Integraal Ontwerpen

IO-Open Mind

IO-Open Mind
Uitvoeringsfase
Leidraad informatiemodel

Datum: 1 juni 2004

Versie: 3

Door: B. Bink (TLO)

INHOUDSOPGAVE

Inleiding

Het informatiemodel is ontstaan onder activiteit 5 ‘scheepsbouw informatiemodel’ uit het plan van aanpak ‘Specificatie vervolgprojecten IO-Open Mind Uitvoeringsfase’, en wel op activiteit 5.1. Doel en scope van deze startversie en de uiteindelijk daaruit volgende definitieve versie van het scheepsbouw informatiemodel zijn gedefinieerd in het ‘Werkplan Generieke Modelbouw’.

Om tot het informatiemodel te komen is als referentiekader gebruik gemaakt van volgend bestaand materiaal:

- Het productmodel
- Het procesmodel
- Functionele eisen voor SeaQuipment 2
- DelftBase 1
- ETIM Plus
- STEP*lib*

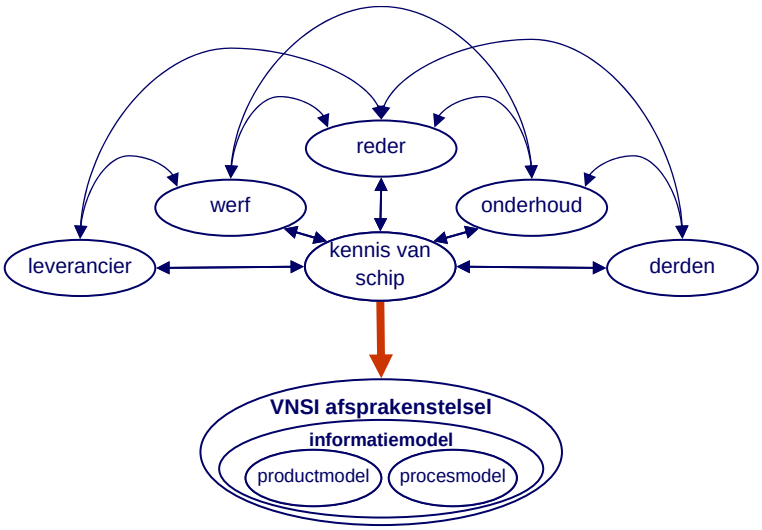
Deze leidraad is bedoeld om de werven te ondersteunen bij het toepassen van het informatiemodel.

Het toepassen van het informatiemodel kan over de hele keten. Binnen de scheepsbouw als keten worden, zoals in iedere keten, veel digitale gegevens uitgewisseld over een te bouwen schip, de processen er om heen en de organisaties. Het informatiemodel biedt een raamwerk om afspraken over het product en procesinformatie in vast te leggen. Dit raamwerk dient als uitwisselingsformaat van die informatie.

Het informatiemodel is bedoeld als basis architectuur voor het ondersteunen van informatie-uitwisseling en eventueel opslag.

1. Doelstelling van het informatiemodel

Een informatiemodel biedt de mogelijkheid digitale gegevens te integreren. Het doel van het informatiemodel is dan ook hergebruik van kennis door het eenduidig en expliciet structureren van kennis van de scheepsbouw ter bevordering van (her)gebruik, uitwisseling en delen ervan.

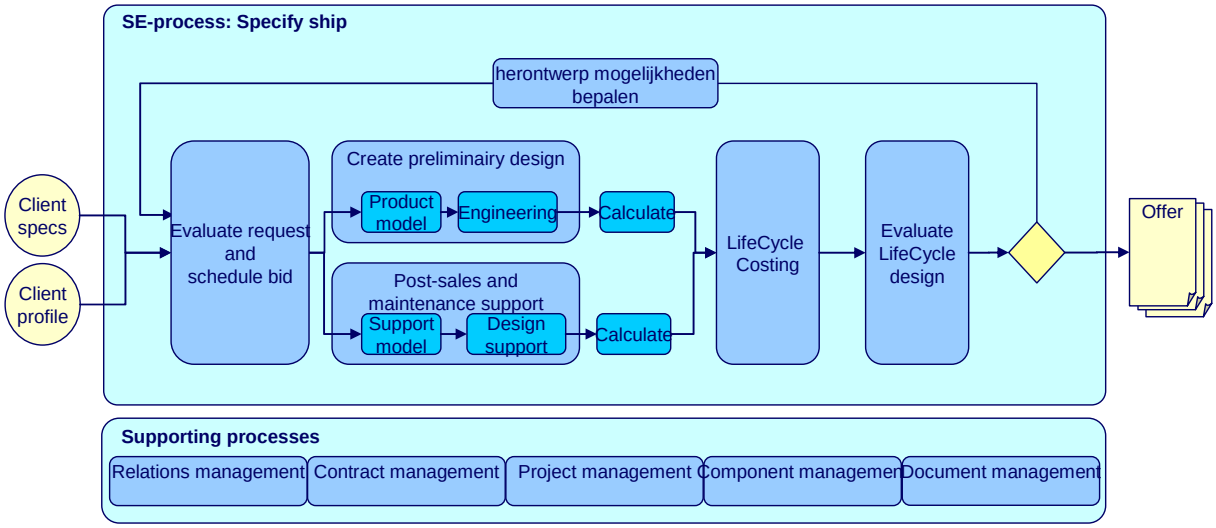


Het informatiemodel is een formele beschrijving van de uit te wisselen informatie ter ondersteuning van het Sales Engineering proces.

2. Uitgangspunten

Er is een aantal uitgangspunten en randvoorwaarden waar vanuit is gegaan bij het maken van het informatiemodel. De randvoorwaarden zijn:

- Het model dient een afsprakenstelsel tussen meerdere partijen.
- Het model dient als een middel voor het uitwisselen van informatie.
- Het model dient onderhoudbaar te zijn
- Het model dient aan te sluiten bij (ISO) standaards
- Het model legt vast wat er wordt uitgewisseld en wat de betekenis ervan is.



Figuur 2-1 Het Sales Engineering proces diagram

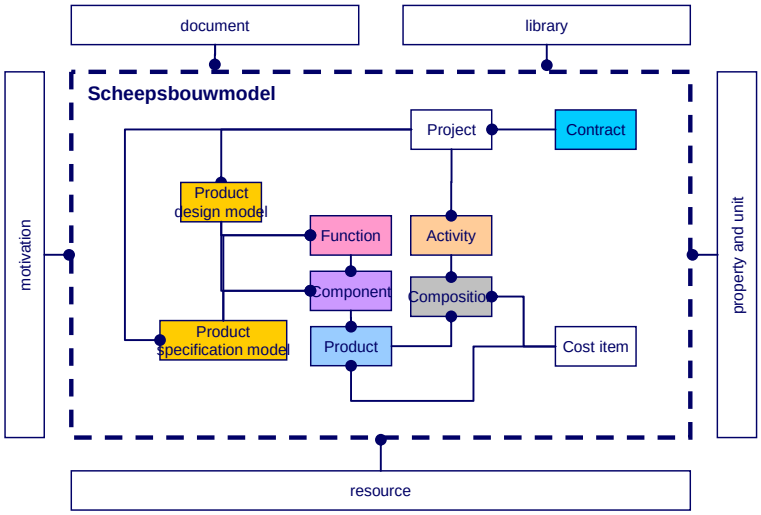
De uitgangspunten zijn de volgende:

- Het model beperkt zich tot de processen binnen Sales Engineering (aanbiedingsfase)
- Binnen Sales engineering worden de volgende processen gedekt
  - Product specificatie
  - Product engineering
  - Product management
  - Cost calculation
  - Procurement
  - Projectmanagement (beperkt)

3. Leeswijzer

3.1. Model organisatie

Hieronder is een zeer basale weergave gegeven van het informatiemodel. De rechthoeken binnen de stippellijn zijn hoofdelementen. De rechthoeken er omheen zijn ondersteunend. De lijnen tussen de rechthoeken zijn associaties met namen als ‘is defined by’ en ‘is grouped by’.



Figuur 3-1 Abstracte weergave van het informatiemodel

Het project dient als kapstok waaraan productspecificaties en een productontwerp, een component genoemd, wordt gehangen. Een component vervult een functie en per component wordt een product gekocht of gemaakt. Het component is dus het ontwerp en het product symboliseert het daadwerkelijke product.

Het informatiemodel bestaat uit 16 onderwerpen

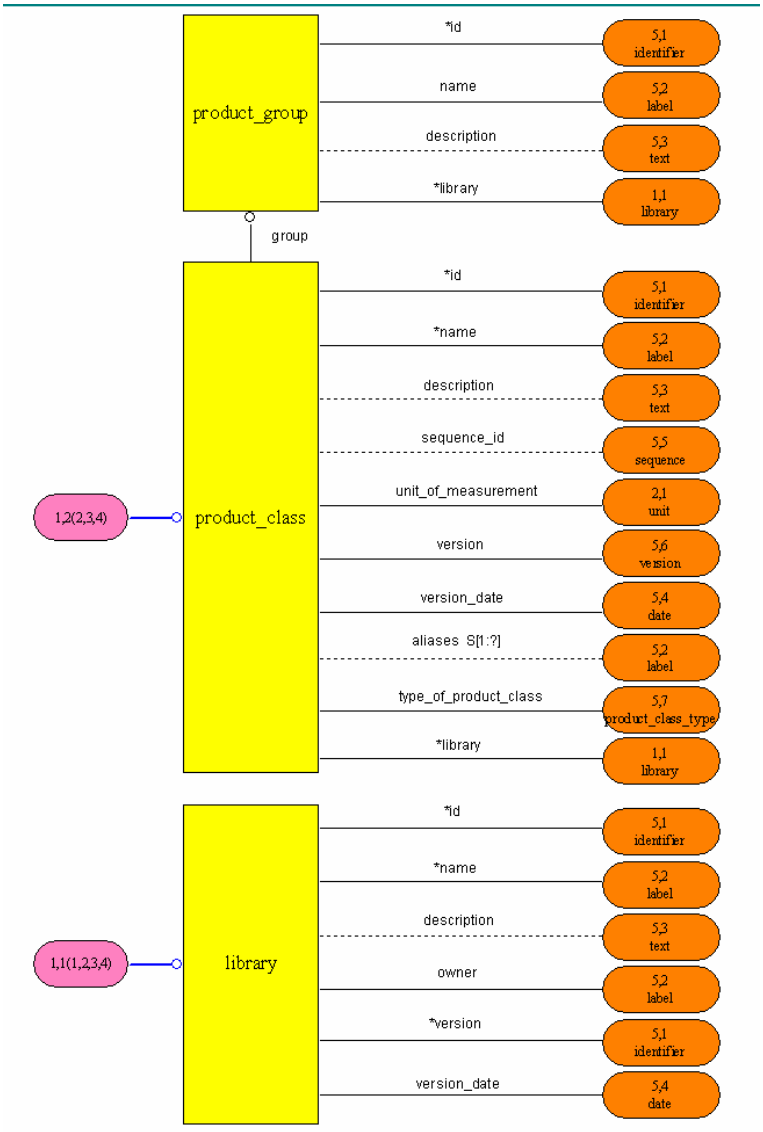
- Activity
- Costing and procurement
- Document
- Engineering
- Library
- Motivation
- Product class
- Product
- Project
- Property
- Resource
- Ship grouping
- Specification
- Symbol
- Terms an synonyms
- Unit

De modulen zijn zo opgezet dat alle objecten per onderwerp gegroepeerd zijn. De onderwerpen zijn zo gekozen dat de elementen van de onderwerpen onafhankelijk van de elementen uit andere onderwerpen kunnen worden toegepast en onderhouden.

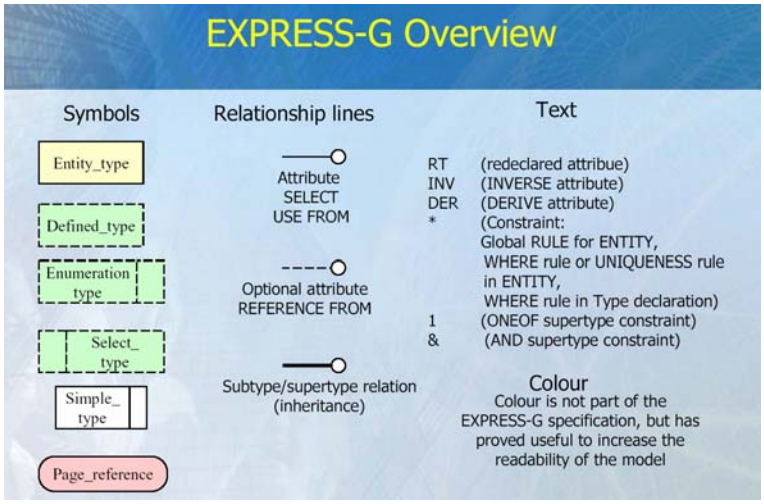
3.2. Korte Express toelichting

Express modellen kunnen middels Express-G worden getoond. Express-G is niets anders dan een grafische weergave van entiteiten en hun onderlinge samenhang. Om een eenduidige weergave te kunnen maken, zijn er een aantal tekenafspraken gemaakt. Een entiteit (geel rechthoekje) heeft attributen. De attributen worden voorgesteld door lijnen. Als een attribuut optioneel is, is de lijn gestippeld. Een attribuut kan worden uitgedrukt in een andere entity of in een defined type. Een defined type is bijvoorbeeld string of number. Dit betekent dat het attribuut kan worden uitgedrukt in respectievelijk een woord of een nummer.

De figuur hiernaast dient puur als voorbeeld.



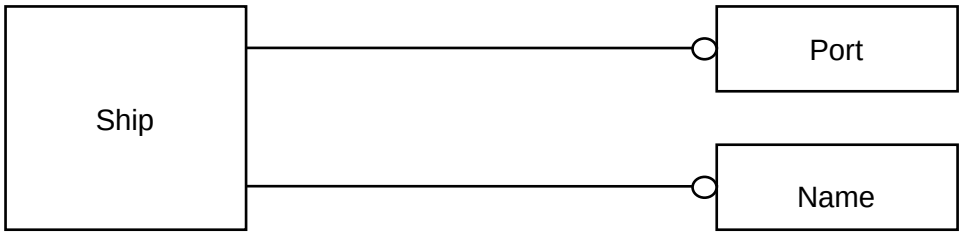
Voor verdere toelichting, zie Figuur 1)



Figuur 1 Toelichting op Express-G

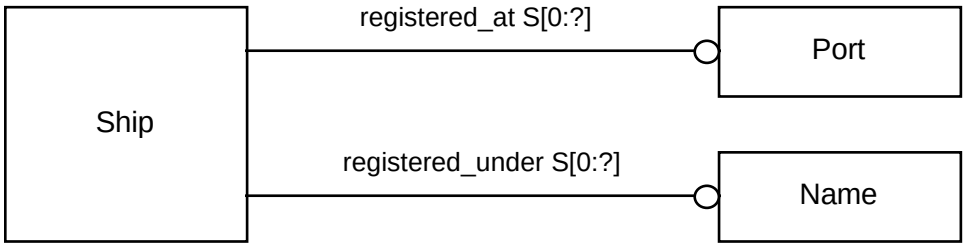
3.3. Modelleertechniek: Associatief modelleren

Figuur 3-2 toont date en schip geregistreerd is bij maximaal één haven en onder maximaal één naam.



