

EN QUOI CONSISTE LE LANGAGE SMALLTALK ?

Michel AUBÉ, Danièle BRACKE

C'est sans doute le représentant le plus connu de ce que l'on appelle les langages orientés-objet. Cette expression signifie que ce sont des objets qui constituent l'unité de représentation dans ces langages. A titre de comparaison, on pourrait considérer qu'en LOGO, l'unité de représentation est la procédure. En effet, programmer en LOGO revient essentiellement à construire des procédures, et c'est d'ailleurs l'information qui est préservée sur disquette à la fin d'une session.

En SMALLTALK, ce que l'on construit, ce que l'on manipule et ce que l'on emmagasine en mémoire, ce sont des objets. Nous examinerons donc ici en quoi consistent ces objets. Il nous faudra pour cela préciser un certain nombre de concepts-clés, comme ceux de classe, sous-classe, instance, méthode et message, qui s'avèrent fondamentaux pour la compréhension du langage.

Nous examinerons ensuite en quoi consiste la programmation en SMALLTALK.

A. OBJETS, CLASSES, INSTANCES

Qu'est-ce donc qu'un objet ? Il s'agit d'une entité caractérisée par un état et capable de répondre à certaines commandes ou opérateurs.

Si nous revenons temporairement à LOGO, nous pouvons considérer la tortue comme un objet dans le sens où elle a un état et répond à certains opérateurs : l'état de tortue est déterminé par sa position, son orientation, l'état de son crayon ; ses opérateurs sont les différentes primitives graphiques de LOGO : Avance, Droite, Lèvecrayon, Cap...

Ces primitives sont définies de telle sorte que LOGO "sait" qu'elles s'adressent implicitement à la tortue. En SMALLTALK, le nom d'une procédure est considéré comme un message et l'objet auquel est adressé ce message doit être spécifié explicitement

| | |
|---------------|--------------------|
| En Logo : | En Smalltalk |
| baisse crayon | tortue 1 down |
| avance 10 | tortue 1 go : 10 |
| droite 90 | tortue 1 turn : 90 |

Les nouvelles versions de LOGO qui disposent de plusieurs tortues obligent d'ailleurs l'utilisateur à s'adresser à chacune d'elles de façon explicite, tout comme il est nécessaire de le faire en SMALLTALK.

Toute programmation en SMALLTALK requiert donc que l'on s'adresse aux objets ayant les propriétés adéquates pour la tâche que l'on souhaite réaliser. Par exemple, si l'on veut faire de l'animation, les objets manipulés seront des formes, et chacune aura des caractéristiques de position, d'orientation, de grandeur, de couleur... Chaque forme saura en outre répondre à certains messages lui signifiant de s'afficher à l'écran, de se déplacer, de se rétrécir, de se dilater...

SMALLTALK contient déjà toute une variété d'objets utiles. Mais il est également possible de définir un nouvel objet. Il faut pour cela :

- identifier, sous forme de variables, l'état de l'objet, c'est-à-dire les caractéristiques qui lui sont essentielles.

- définir, sous forme de procédures, qu'on appelle méthodes en SMALLTALK, les messages auxquels l'objet devra savoir répondre.

Les objets, ce sont donc des entités essentiellement dynamiques. Cela évoque facilement l'idée de spécialistes ayant chacun leurs caractéristiques personnelles et leur savoir-faire. Nous avons alors pensé exploiter ce que nous avons appelé la métaphore des professions pour mieux transmettre les concepts-clés du langage. En effet, l'environnement SMALLTALK, c'est un peu comme une société d'individus où chacun a un métier et sait exécuter un certain nombre de tâches dans sa spécialité. Ainsi, on peut demander à un plombier d'installer la tuyauterie, à un électricien d'installer le système électrique, à un chirurgien de faire des opérations chirurgicales. En outre, ce sont les professions qui déterminent la nature des savoir-faire. Un même terme, "entailler" par exemple, prend une signification tout à fait différente selon que l'on s'adresse à un cultivateur dans son érablière, à un couturier devant le tissu dont il espère tirer une nouvelle mode, ou à un chirurgien près de sa table d'opération.

En SMALLTALK, les objets qui partagent un même savoirfaire sont regroupés en classe, comme des individus rattachés à une même profession : ainsi tous les médecins savent ausculter, faire une injection ou établir un diagnostic. Il y a cependant deux façons d'appartenir à une classe, comme sous-classe et comme instance, et c'est peut-être ici que la métaphore des professions s'avère la plus éclairante.

Une sous-classe, c'est en quelque sorte une spécialisation. Par exemple, pédiatre, urologue et chirurgien seraient trois sous-classes de médecin. Cela signifie qu'en plus de son savoir-faire de médecin, chaque nouvelle classe a des savoir-faire additionnels qui la distinguent des autres. Les classes sont donc organisées en hiérarchie. Les sous-classes héritent des méthodes de leur super-classe. Lorsqu'un message est envoyé à une sous-classe, la méthode correspondante est exécutée. Si cette méthode n'est pas trouvée dans le répertoire de la sous-classe, la recherche se fait automatiquement en remontant dans la hiérarchie. Lorsque deux classes différentes ont des méthodes qui portent le même nom, c'est la première méthode trouvée en remontant dans la hiérarchie qui est utilisée.

