



UNIVERSITE CONSTANTINE 2 ABDELHAMID MEHRI

PROJET DE FIN D'ÉTUDES

Systeme de localisation en intérieur

Seghiri Taki Eddine et Benchetra Hamza

Encadré by
Dr. Belguidoum Meriem.

20 avril 2020

Table des matières

I	État de l’art.	4
1	Étude sur le positionnement en intérieur.	6
1.1	introduction	6
1.2	Définitions et terminologie	6
1.2.1	Géolocalisation par satellites	6
1.2.2	RSSI	7
1.2.3	RTLS	7
1.2.4	GPS	7
1.2.5	Ultra-large bande	7
1.2.6	Bluetooth	7
1.2.7	Wifi	8
1.2.8	Wlan	8
1.2.9	Radio-identification	8
1.2.10	Trilatération	8
1.3	Domaines d’application des systèmes de positionnement en intérieur .	8
1.3.1	Aide à la navigation	8
1.3.2	Sites industriels	8
1.3.3	Supermarchés et Hypermarchés	9
1.3.4	Marketing	9
1.4	Technologies utilisées	9
1.4.1	Les technologies autonomes	9
1.4.2	Les technologies basées sur une infrastructure	11
1.4.3	Synthèse des technologies	13
1.5	Localisation en intérieur par ondes	14
1.5.1	Méthode de mesure des angles d’arrivée	14
1.5.2	Méthode de mesure des temps d’arrivée	15
1.5.3	Méthode de mesure des différences de temps d’arrivée	16
1.5.4	Méthodes basées sur la puissance des signaux reçus	17
1.5.5	Comparatif des techniques	19
1.6	Etude de l’existant	20
1.6.1	L’entreprise française Polestar	20
1.6.2	L’entreprise anglaise Pointr Labs	20
1.6.3	L’aéroport de Londres-Gatwick	20
1.6.4	Hypermarché Carrefour Westfield Euralille Lille, France	21
1.7	Conclusion	21

2	Méthodologie.	22
2.1	Introduction	22
2.2	Diagrammes de SysML	23
2.2.1	Description fonctionnelle	24
2.2.2	Description structurelle	27
2.2.3	Description comportementale	33
2.3	Approche SysML	35
2.4	La relation entre UML et SysML	35
2.5	Synthèse :	36
2.5.1	Le processus unifié UP (Unified Process)	37
3	Outils et matériel	42
3.1	Introduction	42
3.2	Internet des objets (IoT)	42
3.2.1	Types d'objet connectés	44
3.2.2	Domaines d'application de l'IoT :	45
3.2.3	Architecture général d'un system IoT	47
3.2.4	Les plateformes IoT :	48
3.3	Les systèmes embarqués	48
3.3.1	Caractéristiques	49
3.3.2	Systèmes informatiques et Systèmes 574727421embarqués	49
3.3.3	Classification des systèmes embarqués	50
3.3.4	Domaines d'application des systèmes embarqués	50
3.4	Raspberry	50
3.4.1	Composants d'une carte Raspberry	51
3.5	Plateformes et langages :	52
3.5.1	PhpStorm	52
3.5.2	Android Studio	53
3.5.3	PHP	53
3.5.4	Javascript	54
3.5.5	HTML	54
3.5.6	JAVA	55
3.5.7	XML	55
3.5.8	CSS	56
3.5.9	XAMPP	56

Table des figures

1.1	Repère d'étude lié à l'appareil embarqué.	10
1.2	Exemple d'utilisation des signaux Wi-Fi à des fins de localisation. . . .	11
1.3	Configurations possibles dans les systèmes de localisation en intérieur. .	12
1.4	Classification des technologies de positionnement indoor selon l'archi- tecture à déployer.	13
1.5	Synthèse des technologies par ondes radios.	14
1.6	Principe de triangulation à partir de deux antennes réceptrices.	15
1.7	Principe de trilatération utilisé dans la méthode de TOA	16
1.8	Illustration de la méthode de TDoA	17
1.9	Exemple de l'évolution de la force du signal avec la distance (Technolo- gie RFID)	18
1.10	Illustration du principe de <i>Fingerprinting</i>	19
2.1	Diagramme des cas d'utilisation d'un lecteur MP3 (partiel)	24
2.2	Diagramme d'exigence partiel d'un lecteur MP3	26
2.3	Diagramme de contexte d'un lecteur MP3	27
2.4	Diagramme de définition de blocks d'un lecteur MP3	28
2.5	Diagramme de block interne du Block Alimentation (Lecteur MP3) . . .	29
2.6	Diagramme paramétrique d'un lecteur MP3	30
2.7	Diagramme de séquence du cas d'utilisation "Allumer" du lecteur MP3 .	33
2.8	Diagramme état - transition partiel d'un lecteur MP3	35
2.9	35
2.10	Relation entre uml et sysml	36
2.11	UP est itératif	38
2.12	UP est centré sur l'architecture	39
2.13	UP est piloté par les cas d'utilisation	39
2.14	Activité et phases d'UP	41
3.1	Historique de l'IoT.	43
3.2	Evolution de l'internet.	44
3.3	Évolution de l'internet.	51
3.4	Logo PhpStorm.	53
3.5	Logo Android Studio.	53
3.6	Logo PHP.	53
3.7	Logo JavaScript.	54
3.8	Logo HTML.	55
3.9	Logo JAVA.	55
3.10	Logo CSS.	56
3.11	Logo XAMPP.	57

Première partie

État de l'art.

Introduction générale

La localisation géographique d'objets, services ou personnes est devenue un besoin omniprésent dans le quotidien. Évoluant à grande vitesse, les besoins sont toujours plus spécifiques ce qui implique que les techniques et technologies doivent elle aussi évoluer constamment pour satisfaire ces derniers.

De nos jours, le GPS s'impose en *outdoor*, offrant une précision métrique, cependant, dans les espaces clos, cette technologie peine à fournir des résultats fiables à cause des signaux affaiblis voire même bloqués par les obstacles qui les rendent inefficaces au regard des résultats souhaités par les utilisateurs.

Ainsi, avec l'omniprésence d'infrastructures sans fil au sein des espaces clos, de nouvelles technologies de positionnement en intérieur peuvent être déployées pour prendre le relais des technologies existantes, notamment le GPS.

Certains systèmes opèrent dans des lieux publics tels que les musées ou parkings souterrains à des fins de guidage ou suivi de position en temps réel, d'autres dans le commerce, en prenant connaissance des déplacements habituels de l'utilisateur pour créer de l'interactivité en fonction du lieu où il se trouve.

Ces systèmes peuvent également être utilisés dans un but de surveillance. Les technologies de localisation peuvent permettre le suivi de personnes sensibles telles que des enfants ou de personnes âgées (par exemple).

De tels systèmes contribuent aussi au domaine de l'IoT. Les systèmes de positionnement en intérieur permettent de localiser toute entité munie d'un tag et les localiser les uns par rapport aux autres.

La localisation *indoor* soulève donc de nouveaux défis de conception et d'installation. Les systèmes de positionnement, désormais déployés en environnement dynamique, doivent allier robustesse et fiabilité tout en répondant à des exigences de précision, de temps critique, d'efficacité énergétique et de coûts de déploiement.

Chapitre 1

Étude sur le positionnement en intérieur.

1.1 introduction

Dans le passé, pour déterminer sa localisation l'homme utilisait des méthodes primitives telles que le calcul d'angles entre l'horizon et le soleil à midi à l'aide d'outils particuliers, ainsi qu'une horloge pour déterminer la longitude, mais ces techniques n'étaient utilisables qu'à certains moments de la journée et dans des conditions météorologiques adéquates. Après cela, pendant les années 70 les forces militaires ont mis au point un système de localisation reposant sur un ensemble de 32 satellites chacun emmenant un signal unique et en continue à une vitesse constante ce qui fait que chaque signal arrive à une heure différente permettant ainsi de calculer la distance entre l'appareil et le satellite. Cette technique fut mise à la disposition des particuliers aux alentours des années 80 qui l'intégrèrent aux téléphones portables et aux véhicules. La détermination de la position se faisait ensuite à l'aide d'un algorithme de trilatération qui consiste à calculer la position à l'aide de trois distances pour un plan et de quatre pour une sphère (la terre). Avec le temps la localisation avec satellite a évolué au point de fournir une précision à l'ordre de quelques mètres, ce pendant elle n'était efficace qu'en dehors des bâtiments et des espaces clos et c'est ici qu'interviennent les systèmes de localisation en intérieur qui remplacent les systèmes de géolocalisation par satellite dans les milieux fermés.

1.2 Définitions et terminologie

1.2.1 Géolocalisation par satellites

La géolocalisation par satellite est un procédé de localisation qui consiste à émettre des signaux qui sont captés par un terminal qui calcule la latitude la longitude et parfois aussi l'altitude et au moins 4 satellites doivent être visibles du terminal, les exemples les plus connus sont le GPS le système de positionnement universel américain *Navstar* NS-7B où sa variante européenne nommée Galileo.

1.2.2 RSSI

L'indication de la force du signal reçu (en anglais *Received Signal Strength Indication*) est une métrique de radio-localisation, déjà mise à profit dans les systèmes de communication radio tel que les réseaux cellulaires et WLAN, elle s'appuie sur la mesure de la puissance du signal reçu, notée RSS (*Received Signal Strength*) pour décrire la puissance analogique reçue après quantification de cette même puissance par un récepteur réel.(1)

1.2.3 RTLS

RTLS (*Real-time locating systems*) est une technologie de positionnement et d'identification qui a été prouvée dans divers marchés. Elle est utilisée pour identifier et suivre automatiquement des personnes ou des objets en temps réel, en général dans un bâtiment ou une zone limitée. Des tags RFID (*Radio Frequency Identification*) sans fil sont attachés à des objets ou portés par des personnes, et dans la plupart des applications RTLS les points de référence fixés reçoivent des signaux des tags RFID actifs ou passifs pour déterminer leur emplacement(2)

1.2.4 GPS

Le *Global Positioning System* (en français : « Système mondial de positionnement » [littéralement] ou « Géo positionnement par satellite »), originellement connu sous le nom de *Navstar GPS*, est un système de positionnement par satellites appartenant au gouvernement des États-Unis. Mis en place par le département de la Défense des États-Unis à des fins militaires à partir de 1973, le système avec 24 satellites est totalement opérationnel en 1995 et s'ouvre au civil en 2000(3)

1.2.5 Ultra-large bande

La transmission a ultra-large bande (ULB) ou en anglais *ultra Wide Band* (UWB) est une technologie largement utilisée dans les applications de radar et de télédétection et a récemment reçu une grande attention dans les universités et l'industrie pour les applications de communications sans fil. Un système UWB est défini comme tout système radio dont la largeur de bande de 10 dB est supérieure à 25 pour cent de sa fréquence centrale, ou dont la largeur de bande de 10 dB est égale ou supérieure à 1,5 GHz si la fréquence centrale est supérieure à 6 GHz (4)

1.2.6 Bluetooth

Bluetooth est une interface radio universelle dans la bande de fréquences 2,45 GHz qui permet aux appareils électroniques portables de se connecter et de communiquer sans fil via des réseaux ad hoc* à courte portée. Chaque unité peut communiquer simultanément avec jusqu'à sept autres unités par picoréseau (très petit réseau). De plus, chaque unité peut appartenir simultanément à plusieurs piconets (5)

1.2.7 Wifi

le wi-fi ou la fidélité sans fil en anglais (*Wireless Fidelity*) est un ensemble de protocoles de communication qui fonctionne avec des ondes radio dans une bande passante entre [2.4 - 5] GHz , il est destiné à relier des équipements informatiques et de téléphonie mobiles dans un seul réseau haut débit et sans fil (WLAN)

1.2.8 Wlan

Un réseau sans fil (en anglais : *wireless network*) est un réseau informatique numérique qui connecte différents postes ou systèmes entre eux par ondes radio. Il peut être associé à un réseau de télécommunications pour réaliser des interconnexions à distance entre nœuds. (Réseau sans fil)

1.2.9 Radio-identification

La radio-identification, le plus souvent désignée par le sigle RFID (de l'anglais *radio frequency identification*), est une méthode pour mémoriser et récupérer des données à distance en utilisant des marqueurs appelés « radio-étiquettes » (« *RFID tag* » ou « *RFID transponder* » en anglais). (*Radio-identification*)(6)

1.2.10 Trilatération

La trilatération est une méthode mathématique permettant de déterminer la position relative d'un point en utilisant la géométrie des triangles tout comme la triangulation. Mais contrairement à cette dernière, qui utilise les angles et les distances pour positionner un point, la trilatération utilise les distances entre un minimum de deux points de référence.(7)

1.3 Domaines d'application des systèmes de positionnement en intérieur

1.3.1 Aide à la navigation

Dans le cas d'une navigation allant de l'*outdoor* à l'*indoor* (ex : Un client utilisant son application GPS pour se rendre de chez lui à un point précis situé à l'intérieur d'un centre commercial), celle-ci se doit d'être « sans coupures » : le mobile devra immédiatement détecter que l'utilisateur est rentré dans un bâtiment, et passer de manière transparente en mode navigation intérieure.

1.3.2 Sites industriels

Lorsqu'un problème ou une panne est identifiée sur un site industriel, une application de localisation pourrait permettre de guider l'équipe technique mobile la plus proche vers la source du problème dans un minimum de temps. Ensuite, une fonctionnalité d'aide à la résolution de problème type Réalité augmentée ou Télé-assistance pourrait prendre le relais si besoin(8)

1.3.3 Supermarchés et Hypermarchés

Les grandes surfaces comme les supermarchés nécessitent des compétences de localisation pour se situer dans les rayons et de trouver les articles recherchés, un système de localisation dans un tel endroit permettra de montrer aux clients les emplacements des produits ou rayons et faciliter leur expérience d'utilisateur. Le système peut même être utilisé pour des buts publicitaires comme par exemple faire des propositions aux clients selon leurs préférences ou de liquider des produits etc.

1.3.4 Marketing

La principale application du marketing serait le géomarketing : l'optimisation des revenus au mètre carré, en fonction des données de visite des clients (flux, temps de visite, zones de passages, zones de transformation, etc.). Il est possible de personnaliser des offres marketing, non seulement selon les données connues sur le client par la marque (carte de fidélité) mais aussi selon le contexte géographique d'un client en magasin.

1.4 Technologies utilisées

Les technologies mettant à l'œuvre des systèmes de localisation *indoor* se distinguent en deux catégories :

- Des technologies autonomes ne nécessitant aucune infrastructure à déployer, donc sans interaction entre l'utilisateur et des équipements.
- D'un autre côté, celles reposant sur une infrastructure d'équipements inter-connectés.

1.4.1 Les technologies autonomes

-Capteurs inertiels et *dead reckoning* :

Ces premiers systèmes de positionnement s'appuient sur une méthode appelée de « *dead reckoning* »(10) , qui consiste à déduire la position actuelle à partir de la position précédente.

Le système exploite donc les capteurs inertiels (accéléromètres, gyroscopes, etc.) parfois utilisés en combinaison avec la boussole ou le podomètre intégré de l'appareil mobile. Ces différents dispositifs permettent de calculer les déplacements dans l'environnement. L'utilisateur doit disposer d'un *smartphone* ou d'une carte embarquée contenant des capteurs inertiels et le plan du bâtiment doit être connue, aussi, il est nécessaire de connaître l'emplacement initial de l'utilisateur.

Les données de déplacement recueilli pendant le mouvement du porteur du dispositif permet de déterminer la nouvelle position qui calculé de manière relative, la position à l'instant T-1 permettant de déterminer celle à l'instant T. Une communication entre la carte ou l'appareil mobile peut entrer en jeu a fin de centraliser les données et réduire la charge de calcul sur le terminal

Atouts et faiblesse : -Ces systèmes ont l'avantage d'être pratiquement autonomes en présences des conditions adéquates. Mais leur précision de positionnement reste toutefois limitée à causes des erreurs cumulées au fil des déplacements.

-Les capteurs inertiels sont généralement utilisés en combinaison avec d'autres technologies, selon le principe de fusion des données.

Étude des champs magnétiques

Cette technique se base sur des capteurs a effet Hall afin de mesurer l'intensité des champs magnétiques(11) terrestres présents au sein des espaces clos, champs magnétiques dont l'intensité diffère d'un emplacement à un autres à causes des perturbations causées par les structures métalliques composants le bâtiment. En enregistrant les variations de ces derniers, une détermination de la position est possible

Bien qu'elle soit indépendante de toute interaction avec des périphériques externes, cette technique de localisation nécessite toutefois une calibration initiale en plus de l'acquisition d'informations spécifiques à l'environnement de déplacement

Deux approches sont donc possibles :

- pour chaque position du champs , l'utilisation de ses valeurs mesurées sur différents axes

- La reconnaissance de patterns représentatifs d'un élément

La première démarche s'appuie sur le principe de « *Fingerprinting* ». Qui consiste a enregistrer les différentes intensités du champ magnétique sur les axes de déplacements (x, y, z) dans différents points de l'environnement, qui soit dit en passant doit être cartographier au préalable, grâce au capteur a effet Hall et sauvegarder les valeurs recueilli dans une base de données.