Figuur 3-2 Overdraagbare relaties

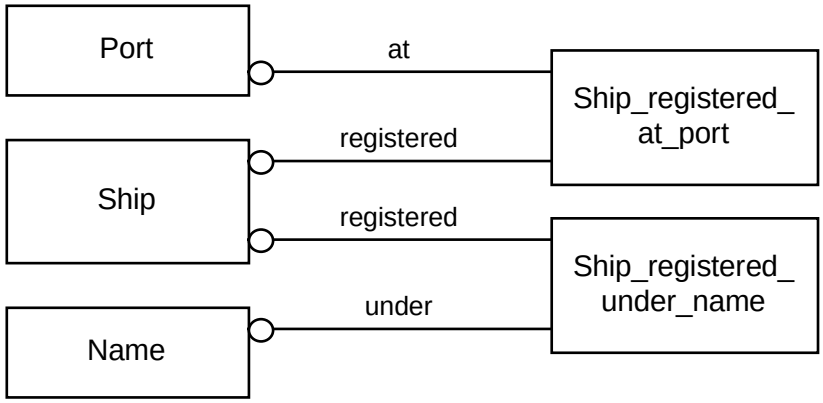
Maar wat gebeurt er als een schip opnieuw wordt geregistreerd onder een andere naam of bij een andere haven? Hoe weet je dan zijn vorige naam? Als de naam verandert of de haven waar het schip is geregistreerd hoe weet je dan dat het refereer dan een oude klant?



Figuur 3-3 Juiste cardinaliteiten voor overdraagbare relaties

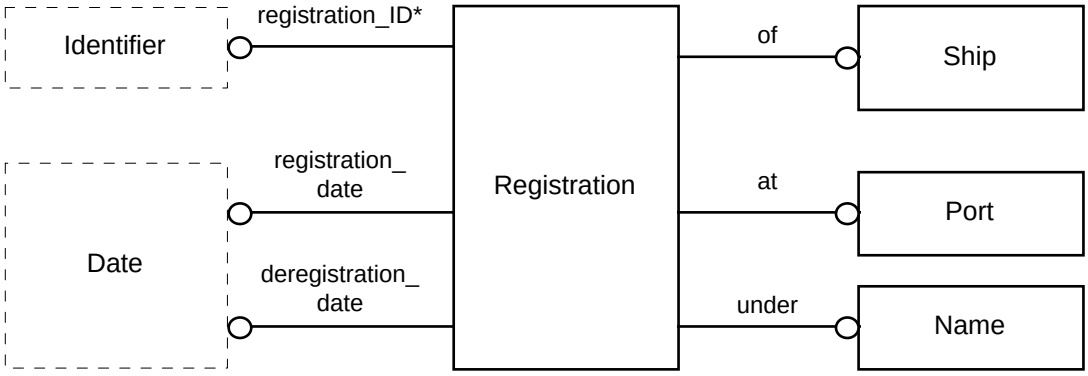
Figuur 3-3 toont de juiste cardinaliteiten, ‘many tot many’, waarin is verwerkt dat een schip een naam heft op een bepaald tijdstip, maar over zijn levensduur meerdere namen kan hebben. Het probleem zat hem in het modelleren. Er was uitgegaan van de zakelijke context waarin een schip wordt aangeduid door zijn naam en haven waar het geregistreerd is in plaats van de context van het principe wat er achter schuilt.

De oplossing voor de ‘many to many’ relaties is het verheffen van de relatie tot een ‘entity’ of object zoals weergegeven in Figuur 3-4. Een tweede voordeel is dat er nu ook extra informatie aan het relatieobject gehangen kan worden.



Figuur 3-4 Oplossing voor 'many to many' relaties

In het voorbeeld is de activiteit van het registreren zelf dat leidt een registratie waarin beide relaties gelegd worden. Daarmee kunnen de twee relaties vervangen worden door het ‘registration’ object weergegeven in Figuur 3-5.



Figuur 3-5 Het begrip dat een activiteit leidt tot een relatie

Het object registratie heeft zijn eigen id. De relaties met de andere objecten zijn benoemd volgens de rol die de andere objecten spleen.

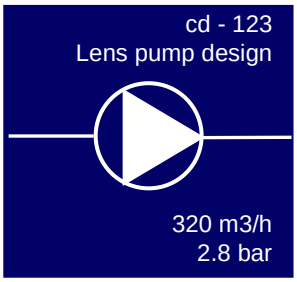
4. Verificaties bij pilots

4.1. Toepassen van het informatiemodel

Hoe het informatiemodel kan worden toegepast kan het beste aan de hand van een voorbeeld duidelijk gemaakt worden.

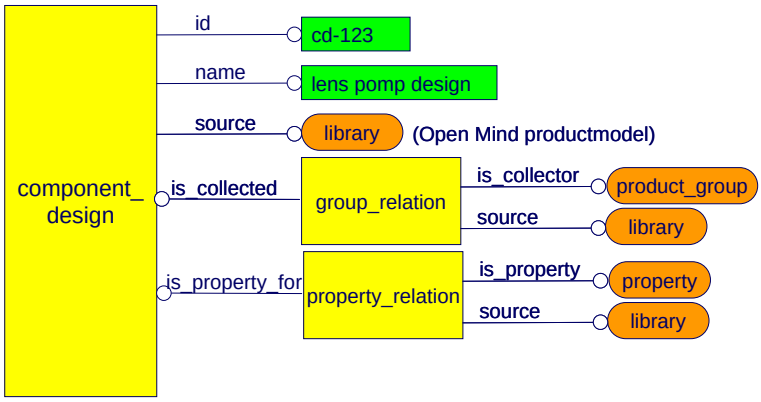
Stel in het Sales Engineeringsproces wordt een pomp ge-allocceerd. Op dat moment zijn de volgende gegevens bekend:

Pomp behoefte werf: 320 m3/h at 2.8 bar. Het gaat hier om een component die de functie verpompen vervult.

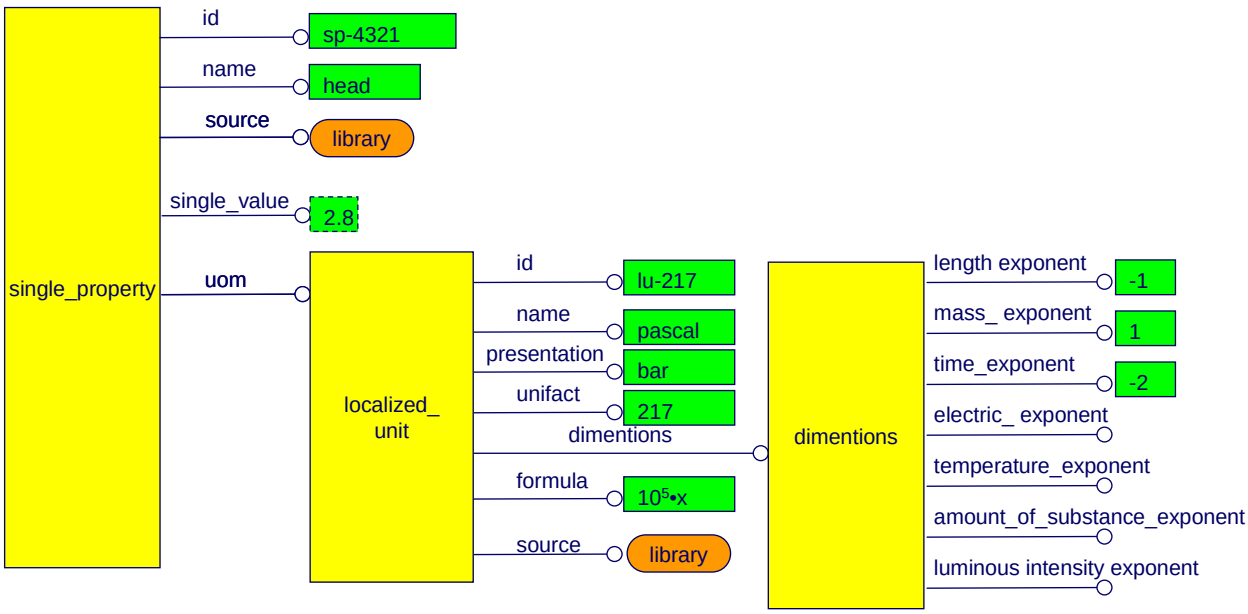


Figuur 4-1 functionele pomp uit lenspompschema

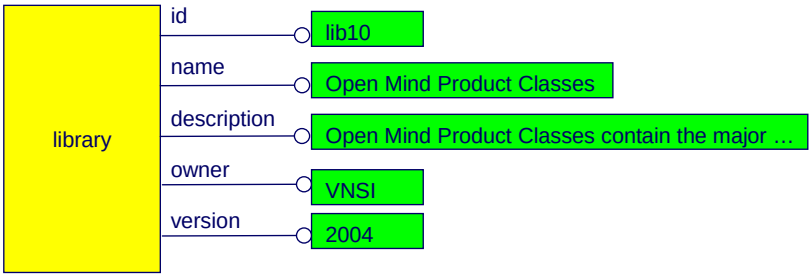
Als de informatie wordt weergegeven conform het informatiemodel ziet het er grafisch uit zoals hieronder weergegeven.



Figuur 4-2 Express view op de pomp component.



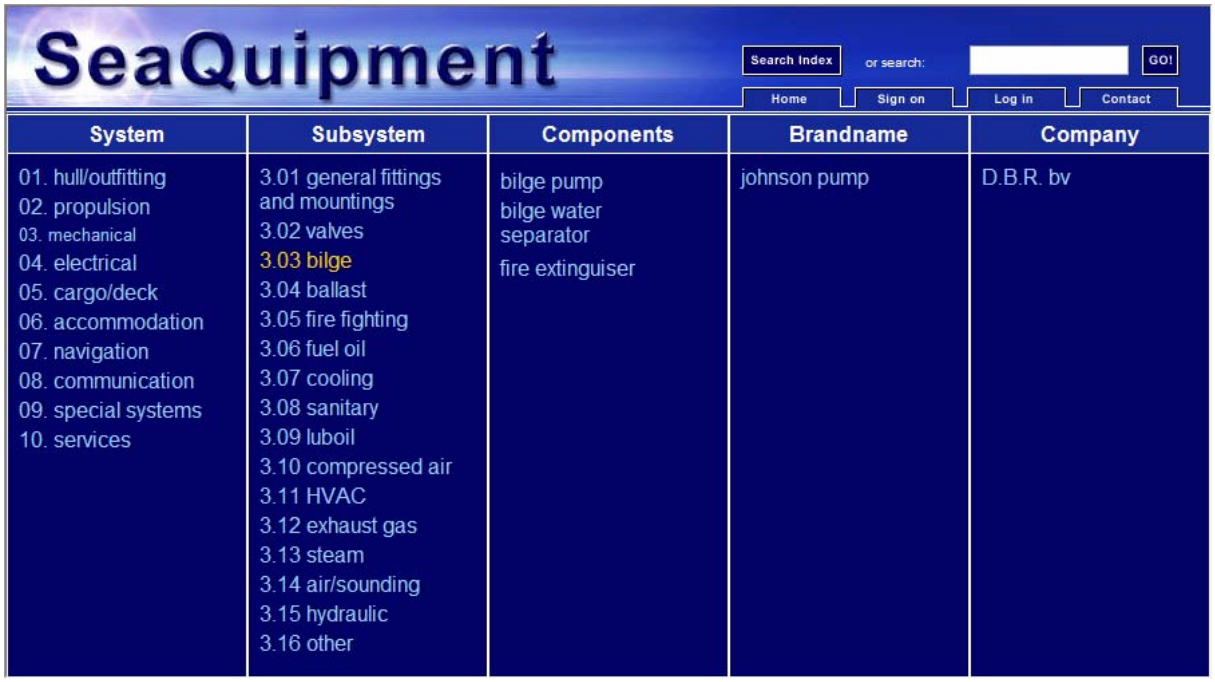
Figuur 4-3 Express view op de opvoerhoogte van de pomp component.



Figuur 4-4 eigenaar van een object wordt vastgelegd in een 'library'

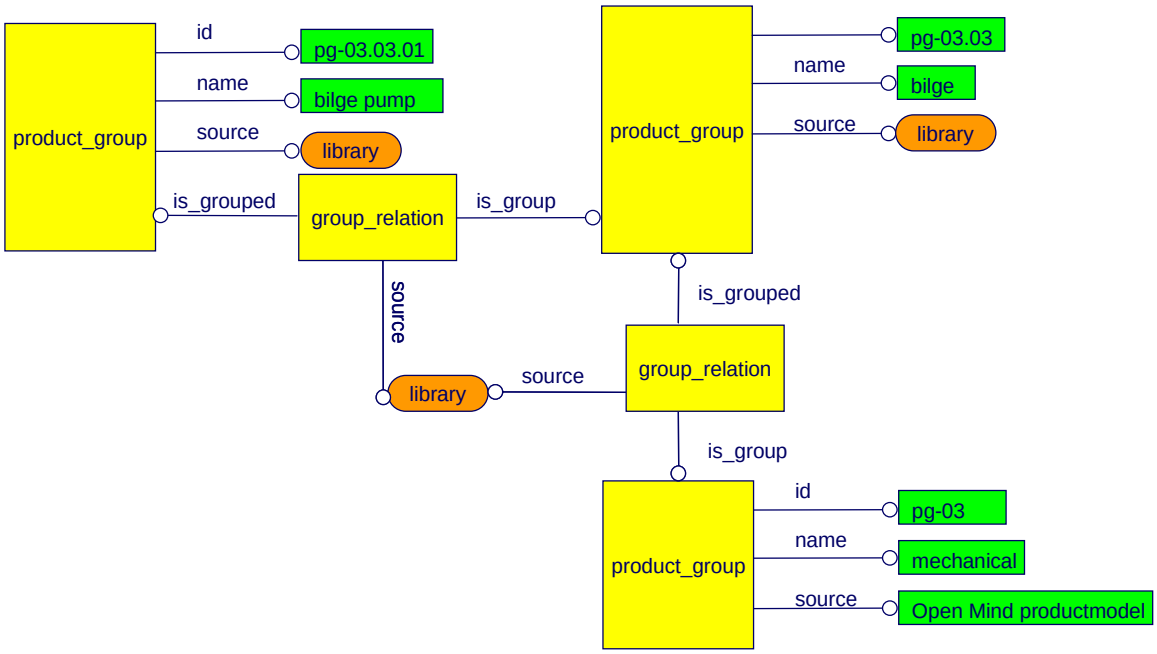
De uiteindelijke pomp die via SeaQuipment wordt besteld is hieronder weergegeven.

