0, E.Z_P_0 \rangle \text{ (le comportement initial)}$$

- ∇ = représente le changement des conditions d'une entité, il existe une fonction définie par l'utilisateur pour détecter le changement de dimensions DLMO
- $E.B_i = \nabla E.B_{i-1} = \langle E.D_i, E.L_i, E.M_i, E.O_i, E.Z_P_i \rangle$ (for $i \geq 1$)

En fonction du comportement d'une entité (E) dans un E_P il est possible de détecter le changement d'une mesure DLMO qui provoquera automatiquement le changement d'autres mesures DLMO et de sa zone proxémique. Par exemple, si un mouvement (M) de E est détecté, cela causera le changement de sa localisation (L) et de sa distance (D) ; De même, si la distance (D) de E change, cela signifie que E se déplace et que sa localisation (L) changera également. Dans ces deux exemples, il est également possible que la Z_P de E change.

Définition 11. Action (Action), une Action représente un événement réalisé par le CPS ou toute autre entité (E ou I), dans un E_P, en réponse à un comportement spécifique (B) d'une entité ou d'un groupe d'entités.

Pour définir un E_P, les conditions sont donc :

1. Distances, définies pour les quatre Z_P fournies par les concepteurs/développeurs en fonction des besoins des utilisateurs.
2. Un E_P existe seulement dans le cas , où il contient une entité cible (CPS), des entités et des distances :

$$E_P = \langle \text{CPS, entités, identités, distances} \rangle, \text{ le CPS n'est pas NULL}$$

3. Le CPS représente la cible d'interaction et a les propriétés suivantes :
 - la localisation d'origine du CPS est indiquée comme CPS.L (0,0), puisque les zones proxémiques et les dimensions DILMO de tous les autres objets d'interaction sont déterminées par rapport au CPS ;
 - Le champ de vision est défini selon les angles suivants : α MinAofV et α MaxAofV. Ces valeurs sont des paramètres fournis par les utilisateurs ou les développeurs.
4. Pour toutes les entités (E) du système, les dimensions DILMO et les zones proxémiques peuvent avoir des conditions initiales :

$$\forall E_i \in E_P \wedge I_j \in E_P, E_i.B_0 = \langle D_0, L_0, M_0, O_0 \rangle \wedge I_j.B_0 = \langle D_0, I_j, L_0, M_0, O_0 \rangle$$
 et L_0, M_0, O_0 peuvent être NULLE.

Cela signifie que la distance (D) est la seule dimension obligatoire pour les entités. L'orientation est une dimension non pertinente si le CPS n'a pas de champs de vision c'est à dire que α MinAofV et α MaxAofV sont égaux à zéro. Par exemple si nous définissons une action qui est basée sur la distance entre les appareils avec l'utilisation du Bluetooth seulement.

Ces définitions forment la base formelle de notre DSL. Elles sont utilisées pour spécifier l'E_P, avec ses entités et leurs actions, qui sont définies à partir de conditions initiales (Z_P et DILMO).

3.2. Notation graphique, à base de symboles, pour les environnements proxémiques

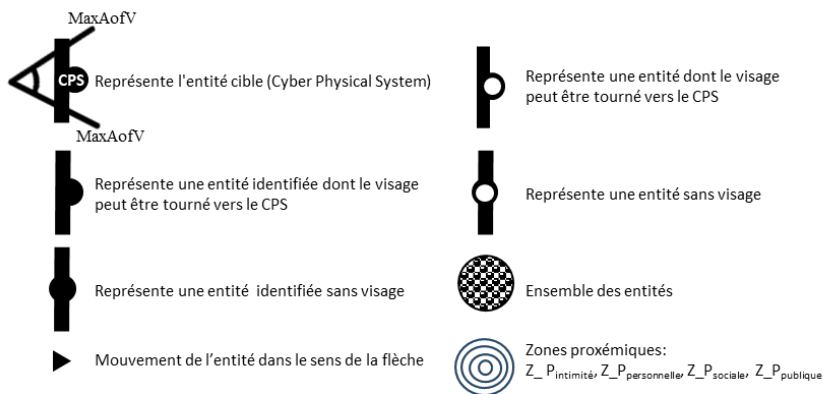


Figure 1. Symboles de la notation Graphique pour les env. proxémiques

Notre modèle graphique s'inspire du système de notation proposé par Edward T. Hall dans (Hall, 1963; 1966) qui fournit un code de notation permettant la représentation des relations humaines. La figure 1 décrit les éléments graphiques que nous proposons. Nous illustrons le formalisme de la notation graphique sur l'exemple suivant :