FIGURE 1.1 – Repère d'étude lié à l'appareil embarqué.

ensuite, a la phase de déplacement libre , le dispositif (téléphone par exemple) compare les valeurs actuelles avec les valeurs qui se trouve dans la base de données précédemment constituée. Il repère alors l'utilisateur à la position correspondante.

Atouts et faiblesses : Bien que cette technique ne soit pas très populaire, la localisation via champ magnétique prometteuse et pas très couteuse. Elle offre une précision jusqu'à maintenant inférieure à un mètre, et tout ça, sans l'exigence d'une infrastructure.

Toutefois l'utilisation d'une telle technologie se limite à des situations bien spécifiques. En effet, pour pouvoir l'utiliser, les déplacements de l'utilisateur doivent être

prévisibles et former un pattern plus ou moins précis par exemple un supermarché avec des rayons ou un local avec des couloirs étroits.

De fait, la mobilité d'usager doit être limitée, sous soucis d'avoir une complexité élevée

1.4.2 Les technologies basées sur une infrastructure

Les techniques d'imagerie

Cette méthode consiste à équiper l'environnement de capteurs visuels comme des caméras, dans des positions fixes et connues et qui identifieront les objets à localiser, elle repose sur le concept d'analyse environnementale afin d'extraire les anomalies.

Atouts et faiblesses : Les systèmes d'imagerie atteignent souvent une très bonne précision en espace clos, de plus ils ne nécessitent aucun dispositif rattaché à l'utilisateur, ce qui constitue un avantage certain (le système est dit « *device free* »).

Toute fois elles sont limitées en ce qui concerne le champ de vision, et en présence d'un nombre importants d'individus, il devient plus difficile de les identifier, bien sur des solutions existent, mais cela requiert des capacités de calculs élevées, venant augmenter les coûts.

L'exploitation des ondes radio

Avec leurs caractéristiques impressionnantes, les ondes radio-fréquences peuvent traverser des cloisons, murs ou tout autre obstacle pouvant être présent au sein d'un bâtiment ou n'importe quel espace clôt. En plus, leur omniprésence au quotidien fait d'eux la base de nombreuses technologies de positionnement.

Plusieurs standards, et que l'on va citer dans ce qui suit, existent et chacun s'adapte à de nombreux cas d'utilisations. En général, l'architecture utilisée consiste en un ensemble de *tag* attachés aux personnes ou aux objets à localiser et d'un ensemble d'antennes déployer dans l'espace clos.

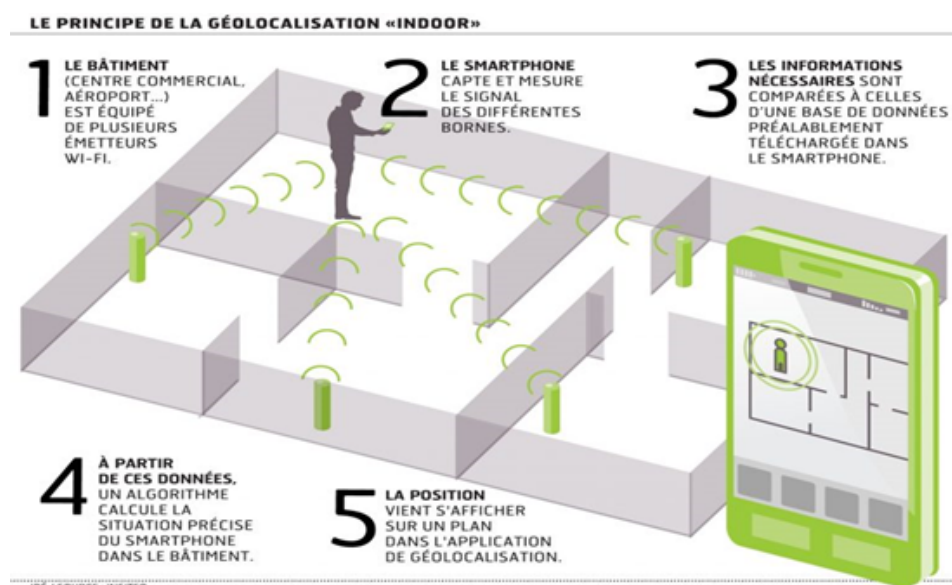


FIGURE 1.2 – Exemple d'utilisation des signaux Wi-Fi à des fins de localisation.

Deux cas d'architecture sont alors possible : -Soit le dispositif (*tag*) émet et les capteurs (antennes) sont à l'écoute
-Soit chaque capteur (antenne) émet à destination du dispositif (*tag*), qui se charge de traiter l'information

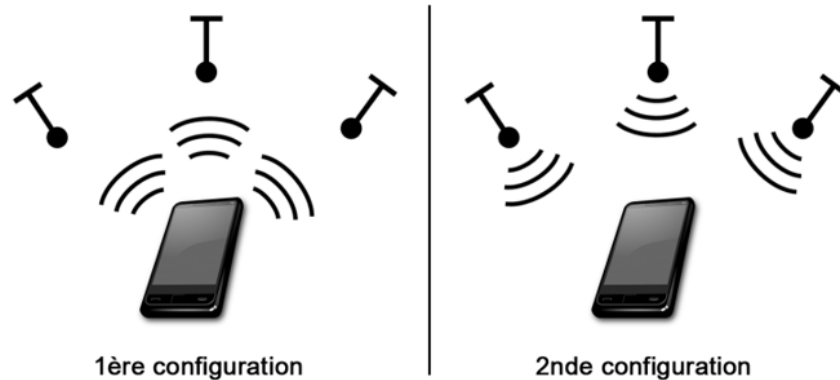


FIGURE 1.3 – Configurations possibles dans les systèmes de localisation en intérieur.

-Chacune de ces configurations offre plusieurs avantages :

La première permet de réduire les éléments actifs au sein du système. En effet, il n'y a que le dispositif (*tag*) qui est chargé d'émettre les signaux vers les antennes environnantes, ce qui facilite énormément la synchronisation. Après réception du signal les antennes communiquent avec le serveur chargé du traitement algorithmique.

Grâce à cette approche, les ressources chargées du calcul nécessaires au niveau du dispositif (*tag*) sont minimisées, la position de l'utilisateur est par contre rendue accessible pour tous les utilisateurs du serveur.

La deuxième configuration concentre tout le traitement au niveau de l'équipement à localiser, cela permet une position à temps réel sans avoir à un accès secondaire et donc l'utilisation d'une connexion réseau n'est plus utile. En pratique le choix de la solution est à la charge du concepteur, selon l'utilisation souhaitée et la technologie support. En somme, la figure suivante synthétise la classification du chapitre :

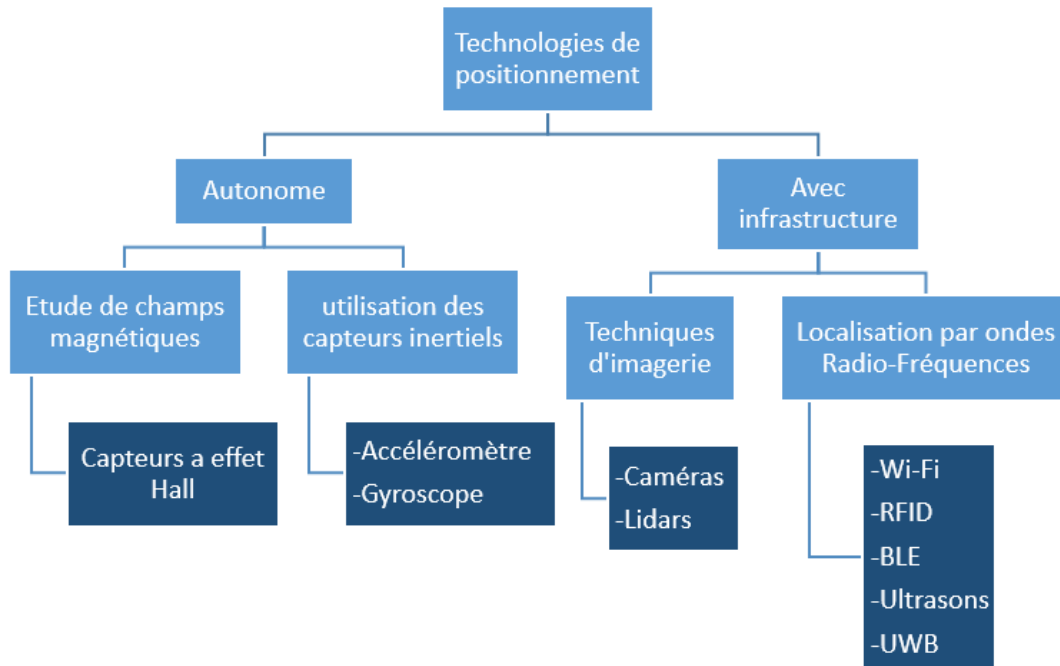


FIGURE 1.4 – Classification des technologies de positionnement indoor selon l'architecture à déployer.

La majorité des systèmes de localisation en intérieur utilisent des ondes radio comme moyen de communication. La transmission se fait entre un dispositif (*tag*) mobile et un réseau d'antennes fixés dans les angles d'endroit étudié.

selon la nature de l'information utile, les algorithmes de positionnement se subdivisent en plusieurs catégories, chacun d'eux s'adapte à : "la technologie supportée et l'espace ciblé".

1.4.3 Synthèse des technologies

Les forces et faiblesses des technologies précédentes peuvent être synthétisées à travers les tableaux suivants, le premier pour **les technologies autonomes** et le second pour **les technologies basées sur les ondes**.

Les paramètres de exactitude et de portée sont en comparaison. Ces indices sont dépendants de la densité de l'espace étudié, la nature des matériaux, ...)

<i>Technologie support</i>	<i>Principe de positionnement</i>	<i>Avantages</i>	<i>Inconvénients</i>	<i>Cas d'utilisation courants</i>
<i>Points d'accès wifi</i>	-Fingerprinting	-Grande portée - Infrastructure déjà existante -Chips wifi très répandus	-Fluctuation de la puissance des signaux -Faible précision (de 2 à 5 mètres seulement)	-Couverture de grands espaces Localisation sur plusieurs étages
<i>Bluetooth Low Energy (BLE)</i>	-Détection de cellule possiblement couplée à du RSS	-très faible consommation énergétique -technologie Bluetooth très répandue	-Portée limitée (quelques dizaines de mètres)	-Geofencing
<i>RFID</i>	-Détection de cellule **possiblement couplée à du RSS	-très faible consommation énergétique	-Faible portée (<10m)	-Contrôle d'accès surveillance Identification et localisation de biens
<i>UWB</i>	-TOA -TWR	-Excellente précision (dizaine de centimètres) -Pas d'interférences avec les autres signaux -Bonne pénétration a travers les matériaux -Très faible consommation énergétique.	-Portée limitée (<15m selon la norme) -Possibles interférences avec les milieux liquides ou métalliques	-Applications industrielles -Logistique
<i>Ultrasons</i>	-TOA -Tdoa	-précision d'ordre centimétrique -Faible consommation énergétique	-LOS* requise -Très faible portée (quelques mètres)	-Intégration a la robotique -Espaces très réduits

FIGURE 1.5 – Synthèse des technologies par ondes radios.

1.5 Localisation en intérieur par ondes

Étant donné notre intérêt a la géolocalisation *indoor* par ondes radio-fréquence nous présenterons dans cette section ses divers principes et technologies. Une comparaison des différentes technologies sera par la suite effectuée, permettant ainsi de préciser leurs différences en terme de caractéristiques et les cas d'utilisation courants. Dans cette architecture les systèmes seront composée d'un dispositif mobile (*tag*) à localiser, ainsi qu'un réseau d'antennes, déployées au sein de l'espace étudié à des positions connues et préalablement enregistrées dans la base de données.

1.5.1 Méthode de mesure des angles d'arrivée

Cette méthode nommé "l'angle d'arrivée (*Angle Of Arrival* : AoA)" utilise les angles des signaux émis par le dispositif mobile vers deux antennes environnantes au minimum. Après détermination de ces angles, la position du dispositif est obtenue en intersectant les droites passantes par chaque point de référence ce qui est appelé **la triangulation**

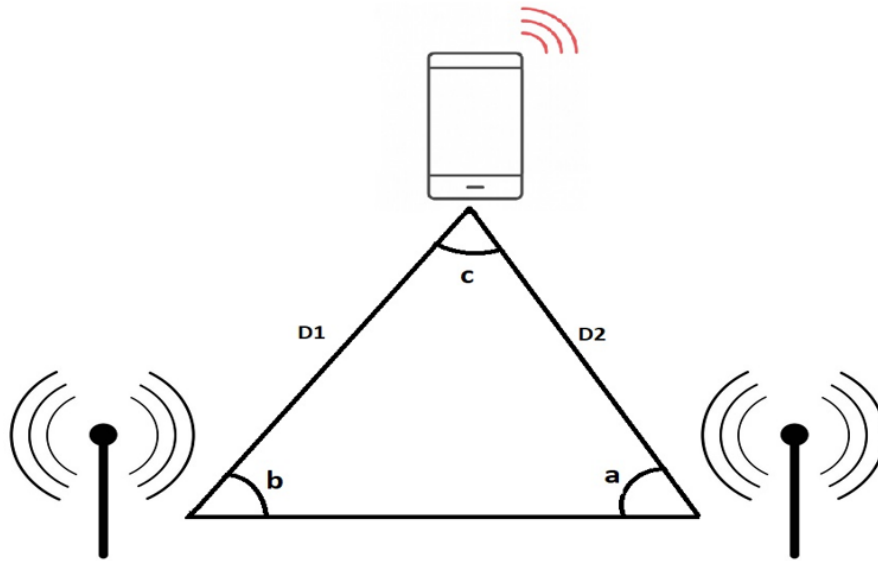


FIGURE 1.6 – Principe de triangulation à partir de deux antennes réceptrices.

Le principal avantage de ce processus est qu'il ne nécessite que deux antennes de référence pour qu'il détermine une position en espace 2D. Mais reste incapable lorsqu'il s'agit d'un dispositif émetteur qui se trouve dans la même ligne qu'un récepteur

1.5.2 Méthode de mesure des temps d'arrivée

Cette technique se base sur le temps de propagation d'un signal, entre l'émission par le dispositif (*tag*) mobile au moment T_i et la réception au sein du capteur (antenne) au moment T_f . Ces deux instants justifient la nomination de la technique, appelée "*Time Of Arrival*" : ToA.

Ce temps de propagation est lié à une distance d tel que $d = c * (T_f - T_i)$,

Pour chaque capteur, une position est préalablement connue, l'utilisateur se repère alors quelque part sur le cercle de rayon " d " et le point du centre " c " où est situé le capteur (antenne) en question.

Cette opération est faite pour trois antennes au minimum, ce qui permet d'obtenir une position en espace bi-dimensionnelle (à l'intersection des 3 cercles construits). c'est la « **trilatération** ».

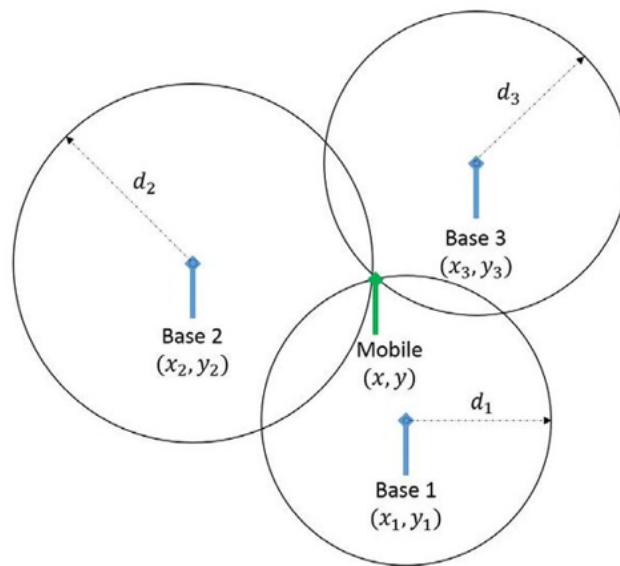


FIGURE 1.7 – Principe de trilatération utilisé dans la méthode de TOA

l'inconvénient majeur de cette technique est que la synchronisation entre les nœuds doit être extrêmement précise, ainsi que l'horodatage exact doit être transmis à toutes les antennes du réseau une fois un dispositif (*tag*) émet une trame.