Figuur 4-5 Foto van de bestelde pompsoort.



Figuur 4-6 Het zoekmechanisme van SeaQuipment

In het informatiemodel is de SeaQuipment 'classificatie' middels een product group constructie gedekt. In Express ziet de bovenstaande er grafisch als volgt uit:



Figuur 4-7 SeaQuipment groepering van de ballast pump

De gegevens die SeaQuipment over de pompgroep geeft is:

- 350 m3/h at 3 bar
- e-motor 45 KW at 1450 rpm
- Groups: mechanical>bilge>bilge

De extra informatie betreft hier de gegevens over de elektromotor.

De leverancier heeft uiteraard nog veel meer informatie over de pomp:

- leverancier: REIKON B.V.
- fluid: seawater
- purpose: ballast, sea cooling water, fire-fighting, etc.
- execution: vertical
- body material: bronze
- impeller material: bronze
- shaft material: stainless steel
- seal: mechanical
- flow rate: 350 m3/h
- head: 3 bar
- e-motor: 45 KW at 1450 rpm

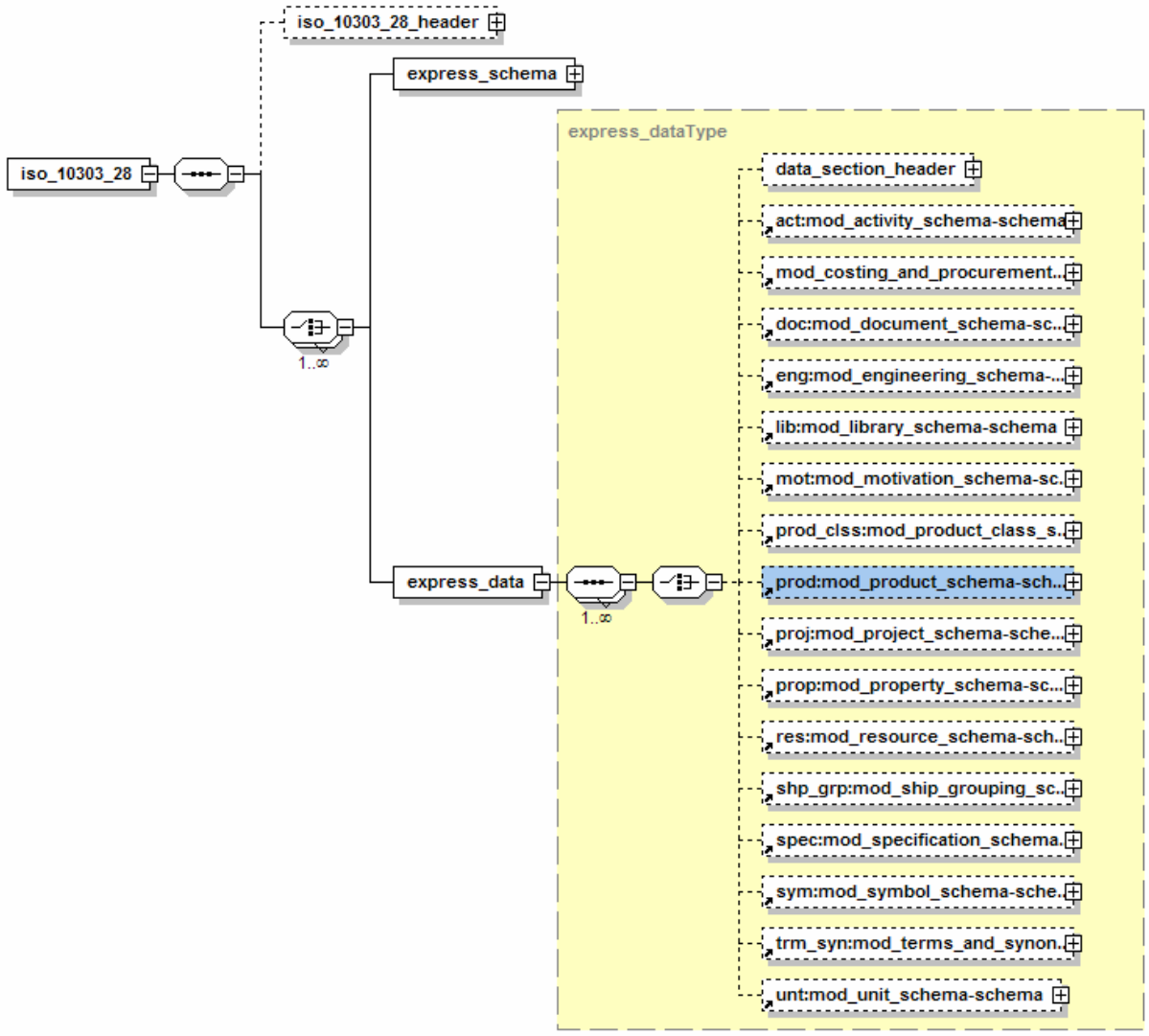
Al deze gegevens kunnen worden vastgelegd conform het informatiemodel.

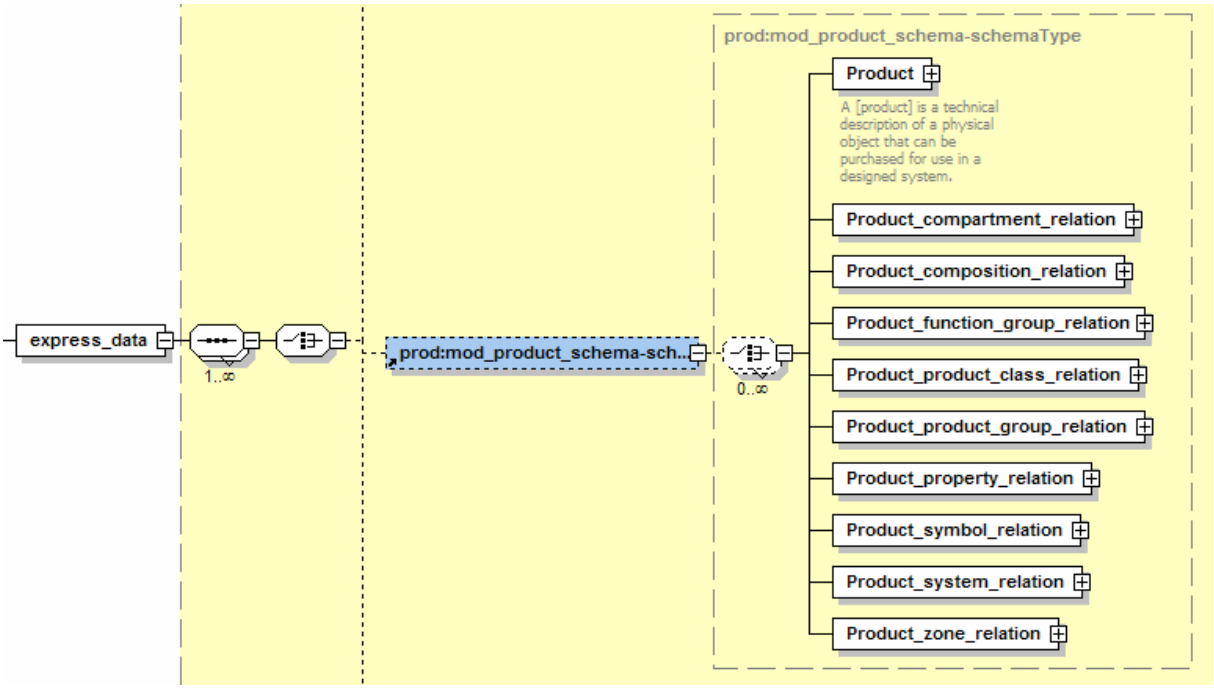
4.2. Vertalingen van Express

Het voordeel van Express is dat het door de computer geïnterpreteerd kan worden en vertaald. Een aantal vertalingen is vastgelegd in de ISO STEP standaard. Hieronder valt ook de XML binding. De opbouw is hieronder weergegeven. Het geselecteerde schema is de ‘root’.

import	loc:MOD_COSTING_AND_PROCUREMENT_SCHEMA.xsd
import	loc:MOD_DOCUMENT_SCHEMA.xsd
import	loc:MOD_SYMBOL_SCHEMA.xsd
import	loc:MOD_ENGINEERING_SCHEMA.xsd
import	loc:MOD_ACTIVITY_SCHEMA.xsd
import	loc:MOD_PROJECT_SCHEMA.xsd
import	loc:MOD_PRODUCT_SCHEMA.xsd
import	loc:MOD_TERMS_AND_SYNONYMS.xsd
import	loc:MOD_PROPERTY_SCHEMA.xsd
import	loc:MOD_PRODUCT_CLASS_SCHEMA.xsd
import	loc:MOD_UNIT_SCHEMA.xsd
import	loc:MOD_SPECIFICATION_SCHEMA.xsd
import	loc:MOD_LIBRARY_SCHEMA.xsd
import	loc:MOD_MOTIVATION_SCHEMA.xsd
import	loc:MOD_SHIP_GROUPING_SCHEMA.xsd
import	loc:MOD_RESOURCE_SCHEMA.xsd
element	**iso_10303_28**
complexType	**iso_10303_28_headerType**
complexType	**data_section_headerType**
complexType	**express_dataType**
complexType	**external_refidType**
complexType	**express_schemaType**

Deze ziet er grafisch als vogt uit:


iso_10303_28_header
express_schema
express_dataType
iso_10303_28
data_section_header
act:mod_activity_schema-schema
mod_costing_and_procurement...
doc:mod_document_schema-sc...
eng:mod_engineering_schema-...
lib:mod_library_schema-schema
mot:mod_motivation_schema-sc...
prod_clss:mod_product_class_s...
express_data
prod:mod_product_schema-sch...
proj:mod_project_schema-sche...
prop:mod_property_schema-sc...
res:mod_resource_schema-sch...
shp_grp:mod_ship_grouping_sc...
spec:mod_specification_schema...
sym:mod_symbol_schema-sche...
trm_syn:mod_terms_and_synon...
unt:mod_unit_schema-schema
1..∞
1..∞



De XML vertaling van het Express model wordt uiteindelijk gebruikt voor de uitwisseling van de informatie. De technische voorkeur gaat echter uit naar een uitwisseling in ISO 10303 part 11. Doordat XML algemeen goed is geworden en laagdrempelig is als uitwisselingstaal, is gekozen voor de XML variant. Dit is de ISO 10303 part 28 ETEB binding.

Bijlage 1: Opmerkingen ten aanzien van het informatiemodel

Het informatiemodel is opgebouwd uit bestaande ISO 10303 parts en aangevuld om de behoefte te dekken voor het Sales Engineering proces. Het is echter zo dat de ISO 10303 standaard (AP221) waarop veel is gebaseerd, een eigen ISO data model krijgt. AP221 is gebouwd op het Epistle core model versie 3.1. welke nu als versie 4.5 een ISO certificaat zal krijgen onder nummer 15926.

Tevens is ISO 10303 AP233 in ontwikkeling die een framework moet worden voor de uit te wisselen informatie in de keten. In het bijzonder worden de ontwerpspecificaties gedekt. Het Open Mind informatie model zal hier op aan moeten gaan sluiten.

Op de CD-ROM zal het informatiemodel in Express en ISO 10303 part 28 ETEB binding vertaling beschikbaar zijn. Ook is er een SQL binding beschikbaar. Deze laatste is door Open Mind zelf bepaald en dient louter als voorbeeld.