Soit un E_P , avec les identités I_1 et I_2 se dirigeant vers le CPS depuis la zone personnelle ($Z_P_{personnel}$) et faisant face au CPS et l'entité E_1 se tenant devant le CPS dans la zone sociale ($Z_P_{sociale}$). La figure 2 montre la représentation graphique de ce scénario avec les conditions initiales suivantes :

- Les quatre tuples de E_P : $E_P \ll \text{CPS}$, entités = $\{E_1\}$, identités = $\{I_1, I_2\}$, distances = $\{0, 4, 1, 4, 7\}$ >
- $\text{CPS.L} = (0, 0)$; $\nexists \text{MinAofV} = 0^0$ et $\text{MaxAofV} = 60^0$;
- Distance des entités ; $I_1.D = 0.85$, $I_2.D = 0.80$, $E_1.D = 3$;
- Localisation de l' $I_1.L = (0.8, 0.3)$;
- Mouvement de l' I_1 et l' I_2 (vers CPS) : $I_1.M(l_1, 1)$ et $I_2.M(l_2, 1)$; Mouvement de l' E_1 (statique) : $E_1.M=(l_1, 0)$;
- Orientation de l'entités (les visages des entités sont dans le champs de vision du CPS): $I_1.O=20^0$, $I_2.O=35^0$, $E_1.O= 75^0$ ainsi : $\nexists \text{MinAofV} \nexists I_1.O, E_1.O \nexists \text{MaxAofV}$;
- Zones proxémiques des entités $I_1.Z_P$, $I_2.Z_P = Z_P_{personnelle}$, $E_1.P = Z_P_{sociale}$;
- Ainsi, les comportements initiaux des entités sont :
- $I1.B_0 = \langle 0.85, (0.8, 0.3), (l_1, 1), 20^0, Z_P_{personnelle} \rangle$
- $I2.B_0 = \langle 0.8, \text{NULL}, (l_2, 1), 35^0, Z_P_{personnelle} \rangle$
- $E1.B_0 = \langle 3, \text{NULL}, (l_3, 0), 75^0, Z_P_{sociale} \rangle$

À partir de ces conditions initiales, le CPS peut réagir par des actions qui peuvent être désignées de plusieurs façons, par exemple : $Z_P_{sociale}$

Si $\exists E$ dans $Z_P_{sociale}$, alors CPS.Action_1 .

$\forall I \in P \text{ E.identités} \wedge I_j$ dans $Z_P_{personnelle}$ puis CPS.Action_1 . Ainsi, selon les comportements des entités détectées (E ou I), les actions du CPS peuvent changer. Elles sont représentées par des notations formelles.

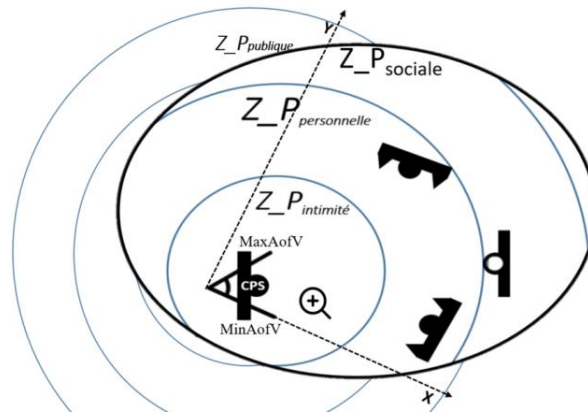


Figure 2. Modèle graphique d'un E_P

Une fois les comportements modélisés dans ce scénario (concepteur), les actions doivent être implémentées (développeur). Cette étape nécessite la production de code. De même, la manière dont les comportements peuvent être détectés en fonction de la technologie utilisée dans le E_P (par exemple, BLE Bluetooth à base consommation, capacités des capteurs de l'appareil, vision par ordinateur) nécessitent l'intervention et la production de code. Lorsqu'un E_P possède plusieurs CPS, il sera nécessaire d'itérer la modélisation pour chaque CPS.