1.5.3 Méthode de mesure des différences de temps d'arrivée

En anglais *Time difference of arrival* (TDOA) et son fonctionnement se résume en ces étapes :

- Un capteur (antenne) maître est en charge de partager son horloge aux autres récepteurs qui se trouvent à son voisinage.

- Ensuite le dispositif (*tag*) émet régulièrement de courts messages diffusés en *broadcast*. La différence entre les temps d'arrivée du signal au paire de capteurs introduit une différence de distance entre l'utilisateur et cet même paire. Celui-ci se situe alors sur une hyperbole ayant pour foyer les deux récepteurs. Sa position exacte est finalement obtenue après répétition du processus, à l'intersection des hyperboles construites comme la figure ci-dessous l'indique

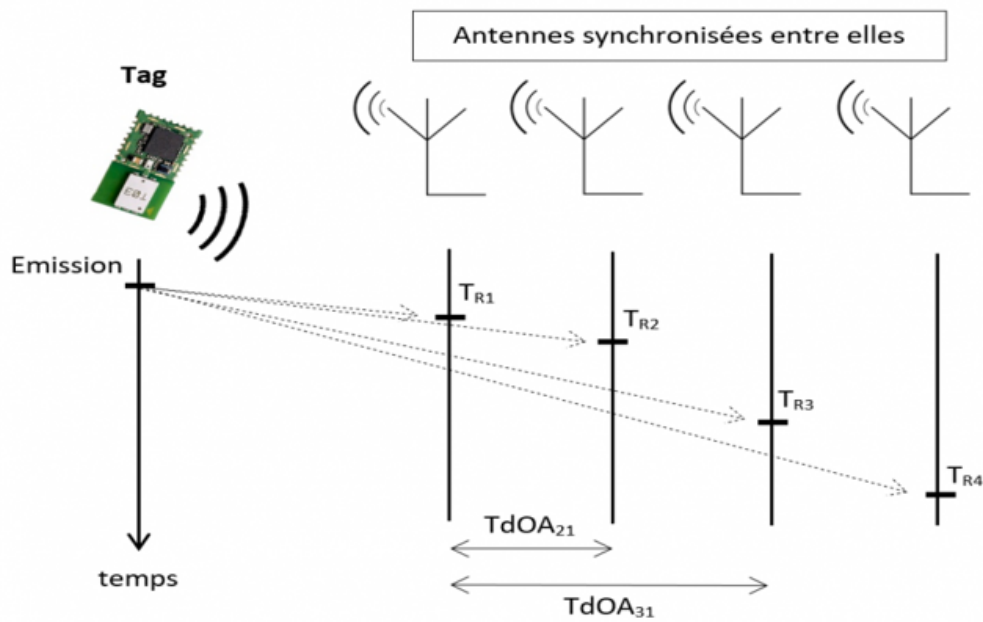


FIGURE 1.8 – Illustration de la méthode de TDoA

1.5.4 Méthodes basées sur la puissance des signaux reçus

Ces méthodes se basent sur la puissance du signal reçu (soit au niveau du dispositif cible (*tag*) soit au niveau des nœuds de référence (antennes fixes), selon le type d'émetteur. Les algorithmes qui se basent sur la RSS (*Received Signal Strength*) se subdivisent en deux catégories.

- ceux qui se basent sur une correspondance directe **Puissance = fonction(distance)**.

La puissance mesurée, au niveau du dispositif (*tag*) par exemple, est dite représentative de la distance des antennes de référence, leur position est à présent connue. Une fois ces distances sont tenues en compte pour un minimum de trois antennes, le système utilise le principe de **trilatération** afin de marquer la position exacte du dispositif mobile. Il existe plusieurs équations mathématiques qui permettent de modéliser le comportement du canal, en quantifiant la puissance du signal en fonction de la distance parcourue. L'évolution est souvent logarithmique.

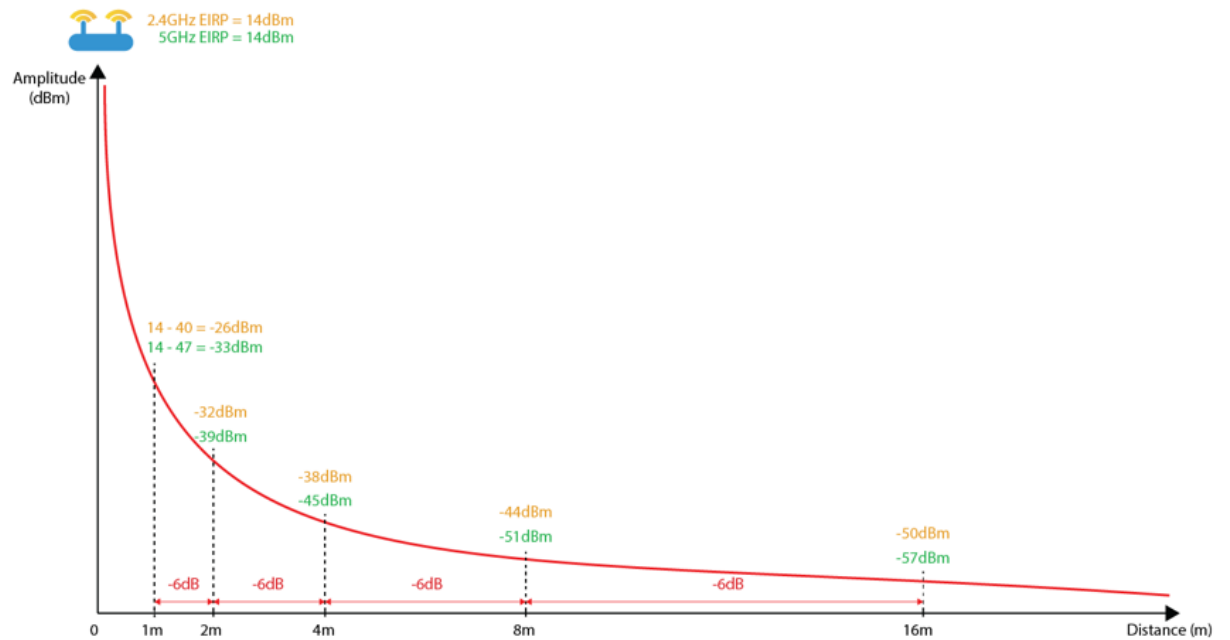


FIGURE 1.9 – Exemple de l'évolution de la force du signal avec la distance (Technologie RFID)

- cette catégorie nécessite la création d'un jeu de donnée lors d'initialisation, nommé le « *Fingerprinting* ». la base de données est constituée en amont de la localisation et contient tous les relevés de puissance pour des positions fixes dans l'environnement étudié, en supposant que les puissances de N *tag* émetteur peuvent être observées à la position (x,y) , ces informations sont sauvegardées et forment une empreinte '*Fingerprint*' :

$$RSS(x, y) = \begin{bmatrix} RSS_1(x, y) \\ RSS_2(x, y) \\ \vdots \\ RSS_N(x, y) \end{bmatrix}$$

Ce processus se répète régulièrement ce qui permet d'établir une cartographie de l'environnement.

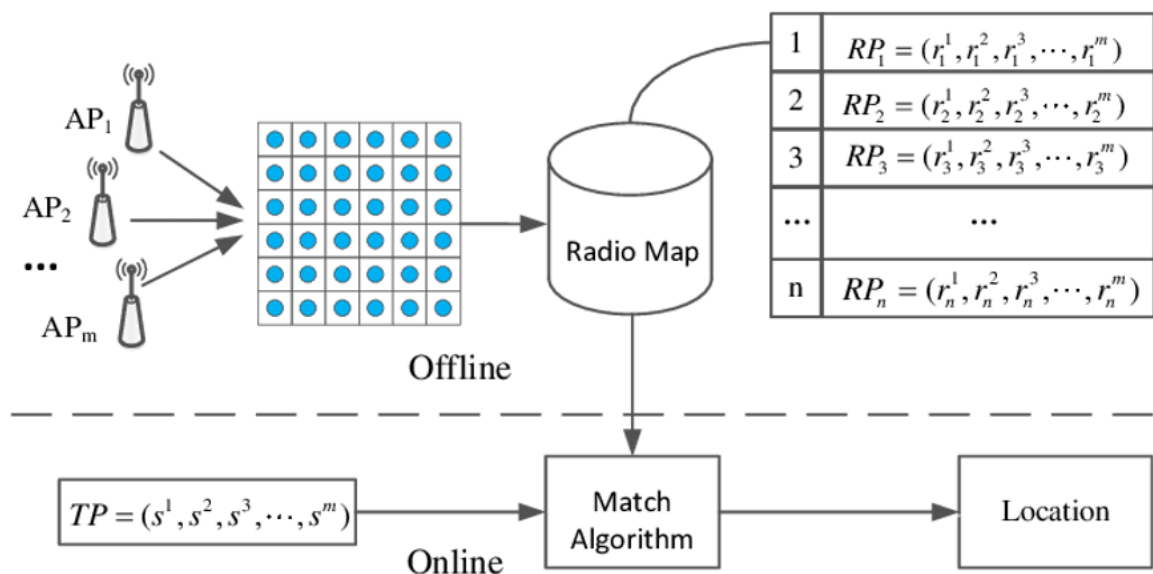


FIGURE 1.10 – Illustration du principe de *Fingerprinting*

Lors du passage suivant à la période de localisation, l'appareil interroge la base de donnée pour affecter l'empreinte observée à une position dans l'environnement. La précision du positionnement est en relation directe avec le nombre de positions enregistrées et le nombre d'antennes déployées.

Cette seconde méthode de cartographie a pour mérite de tenir compte des spécificités du milieu. Les puissances observées reflètent les atténuations des murs ou du mobilier. La construction préalable de la base de données permet alors de répertorier les singularités de chaque position. Ainsi le processus offre usuellement de meilleurs résultats en milieu dense, ce qui justifie son importante utilisation. A l'inverse, une simple corrélation distance/puissance est plus susceptible d'engendrer de lourdes imprécisions.

1.5.5 Comparatif des techniques

Les méthodes précédemment décrites admettent des atouts et faiblesses justifiant leur emploi selon le contexte. Ces caractéristiques sont résumées dans le tableau suivant :

<i>Technologie support</i>	<i>Principe de positionnement</i>	<i>Avantages</i>	<i>Inconvénients</i>
<i>Angle d'arrivée (AOA)</i>	Triangulation à partir des angles de réception.	-Moins de stations fixes requises. -Algorithme de positionnement simple. -Pas de synchronisation d'horloges nécessaire.	-Exige un large et coûteux réseau d'antennes -Même de petits écarts d'angles peuvent affecter les calculs. -Sensible aux phénomènes de trajets multiples.
<i>Temps d'arrivée (TOA)</i>	Trilatération (distances déduites des temps de propagation des signaux).	-Algorithme de positionnement simple. -Bonne précision si résolution temporelle suffisante.	-Synchronisation extrême entre le mobile et les stations fixes. -Nécessite un trajet en ligne directe.
<i>Le décalage horaire d'arrivée (Tdoa)</i>	Trilatération (distances déduites des différences de temps d'arrivée)	- La synchronisation de l'émetteur n'est plus nécessaire.	-Synchronisation entre la station fixe toujours nécessaire. -Nécessite un trajet en ligne directe.
<i>Puissance du signal reçu (RSS) model mathématique</i>	Trilatération (distance = f(puissance))	-Algorithme de positionnement simple. - Existence de modèles mathématiques d'atténuations.	-Faible précision en milieu dense. -Courtes distance entre les nœuds pour garder la puissance du signal.
<i>Puissance du signal reçu (RSS) Fingerprinting</i>	Reconnaissance d'empreintes à partir d'une BDD.	-Prise en compte des caractéristiques du milieu. -Adapté aux milieux denses.	-Longue phase offline avec MàJ en cas de changements du milieu. -Empreintes parfois non uniques si il y a peu de balises.

1.6 Etude de l'existant

1.6.1 L'entreprise française Polestar

Cette entreprise offre des services de localisation en intérieur en utilisant les réseaux sans fil disponibles à l'intérieur des bâtiments tel que le wifi ou le Bluetooth, elle combine aussi l'envoi d'information ciblées avec ses systèmes.

1.6.2 L'entreprise anglaise Pointr Labs

Cette entreprise utilise une approche de deep learning pour raffiner ses systèmes de localisation en intérieur, elle utilise aussi la technologie Li-Fi qui consiste à transférer les données via la lumière, une technologie précise mais gourmande en terme de calcul et d'argent.

1.6.3 L'aéroport de Londres-Gatwick

« En 2017, L'aéroport Gatwick près de Londres en Angleterre a installé un système de « GPS intérieur » pour permettre aux voyageurs de trouver leur chemin dans ses terminaux grâce à leur téléphone multifonction. Environ 2000 balises communiquant en Bluetooth ont été installées au début de l'année sur une période de trois mois permettent ainsi d'indiquer une position géographique, à plus au moins trois mètres

près. En comparaison, les points d'accès Wi-Fi, qui pourraient servir à localiser un téléphone, offrent une précision moindre, d'entre cinq et quinze mètres.(12)

1.6.4 Hypermarché Carrefour Westfield Euralille Lille, France

L'hypermarché Carrefour Euralille a équipé ses locaux d'un système de géolocalisation interne à base de lumière pour mieux orienter les clients. Pour mettre en œuvre ce système, il a fallu équiper le magasin de 7 800 M² de 2,5 km de rails de LED. Il y a près de 800 luminaires dans le magasin, ce système fonctionne comme suite :

Les LED font office de balises de localisation. Elles sont entreposées d'une manière dense et précise. Chaque LED représente une localisation, et peut communiquer sa position à l'application mobile de l'utilisateur. Les diodes émettent des fréquences de rayonnement qui sont détectées par les caméras des smartphones. Le décodage des signaux des signaux se fait de façon binaire, si le LED est allumé c'est un 1 s'il est éteint c'est un 0, et se base sur le principe MIMO (Multiple Input, Multiple Output) qui permet de transférer des sur une grande distance et plus rapidement à travers des antennes d'un réseau de communication sans fil.

Un tel système à l'avantage de pouvoir utiliser un système d'éclairage à base de LED déjà présent au sein de l'infrastructure.

1.7 Conclusion

Les travaux cités précédemment sont des exemples de travaux liés au sujet de notre mémoire chacun d'entre eux utilise une technologie différente et adaptée à son contexte. Dans notre cas, nous allons mettre en place un système standard pouvant être utilisé dans n'importe quel infrastructure du moment où l'en a accès à la cartographie de cette dernière. Nous allons maitre au point un système de localisation qui combine à la fois technologie autonome et la technologie basées sur les ondes wifi, et ce, pour améliorer au maximum la précision de notre système et l'algorithme de trilatération vu que c'est l'algorithme le plus utilisé dans le domaine de localisation que ce soit en interne ou en externe et aussi a cause de sa simplicité et efficacité.

Chapitre 2

Méthodologie.

2.1 Introduction

D'abord pour entamer un projet on doit établir un plan de travail ainsi que sa conception, dans le monde informatique, les processus de travail définissent plusieurs langages de conceptions, d'où on a choisi le processus UP comme processus de développement qui sera défini par le langage SysML. Les méthodes de l'Ingénierie Système (IS) reposent sur des approches de modélisation et de simulation pour valider les exigences, et pour vérifier ou évaluer le système. La modélisation a donc couramment été utilisée pour l'IS, que ce soit pour des représentations concrètes avec des plans ou modèles réduits, ou plus abstraites avec des systèmes d'équations.

En général l'IS tend à modéliser les aspects suivants du système : décomposition fonctionnelle, flux de données, et décomposition structurelle. Exemples de techniques de modélisation employées :

Le diagramme de flux de données (DFD ou Data Flow Diagram) pour définir les données traversant un système et leurs traitements éventuels

Le diagramme de flux fonctionnel de bloc (FFBD ou Functional Flow Block Diagram) proche du diagramme UML d'activité ou du "flowchart".

La modélisation avec le langage UML est une pratique bien établie dans l'industrie logicielle. Bien que le langage UML permette par son caractère à usage général d'adresser de nombreux besoins pour l'IS, il est nécessaire d'adapter ce langage de modélisation par la définition de "profils UML".

Le besoin de définir un langage basé sur UML pour l'IS a été initié en 2001 par l'organisation internationale de l'ingénierie système INCOSE (International Council on Systems Engineering).

