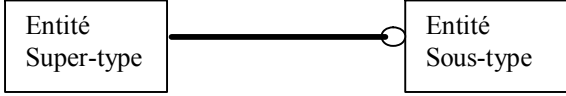
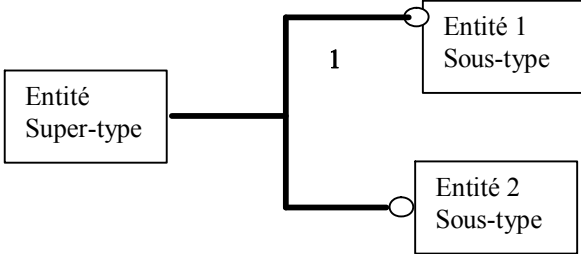
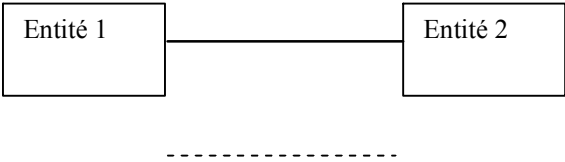
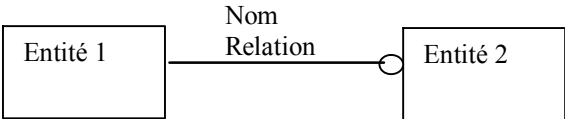
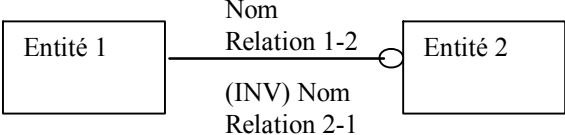
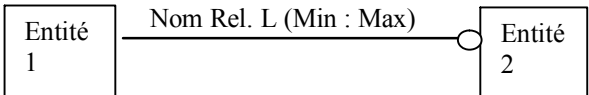
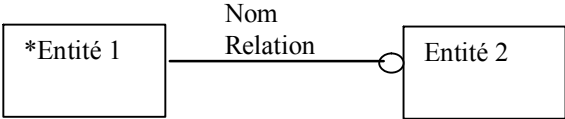


Concept et syntaxe Express-G	Commentaires
<p><u>Entité(entity-name)</u></p> <div style="border: 1px solid black; width: 150px; height: 30px; margin: 10px auto; text-align: center;">Nom de l'entité</div>	<p><u>Classe</u> d'objets dans UML. <u>Concept</u> dans NIAM</p> <p>L'entité peut être un <u>attribut</u></p>
<p><u>Type défini (new-defined-data-type)</u></p> <div style="border: 1px dashed black; width: 200px; height: 30px; margin: 10px auto; text-align: center;">Nom du type défini</div>	<p>Type d'entité très utilisé dans les IFC Correspond au type défini de l'arborescence en annexe, qui est une spécialisation de la classe décrite, et dont le nom est normalisé par les IFC ou les utilisateurs.</p>
<p><u>Type de données construit :</u> SELECT et ENUMERATION</p> <div style="border: 1px dashed black; width: 150px; height: 30px; margin: 10px auto; text-align: center;">Nom donnée SELECT</div> <div style="border: 1px dashed black; width: 150px; height: 30px; margin: 10px auto; text-align: center;">Nom donnée ENUMERATION.</div>	<p>Donne des indications sur la manière dont on construit ces données par sélection ou énumération à partir d'une classe.</p> <p>Pour les informaticiens</p>
<p><u>Type de données simple</u> Par exemple :</p> <div style="border: 1px solid black; width: 150px; height: 25px; margin: 10px auto; text-align: center;">INTEGER</div>	<p>Indication sur le type de données pour les informaticiens (type entier, réel, chaîne de caractère)</p>
<p><u>Relations</u> On distingue dans l'ordre d'importance : Les relations d'héritage Les relations entre entité Les relations de type attribut</p>	<p>On retrouve les mêmes concepts dans NIAM et UML, avec des nuances</p>

Concept et syntaxe Express-G	Commentaires
<p><u>Héritage simple</u></p> 	<p>le sens de la relation est donné par Le petit cercle placé contre Le rectangle de l'entité sous-type</p> <p>Le trait de la relation est épais</p>
<p><u>Héritage avec contrainte d'exclusion</u></p> 	<p>Pour exprimer que l'on doit choisir un seul des deux descendants à la fois, (contraintes d'exclusion dans NIAM) On marque le chiffre un sur l'arborescence Comme dans NIAM et UML, l'héritage Multiple n'est pas formalisé.</p>
<p><u>Relation simple</u></p> 	<p>Dans le cas général, c'est un trait Mince entre les deux entités. Sans autre indication, il n'y a pas de sens privilégié dans la relation Exemple : relation entre un objet et un Attribut. Pas de nom à cette relation. Trait en pointillé si attribut optionnel.</p>
<p><u>Relation à sens prioritaire</u></p> 	<p>C'est la majorité des cas. On précise toujours le nom de la relation dans le sens prioritaire, qui s'identifie ainsi à l'un des deux Rôle dans NIAM</p>
<p><u>Relation qualifiée dans les 2 sens</u></p> 	<p>On précise en plus le nom de la relation dans le sens inverse au sens prioritaire. Ainsi l'information est équivalente aux deux Rôle de NIAM ou UML. Mettre le nom sous le trait, précédé du sigle (INV)</p>
<p><u>Contraintes de cardinalités</u></p> 	<p>Par défaut, au moins 1, donc plusieurs. C'est différent de NIAM (0 autorisé). En général, après le nom de la relation, on place entre parenthèse les cardinalités minimales et maximales, précédé d'une lettre dont le sens est réservé aux informaticiens : Tableau, Liste ... ? signifie que le Max n'est pas défini.</p>

Concept et syntaxe Express-G	Commentaires
<p data-bbox="284 338 756 371"><u>Contrainte autres que cardinalités</u></p>  <pre data-bbox="274 450 841 568">graph LR; E1[*Entité 1] --- Nom Relation E2[Entité 2];</pre>	<p data-bbox="948 338 1394 483">Quand une entité fait l'objet d'une Règle dans une relation, on inscrit un astérisque précède le nom de de l'entité objet ou attribut</p>