Le choix entre E et I dépend des entités manipulées, des actions à réaliser et de leur action générique. Prenons un scénario simple : lorsque deux personnes (E) s'approchent d'un panneau publicitaire (CPS), il s'allume et une publicité est affichée.

En remplaçant E par I et ainsi pouvant connaître les relations entre les deux I, il est possible de cibler et différencier la publicité lorsqu'il s'agit de couples ou d'amis par exemple.

La Figure 3 présente en résumé le diagramme de classe correspondant :

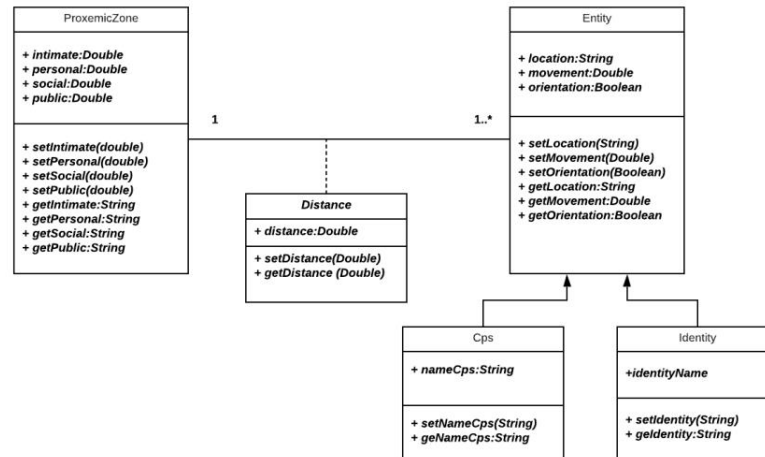


Figure 3 : Diagramme de classes

Dans la section suivante, nous verrons comment se traduit la représentation formelle en un fichier XML de type Schéma Design (XSD).

3.3. XSD pour les environnements proxémiques

A partir des spécifications formelles et de la conception graphique d'un environnement proxémique, il est possible de générer automatiquement un fichier XML de type Schéma Design (XSD). Il contient les informations relatives aux conditions initiales d'un E_P, les dimensions DILMO, les actions et les zones. Le XSD permet de traduire le modèle de conception dans un langage informatique (Daum,2003 ; Machkour,2016). Nous utilisons ensuite le générateur de classes JAXB (Oracle, 2019) qui permet de créer des interfaces et d'implémenter des classes à partir de cet XSD. Notre proposition de DSL graphique constitue un outil qui supporte à la fois les phases de conception et de développement des applications proxémiques.

- Conception de l'environnement proxémique : Cela signifie que les zones proxémiques (Z_P), le CPS, les entités (E et I), l'ensemble des dimensions DILMO à prendre en compte et les conditions initiales sont identifiées. A partir de la définition formelle, il est possible d'établir les règles d'inférence qui peuvent générer des comportements spécifiques et des actions liées aux déplacements des entités dans l'environnement proxémique.
- Modélisation de l'environnement proxémique : la notation graphique E_P est représentée dans un modèle graphique intuitif.

- Modélisation des comportements proxémiques : une fois l'E_P défini et modélisé, il est possible de modéliser les comportements proxémiques dans le temps en fonction des dimensions DILMO considérées pour représenter les actions proxémiques du CPS et des entités.
- Génération du XSD : les informations relatives aux conditions initiales d'un E_P ainsi que les comportements et actions proxémiques sont stockées dans un fichier XML. Le XSD représente chaque composant de l'environnement proxémique tel que le concepteur l'a créé. Les interfaces et classes Java sont ensuite automatiquement générées à partir du XSD. Cette génération de code est basée sur des règles one-to-one (une entité génère une classe, une propriété génère un attribut).
- Implémentation d'une application proxémique : les interfaces et classes générées à l'étape précédente peuvent ensuite être intégrées, et utilisées. Dans la phase suivante de mise en œuvre, les développeurs décident comment les dimensions DILMO sont capturées en fonction de la technologie des capteurs disponibles (par exemple, BLE, caméra, , boussole, etc.) à l'aide de l'API développées et accessible sur : <https://github.com/paulocpd76/DSLGUI003-L>.