SysML (Systems Modeling Language) est basé sur UML et remplace la modélisation de classes et d'objets par la modélisation de blocs pour un vocabulaire plus adapté à l'Ingénierie Système. Un bloc englobe tout concept logiciel, matériel, données, processus, et même la gestion des personnes.

Comme représenté sur le diagramme suivant, SysML réutilise une partie d'UML, et apporte également ses propres définitions (extensions SysML) : 4 structurels, 4 dynamiques, et le diagramme d'exigences.

Structurels :

1. Le " Block Definition Diagram " (BDD) remplace le diagramme de classes
2. L' "Internal Block Diagram" (IBD) remplace le diagramme de structure composite
3. Le diagramme de paquetage reste inchangé
4. Le diagramme paramétrique est une extension SysML pour l'analyse de paramètres critiques
5. du système.

Dynamiques :

1. Le diagramme d'activités est légèrement modifié pour SysML
2. Les diagrammes de séquence, d'états, et de cas d'utilisations restent inchangés
3. Le diagramme d'exigences est une extension SysML

Ainsi SysML permet de fournir un référentiel central qui supporte les analyses du système requises

par l'IS, à savoir la décomposition fonctionnelle, les flux de données, et la décomposition structurelle.

Chaque diagramme est nommé d'une façon bien précise et constitue un élément nommé du modèle.

Pour cela SysML définit une en-tête standard à chaque diagramme qui contient obligatoirement :

De plus, chaque diagramme dispose d'une description :

1. Version.
2. Description.
3. Statut ou niveau d'avancement.
4. Référence, etc...

2.2 Diagrammes de SysML

Le langage sysML propose 9 types de diagrammes destinés à représenter les aspects fonctionnel, structurel et comportemental d'un système.

Pour modéliser l'aspect fonctionnel :

1. Le diagramme des cas d'utilisation (UCD)
2. Le diagramme des exigences (RD)

Pour représenter l'aspect structurel :

1. Du diagramme de définition de blocks (BDD)
2. Du diagramme de block interne (IBD)

Pour modéliser l'aspect comportemental :

1. Le diagramme de séquence (SD)
2. Le diagramme d'état (STM)

2.2.1 Description fonctionnelle

Trois diagrammes interviennent dans la modélisation fonctionnelle d'un système.

Le diagramme des cas d'utilisation (UCD) :

Tout comme avec UML, on utilise les diagrammes de cas d'utilisations :

1. Basé sur les interactions acteurs/système, pour identifier les acteurs et les cas d'utilisations d'un point de vue utilisation du système
2. Pour choisir et identifier les frontières, le périmètre fonctionnel du système

Une description textuelle pour chacun des cas d'utilisations peut être rédigée avec les pre conditions, post conditions, scénario nominal, scénarios alternatifs et d'erreur.

1. Il délimite la frontière entre ceux qui interagissent avec le système, les acteurs (humains, systèmes, flux d'énergie, de matière, etc.), et le système lui-même ;
2. Les cas d'utilisation présentent de façon organisée les fonctionnalités métier attendues du point de vue de l'utilisateur final ;
3. Il associe les cas d'utilisation aux acteurs concernés.
4. Chaque cas d'utilisation fait l'objet d'une description soit textuel, soit par un diagramme comportemental (diagramme d'état ou de séquence).

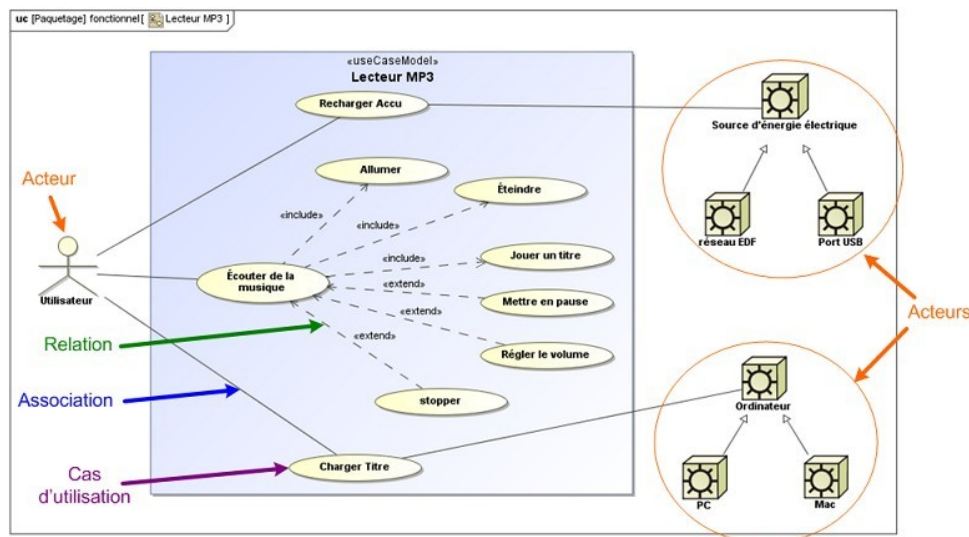
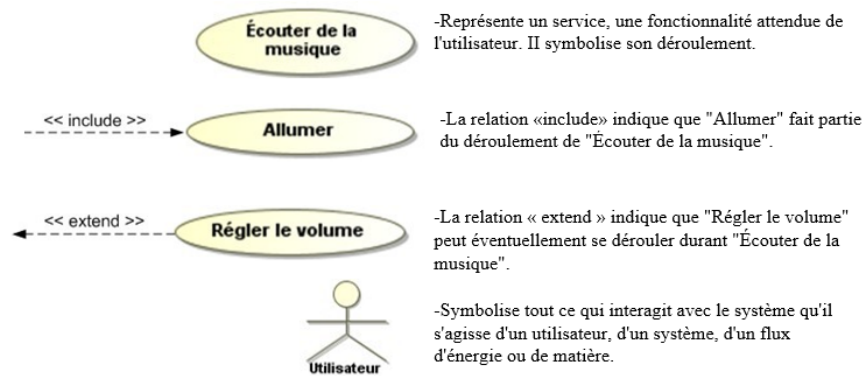


FIGURE 2.1 – Diagramme des cas d'utilisation d'un lecteur MP3 (partiel)



Un cas d'utilisation représente les interactions entre un acteur et le système d'un point de vue « boîte noire », et comprend l'ensemble des scénarios identifiés (nominal, alternatifs, et d'erreurs). Ces scénarios peuvent être détaillés par une description textuelle et/ou par un diagramme de séquence.

Le diagramme des exigences

Que ce soit pour l'ingénierie système ou pour des réalisations logicielles, les exigences sont couramment utilisées pour formaliser les prérequis du système, se traduisant par des fonctionnalités ou conditions qui doivent ou devraient être satisfaites par le système (selon les éventuelles priorités associées aux exigences). Pour la maîtrise d'ouvrage (MOA), les exigences ont pour objectif d'assurer l'adéquation de la solution (le système réalisé) avec les besoins. Les exigences peuvent être formalisées et catégorisées par centre d'intérêts ou aspects, par exemple différenciant les exigences fonctionnelles des exigences techniques (performance, fiabilité, ergonomie, etc.).

La formalisation des exigences peut être effectuée avec une feuille Excel, ou avec un outil spécialisé tel que DOORS ou CaliberRM. L'intérêt qu'offrent ces outils est la gestion des exigences dans une organisation structurée. Les exigences sont également utilisées pour la modélisation, par la création d'associations entre exigences et cas d'utilisations, blocs ou tout type d'élément du modèle, établissant une traçabilité. Il est possible avec l'outil Enterprise Architect de définir les exigences ou dès les importer depuis un outil tel que DOORS, et de les associer avec les éléments du modèle.

SysML formalise les exigences et leur représentation, s'inspirant des fonctionnalités des outils actuellement disponibles sur le marché, par exemple EA, DOORS, CaliberRM. Ainsi SysML définit une représentation graphique et visuelle des exigences textuelles, permet une organisation hiérarchique et l'association avec les éléments du modèle.

SysML définit de nouveaux types de d'associations (liens de dépendance stéréotypés) :

1. Derive : une ou plusieurs exigences sont dérivées d'une exigence
2. Satisfy : un ou plusieurs éléments du modèle (par exemple un bloc) permettent de satisfaire une exigence

3. Verify : un ou plusieurs éléments du modèle (par exemple un « test case ») permettent de vérifier et valider une exigence
4. Refine : un ou plusieurs éléments du modèle, par exemple un cas d'utilisation, redéfinit une exigence
5. SysML définit de nouveaux commentaires stéréotypés permettant d'associer une explication à des associations ou éléments du modèle :
6. Problem : commentaire dont la description pose le problème ou le besoin qui a donné lieu à la création de l'association ou de l'élément associé
7. Rationale : commentaire dont la description indique la raison ou la justification par rapport à l'élément ou l'association associé

Exemple d'associations et de commentaires des spécifications officielles de SysML (OMG) :

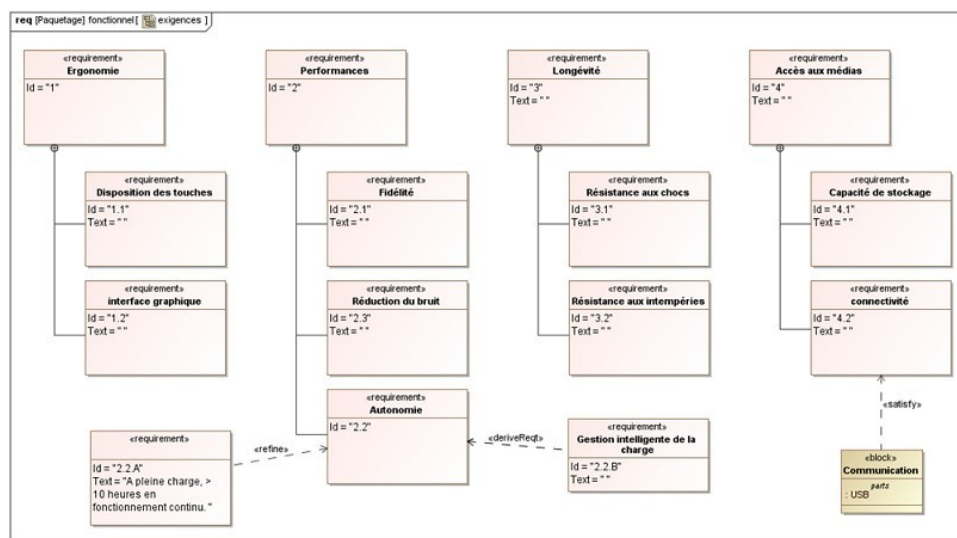
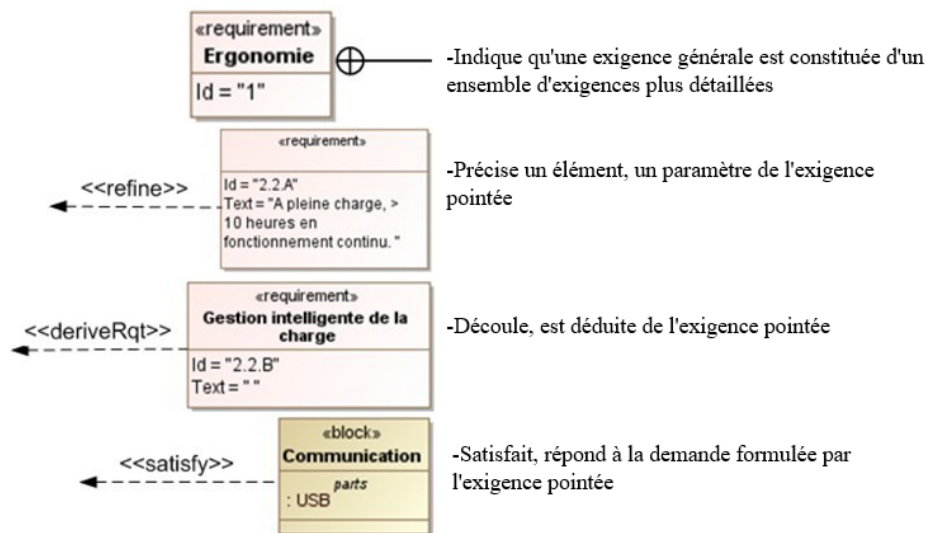


FIGURE 2.2 – Diagramme d'exigence partiel d'un lecteur MP3



Le diagramme de contexte

Il complète éventuellement la description fonctionnelle en présentant tous les éléments externes qui influencent le système étudié et le système lui-même.

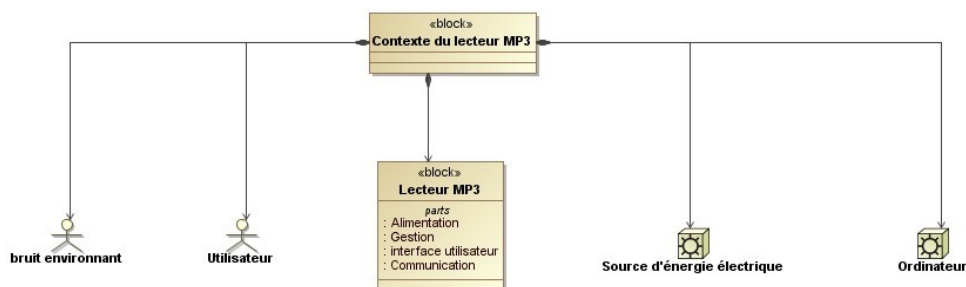


FIGURE 2.3 – Diagramme de contexte d'un lecteur MP3

2.2.2 Description structurelle

Le langage sysML propose 2 diagrammes destinés à décrire la composition du système.

Le diagramme de définition des blocks (BDD)

Le diagramme de définition de bloc (BDD, ou Block Definition Diagram en anglais) représente la vue boîte noire d'un bloc. Ainsi le bloc principal et la hiérarchie des blocs qui le composent, qu'ils soient logiciels ou matériels, sont spécifiés dans ce diagramme.

Par rapport à UML, le BDD de SysML redéfinit le diagramme de classe en remplaçant les classes par des blocs.

Le BDD est similaire à la première page d'une notice de montage d'un meuble, indiquant la liste des éléments et des pièces à assembler avec leurs quantités respectives.

Il répertorie les constituants du système ou d'un block en précisant éventuellement leur rôle et leur quantité. Chaque block peut faire l'objet d'une description plus précise en indiquant ses constituants, ses propriétés, les opérations qu'il peut effectuer ainsi que les contraintes ou limites auxquelles il est soumis. Exemple :

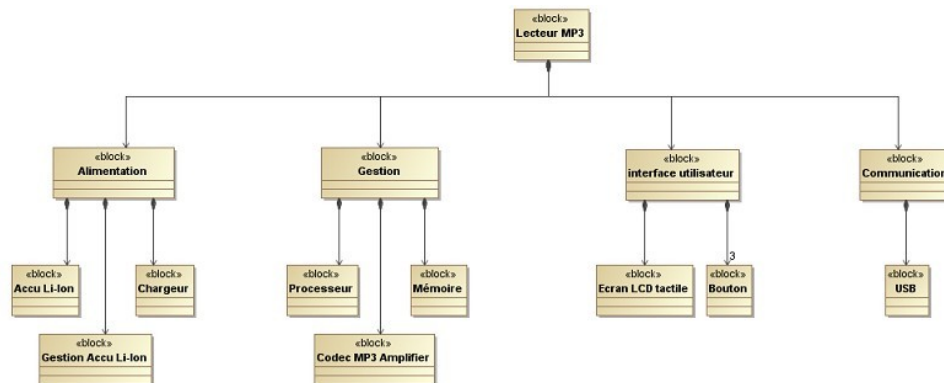


FIGURE 2.4 – Diagramme de définition de blocs d'un lecteur MP3

Remarque : Dans le cas présent, Le lecteur MP3 dispose d'une interface utilisateur constituée d'un écran LCD tactile et de 3 boutons.

Le diagramme de block interne (IBD)

Le diagramme de bloc interne (IBD, ou Internal Block Diagram) décrit la vue interne d'un bloc ou vue boîte blanche, et se base sur le BDD pour assembler les blocs qui composent le bloc principal. Il représente les liens, les flux et les informations échangées entre les parties d'un block ou du système. Le cadre du diagramme représente le block lui-même ou le système.

Le bloc principal peut être représenté comme conteneur sur l'IBD ou être absent de ce diagramme. Les blocs qui le composent, définis dans le BDD, sont instanciés en parties (tout comme les parties dans un diagramme de structure composite avec UML2). Ces parties sont assemblées par des connecteurs qui relient leurs ports (ports standards avec interfaces exposées et/ou ports de flux).

Il est également possible de relier des parties directement entre elles, l'utilisation des ports étant optionnelle.

Par rapport à UML2, l'IBD de SysML redéfinit le diagramme de structure composite en ajoutant entre autre les ports de flux (« flow ports » en anglais).