Bijlage 2: Informatiemodel in Express

EXPRESS-G diagrams MOD_ACTIVITY_SCHEMA

TLO Holland Controls B.V- 06.05.2004

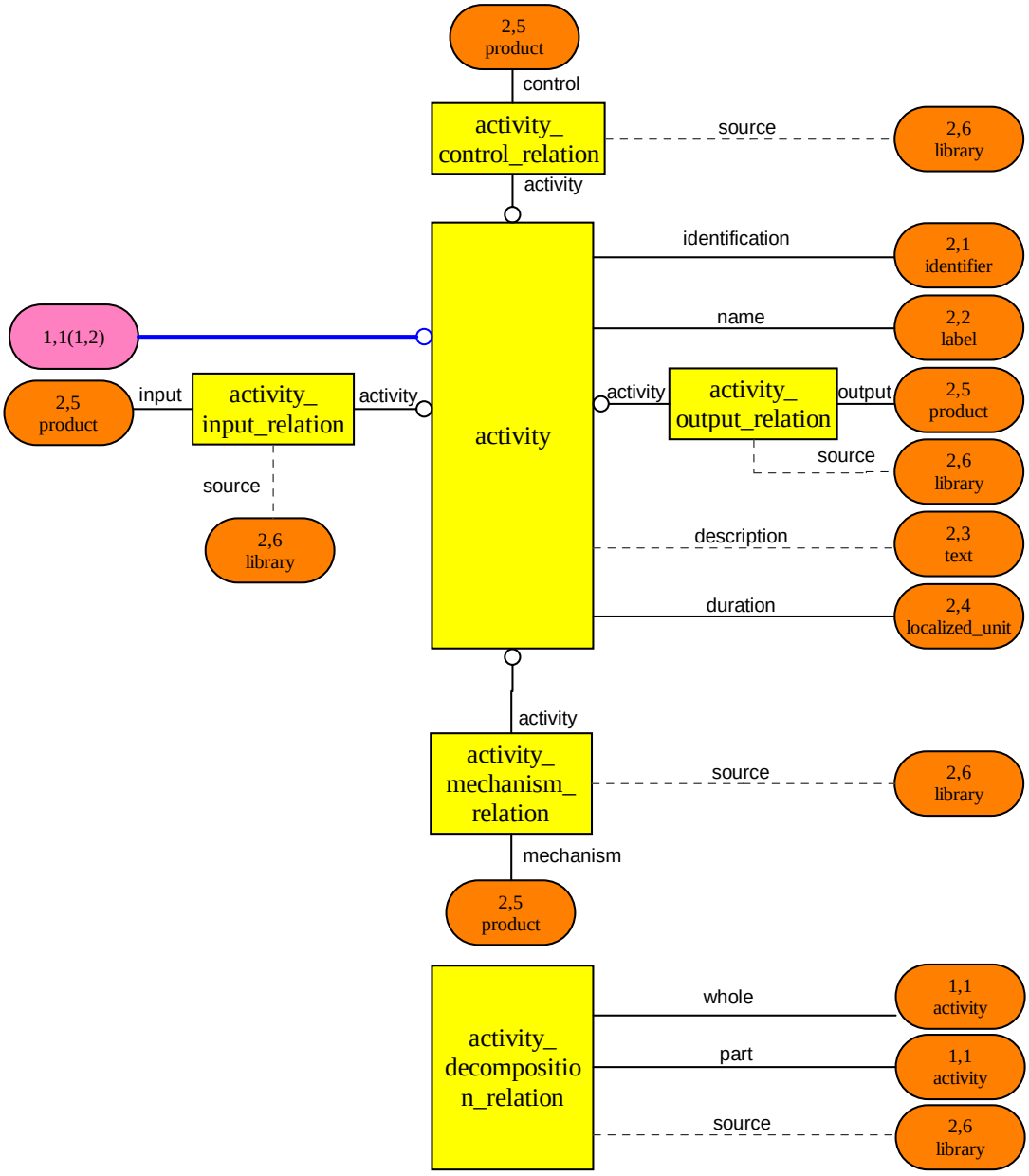


Figure D.1 - MOD_ACTIVITY_SCHEMA EXPRESS-G diagram 1 of 2

TLO Holland Controls B.V- 06.05.2004

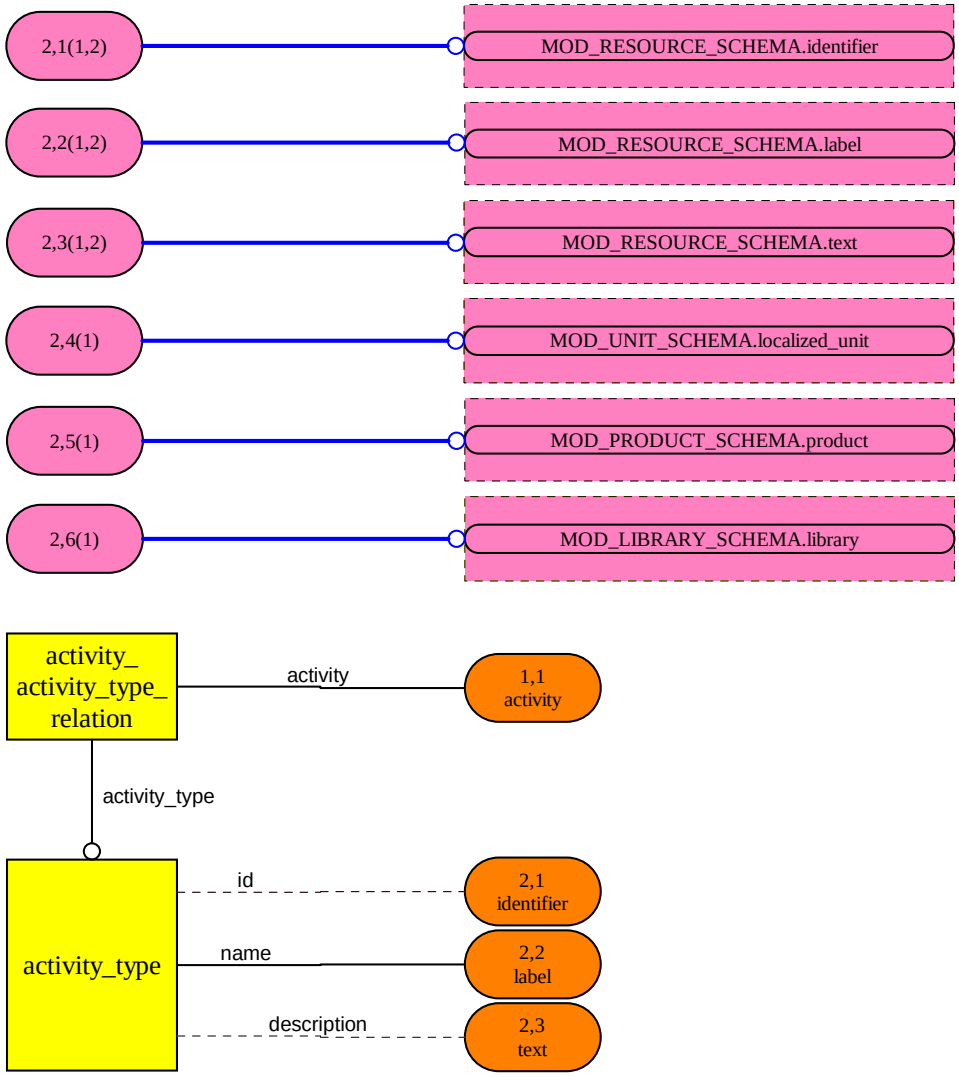


Figure D.2 - MOD_ACTIVITY_SCHEMA EXPRESS-G diagram 2 of 2

*)
(*

5. MOD_ACTIVITY_SCHEMA

EXPRESS specification:

```
SCHEMA MOD_ACTIVITY_SCHEMA;
  REFERENCE FROM MOD_LIBRARY_SCHEMA
    (library);
  REFERENCE FROM MOD_PRODUCT_SCHEMA
    (product);
  REFERENCE FROM MOD_RESOURCE_SCHEMA
    (identifier,
    label,
    text);
  REFERENCE FROM MOD_UNIT_SCHEMA;
(*
```

5.1. activity

An [activity] is performing an operation.

EXPRESS specification:

```
*)
  ENTITY activity;
    identification            :identifier;
    name                      :label;
    description               :OPTIONAL text;
    duration                  :localized_unit;
  END_ENTITY;
(*
```

5.2. activity_activity_type_relation

EXPRESS specification:

```
*)
  ENTITY activity_activity_type_relation;
    activity_type             :activity_type;
    activity                  :activity;
  END_ENTITY;
(*
```

5.3. activity_control_relation

EXPRESS specification:

```
*)
  ENTITY activity_control_relation;
    control                   :product;
    activity                  :activity;
    source                    :OPTIONAL library;
```

```
  END_ENTITY;
(*
```

5.4. activity_decomposition_relation

EXPRESS specification:

```
*)
  ENTITY activity_decomposition_relation;
    whole                       :activity;
    part                        :activity;
    source                      :OPTIONAL library;
  END_ENTITY;
(*
```

5.5. activity_input_relation

EXPRESS specification:

```
*)
  ENTITY activity_input_relation;
    input                       :product;
    activity                    :activity;
    source                      :OPTIONAL library;
  END_ENTITY;
(*
```

5.6. activity_mechanism_relation

EXPRESS specification:

```
*)
  ENTITY activity_mechanism_relation;
    mechanism                   :product;
    activity                    :activity;
    source                      :OPTIONAL library;
  END_ENTITY;
(*
```

5.7. activity_output_relation

EXPRESS specification:

```
*)
  ENTITY activity_output_relation;
    output                      :product;
    activity                    :activity;
    source                      :OPTIONAL library;
  END_ENTITY;
(*
```

5.8. activity_type

An [activity_type] is a typical activity.

EXPRESS specification:

```
*)
   ENTITY activity_type;
     id                          :OPTIONAL identifier;
     name                        :label;
     description                 :OPTIONAL text;
   END_ENTITY;
END_SCHEMA;
(*
```

EXPRESS-G diagrams
MOD_COSTING_AND_PROCUREMENT_SCHEMA

TLO Holland Controls B.V- 06.05.2004

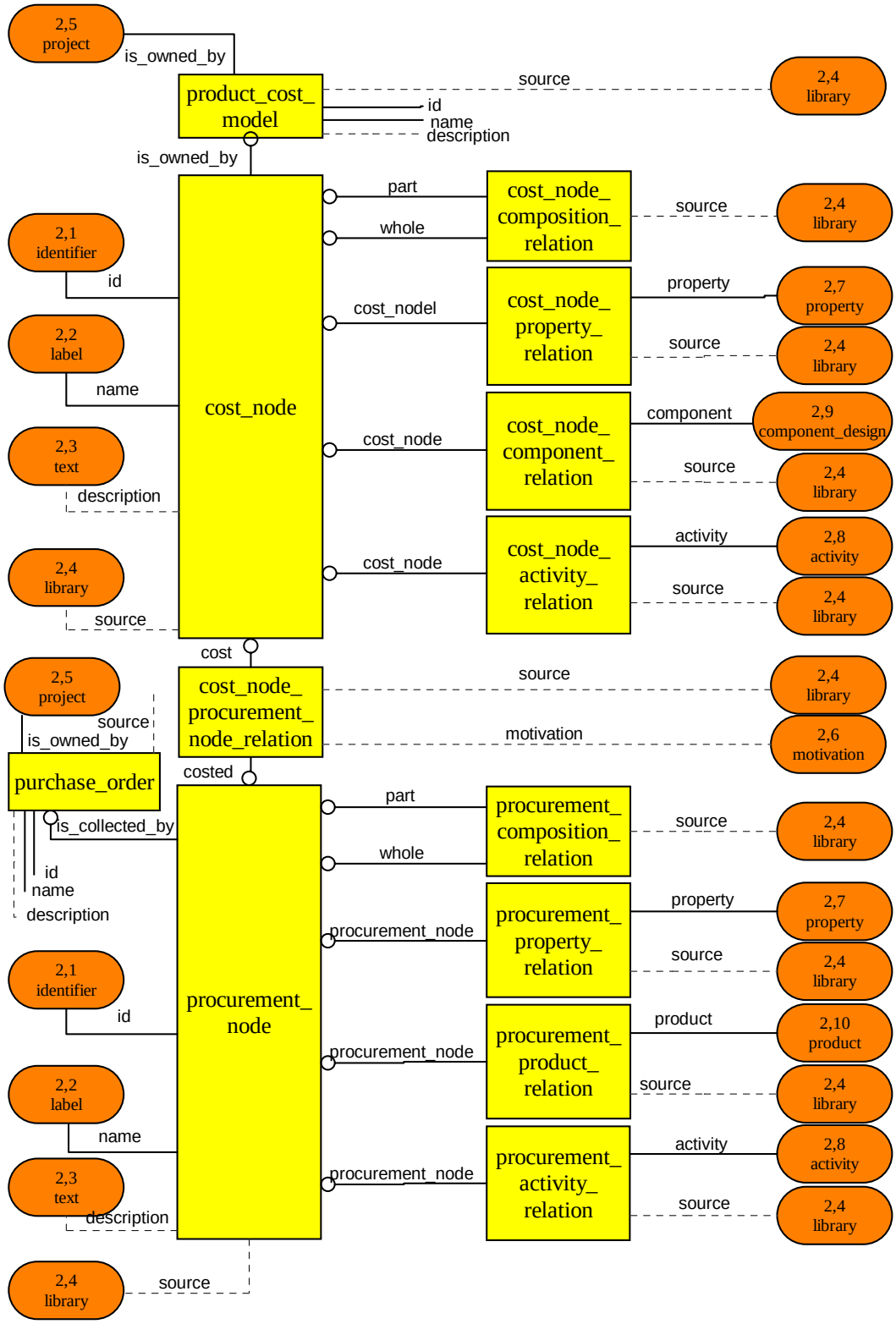


Figure D.1 - MOD_COSTING_AND_PROCUREMENT_SCHEMA EXPRESS-G diagram 1 of 2

TLO Holland Controls B.V- 06.05.2004

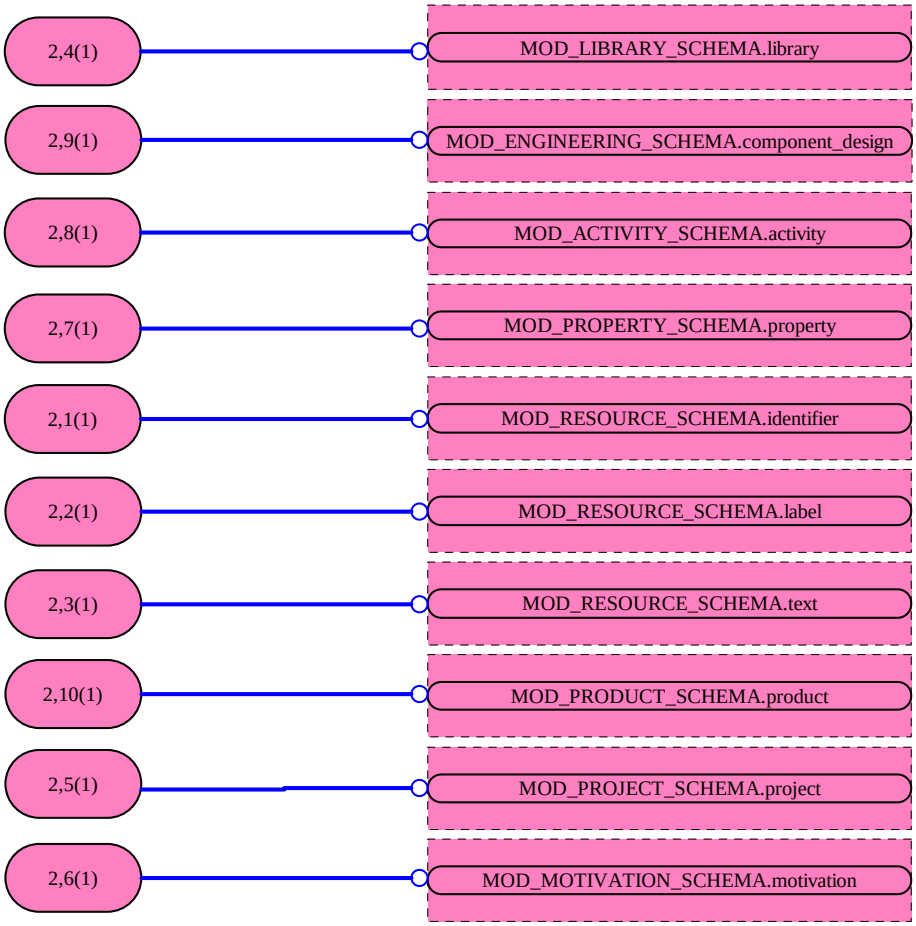


Figure D.2 - MOD_COSTING_AND_PROCUREMENT_SCHEMA EXPRESS-G diagram 2 of 2

*)
(*

6. MOD_COSTING_AND_PROCUREMENT_SCHEMA

EXPRESS specification:

```
*)
SCHEMA MOD_COSTING_AND_PROCUREMENT_SCHEMA;
  REFERENCE FROM MOD_ACTIVITY_SCHEMA
    (activity);
  REFERENCE FROM MOD_ENGINEERING_SCHEMA
    (component_design);
  REFERENCE FROM MOD_LIBRARY_SCHEMA
    (library);
  REFERENCE FROM MOD_MOTIVATION_SCHEMA
    (motivation);
  REFERENCE FROM MOD_PRODUCT_SCHEMA
    (product);
  REFERENCE FROM MOD_PROJECT_SCHEMA
    (project);
  REFERENCE FROM MOD_PROPERTY_SCHEMA
    (property);
  REFERENCE FROM MOD_RESOURCE_SCHEMA
    (identifier,
    label,
    text);
(*
```

6.1. cost_node

A [cost_node] is a hierarchical element of the estimate.