4. Un premier prototype du DSL

La figure 4 montre la représentation des conditions initiales de l'E_P décrites dans la section 3.2 à l'aide de notre DSL graphique. Le concepteur précise les distances qui définissent les quatre zones proxémiques, ajoute des entités ou des identités sur chaque zone (représentées dans les carrés noirs), ajuste les dimensions DILMO dans l'icône qui représente chaque entité (dans la zone verte), et spécifie qu'une action est effectuée par le CPS (en appuyant sur le bouton « action »). Le curseur au-dessus de chaque entité permet de simuler les changements de distance entre l'entité et le CPS. D'autres comportements d'entités sont simulés en modifiant leurs dimensions DILMO sur leur icône respective dans la zone verte. Les fichiers XSD et XML sont générés automatiquement pour stocker, lire et modifier les conditions initiales, ainsi que tous les autres comportements simulés dans l'interface graphique. Ce prototype est librement téléchargeable sur github mentionné précédemment.

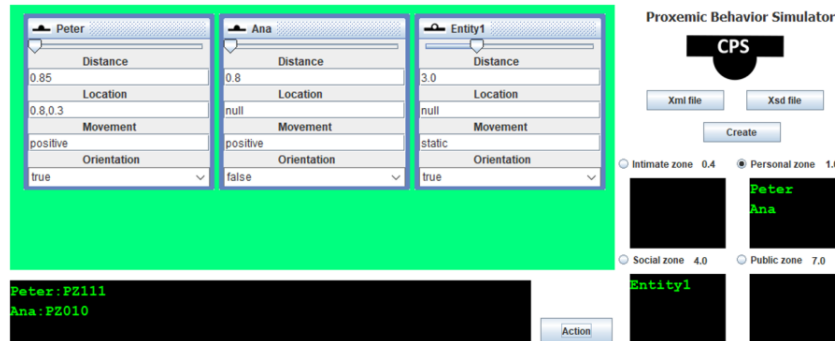


Figure 4 : Modélisation d'un E_P avec DSL

Il est possible de télécharger différents fichiers XML et XSD afin de réaliser des simulations et afin de définir les comportements et les actions souhaités dans un E_P (permet de s'affranchir de la phase de conception interactive). Dans ce premier prototype, l'orientation des entités est représentée comme un paramètre booléen indiquant si l'entité est ou non dans le champ de vision du CPS. Pour la prochaine version, il est prévu d'améliorer et d'affiner cette fonctionnalité en indiquant les angles dans la zone de vision du CPS et l'orientation des entités. Nous prévoyons également d'ajouter tous les éléments graphiques présentés à la section 3.2 et de permettre leur utilisation par simple glisser/déposer.

5. Conclusion

Dans cet article, nous avons présenté une modélisation graphique basée sur un DSL permettant de mieux appréhender et de mieux comprendre les environnements proxémiques. Contrairement aux approches plus traditionnelles des règles de type ECA (Event-Condition-Action) centrée sur l'occurrence d'un événement dans une perspective ensembliste, la conception est ici centrée sur des entités (ici proxémiques), ce qui simplifie le travail d'identification des conditions de déclenchement mais également rend la conception plus intuitive, quitte à la dupliquer pour chaque CPS. A l'aide d'un fichier XSD (fichier XML de type Schéma Design) généré par le DSL, nous générons des classes afin de faciliter le travail in fine des développeurs pour rendre plus rapide et plus simple l'intégration dans le code de la capture des données via les capteurs, et les actions associées aux événements. Ce travail est grandement facilité par l'utilisation d'une API (Perez, 2018), fonctionnant pour dispositifs mobiles équipés du système Android. Nous avons décliné la modélisation graphique en une interface tangible qui incarne les données symboliques par des objets physiques. Cette interface semble naturellement faciliter la collaboration au moment de la conception, par l'appropriation des mouvements dans l'espace, mais aussi la compréhension des symboles grâce à leur incarnation. Nos travaux futurs devront montrer s'il y a un réel apport à la physicalisation du DSL. La méthode de conception et son adéquation à l'API sont (au moment de l'écriture de cet article) en train d'être testées par des élèves ingénieurs et par les développeurs de l'entreprise Dev 1.0, offrant ainsi une évaluation in-vivo du concept et des outils proposés.