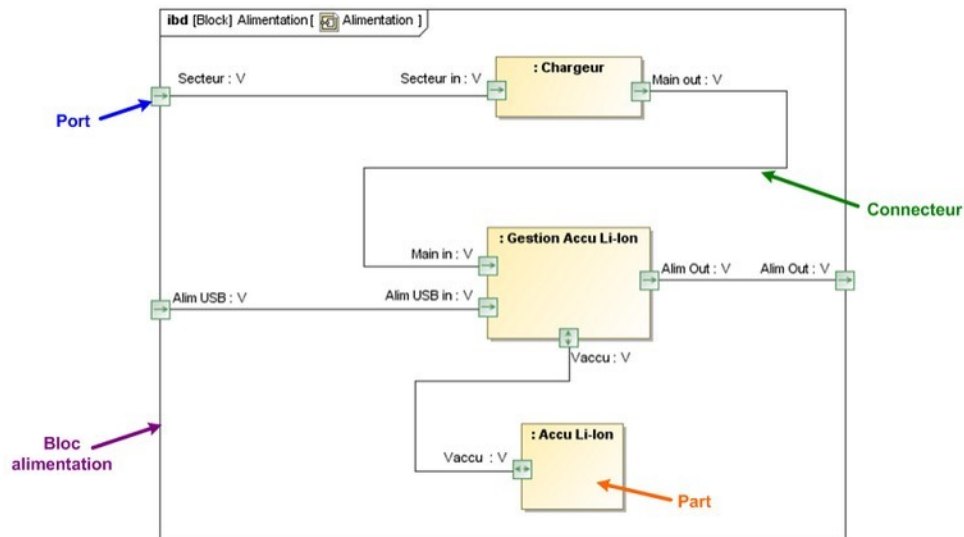




FIGURE 2.5 – Diagramme de block interne du Block Alimentation (Lecteur MP3)

Un port symbolise ce qui peut entrer/sortir d'un block. On en distingue 2 types :

-Le port de flux (flow port)  qui correspond à l'entrée/sortie d'un flux de matière, d'énergie, de données, etc. Le sens de circulation peut être précisé par une flèche.

-Le port standard  qui représente un point de communication lié à un service :

1. une entrée/sortie véhiculant des informations (ou des ordres) logiques/numériques comme l'état d'un bouton poussoir ;
2. une communication plus élaborée entre 2 parts via un réseau.

Les connecteurs représentent les liaisons entre les ports ou les parts, en précisant éventuellement la nature du lien ou ce qui est réellement véhiculé.

Le diagramme paramétrique

Le diagramme paramétrique est une nouveauté SysML qui redéfinit le diagramme interne de bloc SysML (lui-même basé sur le diagramme de structure composite UML2), et permet d'intégrer des analyses systèmes (performance, fiabilité, etc.) avec des blocs de contrainte.

Un bloc de contrainte représente une expression mathématique dont les paramètres peuvent faire référence à des éléments du système, par exemple des propriétés de blocs.

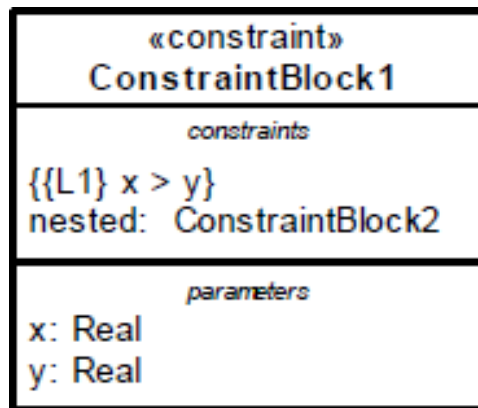
Exemples de blocs de contraintes : $\{F=m \cdot a\}$, $\{a=dv/dt\}$

Dans un premier temps, de façon similaire à la création du diagramme BDD, les blocs de contraintes sont définis dans un diagramme de classe, et représentés comme suit :

Après avoir défini les blocs de contrainte, il faut générer un diagramme paramétrique :

Des blocs de contraintes sont instanciés, donnant lieu aux propriétés de contrainte (ou constraint property), héritant ainsi des paramètres du bloc de contrainte (note : il n'y a pas de différenciation entre paramètres d'entrée et paramètres de sortie)

Des propriétés systèmes (optionnellement liées à des blocs)



Des connecteurs reliant les propriétés systèmes aux paramètres des propriétés de contrainte, ou reliant uniquement des paramètres des propriétés de contrainte

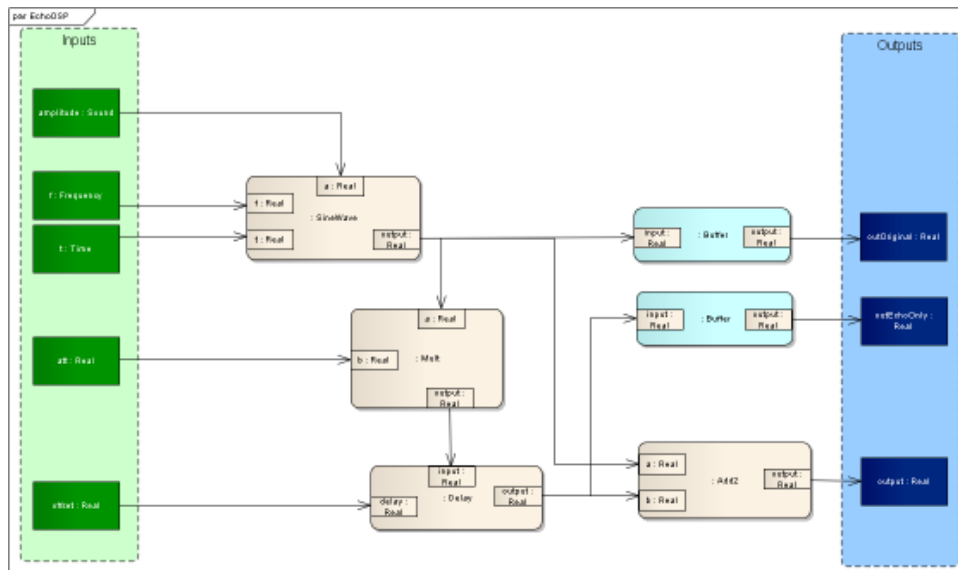


FIGURE 2.6 – Diagramme paramétrique d'un lecteur MP3

1. (a) Informations liées au diagramme paramétrique :

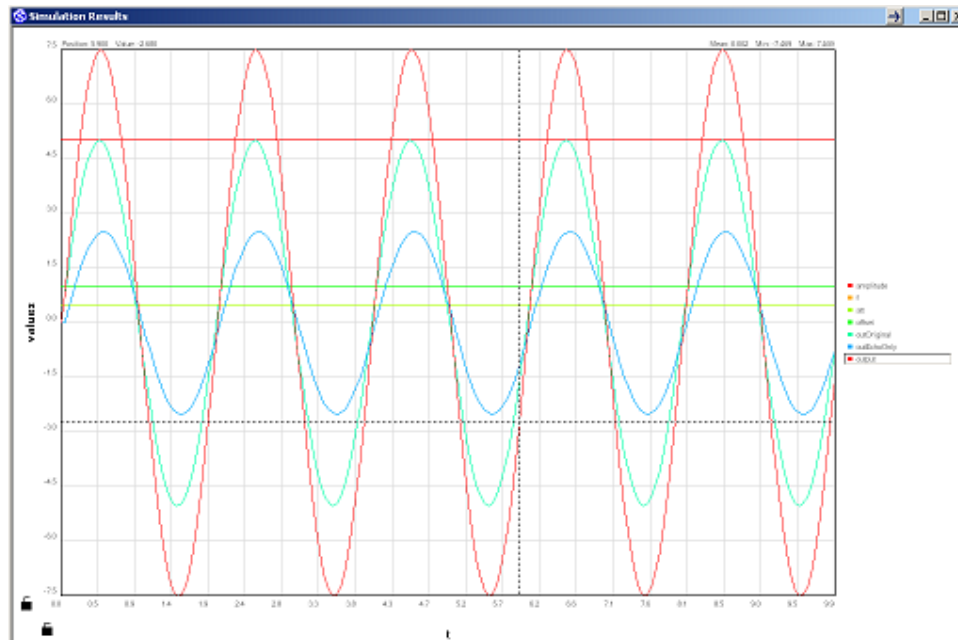
Le diagramme comporte six propriétés de contrainte, dont deux propriétés instanciées du même bloc de contrainte : Buffer.

Les paramètres systèmes sont catégorisés entre entrées et sorties (ex : amplitude et fréquence en entrée, output en sortie). Ces paramètres sont associés aux paramètres de propriétés de contrainte (par exemple la fréquence d'entrée *f* est reliée au paramètre *f* de la propriété de contrainte SineWave)

Certaines propriétés de contrainte sont reliées entre elle (ex : le paramètre SineWave.output est relié à Buffer.input et à Add.a)

Avec certains outils, ce diagramme peut être exécuté dans le cadre de simulation du système.

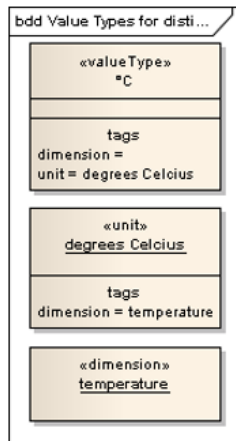
Il est possible sous Enterprise Architect de saisir des expressions pour chacun des blocs de contraintes (par exemple en VBScript ou JavaScript), ainsi que de renseigner différentes valeurs pour les paramètres systèmes. Ces expressions et valeurs peuvent être alors exécutées par le module de simulation Enterprise Architect comme illustré ci-dessous :



Value Types

Les Value Types sont une nouveauté SysML pour définir des types de valeurs réutilisables pour des propriétés du modèle, par exemple des blocs. De façon similaire à la modélisation UML où les attributs de classes peuvent être typés par d'autres classes, SysML permet de définir des propriétés de blocs typées avec des Value Type.

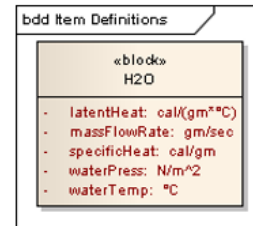
La Value Type a la particularité de contenir 2 propriétés optionnelles : une unité et une dimension. Les unités et dimensions sont des classes spécialisées et définies dans le modèle, par exemple :



Value Type « Distance » avec unit = mètres et dimension = longueur. L'unit « mètres » est définie avec dimension = nulle.

Value Type « °C » avec unit = degrés Celsius et dimension = nulle. L'unit « degrés Celsius » est définie avec dimension = température :

L'exemple suivant montre l'utilisation de la value type « °C » par une propriété du bloc H2O (waterTemp : °C).



2.2.3 Description comportementale

Pour modéliser l'aspect dynamique d'un système et/ou de ses constituants, le langage sysML propose des diagrammes de comportement parmi lesquels on trouve :

Le diagramme de séquence (sd)

Le diagramme de séquence représente les éléments intervenant dans le scénario, ainsi que les échanges de messages entre le système et des acteurs, ou entre des parties du système, de manière chronologique en précisant d'éventuelles contraintes de temps. La lecture d'un tel diagramme se fait de haut en bas.

Dans un premier temps, on peut choisir de représenter les interactions entre l'acteur et le système (vue boîte noire). Par la suite il est possible de rentrer en détails avec un diagramme de séquence qui représente les blocs internes du système intervenant dans un scénario (pour un message émis par l'acteur, le diagramme décrit l'enchaînement des messages échangés entre les blocs internes du système); on parle ainsi de la vue boîte blanche (comportement du système).

Les éléments intervenant sont représentés par des lignes de vie (lifetime en anglais). Ces lignes de vies peuvent être des instances de blocs, établissant un lien avec la modélisation structurelle du système, permettant ainsi d'établir une cohérence dans le modèle :

On peut accéder aux propriétés du bloc d'une ligne de vie (et également aux diagrammes de BDD et d'IBD de ce bloc)

Chaque message échangé peut être utilisé pour l'identification des opérations de blocs

L'ensemble des propriétés du diagramme de séquence utilisées en UML sont également disponibles avec SysML : messages synchrones ou asynchrones, opérateurs (ex : alt, loop, opt, par), références vers d'autres diagrammes de séquence (ex : naviguer de la vue boîte noire du scénario vers la vue boîte blanche), etc.

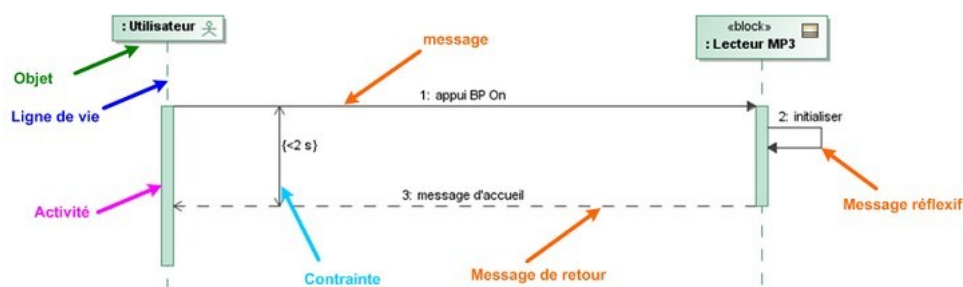


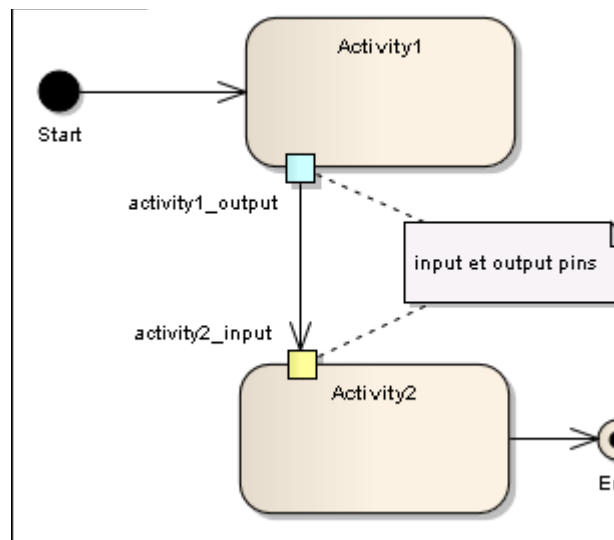
FIGURE 2.7 – Diagramme de séquence du cas d'utilisation "Allumer" du lecteur MP3



Lorsque une séquence de message n'est pas linéaire (conditionnelle, répétitive, simultanée), les messages concernés sont encadrés par des fragments combinés :

Le diagramme d'activité

Le diagramme d'activité est utilisé pour représenter les étapes d'un traitement. Avec SysML, les « input et output pins » sont particulièrement utilisés pour représenter le type d'élément qui est requis en entrée d'une activité ou action, et ce qui est généré en sortie.



Si une action ou activité représente l'opération d'un bloc, on peut garantir la cohérence du modèle en s'assurant que ce qui est défini en entrée d'une activité corresponde à la signature de l'opération du bloc ou de son interface.

Toutes les propriétés des diagrammes d'activités UML sont également disponibles avec SysML. SysML a rajouté quelques spécificités :

1. Notion de contrôle pour activer ou désactiver les actions en cours.
2. Spécification de la nature du débit sur le flot : système continu ou discret
3. Définition de taux et de probabilité sur les flots

Le diagramme d'état (stm)

Le diagramme d'états est utilisé avec SysML de la même manière qu'avec UML2, c'est-à-dire qu'il permet de représenter le cycle de vie auquel doivent se conformer toutes les instances d'un bloc donné, ce au travers de la modélisation de tous les états possibles. Il modélise l'évolution de l'état d'un block en fonction des événements qui peuvent se produire.