Une instance, par contre, cela correspond à un individu. Par exemple, Mme Gagnon et M. Hivon peuvent être tous les deux médecins et on pourrait alors s'adresser à l'un ou l'autre indifféremment pour n'importe quel geste qui requiert l'expertise d'un médecin. Le concept d'instance capte l'idée que l'on ne peut s'adresser à la médecine ou à la classe médecin en soi pour obtenir un traitement : c'est toujours à un individu particulier, représentant de sa profession, qu'il faut s'adresser. C'est pourquoi on appellera méthodes d'instance toutes celles qui captent le savoir-faire partagé par tous les individus d'une classe. Il est important de réaliser ici que l'instance d'une sous-classe est automatiquement une instance de la super-classe. Ainsi, un individu pédiatre est aussi un médecin qui sait répondre à toutes les méthodes d'instance de médecin.

Il existe également des méthodes de classe : celles-ci, adressées exclusivement aux classes, permettent non pas d'exécuter l'un ou l'autre des savoir-faire de la classe mais de créer de nouvelles instances. Dans la métaphore des professions, cela correspondrait en quelque sorte à la Faculté de médecine qui peut former de nouveaux médecins, lesquels seront alors en mesure d'exécuter les méthodes d'instance :

Un objet, nous l'avons vu, se caractérise par ses méthodes mais également par son état, qui est spécifié par la valeur de ses variables. Ici

LE BULLETIN DE L'EPI EN QUOI CONSISTE LE LANGAGE SMALLTALK ?

encore, il importe de distinguer variables d'instance et variables de classe.

Les variables de classe, partagées par toutes les instances d'une classe, gardent la même valeur pour chacune d'entre elles. A titre d'exemple, dans la classe Date, les noms des mois de l'année et des jours de la semaine sont des variables de classe dont la valeur est déterminée une fois pour toutes pour chacune des instances.

Les variables d'instance ont une valeur particulière pour chaque instance. Elles caractérisent en propre l'état de celle-ci. Une instance de la classe Date, par exemple, sera spécifiée complètement par deux variables : l'année et la position du jour dans l'année.

B. PROGRAMMER EN SMALLTALK

Programmer en SMALLTALK peut signifier trois choses différentes : modifier un objet existant, créer un nouvel objet ou utiliser dans un espace de travail les objets déjà définis.

1) On peut en effet ajouter une méthode à une classe ou modifier un méthode déjà existante. Cela ressemble à la définition ou à l'édition d'une procédure dans les langages plus classiques. On peut par exemple doter la classe Integer d'une méthode permettant d'interroger si l'une de ses instances est un nombre premier.

2) On peut également créer un nouvel objet, ce qui, en SMALLTALK ; signifie toujours rattacher une sous-classe à une classe existante. Il faut alors se demander à quelle classe ce nouvel objet sera rattaché et par quelles variables ou caractéristiques ses instances seront identifiées. C'est un peu comme construire une structure de données. Supposons par exemple qu'il existe déjà une classe Droite, dont chaque instance est caractérisée par un point et une orientation. Si l'on veut définir une classe Segment, il est intéressant d'en faire une sous-classe de Droite, puisque l'on pourra profiter automatiquement de tous les savoir-faire de la super-classe : normale à une droite, distance à une droite, intersection avec une autre droite... Il faut également se demander quelles variables d'instance, comme par exemple origine et extrémité, pourront servir à distinguer chaque segment. On lui ajoutera finalement des méthodes spécifiques, comme médiatrice, qui ne pourraient avoir de sens pour les instances de la super-classe.

3) Finalement, on peut faire surgir dans l'espace de travail un certain nombre d'objets. Ce type de programmation correspond à la réalisation d'un projet, comme celui de produire une animation à l'écran. Il faut alors identifier les classes dont on aura besoin Display, Rectangle, Form, OpaqueForm, Delay, OrderedCollection... Ces classes existent-elles toutes déjà en SMALLTALK ? Sinon, il faut créer celles qui manquent... on rédige ensuite la séquence d'instructions constituées essentiellement de messages envoyés aux divers objets retenus. On doit en général faire appel à un certain nombre de variables temporaires pour désigner les instances des classes utilisées.

Plus qu'un langage, SMALLTALK constitue en fait un environnement dynamique extrêmement vaste et diversifié. C'est dire qu'en plus de permettre la création et la manipulation d'objets, il met à la disposition de l'utilisateur toute une panoplie d'instruments lui facilitant l'exploration de cet univers complexe, notamment les outils permettant d'interroger le système et la hiérarchie des classes, d'assurer une édition souple et rapide, de repérer facilement les erreurs et d'examiner à tout moment en profondeur les caractéristiques de n'importe quel objet.

Michel Aubé
Danièle Bracke

NDLR. Nous avons extrait cette présentation particulièrement claire du langage SMALLTALK du rapport de recherche. "APPRENDRE À UTILISER L'ENVIRONNEMENT SMALLTALK-80 AU SECONDAIRE" présenté, par Michel Aubé et Danièle Bracke, dans le numéro 47 (déc. 87) de *BIP-BIP* (Direction des ressources didactiques - 600 rue Fullum - MONTRÉAL H2K-4L1).