Remerciements

Travaux soutenus par la région Nouvelle Aquitaine et la Communauté Pays Basque dans le cadre du projet PISCO, réalisé en collaboration par l'UPPA, l'ESTIA INSTITUTE OF TECHNOLOGY et la société Dev 1.0.

Bibliographie

- Ballendat, T., Marquardt, N., Saul, G. (2010). Proxemic interaction: designing for a proximity and orientation-aware environment. In *acm International Conference on Interactive Tabletops and Surfaces*, p. 121-130
- Bhagya, S., Samarakoon, P., Sirithunge, H.C., Viraj, M., Muthugala, J., Buddhika, A., Jayasekara, P. (2018). Proxemics and approach evaluation by service robot based on user behavior in domestic environment. *International Conference on Intelligent Robots and Systems (IROS)*. IEEE. p. 8192-8199
- Brock, M., Quigley, A., Kristensson, P.O. (2018). Change blindness in proximity-aware mobile interfaces. *CHI conf. on Human Factors in Computing Systems*. p. 1-7
- Brudy, F., Holz, C., Radle, R., Wu, C.J., Houben, S., Klokmose, C.N., Marquardt, N. (2019). Cross-device taxonomy: Survey, opportunities and challenges of interactions spanning across multiple devices. *CHI conf. on Human Factors in Computing Systems*, p. 1-28
- Brudy, F., Suwanwatcharachart, S., Zhang, W., Houben, S., Marquardt, N. (2018). Eagleview: A video analysis tool for visualising and querying spatial interactions of people and devices. *International conf. on Interactive Surfaces and Spaces*. p.61-72
- Cardenas, C., Garcia-Macias, J.A. (2017). Proximithings: Implementing proxemic interactions in the internet of things. *Procedia Computer Science* vol, 113, p. 49-56.
- Cho, Y., Bianchi-Berthouze, N., Marquardt, N., Julier, S.J. (2018). Deep thermal imaging: Proximate material type recognition in the wild through deep learning of spatial surface temperature patterns. *CHI conf. on Human Factors in Computing Systems*. p. 1-13
- Daum, B. (2003) Modeling business objects with XML schema. Morgan Kaufmann, San Francisco, Calif. Heidelberg, Germany
- Dingler, T., Funk, M., Alt, F. (2015). Interaction proxemics: Combining physical spaces for seamless gesture interaction. In *Proc of Internat Symposium on Pervasive Displays*, vol.107, p. 114
- Dostal, J., Hinrichs, U., Kristensson, P.O., Quigley, A. (2014). Spidereyes: designing attention-and proximity-aware collaborative interfaces for wall-sized displays. *International conf. on Intelligent User Interfaces*, p. 143-152
- Dostal, J., Kristensson, P.O., Quigley, A. (2013). Multi-view proxemics: distance and position sensitive interaction. In *the acm Internat Symposium on Pervasive Displays*, p. 1-6.
- Evans, G.W., Lepore, S.J., Allen, K.M. (2000), Cross-cultural differences in tolerance for crowding: Fact or fiction? *Journal of Personality and Social Psychology*, vol. 79, p. 204.
- Garcia-Macias, J.A., Ramos, A.G., Hasimoto-Beltran, R., Hernandez, S.E.P. (2019): Uasisi: a modular and adaptable wearable system to assist the visually impaired. *Procedia Computer Science*, vol. 151, p. 425-430.
- Greenberg, S., Marquardt, N., Ballendat, T., Diaz-Marino, R., Wang, M. (2011). Proxemic interactions: the new ubicomp? *Interactions*, vol. 18, p. 42-50.
- Gørnbk, J.E., Linding, C., Kromann, A., Jensen, T.F.H., Petersen, M.G. (2019). Proxemics play: Exploring the interplay between mobile devices and interiors. In *Companion Publication on Designing Interactive Systems Conference*, p. 177-181