Seuls les blocs qui sont importants d'un point de vue métier, ou qui sont de nature complexe, possèdent un diagramme d'état. Exemple :

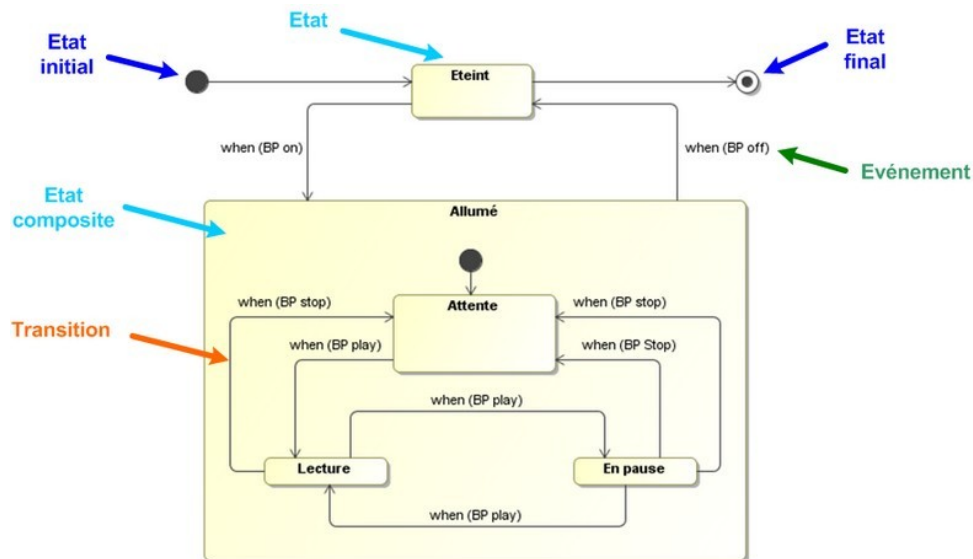


FIGURE 2.8 – Diagramme état - transition partiel d'un lecteur MP3

Les propriétés du diagramme d'état UML sont également disponibles avec SysML : conditions sur événements, effets, activité durable, transitions, états composites, modularité, régions concurrentes, etc.

2.3 Approche SysML

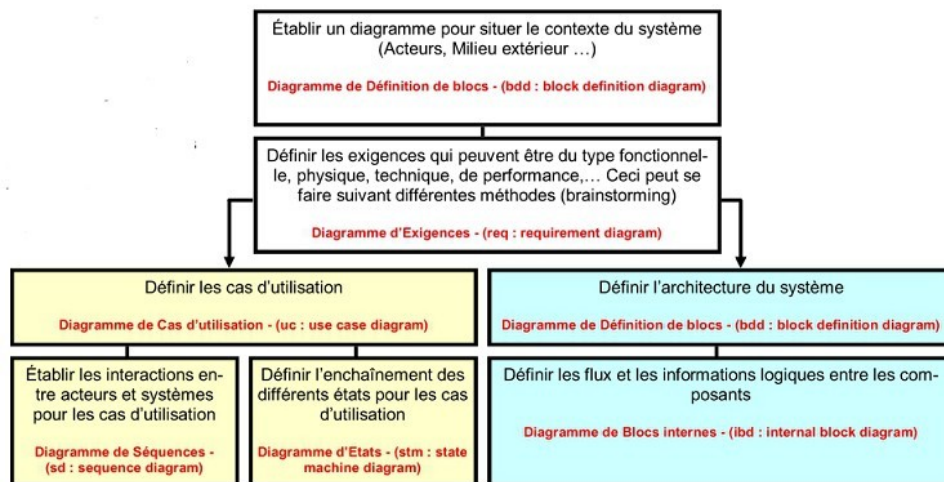


FIGURE 2.9 –

Le SysML est un langage et non une méthode. Il est néanmoins possible d'établir un ordre pour comprendre et expliquer le fonctionnement des systèmes.

2.4 La relation entre UML et SysML

SysML réutilise un sous-ensemble de UML 2.1, appelé "UML4SysML", qui représente environ la moitié du langage UML. Une partie importante de l'UML des concepts ont été écartés car ils n'étaient pas considérés comme pertinents pour les besoins du modèle de SE. Certains des diagrammes réutilisés ont été inclus comme dans UML 2.1.1 . Ceux-ci incluent la machine d'état, la séquence et les diagrammes de cas d'utilisation. D'autres diagrammes ont été étendus, tels que les diagrammes d'activité, afin de répondre aux besoins spécifiques 32 des SE. De plus, SysML a omis certains diagrammes UML, à savoir d'objet, de composant, de déploiement, de communication, de temps et d'interactions. Les diagrammes de structure dont diagramme de Classe et composition ont été fondamentalement modifiés et remplacés par des diagrammes de définition de blocs et diagrammes de blocs internes. Ces extensions sont basées sur le mécanisme de profilage UML standard, qui inclut les stéréotypes UML, extensions de diagramme UML et la bibliothèque de modèles. Le mécanisme de profilage a été choisi sur d'autres mécanismes d'extension afin de tirer parti des outils existants basés sur UML pour la modélisation de systèmes. En outre, SysML ajoute deux nouveaux diagrammes, ceux-ci étant diagrammes d'exigences et diagrammes paramétriques, et intégré de nouvelles capacités de spécification liens tels que l'allocation

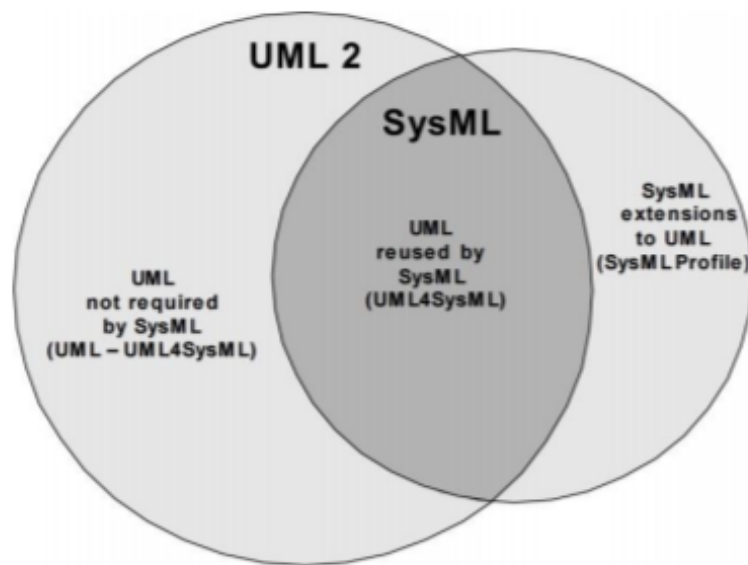


Figure 15: La relation entre UML et SysML

FIGURE 2.10 – Relation entre uml et sysml

2.5 Synthèse :

Dans cette section, nous avons mis en évidence les points suivants :

1. SysML permet de modéliser un produit et son environnement par un système, c'est-à-dire un ensemble de composants qui interagissent entre eux.
2. Cette modélisation est multipoints de vue et inclut notamment des aspects fonctionnels, structurels et comportementaux.
3. Un modèle SysML est représenté par un ensemble de boîtes et de lignes, munies de types, de nom permettant de les identifier, et d'une hiérarchie (une boîte contient d'autres boîtes et des lignes).
4. Les boîtes et les lignes sont assemblées au sein de diagrammes interdépendants : un même élément, repéré par son nom, peut figurer sur des diagrammes différents, qui apportent alors des informations complémentaires.
5. Il existe neuf types de diagrammes. Ces types, ainsi que les relations possibles entre les éléments représentés par ces diagrammes, sont récapitulés sur la figure 14 (le diagramme des paquets a été omis dans un souci de simplicité).
6. Un diagramme doit toujours représenter l'intérieur d'une boîte. Cependant, cette représentation peut être partielle (on ne représente que ce dont on a besoin)

2.5.1 Le processus unifié UP (Unified Process)

Le processus unifié noté UP est un processus de développement logiciel moderne, très complet et construit sur UML. Il est itératif, centré sur l'architecture, piloté par des cas d'utilisation et orienté vers la diminution des risques. Il regroupe les activités à mener pour transformer les besoins d'un utilisateur en système logiciel. C'est un patron de processus pouvant être adapté à une large classe de systèmes logiciels, à différents domaines d'application, à différents types d'entreprises, à différents niveaux de compétences et à différentes tailles de l'entreprise.

UP est itératif

L'itération est une répétition d'une séquence d'instructions ou d'une partie de programme un nombre de fois fixé à l'avance ou tant qu'une condition définie n'est pas remplie, dans le but de reprendre un traitement sur des données différentes. Elle qualifie un traitement ou une procédure qui exécute un groupe d'opérations de façon répétitive jusqu'à ce qu'une condition bien définie soit remplie.

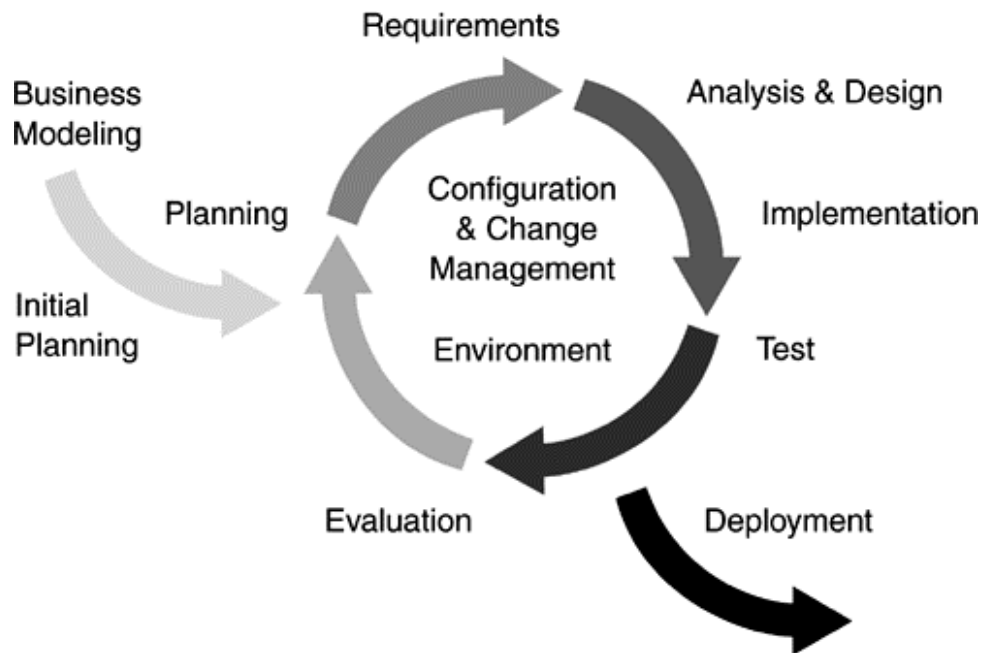


FIGURE 2.11 – UP est itératif

Une itération prend en compte un certain nombre de cas d'utilisation et traite en priorité les risques majeurs.

UP est centré sur l'architecture

Dès le démarrage du processus, on aura une vue sur l'architecture à mettre en place. L'architecture d'un système logiciel peut être décrite comme les différentes vues du système qui doit être construit. L'architecture logicielle équivaut aux aspects statiques et dynamiques les plus significatifs du système. L'architecture émerge des besoins de l'entreprise, tels qu'ils sont exprimés par les utilisateurs et autres intervenants et tels qu'ils sont reflétés par les cas d'utilisation .

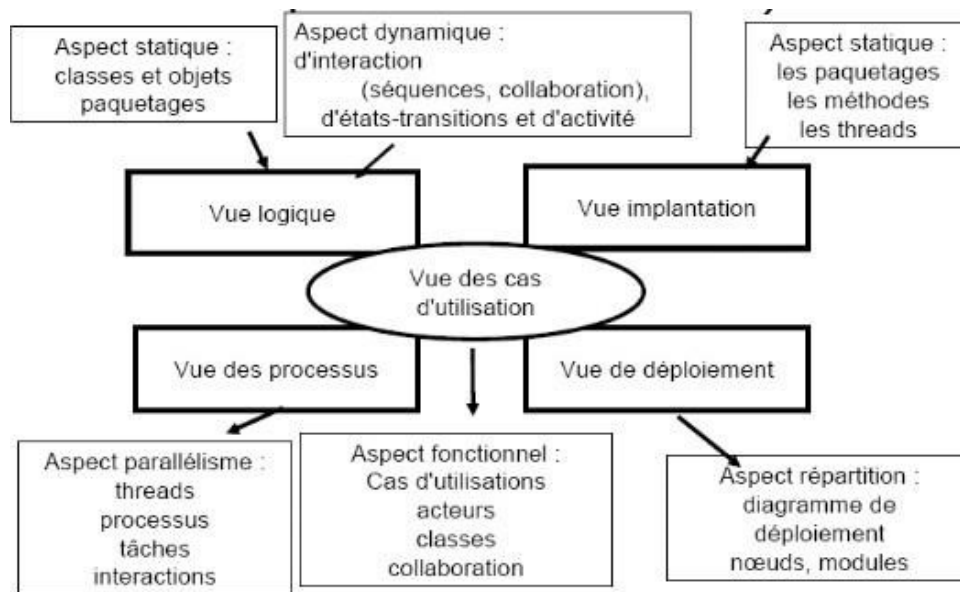


FIGURE 2.12 – UP est centré sur l'architecture

UP est piloté par les cas d'utilisation

Le but principal d'un système informatique est de satisfaire les besoins du client. Le processus de développement sera donc axé sur l'utilisateur. Les cas d'utilisation permettent d'illustrer ces besoins.

Ils détectent puis décrivent les besoins fonctionnels (du point de vue de l'utilisateur), et leur ensemble constitue le modèle de cas d'utilisation qui dicte les fonctionnalités complètes du système.

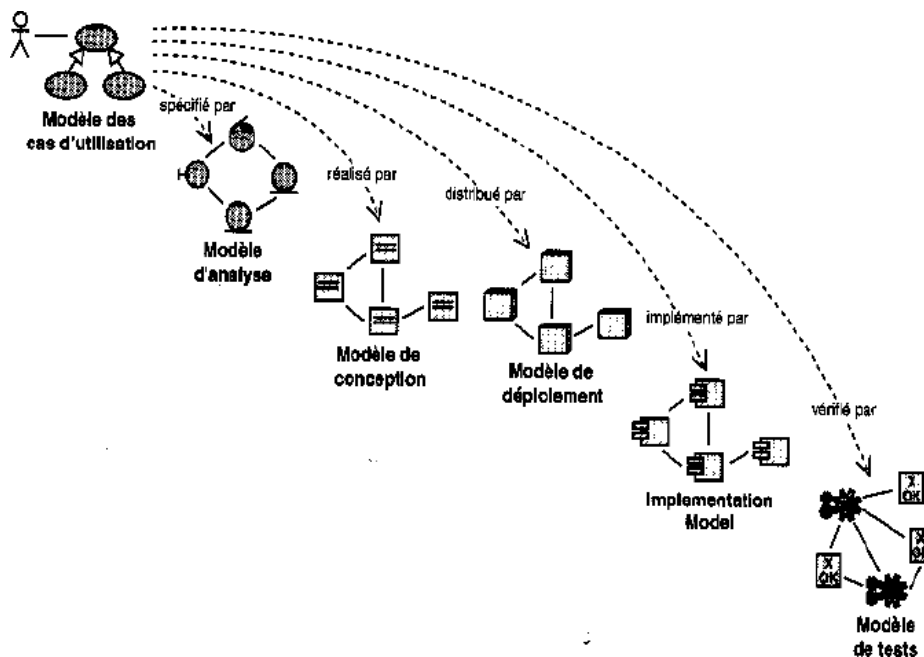


FIGURE 2.13 – UP est piloté par les cas d'utilisation

Les activités

Expression des besoins : permet de définir les différents besoins :

1. Besoins principaux et fournir une liste de leurs fonctions.
2. Besoins fonctionnels du point de vue de l'utilisateur.
3. Besoins non fonctionnels ou technique.

Analyse : Un modèle d'analyse livre une spécification complète des besoins issus des cas d'utilisation et les structure sous une forme qui facilite :

1. La compréhension (scénarios).
2. La préparation (définition de l'architecture).
3. La modification et la maintenance du futur système.

Conception :

1. Elle détermine les principales interfaces et les transcrit à l'aide d'une notation commune.
2. Elle constitue un point de départ à l'implémentation :
3. Elle décompose le travail d'implémentation en sous-système
4. Elle crée une abstraction transparente de l'implémentation

Implémentation : Les objectifs principaux de l'implémentation sont :

1. Planifier les intégrations des composants pour chaque itération.
2. Produire les classes et les sous-systèmes sous formes de codes source.

Test : Les tests sert à :

1. Vérifier des résultats de l'implémentation en testant la construction.

Les phases

Analyse des besoins : cette phase porte essentiellement sur :

1. Les besoins principaux du point de vue de l'utilisateur.
2. L'architecture du système, les risques majeurs.
3. Les délais et les coûts.

Elaboration : l'élaboration reprend les éléments de la phase d'analyse des besoins et les précise pour arriver à une spécification détaillée de la solution à mettre en œuvre. Les tâches à effectuer dans la phase élaboration sont les suivantes :

1. Créer une architecture de référence.
2. Identifier les risques.
3. Définir les niveaux de qualité à atteindre.
4. Formuler les cas d'utilisation pour couvrir les besoins fonctionnels et planifier la phase de construction.
5. Élaborer une offre abordant les questions de calendrier, de personnel et de budget.