EXPRESS specification:

```
*)
  ENTITY cost_node;
    id                          :identifier;
    name                        :label;
    description                 :OPTIONAL text;
    is_owned_by                 :product_cost_model;
    source                      :OPTIONAL library;
  END_ENTITY;
(*
```

6.2. cost_node_activity_relation

EXPRESS specification:

```
*)
  ENTITY cost_node_activity_relation;
    cost_node                   :cost_node;
    activity                    :activity;
    source                      :OPTIONAL library;
  END_ENTITY;
```

(*

6.3. cost_node_component_relation

EXPRESS specification:

```
*)
  ENTITY cost_node_component_relation;
    cost_node                  :cost_node;
    component                  :component_design;
    source                     :OPTIONAL library;
  END_ENTITY;
(*
```

6.4. cost_node_composition_relation

EXPRESS specification:

```
*)
  ENTITY cost_node_composition_relation;
    whole                      :cost_node;
    part                       :cost_node;
    source                     :OPTIONAL library;
  END_ENTITY;
(*
```

6.5. cost_node_procurement_node_relation

EXPRESS specification:

```
*)
  ENTITY cost_node_procurement_node_relation;
    cost                       :cost_node;
    costed                     :procurement_node;
    source                     :OPTIONAL library;
    motivation                 :OPTIONAL motivation;
  END_ENTITY;
(*
```

6.6. cost_node_property_relation

EXPRESS specification:

```
*)
  ENTITY cost_node_property_relation;
    cost_nodel                 :cost_node;
    property                   :property;
    source                     :OPTIONAL library;
  END_ENTITY;
(*
```

6.7. procurement_activity_relation

EXPRESS specification:

```
*)
  ENTITY procurement_activity_relation;
```

```
    procurement_node          :procurement_node;
    activity                  :activity;
    source                    :OPTIONAL library;
  END_ENTITY;
(*
```

6.8. procurement_composition_relation

EXPRESS specification:

```
*)
  ENTITY procurement_composition_relation;
    part                      :procurement_node;
    whole                     :procurement_node;
    source                    :OPTIONAL library;
  END_ENTITY;
(*
```

6.9. procurement_node

A [procurement_node] is an item that groups one ore more items that can be ordered from suppliers. This may be a product, a rental or a service.

EXPRESS specification:

```
*)
  ENTITY procurement_node;
    is_collected_by           :purchase_order;
    id                        :identifier;
    name                      :label;
    description               :OPTIONAL text;
    source                    :OPTIONAL library;
  END_ENTITY;
(*
```

6.10. procurement_product_relation

EXPRESS specification:

```
*)
  ENTITY procurement_product_relation;
    procurement_node          :procurement_node;
    source                    :OPTIONAL library;
    product                   :product;
  END_ENTITY;
(*
```

6.11. procurement_property_relation

EXPRESS specification:

```
*)
  ENTITY procurement_property_relation;
    procurement_node          :procurement_node;
    property                  :property;
    source                    :OPTIONAL library;
```

```
  END_ENTITY;
(*
```

6.12. product_cost_model

A [product_cost_model] is a container which holds all the cost estimate information.

EXPRESS specification:

```
*)
  ENTITY product_cost_model;
    id                          :identifier;
    name                        :label;
    description                 :OPTIONAL text;
    source                      :OPTIONAL library;
    is_owned_by                 :project;
  END_ENTITY;
(*
```

6.13. purchase_order

A [purchase_order] is a request for products, rentals or services of a supplier.

EXPRESS specification:

```
*)
  ENTITY purchase_order;
    id                          :identifier;
    name                        :label;
    description                 :OPTIONAL text;
    source                      :OPTIONAL library;
    is_owned_by                 :project;
  END_ENTITY;
END_SCHEMA;
(*
```

EXPRESS-G diagrams MOD_DOCUMENT_SCHEMA

TLO Holland Controls B.V- 06.05.2004

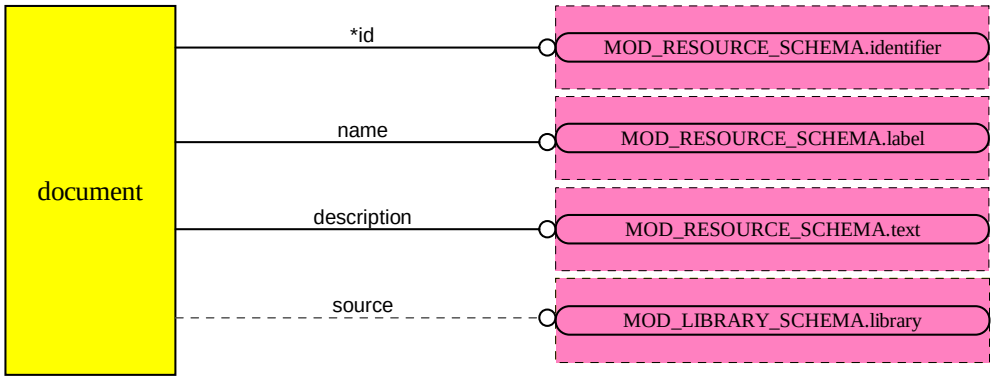


Figure D.1 - MOD_DOCUMENT_SCHEMA EXPRESS-G diagram 1 of 1*)

```
*)
(*
```

7. MOD_DOCUMENT_SCHEMA

EXPRESS specification:

```
*)
SCHEMA MOD_DOCUMENT_SCHEMA;
  REFERENCE FROM MOD_LIBRARY_SCHEMA
    (library);
  REFERENCE FROM MOD_RESOURCE_SCHEMA
    (identifier,
    label,
    text);
(*
```

7.1. document

A [document] is a collection of information on a subject to be read by humans.

EXPRESS specification:

```
*)
  ENTITY document;
    id                          :identifier;
    name                        :label;
    description                 :text;
    source                      :OPTIONAL library;
  UNIQUE
    u1       :id;
  END_ENTITY;
END_SCHEMA;
(*
```