- Gørnbc, J.E., O'Hara, K. (2016). Built-in device orientation sensors for ad-hoc pairing and spatial awareness. In Proc. of Cross-Surface Workshop
- Hall, E.T. (1963). A system for the notation of proxemic behavior. *American Anthropologist*, vol. 65, p.1003-1026.
- Hall, E.T. (1966). *The Hidden Dimension: An anthropologist examines man's use of space in private and public*. New York: Anchor Books; Doubleday & Company, Inc.
- Kim, H.J., Kim, J.W., Nam, T.J. (2016). ministudio: Designers' tool for prototyping ubicomp space with interactive miniature. CHI conf. on Human Factors in Comp. Syst, p. 213-224.
- Ledo, D., Greenberg, S., Marquardt, N., Boring, S. (2015). Proxemic-aware controls: Designing remote controls for ubiquitous computing ecologies. In Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services, p. 187-198
- Machkour, M., Afdel, K. (2016). Transforming xml into object-relational schema. *IOSR Journal of Computer Engineering*, vol. 18, p. 40-52.
- Marquardt, N. (2013). Proxemic interactions in ubiquitous computing ecologies. Ph.D. thesis, University of Calgary.
- Marquardt, N., Diaz-Marino, R., Boring, S., Greenberg, S. (2011). The proximity toolkit: prototyping proxemic interactions in ubiquitous computing ecologies. In Proceedings of the 24th annual ACM symposium on User interface software and technology, p. 315-326.
- Mentis, H.M., O'Hara, K., Sellen, A., Trivedi, R. (2012). Interaction proxemics and image use in neurosurgery. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, p. 927-936.
- Mojgan, G., Marvin, P., Wong, C., Wallace, J.R., Scott, S.D. (2018). Increasing passersby engagement with public large interactive displays: A study of proxemics and conation. International Conference on Interactive Surfaces and Spaces, p. 1-14.
- Oracle: Using the jaxb class generator and jaxb users guide (2019) <https://docs.oracle.com/javase/8/docs/technotes/guides/xml/jaxb/index.html>
- Perez, P., Roose, P., Marc, D., Couture, N., Cardinale, Y., Masson, D. (2018). Proxemics for first aid to unconscious injured person. In Proceedings of the 30th Conference on l'Interaction Homme-Machine, p. 156-162
- Rector, K., Salmon, K., Thornton, D., Joshi, N., Morris, M.R. (2017). Eyes-free art: Exploring proxemic audio interfaces for blind and low vision art engagement. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol, 1, p. 1-21
- Roussel, N., Evans, H., Hansen, H. (2004). Mirrorspace: using proximity as an interface to video-mediated communication. In Int'l Conference on Pervasive Computing, p. 345-350.
- Sørensen, H., Kristensen, M.G., Kjeldskov, J., Skov, M.B. (2013). Proxemic interaction in a multi-room music system. In Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration, p. 153-162.
- Vermeulen, J., Luyten, K., Coninx, K., Marquardt, N., Bird, J. (2015). Proxemic flow: Dynamic peripheral floor visualizations for revealing and mediating large surface interactions. In IFIP Conference on Human-Computer Interaction, p. 107-114.
- Wang, M., Boring, S., Greenberg, S. (2012). International symposium on pervasive displays, p. 1-7.
- Wolf, K., Abdelrahman, Y., Kubitzka, T., Schmidt, A. (2016). Proxemic zones of exhibits and their manipulation using floor projection. In Proceedings of the 5th acm International Symposium on Pervasive Displays, p. 33-37.
- Wolf, K., Lischke, L. (2014). Urban proxemics for public guidance In Proceedings of the NordiCHI Workshop on Ubicomp beyond Devices: People, Objects, Space and Meaning, acm, vol.10 p. 2639189-2654842.