Construction : La construction est le moment où l'on construit le produit.

L'architecture de référence se métamorphose en produit complet.

Le produit contient tous les cas d'utilisation que les chefs de projet, en accord avec les utilisateurs ont décidé de mettre au point pour cette version.

Transition : Un groupe d'utilisateurs essaye le produit et détecte les anomalies et défauts.

La mise en œuvre d'un service d'assistance et la correction des anomalies constatées.

[10]

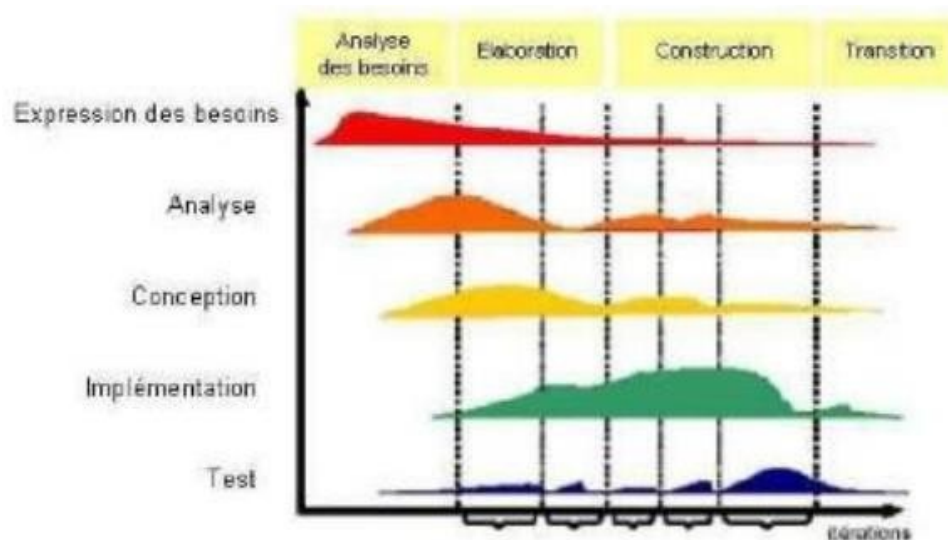


FIGURE 2.14 – Activité et phases d'UP

Chapitre 3

Outils et matériel

3.1 Introduction

Dans ce chapitre nous parlerons des outils et matériel utilisé dans la mise au point d'un système de localisation en intérieur en général puis on terminera avec une comparaison

3.2 Internet des objets (IoT)

L'Internet des objets (IoT) est une nouvelle technologie qui se base sur des dispositifs informatiques interdépendants, dotés d'identifiants uniques (UID) et de la possibilité de transférer des données sur un réseau sans nécessiter l'interaction humaine.

La définition de l'Internet des objets a évolué en raison de la convergence de plusieurs technologies, de l'analyse en temps réel, de l'apprentissage automatique, des capteurs de produits et des systèmes intégrés. [1] Les domaines traditionnels des systèmes embarqués, des réseaux de capteurs sans fil, des systèmes de contrôle, de l'automatisation (y compris l'automatisation de la maison et du bâtiment) et d'autres contribuent tous à permettre l'Internet des objets.

Utilisateur
Windows
<https://accueil.edcloud.fr/snt-n1-5/>

Historique de la Technologie: la Connectivité des choses

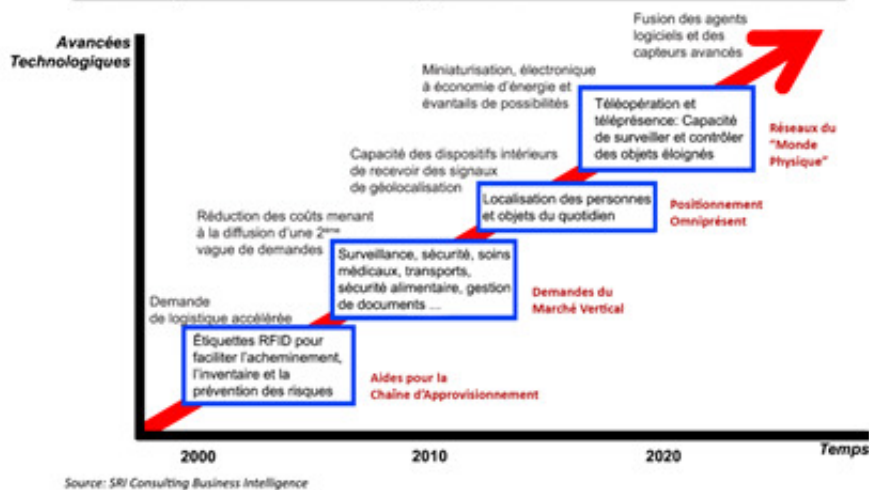


FIGURE 3.1 – Historique de l’IoT.

L’Internet est défini comme :

« Réunir des personnes, des processus, des données et des éléments pour établir des connexions en réseau plus pertinente et plus utile que jamais, en transformant les informations en actions qui créent de nouvelles capacités, des expériences plus riches et des opportunités économiques sans précédent pour les entreprises, les particuliers et les pays. » [IoT 15]. Par CISCO

Comme indiqué dans la définition des IoT, Un objet dans l’Internet des objets a deux caractéristiques :

- ✓ Un identifiant unique : une adresse IP dans un réseau.
- ✓ Il doit être capable de transférer des données sur le réseau pour communiquer avec son environnement

Plusieurs événements qui précèdent l’Internet des objets ont accéléré son apparition, la figure X ci-dessous montre l’évolution d’Internet

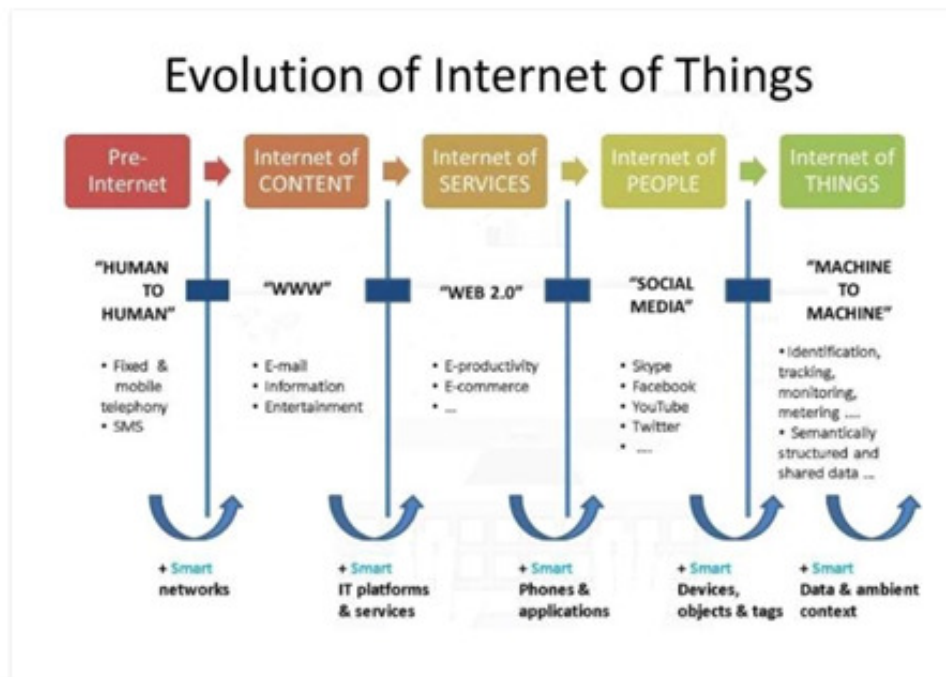


FIGURE 3.2 – Evolution de l'internet.

- 1- la phase pré-internet, la communication « Humain à Humain » qui se base sur la téléphonie fixe et mobile et les services de messageries SMS
- 2- la phase Internet du contenu, l'apparition du réseau internet et les méthodes d'accès associées dénommées World Wide Web « WWW », qui a engendré les courriers électronique « Emails » et les sites informatiques
- 3- la phase internet du service , L'évolution des réseaux Internet a créé des plates-formes et des services informatiques qui ont naissance au WEB 2.0 avec les différentes plateformes de commerce électronique et de productivité électronique qui nous permettent par exemple de faire des achats en ligne.
- 4- la phase internet des gens, l'invention des smartphones et le nombre énormes de l'application qui l'a suivi « Facebook, twitter, Skype ...etc. »
- 5- Enfin, l'augmentation et la diversité des technologies mobiles et embarquées. L'interaction machine à machine est devenue possible, une nouvelle génération de l'utilisation d'Internet était née : l'Internet des objets.

3.2.1 Types d'objet connectés

En mars 2015, le comité Internet Architecture Board (IAB) 9 édite la RFC10 7452. Il propose quatre modèles communs d'interactions entre des acteurs de l'IdO11 [voir ROX 2017, p. 64] :

1. **La communication entre objets**, ce modèle est basé sur une communication sans-fil entre deux objets Les informations sont transmises grâce à l'intégration d'une technologie de communication sans fil comme le Bluetooth Low Energy (BLE), wifi ad-hoc sans la nécessité d'être connectés à internet

1. La communication des objets vers une passerelle : ce modèle est basé sur un intermédiaire « Passerelle » qui fait le lien entre les capteurs et l'objet communicant (Smartphone par exemple)
1. La communication par réseau IoT : Les objets connectés dans ce type utilisent les protocoles M2M (machine to machine) et sont directement connectés à internet grâce à une technologie embarquée dans l'objet en question.

3.2.2 Domaines d'application de l'IoT :

LA MAISON

La maison connectée dispose de systèmes automatiques très avancés comme le contrôle de l'éclairage, de la sécurité, de la fermeture ou de l'ouverture d'issues (portes+ fenêtres) ainsi que d'autres fonctions. Elle est aussi capable de participer à de nombreux aspects de la vie quotidienne comme le réfrigérateur de Samsung, Family Hub, qui peut effectuer l'inventaire de son stock ; proposer des menus ou commander du réapprovisionnement.

LE SPORT

Des capteurs en tout genre ont fait leur apparition dans le monde du sport, qui est un des domaines privilégiés dans le secteur des objets connectés. Le smartphone peut maintenant, grâce à de nombreuses applications, aider les personnes à contrôler leur activité sportive, leurs progrès, leur alimentation et bien d'autre. Nous pouvons voir parmi les objets connectés les fixations de snowboard Cerevo qui permettent de mesurer les performances et de les combiner à une vidéo qui est filmée grâce à une caméra intégrée. Nous retrouvons aussi Xensr qui est un capteur pour le sport de glisse récoltant une multitude d'informations sur la vitesse des longueurs de sauts et le temps passé en l'air.

LE LOISIR

Les loisirs profitent aussi des objets connectés et deviennent des loisirs connectés qui nous facilitent la vie ; nous pouvons y retrouver les drones connectés ; les jouets connectés ainsi que les voyage connectés qui facilitent les départs en vacances et sécurisent le transport grâce à des valises de haute sécurité comme la Blue Smart qui permet aux voyageurs de pouvoir retrouver leur valise à l'aide de leur téléphone portable mais aussi de pouvoir contrôler son poids et de la verrouiller pour empêcher les vols.

LA SANTÉ ET LE BIEN-ÊTRE

La santé connectée est un ensemble de système technologique visant à préserver durablement la santé. Le dispositif va enregistrer toutes les activités du corps afin d'en faire un compte rendu détaillé sur le téléphone, la tablette ou l'ordinateur de son utilisateur.

Contrôler sa santé devient alors un geste banal et rassurant, car les données sont obtenues en temps réel et peuvent être transmises immédiatement au médecin. La balance connectée, notamment celle proposée par withings, permet de mesurer le poids et l'IMC. Elle peut également être utilisée comme unité de mesure de la masse grasse, maigre, musculaire, osseuse, hydrique, viscérale et être en bonne indicatrice de la température ambiante et du pourcentage d'humidité.

LA MODE ET ACCESSOIRES

Les objets connectés concernant la mode se développent, ainsi grâce à la gamme de bijoux Altruis vous pourrez voir le plus important de vos notifications smartphone sous forme de vibrations.

Les ceintures connectées, elles, permettent de nous faire adopter une bonne posture tout enregistrant sur le téléphone les progrès réalisés ou de contrôler le tour de taille afin de prévenir lors d'une prise ou d'une perte de poids.

LE TRANSPORT

Les objets connectés dans l'automobile vont combler les passionnés cependant il faudra encore quelques années avant de pouvoir développer complètement ces technologies de manière efficaces. Quelques produits sont déjà sur le marché comme le casque de moto connecté qui régularise toutes les difficultés rencontrées (recevoir des appels, vision total de l'entourage, envoyer des messages par voie orale). Nous connaissons aussi la voiture connectées qui retranscrit le menu de votre portable sur son tableau de bord dès que l'on y rentre.

LES MULTIMÉDIA

Le multimédia est un environnement propice au développement des objets connectés et de toutes les technologies connectées.

La télévision connectée peut accéder à internet et à de nombreux réseaux sociaux et d'applications tels que YouTube, Dailymotion ou encore Facebook ou Twitter. Un bracelet connecté, KipstR, sera aussi bientôt commercialisé, il servira à repérer le moment où une personne s'endort devant la télévision et programme un enregistrement pour rattraper les choses manquées.

LES ANIMAUX

De nombreux objets connectés existent pour rester en contact avec les animaux lors de déplacement, c'est notamment le but de la caméra en forme de cube, Petcube, équipée d'un micro et d'un haut-parleur. Il existe aussi des jeux connectés, équipés de trois touches tactiles afin d'occuper les animaux la journée.

LES BÉBÉS

Même les nouveaux nés sont touchés par la révolution technologique, ainsi il existe des baby phones, comme l'écoute bébé de Philips, connectés au portable avec des alertes quand le bébé pleure ou encore des pèse-bébés connectés qui permettent de garder un œil sur sa courbe.

3.2.3 Architecture général d'un system IoT

Pour concevoir un système IoT, on doit s'intéresser principalement aux aspects suivants :

Collecte de données (collecte des données à partir de périphériques électroniques comme les capteurs, caméras, ...etc.).

1. (a) Traitement des données.
- (b) Analyses de données.
- (c) Toute architecture IoT doit comporter les composants de base suivants :

Instruments : (Ceci est principalement lié à l'électronique) qui peuvent être branchés sur un réseau informatique avec une adresse IP particulière.

Infrastructure : besoin d'un serveur et une connexion réseau pour garder l'ensemble opérationnel. Il est très similaire aux LAN, WAN ou Internet, et doit avoir ses serveurs connectés à Internet, ou lié à un environnement de service Cloud virtuel. Microsoft, IBM, Amazon et la plupart des grands acteurs fournissent des infrastructures pour IoT. Le choix dépend du besoin particulier.

Codage : cette partie concerne l'utilisation des langages de programmation pour réaliser le logique métier associé au système IoT.

Sécurité : Cela revêt une importance capitale en ce qui concerne l'IoT. Lorsqu'à la Connection d'un plus grand nombre de périphériques ensemble, cela ouvre beaucoup de vulnérabilités. Il existe toujours une menace pour les données personnelles, car cela pourrait être compromis si elles ne sont pas prises en charge correctement.

Gestion des données : Le traitement des quantités massives de données dans l'IoT. Besoin d'un grand système de données comme Hadoop et Apache Storm pour être en place pour la gestion et le traitement des données.

Analyse des données : Une fois que les données sont traitées et gérées, Il existe plusieurs éléments essentiels qui peuvent être déchiffrés des « données inutilisés ». L'intelligence des données, l'exploration de données, ...etc. sont quelques-unes des techniques que pouvait utiliser pour analyser les données et obtenir des informations significatives qui conduiraient l'écosystème IoT [8].

3.2.4 Les plateformes IoT :

La plateforme IoT rassemble un ensemble de services qui permettent de collecter, stocker, corréler, analyser et exploiter des données. Elle fait ainsi référence au logiciel de support qui connecte l'ensemble du système IoT, en facilitant la communication, le flux de données, la gestion des périphériques et la fonctionnalité des applications. La plateforme connecte les appareils à un cloud grâce à des options de connectivité flexibles.

Pour les développeurs, une plateforme IoT fournit un ensemble de fonctionnalités prêtes à l'emploi qui accélère le développement d'applications pour les périphériques connectés, tout en prenant en charge l'évolutivité et la compatibilité entre périphériques. Elle peut également servir de middleware lorsqu'elle connecte les périphériques distants aux applications utilisateurs (ou autres périphériques) tout en gérant les interactions entre le matériel et les couches des applications.