EXPRESS-G diagrams MOD_ENGINEERING_SCHEMA

TLO Holland Controls B.V- 06.05.2004

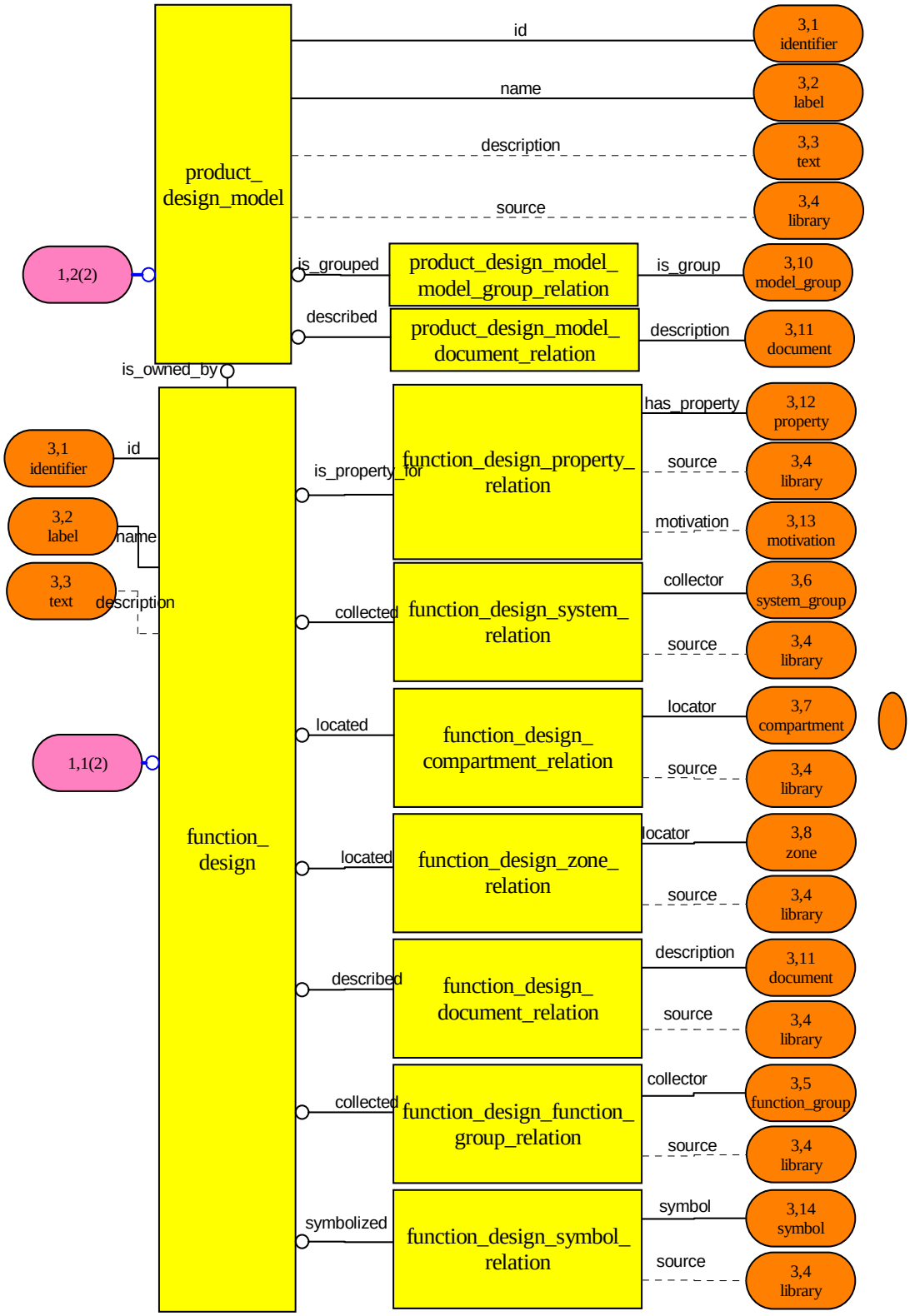


Figure D.1 - MOD_ENGINEERING_SCHEMA EXPRESS-G diagram 1 of 3

TLO Holland Controls B.V- 06.05.2004

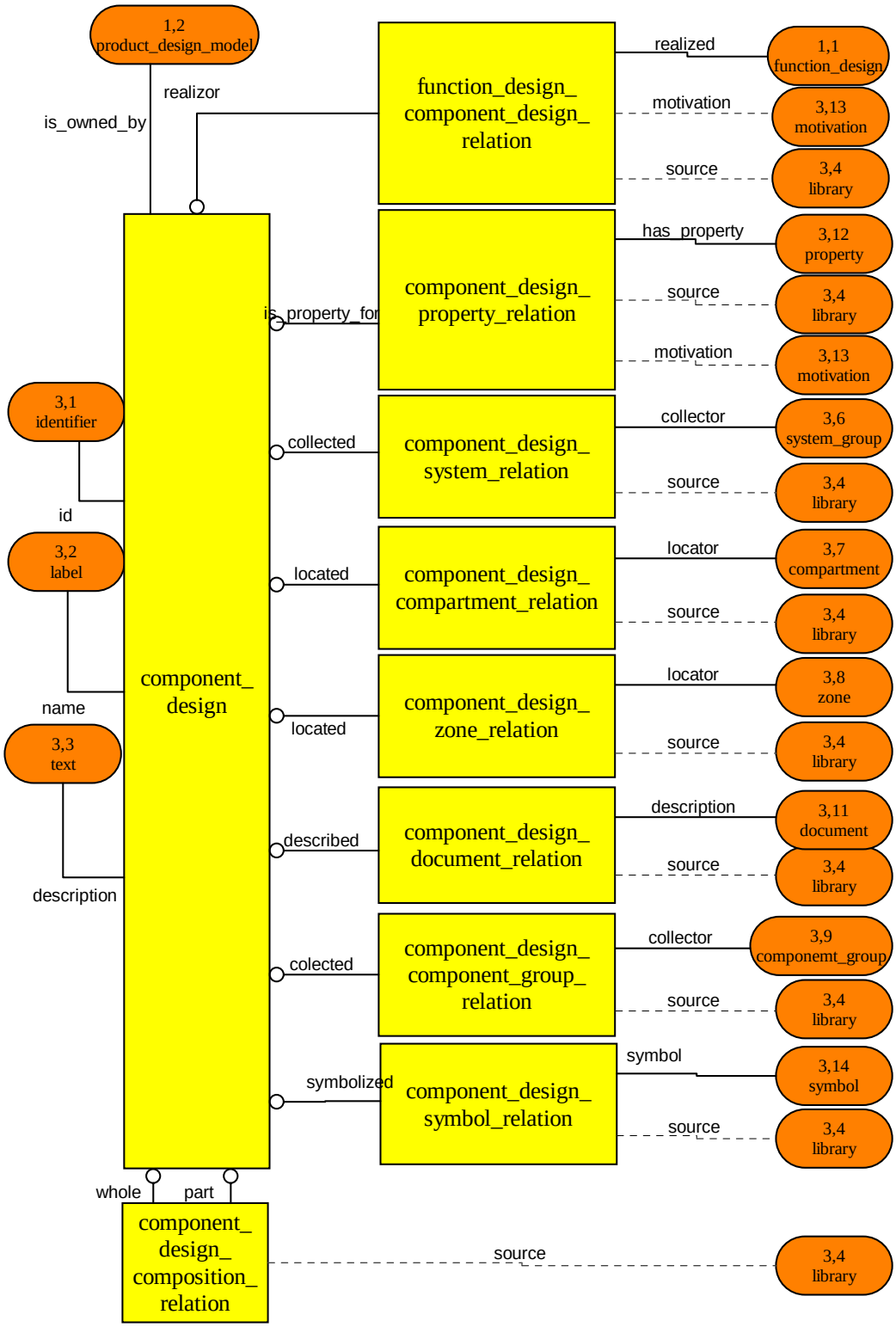


Figure D.2 - MOD_ENGINEERING_SCHEMA EXPRESS-G diagram 2 of 3

TLO Holland Controls B.V- 06.05.2004

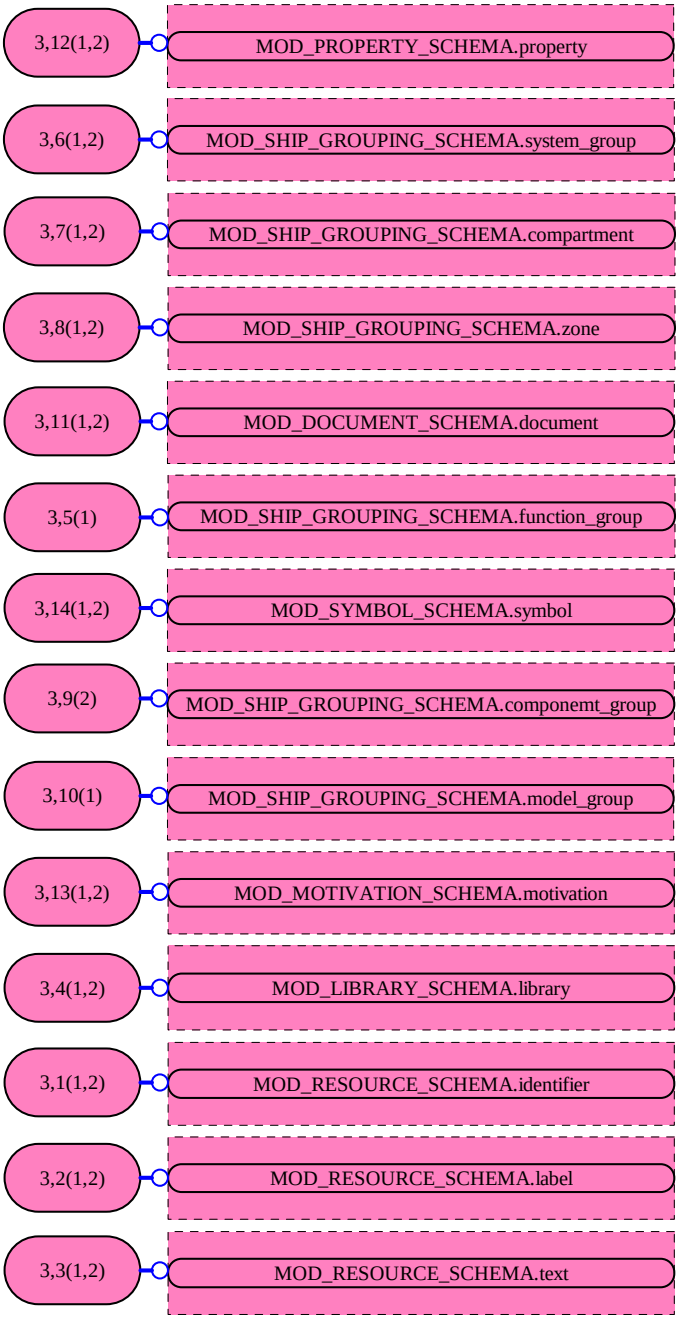


Figure D.3 - MOD_ENGINEERING_SCHEMA EXPRESS-G diagram 3 of 3

```
*)
(*
```

8. MOD_ENGINEERING_SCHEMA

EXPRESS specification:

```
*)
SCHEMA MOD_ENGINEERING_SCHEMA;
  REFERENCE FROM MOD_DOCUMENT_SCHEMA
    (document);
  REFERENCE FROM MOD_LIBRARY_SCHEMA
    (library);
  REFERENCE FROM MOD_MOTIVATION_SCHEMA
    (motivation);
  REFERENCE FROM MOD_PROPERTY_SCHEMA
    (property);
  REFERENCE FROM MOD_RESOURCE_SCHEMA
    (identifier,
    label,
    text);
  REFERENCE FROM MOD_SHIP_GROUPING_SCHEMA
    (system_group,
    compartment,
    zone,
    function_group,
    componemt_group,
    model_group);
  REFERENCE FROM MOD_SYMBOL_SCHEMA
    (symbol);
(*
```

8.1. component_design

A [component_design] is a description of the physical properties of a designed part of a system.

EXPRESS specification:

```
*)
  ENTITY component_design;
    is_owned_by               :product_design_model;
    id                        :identifier;
    name                      :label;
    description               :text;
  END_ENTITY;
(*
```

8.2. component_design_compartment_relation

EXPRESS specification:

```
*)
  ENTITY component_design_compartment_relation;
    located                   :component_design;
    locator                   :compartment;
```

```
    source                      :OPTIONAL library;
  END_ENTITY;
(*
```

8.3. component_design_component_group_relation

EXPRESS specification:

```
*)
  ENTITY component_design_component_group_relation;
    colected                    :component_design;
    collector                   :componemt_group;
    source                      :OPTIONAL library;
  END_ENTITY;
(*
```

8.4. component_design_composition_relation

EXPRESS specification:

```
*)
  ENTITY component_design_composition_relation;
    part                        :component_design;
    whole                       :component_design;
    source                      :OPTIONAL library;
  END_ENTITY;
(*
```

8.5. component_design_document_relation

EXPRESS specification:

```
*)
  ENTITY component_design_document_relation;
    described                   :component_design;
    description                 :document;
    source                      :OPTIONAL library;
  END_ENTITY;
(*
```

8.6. component_design_property_relation

EXPRESS specification:

```
*)
  ENTITY component_design_property_relation;
    is_property_for             :component_design;
    has_property                :property;
    motivation                  :OPTIONAL motivation;
    source                      :OPTIONAL library;
  END_ENTITY;
(*
```

8.7. component_design_symbol_relation

EXPRESS specification:

```
*)
  ENTITY component_design_symbol_relation;
    symbolized                  :component_design;
    symbol                      :symbol;
    source                      :OPTIONAL library;
  END_ENTITY;
(*
```

8.8. component_design_system_relation

EXPRESS specification:

```
*)
  ENTITY component_design_system_relation;
    collected                   :component_design;
    collector                   :system_group;
    source                      :OPTIONAL library;
  END_ENTITY;
(*
```

8.9. component_design_zone_relation

EXPRESS specification:

```
*)
  ENTITY component_design_zone_relation;
    located                     :component_design;
    locator                     :zone;
    source                      :OPTIONAL library;
  END_ENTITY;
(*
```

8.10. function_design

A [function_design] is a description of the functional behaviour of a designed part of a system.

EXPRESS specification:

```
*)
  ENTITY function_design;
    is_owned_by                 :product_design_model;
    id                          :identifier;
    name                        :label;
    description                 :OPTIONAL text;
  END_ENTITY;
(*
```

8.11. function_design_compartment_relation

EXPRESS specification:

```
*)
  ENTITY function_design_compartment_relation;
    located                     :function_design;
    locator                     :compartment;
    source                      :OPTIONAL library;
  END_ENTITY;
```

(*

8.12. function_design_component_design_relation

EXPRESS specification:

*)

```
  ENTITY function_design_component_design_relation;
    realized                  :function_design;
    realizor                  :component_design;
    motivation                :OPTIONAL motivation;
    source                    :OPTIONAL library;
  END_ENTITY;
```

(*

8.13. function_design_document_relation

EXPRESS specification:

*)

```
  ENTITY function_design_document_relation;
    described                 :function_design;
    description               :document;
    source                    :OPTIONAL library;
  END_ENTITY;
```

(*

8.14. function_design_function_group_relation

EXPRESS specification:

*)

```
  ENTITY function_design_function_group_relation;
    collected                 :function_design;
    collector                 :function_group;
    source                    :OPTIONAL library;
  END_ENTITY;
```

(*

8.15. function_design_property_relation

EXPRESS specification:

*)

```
  ENTITY function_design_property_relation;
    is_property_for           :function_design;
    has_property              :property;
    source                    :OPTIONAL library;
    motivation                :OPTIONAL motivation;
  END_ENTITY;
```

(*

8.16. function_design_symbol_relation

EXPRESS specification:

*)

```
  ENTITY function_design_symbol_relation;
    symbolized                  :function_design;
    symbol                      :symbol;
    source                      :OPTIONAL library;
  END_ENTITY;
(*
```

8.17. function_design_system_relation

EXPRESS specification:

```
*)
  ENTITY function_design_system_relation;
    collected                   :function_design;
    collector                   :system_group;
    source                      :OPTIONAL library;
  END_ENTITY;
(*
```

8.18. function_design_zone_relation

EXPRESS specification:

```
*)
  ENTITY function_design_zone_relation;
    located                     :function_design;
    locator                     :zone;
    source                      :OPTIONAL library;
  END_ENTITY;
(*
```

8.19. product_design_model

A [product_design_model] is a container which holds all the functional and physical design information.

EXPRESS specification:

```
*)
  ENTITY product_design_model;
    id                          :identifier;
    name                        :label;
    description                 :OPTIONAL text;
    source                      :OPTIONAL library;
  END_ENTITY;
(*
```

8.20. product_design_model_document_relation

EXPRESS specification:

```
*)
  ENTITY product_design_model_document_relation;
    described                   :product_design_model;
    description                 :document;
  END_ENTITY;
```

(*

8.21. product_design_model_model_group_relation

EXPRESS specification:

*)

```
  ENTITY product_design_model_model_group_relation;
    is_group                    :model_group;
    is_grouped                  :product_design_model;
  END_ENTITY;
END_SCHEMA;
```

(*

EXPRESS-G diagrams MOD_LIBRARY_SCHEMA

TLO Holland Controls B.V- 06.05.2004

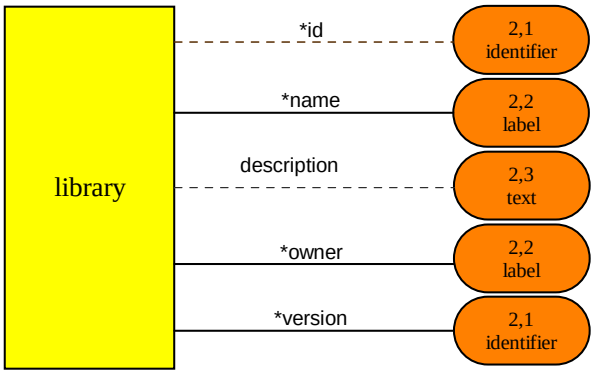


Figure D.1 - MOD_LIBRARY_SCHEMA EXPRESS-G diagram 1 of 2

TLO Holland Controls B.V- 06.05.2004

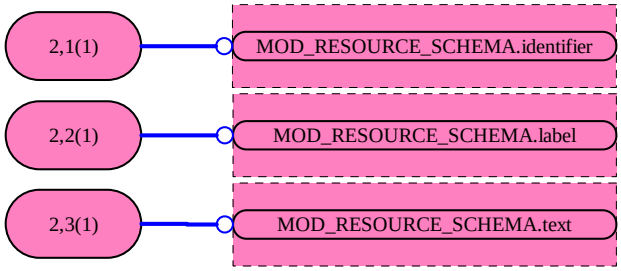


Figure D.2 - MOD_LIBRARY_SCHEMA EXPRESS-G diagram 2 of 2

```
*)
(*
```

9. MOD_LIBRARY_SCHEMA

EXPRESS specification:

```
*)
SCHEMA MOD_LIBRARY_SCHEMA;
  REFERENCE FROM MOD_RESOURCE_SCHEMA
    (identifier,
    label,
    text);
(*
```

9.1. library

A [library] is an official published set of information.

EXPRESS specification:

```
*)
  ENTITY library;
    id                          :OPTIONAL identifier;
    name                        :label;
    description                 :OPTIONAL text;
    owner                       :label;
    version                     :identifier;
  UNIQUE
    name_uniqueness    :     name,
              owner,
              version;id_uniqueness     :     id,
              owner,
              version;
  END_ENTITY;
END_SCHEMA;
(*
```

EXPRESS-G diagrams MOD_MOTIVATION_SCHEMA

TLO Holland Controls B.V- 06.05.2004

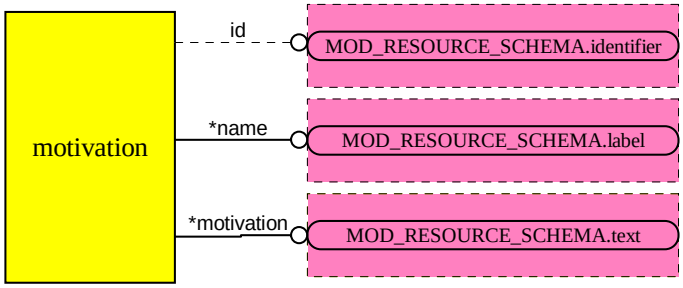


Figure D.1 - MOD_MOTIVATION_SCHEMA EXPRESS-G diagram 1 of 1

```
*)
(*
```

10. MOD_MOTIVATION_SCHEMA

EXPRESS specification:

```
*)
SCHEMA MOD_MOTIVATION_SCHEMA;
  REFERENCE FROM MOD_RESOURCE_SCHEMA
    (identifier,
    label,
    text);
(*
```

10.1. motivation

[motivation] is used to describe the context in which a relation is made.