Windows 5747274171736735644 <https://www.journaldunet.fr/web-tech/dictionnaire-de-l-iot/1440690-plateforme-iot-definition-comparatif-device-management/>

La plate-forme IoT garantit une intégration transparente avec différents matériels en utilisant une gamme de protocoles de communication populaires, en appliquant différents types de topologie (connexion directe ou passerelle) et en utilisant des SDK (Software Développement Kit) si nécessaire.

Exemple de plateformes IoT :

- ✓ AWS IoT Platform.
- ✓ Blynk IoT Platform.
- ✓ EVERYTHING - IoT Smart Products Platform.
- ✓ IBM IoT Foundation Device Cloud.
- ✓ ParStream - IoT Analytics Platform.
- ✓ Kaa IoT platform.
- ✓ Google IOT platform.
- ✓ Microsoft azure IoT platform.
- ✓ CarrIoTs IoT platform.

3.3 Les systèmes embarqués

Définition 1 : 574727418 Un **système embarqué** est un système complexe qui intègre du logiciel et du matériel conçus ensemble afin de fournir des fonctionnalités données. Il contient généralement un ou plusieurs **microprocesseurs** destinés à exécuter un ensemble de **programmes** définis lors de la conception et stockés dans des **mémoires**. Le **système matériel** et l'**application (logiciel)** sont intimement liés et immergés dans le matériel et ne sont pas aussi facilement discernables comme dans un **environnement de travail** classique de type ordinateur de bureau PC (Personale Computer).

Définition 2 : 574727419 Les systèmes embarqués sont des systèmes informatiques mais peuvent ne pas comporter d'interface utilisateur par exemple, sur des périphériques dans lesquels le système intégré est conçu pour effectuer une tâche unique vers une interface utilisateur graphique complexe, telle que dans les

appareils mobiles. L'interface utilisateur peut faire référence à des boutons, des voyants, des capteurs tactiles, etc.

3.3.1 Caractéristiques

Les caractéristiques principales d'un système électronique embarqué sont :

1. L'autonomie : Dédiee à une application spécifique.
2. Exécution temps réel : l'importance majeure des temps de réponses
3. . Fiabilité et sécurité de fonctionnement

. Consommation d'énergie maîtrisée

1. Haute école d'ingénierie et de gestion du canton de Vaud

3.3.2 Systèmes informatiques et Systèmes embarqués

Les différences entre les deux systèmes se résument dans le tableau récapitulatif ci-dessous

Système Informatique	Système Embarqué
Processeur standard	Processeur dédié (contrôleur)
Multiples unités fonctionnelles (flottant)	Architecture adaptée
Vitesse élevée (> GHz)	Vitesse faible (~200 MHz)
Consommation électrique élevée	8-32bits : mémoire limitée
Chaleur	Basse consommation
Taille	Petite taille, grand volume => faible coût
MMU (mémoire virtuelle) OS	Processeur DSP (traitements)
Cache	Très puissants Quelques Mo de mémoire
Grand nombre de périphériques	RTOS

3.3.3 Classification des systèmes embarqués

Les systèmes embarqués sont classifiés en trois systèmes :

Système Transformationnel

Qui lit ses données et ses entrées lors de son démarrage, qui fournit ses sorties, puis meurt

Système Interactif

Système en interaction quasi permanente avec son environnement, y compris après l'initialisation du système ; la réaction du système est déterminée par les événements reçus et par l'état courant (fonction des événements et des réactions passés), le rythme de l'interaction est déterminé par le système et non par l'environnement

Système Réactif ou Temps Réel

Système en interaction permanente avec son environnement, y compris après l'initialisation du système ; la réaction du système est déterminée par les événements reçus et par l'état courant (fonction des événements et des réactions passées), mais le rythme de l'interaction est déterminé par l'environnement et non par le système

3.3.4 Domaines d'application des systèmes embarqués

Domaine grand public

. Smart phone, console de jeux, appareil photos, lecteur audio, ...

Moyens de transport

. Gestion moteur/entraînement, ordinateur de bord, ABS, GPS, système navigation, système d'aide (EPS, ..), => automobiles, avions, trains, bateau, véhicule électrique, ...

Équipement médicaux (diagnostic, thérapeutique, vital)

. Imagerie (rayon X, ultra-sons, IRM), endoscopie, caméra, monitoring, perfusion, lasers, chirurgie, stimulateur cardiaque, ...

Équipements de télécommunication .

Mobile, routeur, Gateway, satellite, ...

Équipement industriels

. commande, contrôle réparti, capteurs intelligents, ...

3.4 Raspberry

Le Raspberry Pi est un ordinateur de la taille d'une carte de crédit et pouvant être connecté à un ordinateur. Moniteur ou téléviseur et utilise un clavier et une souris standard. Il est capable de faire tout ce qu'on attend d'un ordinateur de bureau à faire, de naviguer sur Internet et de lire des vidéos haute définition, éditer des feuilles de calcul, du traitement de texte et jouer à des jeux. [Rasb].

De plus, le Raspberry Pi a la capacité d'interagir avec le monde extérieur et a été utilisé dans un large éventail de projets de fabrication numérique, allant des appareils de musique aux détecteurs des stations météo.

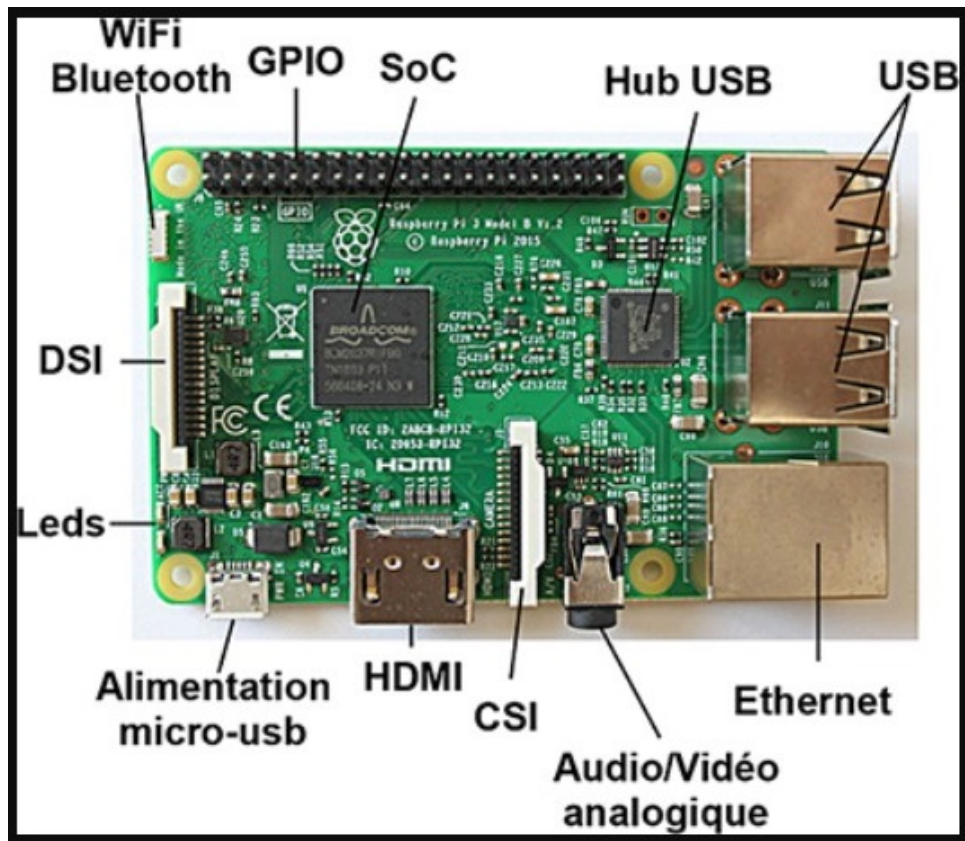


FIGURE 3.3 – Évolution de l'internet.

3.4.1 Composants d'une carte Raspberry

Processeur / GPU ARM : Il s'agit d'un système Broadcom BCM2835 sur puce (SoC) composé d'une unité centrale de traitement ARM (CPU) et d'un graphique Videocore 4 unité de traitement (GPU). La CPU gère tous les calculs qui font un travail sur ordinateur (prendre des entrées, faire des calculs et produire des sorties), et le GPU gère la sortie graphique.

GPIO : Ce sont des points de connexion d'entrée / sortie à usage général qui permettent de connecter d'autres hardware.

RCA : Une prise RCA permet la connexion de téléviseurs analogiques et d'autres périphériques de sortie similaires.

Sortie audio : C'est une prise standard de 3,55 millimètres pour la connexion de la sortie audio des appareils tels que des écouteurs ou des haut-parleurs. Il n'y a pas d'audio dans.

LED Diodes : électroluminescentes, pour tous les besoins en indicateur lumineux.

USB : C'est un port de connexion commun pour les périphériques de tous types (y compris la souris et le clavier). Le modèle A en a un et le modèle B en compte deux. Un concentrateur USB peut être utilisé pour augmenter le nombre de ports USB.

HDMI : Permet de brancher un téléviseur haut définition ou tout autre appareil compatible utilisant un câble HDMI.

Source de courant : Il s'agit d'un connecteur d'alimentation micro USB 5v dans lequel vous pouvez brancher votre appareil compatible.

Lecteur de carte SD (DSI) : Il s'agit d'un compartiment pour carte SD de taille normale. Une carte SD avec un système d'exploitation (OS) installé est requise pour l'amorçage du périphérique. Le système d'exploitation requis est disponible en achat, et téléchargeable également avec une machine Linux.

Ethernet : Permet un accès au réseau filaire et n'est disponible que sur le Modèle B.

3.5 Plateformes et langages :

3.5.1 PhpStorm

PhpStorm est un éditeur pour les langages HTML, CSS, PHP et Javascript. PhpStorm est un éditeur pour PHP3, HTML, CSS et JavaScript4, édité par JetBrains. Une interface utilisateur pour les logiciels de tests tels que PHPUnit; Le débogage pas-à-pas et le profilage de code en dialoguant avec Xdebug. Il permet aussi de visualiser l'architecture de bases de données de différentes sources (MySQL, SQLite, etc.). Enfin, il permet l'intégration d'outils d'opérations serveur comme Vagrant, Docker, une console SSH et bien d'autres outils. PhpStorm est écrit en Java, et ses utilisateurs peuvent lui adjoindre des extensions fournies par JetBrains, une tierce partie ou écrites par eux-mêmes. Toutes sortes d'outils destinés à faciliter et accélérer le travail du développeur : aides à l'apprentissage des touches de raccourci, manipulation de texte, coloration de parenthèses, etc. Enfin, il est possible d'utiliser gratuitement PhpStorm via le Early Access Program¹⁰, qui permet d'évaluer la prochaine version majeure du logiciel.



FIGURE 3.4 – Logo PhpStorm.

3.5.2 Android Studio

Android Studio est un environnement de développement pour développer des applications mobiles Android. Il est basé sur IntelliJ IDEA et utilise le moteur de production Gradle. Il peut être téléchargé sous les systèmes d'exploitation Windows, macOS, Chrome OS et Linux⁴.



FIGURE 3.5 – Logo Android Studion.

3.5.3 PHP

HP : Hypertext Preprocessor⁵, plus connu sous son sigle PHP (sigle auto-référentiel), est un langage de programmation libre⁶, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP⁵, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet. PHP a permis de créer un grand nombre de sites web célèbres, comme Facebook, Wikipédia, etc.⁷ Il est considéré comme une des bases de la création de sites web dits dynamiques mais également des applications web.



FIGURE 3.6 – Logo PHP.

3.5.4 Javascript

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives mais aussi pour les serveurs² avec l'utilisation (par exemple) de Node.js³. C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun équipés de constructeurs permettant de créer leurs propriétés, et notamment une propriété de prototypage qui permet d'en créer des objets héritiers personnalisés. En outre, les fonctions sont des objets de première classe. Le langage supporte le paradigme objet, impératif et fonctionnel. JavaScript est le langage possédant le plus large écosystème grâce à son gestionnaire de dépendances npm, avec environ 500 000 paquets en août 2014.



FIGURE 3.7 – Logo JavaScript.

3.5.5 HTML

Le HyperText Markup Language, généralement abrégé HTML ou dans sa dernière version HTML5, est le langage de balisage conçu pour représenter les pages web. C'est un langage permettant d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et logiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie et des programmes informatiques. Il permet de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l'accessibilité du web. Il est souvent utilisé conjointement avec le langage de programmation JavaScript et des feuilles de style en cascade (CSS). HTML est inspiré du Standard Generalized Markup Language (SGML). Il s'agit d'un format ouvert.



FIGURE 3.8 – Logo HTML.

3.5.6 JAVA

Java est un langage de programmation orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au SunWorld. La société Sun a été ensuite rachetée en 2009 par la société Oracle qui détient et maintient désormais Java. Une particularité de Java est que les logiciels écrits dans ce langage sont compilés vers une représentation binaire intermédiaire qui peut être exécutée dans une machine virtuelle Java (JVM) en faisant abstraction du système d'exploitation.



FIGURE 3.9 – Logo JAVA.

3.5.7 XML

L'Extensible Markup Language, généralement appelé XMLnote 1, « langage de balisage extensible1 » en français, est un métalangage informatique de balisage géné-

rique qui est un sous-ensemble du Standard Generalized Markup Language (SGML). Sa syntaxe est dite « extensible » car elle permet de définir différents langages avec pour chacun son vocabulaire et sa grammaire, comme XHTML, XSLT, RSS, SVG... Elle est reconnaissable par son usage des chevrons (<, >) encadrant les noms des balises. L'objectif initial de XML est de faciliter l'échange automatisé de contenus complexes (arbres, texte enrichi, etc.) entre systèmes d'informations hétérogènes (interopérabilité). Avec ses outils et langages associés, une application XML respecte généralement certains principes :

3.5.8 CSS

Les feuilles de style en cascade¹, généralement appelées CSS de l'anglais Cascading Style Sheets, forment un langage informatique qui décrit la présentation des documents HTML et XML. Les standards définissant CSS sont publiés par le World Wide Web Consortium (W3C). Introduit au milieu des années 1990, CSS devient couramment utilisé dans la conception de sites web et bien pris en charge par les navigateurs web dans les années 2000.



FIGURE 3.10 – Logo CSS.

3.5.9 XAMPP

XAMPP est un ensemble de logiciels permettant de mettre en place un serveur Web local, un serveur FTP et un serveur de messagerie électronique. Il s'agit d'une distribution de logiciels libres (X (cross) Apache MariaDB Perl PHP) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide. Ainsi, il est à la portée d'un grand nombre de personnes puisqu'il ne requiert pas de connaissances particulières et fonctionne, de plus, sur les systèmes d'exploitation les plus répandus.



FIGURE 3.11 – Logo XAMPP.

Bibliographie

- [1] M. J. Maceraudi, Algorithmes de radiolocalisation et traitements adaptés à une architecture de récepteur IR-UWB intégrée. PhD thesis, UNIVERSITÉ DE RENNES 1, 2017.
- [2] https://www.phidata.be/fr/technologies/real_time_location_systems/.
- [3] H. C. Kaplan E.D, C. Understanding GPS : Principles and Applications, 2nd ed. Artech House : Norwood, MA, USA, 2006.
- [4] X. Weihua Zhuang, Centre for Wireless Communications (CWC). Department of Electrical and Computer Engineering, University of Waterloo.
- [5] J. Haartsen, "Bluetooth - the universal radio interface for ad hoc, wireless connectivity," Ericsson review, vol. 75, pp. 110–117, 01 1998.
- [6] <https://fr.wikipedia.org/wiki/Radio-identification>.
- [7] A. D. C. L. Zhao K., Li B., A Comparison of algorithms adopted in fingerprinting indoor positioning systems. Outrigger Gold Coast, Australia, 2013.
- [8] R. Elaire, conception et développement d'une application de géolocalisation indoor. PhD thesis, Université de la Manouba, 2014.
- [9] <https://www.ikomobi.com/la-geolocalisation-indoor-un-enjeu-mobile>.
- [10] L. Ojeda and J. Borenstein, "Personal dead-reckoning system for gps-denied environments," pp. 1 – 6, 10 2007.
- [11] C. Galván Tejada, J. García-Vázquez, and R. Brena, "Magnetic field feature extraction and selection for indoor location estimation," Sensors, vol. 14, pp. 11001–11015, 06 2014.
- [12] "Gps en interieur à l'aéroport de londres-gatwick." <https://www.directioninformatique.com/un-gps-interieur-a-laeroport-de-londres-gatwick/49260>. Consulté le 2020-03-25.
- [13] "Hypermarché carrefour westfield euralille lille, france." <https://www.ictjournal.ch/news/2015-10-20/carrefour-guide-ses-clients-grace-a-la-geolocalisation-par-led>. Consulté le 2020-03-25.