EXPRESS specification:

```
*)
  ENTITY motivation;
    name                        :label;
    motivation                  :text;
    id                          :OPTIONAL identifier;
  UNIQUE
    MOTIVATION_UNIQUENESS   :     name,
              motivation;
  END_ENTITY;
END_SCHEMA;
(*
```

EXPRESS-G diagrams
MOD_PRODUCT_CLASS_SCHEMA

TLO Holland Controls B.V- 06.05.2004

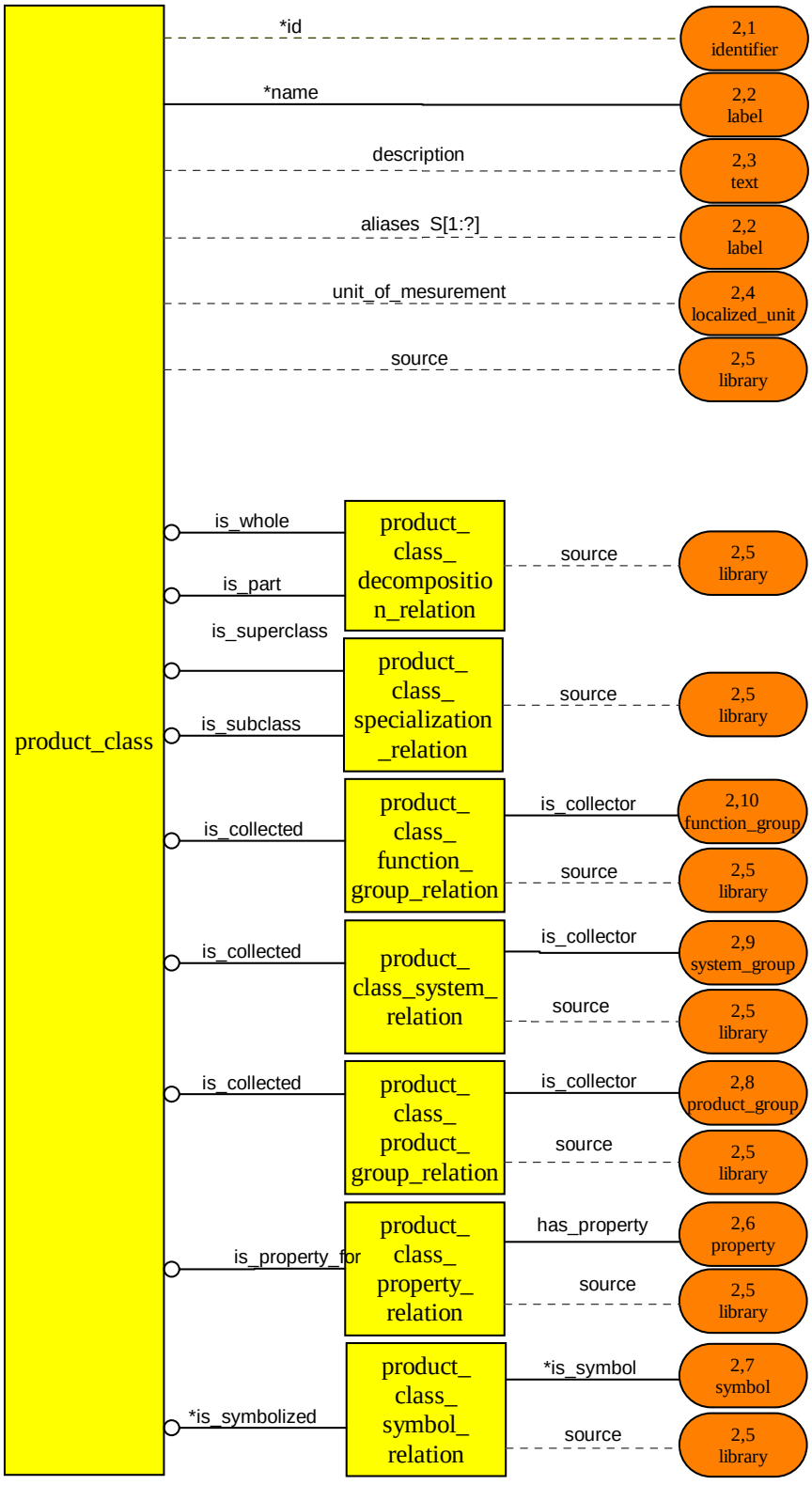


Figure D.1 - MOD_PRODUCT_CLASS_SCHEMA EXPRESS-G diagram 1 of 2

TLO Holland Controls B.V- 06.05.2004

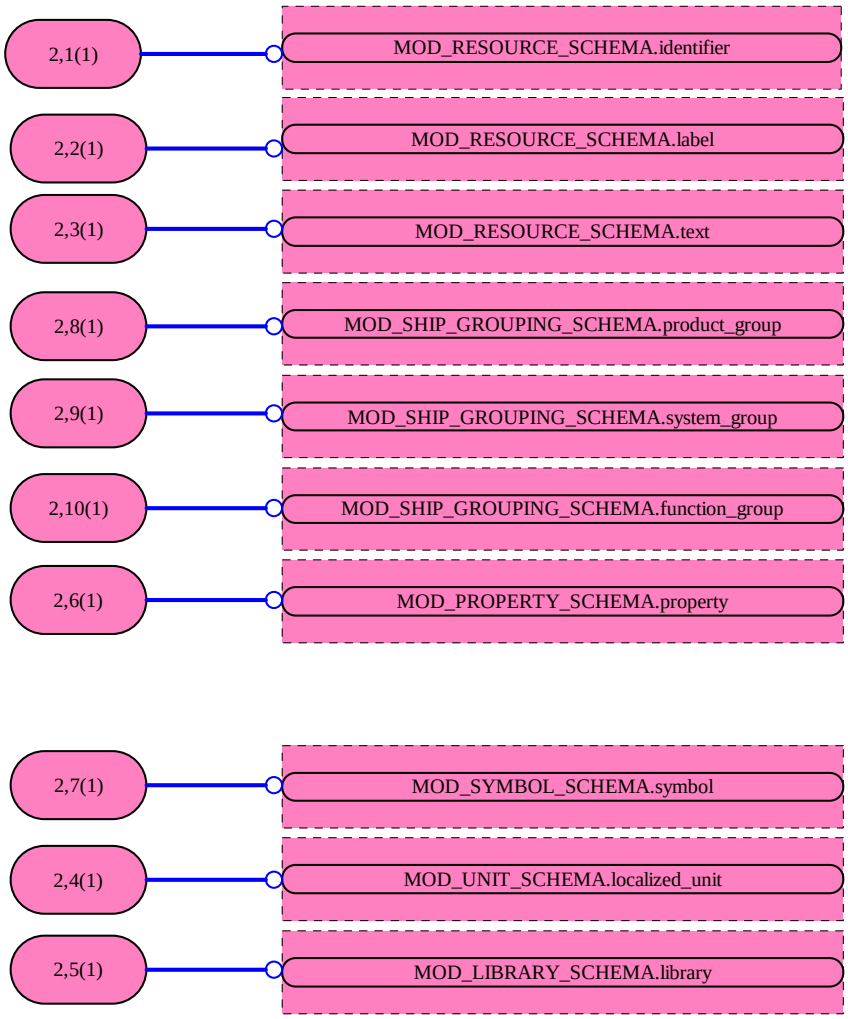


Figure D.2 - MOD_PRODUCT_CLASS_SCHEMA EXPRESS-G diagram 2 of 2

*)
(*

11. MOD_PRODUCT_CLASS_SCHEMA

EXPRESS specification:

```
*)
SCHEMA MOD_PRODUCT_CLASS_SCHEMA;
  REFERENCE FROM MOD_LIBRARY_SCHEMA
    (library);
  REFERENCE FROM MOD_PROPERTY_SCHEMA
    (property);
  REFERENCE FROM MOD_RESOURCE_SCHEMA
    (identifier,
    label,
    text);
  REFERENCE FROM MOD_SHIP_GROUPING_SCHEMA
    (product_group,
    system_group,
    function_group);
  REFERENCE FROM MOD_SYMBOL_SCHEMA
    (symbol);
  REFERENCE FROM MOD_UNIT_SCHEMA
    (localized_unit);
(*
```

11.1. product_class

A [product_class] is a hierarchical grouping abstract of products to a certain viewpoint.

EXPRESS specification:

```
*)
  ENTITY product_class;
    id                          :OPTIONAL identifier;
    name                        :label;
    description                 :OPTIONAL text;
    aliases                     :OPTIONAL SET [1:?] OF label;
    unit_of_mesurement          :OPTIONAL localized_unit;
    source                      :OPTIONAL library;
  UNIQUE
    U1       :id,
              name;
  END_ENTITY;
(*
```

11.2. product_class_decomposition_relation

EXPRESS specification:

```
*)
  ENTITY product_class_decomposition_relation;
    is_whole                    :product_class;
    is_part                     :product_class;
```

```
    source                    :OPTIONAL library;
  END_ENTITY;
(*
```

11.3. product_class_function_group_relation

EXPRESS specification:

```
*)
  ENTITY product_class_function_group_relation;
    is_collected              :product_class;
    is_collector              :function_group;
    source                    :OPTIONAL library;
  END_ENTITY;
(*
```

11.4. product_class_product_group_relation

EXPRESS specification:

```
*)
  ENTITY product_class_product_group_relation;
    is_collector              :product_group;
    is_collected              :product_class;
    source                    :OPTIONAL library;
  END_ENTITY;
(*
```

11.5. product_class_property_relation

EXPRESS specification:

```
*)
  ENTITY product_class_property_relation;
    is_property_for           :product_class;
    has_property              :property;
    source                    :OPTIONAL library;
  END_ENTITY;
(*
```

11.6. product_class_specialization_relation

EXPRESS specification:

```
*)
  ENTITY product_class_specialization_relation;
    is_superclass             :product_class;
    is_subclass               :product_class;
    source                    :OPTIONAL library;
  END_ENTITY;
(*
```

11.7. product_class_symbol_relation

EXPRESS specification:

```
*)
```

```
  ENTITY product_class_symbol_relation;
    is_symbolized              :product_class;
    is_symbol                  :symbol;
    source                     :OPTIONAL library;
  UNIQUE
    U1       :is_symbolized,
              is_symbol;
  END_ENTITY;
(*
```

11.8. product_class_system_relation

EXPRESS specification:

```
*)
  ENTITY product_class_system_relation;
    is_collected               :product_class;
    is_collector               :system_group;
    source                     :OPTIONAL library;
  END_ENTITY;
END_SCHEMA;
(*
```

EXPRESS-G diagrams MOD_PRODUCT_SCHEMA

TLO Holland Controls B.V- 06.05.2004

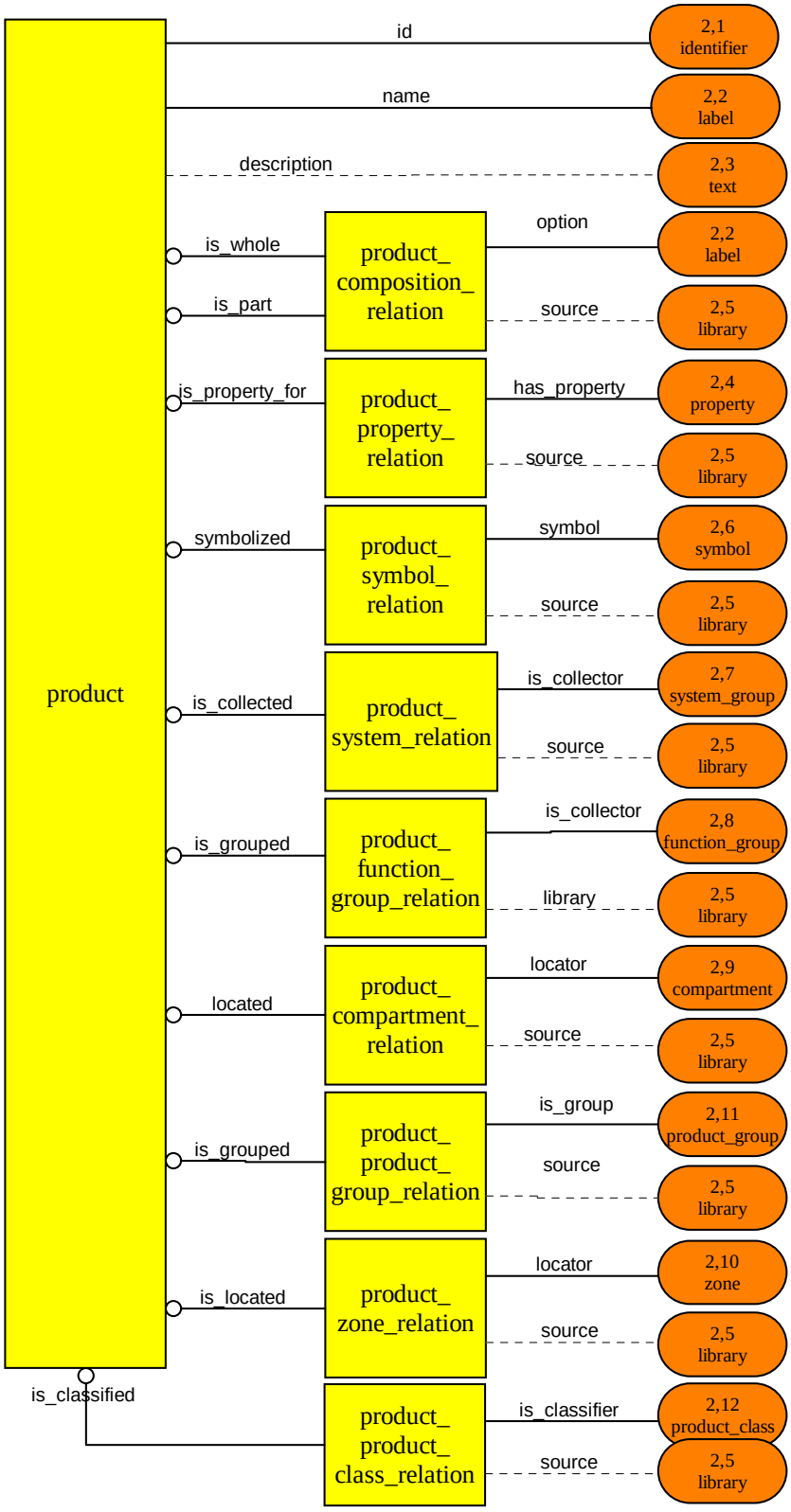


Figure D.1 - MOD_PRODUCT_SCHEMA EXPRESS-G diagram 1 of 2

TLO Holland Controls B.V- 06.05.2004

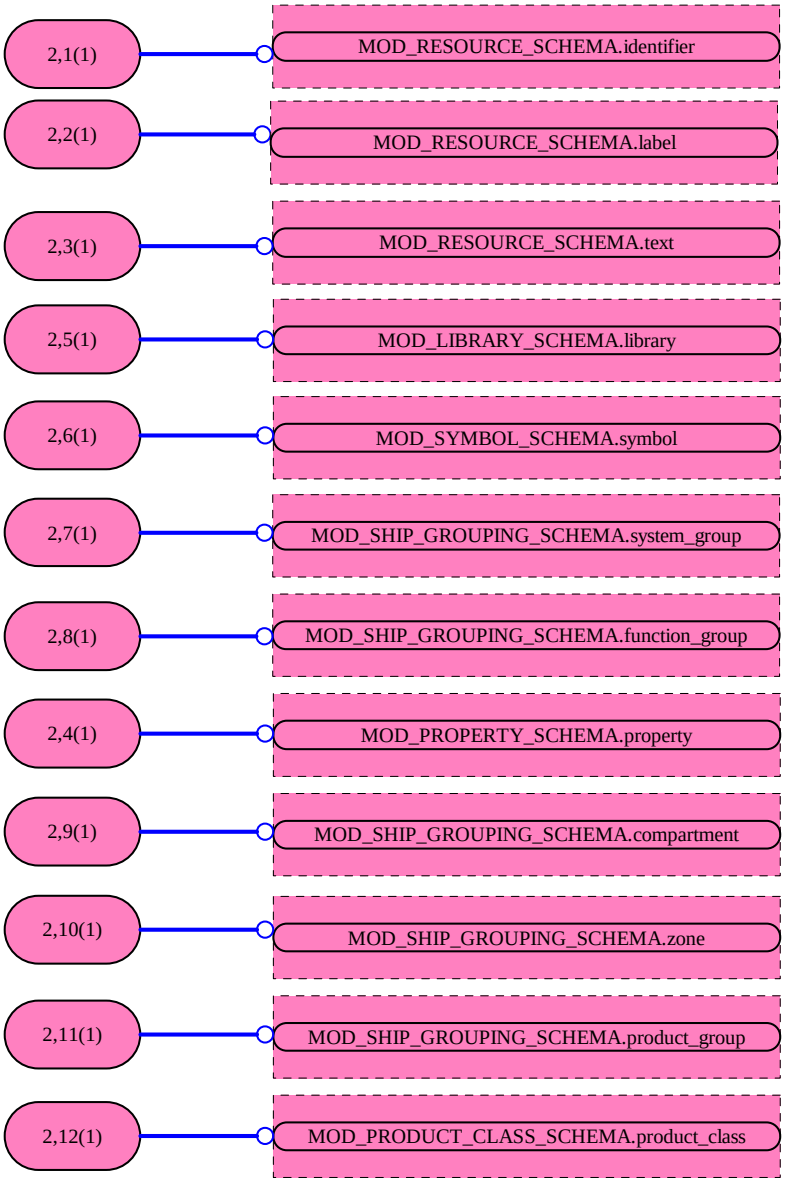


Figure D.2 - MOD_PRODUCT_SCHEMA EXPRESS-G diagram 2 of 2

*)
(*

12. MOD_PRODUCT_SCHEMA

EXPRESS specification:

```
*)
SCHEMA MOD_PRODUCT_SCHEMA;
  REFERENCE FROM MOD_LIBRARY_SCHEMA
    (library);
  REFERENCE FROM MOD_PRODUCT_CLASS_SCHEMA
    (product_class);
  REFERENCE FROM MOD_PROPERTY_SCHEMA
    (property);
  REFERENCE FROM MOD_RESOURCE_SCHEMA
    (identifier,
    label,
    text);
  REFERENCE FROM MOD_SHIP_GROUPING_SCHEMA
    (system_group,
    function_group,
    compartment,
    zone,
    product_group);
  REFERENCE FROM MOD_SYMBOL_SCHEMA
    (symbol);
(*
```

12.1. product

A [product] is a technical description of a physical object that can be purchased for use in a designed system.

EXPRESS specification:

```
*)
  ENTITY product;
    id                          :identifier;
    name                        :label;
    description                 :OPTIONAL text;
  END_ENTITY;
(*
```

12.2. product_compartment_relation

EXPRESS specification:

```
*)
  ENTITY product_compartment_relation;
    located                     :product;
    locator                     :compartment;
    source                      :OPTIONAL library;
  END_ENTITY;
(*
```

12.3. product_composition_relation

EXPRESS specification:

```
*)
  ENTITY product_composition_relation;
    is_whole                  :product;
    is_part                   :product;
    source                    :OPTIONAL library;
    option                    :label;
  END_ENTITY;
(*
```

12.4. product_function_group_relation

EXPRESS specification:

```
*)
  ENTITY product_function_group_relation;
    is_grouped                :product;
    is_collector              :function_group;
    library                   :OPTIONAL library;
  END_ENTITY;
(*
```

12.5. product_product_class_relation

EXPRESS specification:

```
*)
  ENTITY product_product_class_relation;
    is_classified             :product;
    is_classifier             :product_class;
    source                    :OPTIONAL library;
  END_ENTITY;
(*
```

12.6. product_product_group_relation

EXPRESS specification:

```
*)
  ENTITY product_product_group_relation;
    is_grouped                :product;
    is_group                  :product_group;
    source                    :OPTIONAL library;
  END_ENTITY;
(*
```

12.7. product_property_relation

EXPRESS specification:

```
*)
  ENTITY product_property_relation;
    is_property_for           :product;
    has_property              :property;
```

```
    source                    :OPTIONAL library;
  END_ENTITY;
(*
```

12.8. product_symbol_relation

EXPRESS specification:

```
*)
  ENTITY product_symbol_relation;
    symbolized                :product;
    symbol                    :symbol;
    source                    :OPTIONAL library;
  END_ENTITY;
(*
```

12.9. product_system_relation

EXPRESS specification:

```
*)
  ENTITY product_system_relation;
    is_collected              :product;
    is_collector              :system_group;
    source                    :OPTIONAL library;
  END_ENTITY;
(*
```

12.10. product_zone_relation

EXPRESS specification:

```
*)
  ENTITY product_zone_relation;
    is_located                :product;
    locator                   :zone;
    source                    :OPTIONAL library;
  END_ENTITY;
END_SCHEMA;
(*
```

EXPRESS-G diagrams MOD_PROJECT_SCHEMA

TLO Holland Controls B.V- 06.05.2004

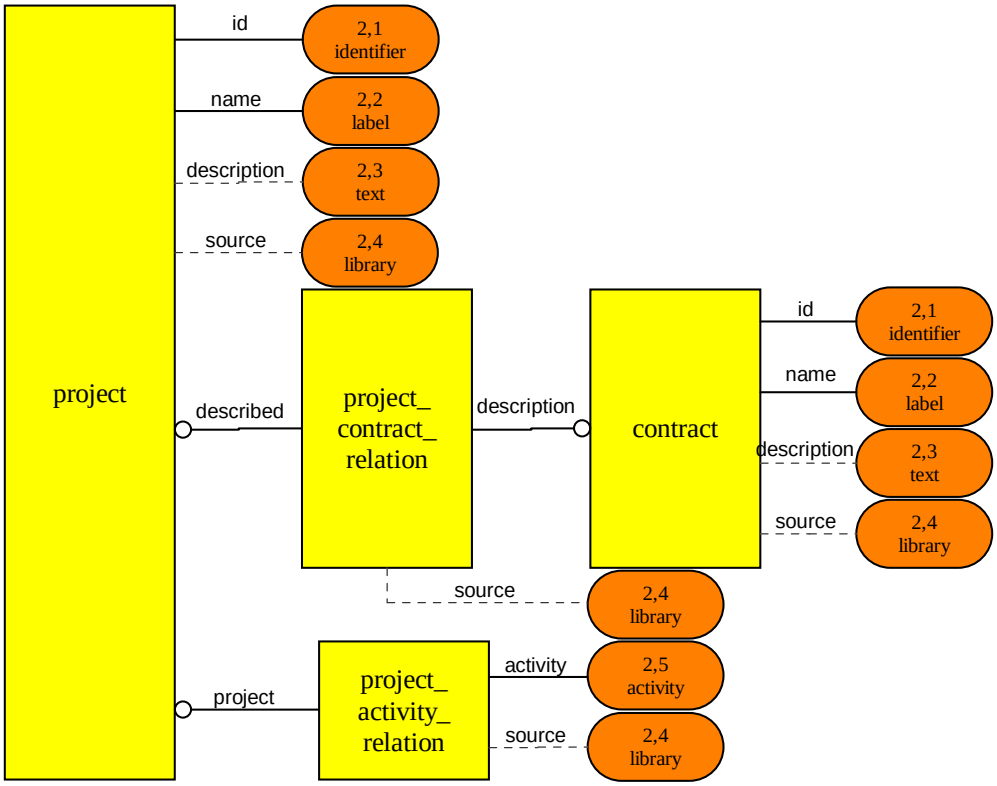


Figure D.1 - MOD_PROJECT_SCHEMA EXPRESS-G diagram 1 of 2

TLO Holland Controls B.V- 06.05.2004

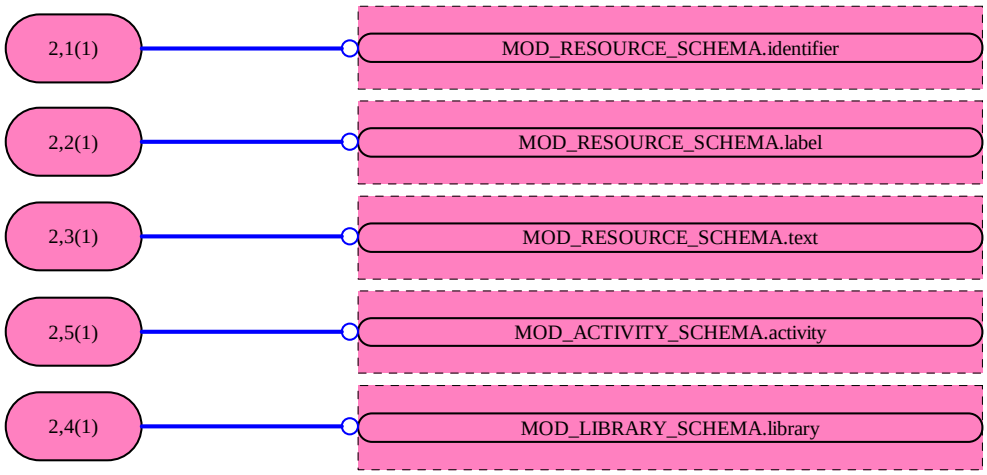


Figure D.2 - MOD_PROJECT_SCHEMA EXPRESS-G diagram 2 of 2

*)
(*

13. MOD_PROJECT_SCHEMA

EXPRESS specification:

```
*)
SCHEMA MOD_PROJECT_SCHEMA;
  REFERENCE FROM MOD_ACTIVITY_SCHEMA
    (activity);
  REFERENCE FROM MOD_LIBRARY_SCHEMA
    (library);
  REFERENCE FROM MOD_RESOURCE_SCHEMA
    (identifier,
    label,
    text);
(*
```

13.1. contract

A [contract] is an agreement between parties to perform certain duties and or products.

EXPRESS specification:

```
*)
  ENTITY contract;
    id                          :identifier;
    name                        :label;
    description                 :OPTIONAL text;
    source                      :OPTIONAL library;
  END_ENTITY;
(*
```

13.2. project

A [project] is the total sum of activities to establish the deliverable within a specific timeframe and budget with the use of men an material, all as optionally specified in a [contract].

EXPRESS specification:

```
*)
  ENTITY project;
    id                          :identifier;
    name                        :label;
    description                 :OPTIONAL text;
    source                      :OPTIONAL library;
  END_ENTITY;
(*
```

13.3. project_activity_relation

EXPRESS specification:

*)

```
  ENTITY project_activity_relation;
    project                    :project;
    activity                   :activity;
    source                     :OPTIONAL library;
  END_ENTITY;
(*
```

13.4. project_contract_relation

EXPRESS specification:

```
*)
  ENTITY project_contract_relation;
    described                  :project;
    description                :contract;
    source                     :OPTIONAL library;
  END_ENTITY;
END_SCHEMA;
(*
```

EXPRESS-G diagrams MOD_PROPERTY_SCHEMA

TLO Holland Controls B.V- 06.05.2004

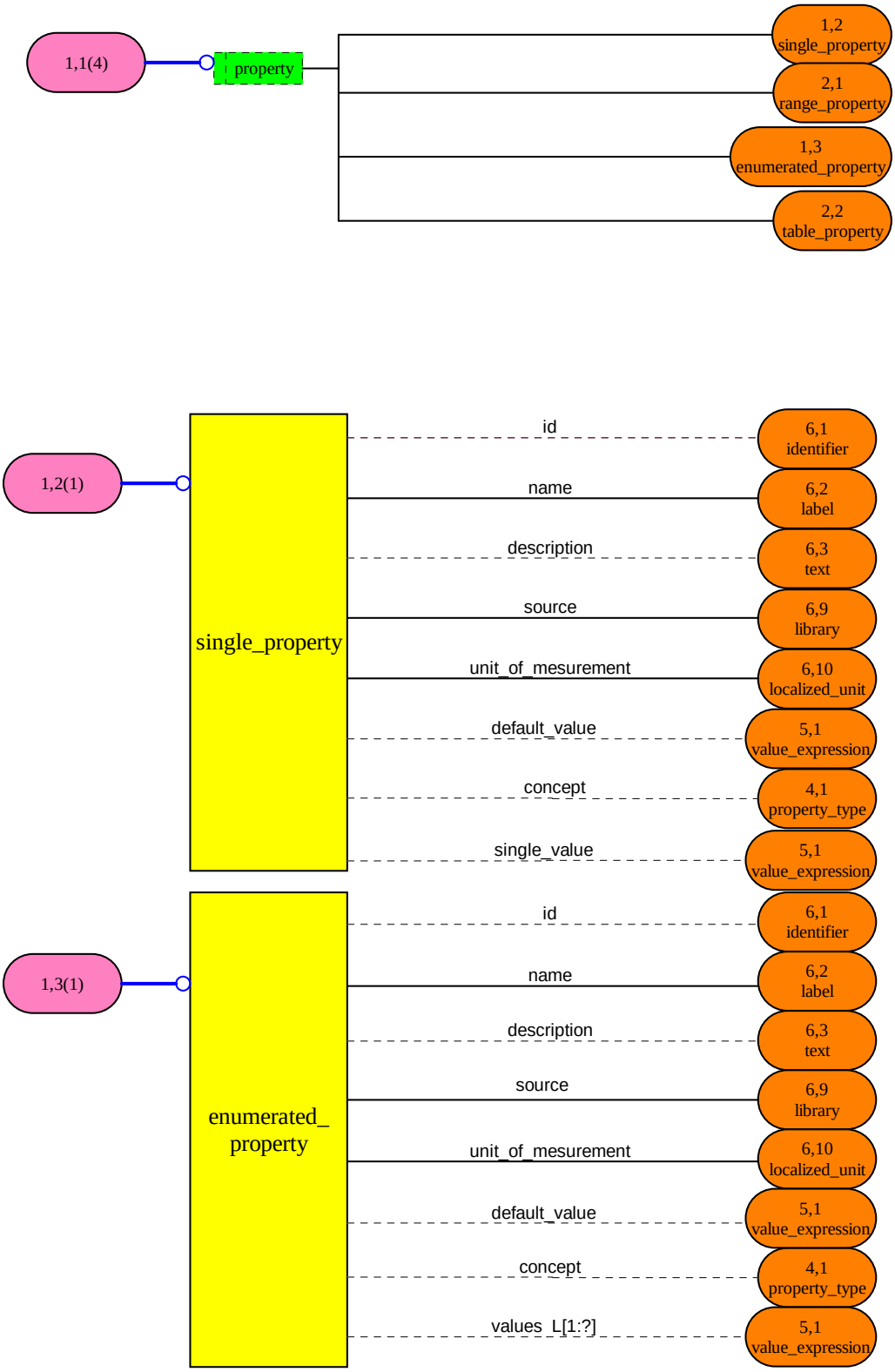


Figure D.1 - MOD_PROPERTY_SCHEMA EXPRESS-G diagram 1 of 6

TLO Holland Controls B.V- 06.05.2004

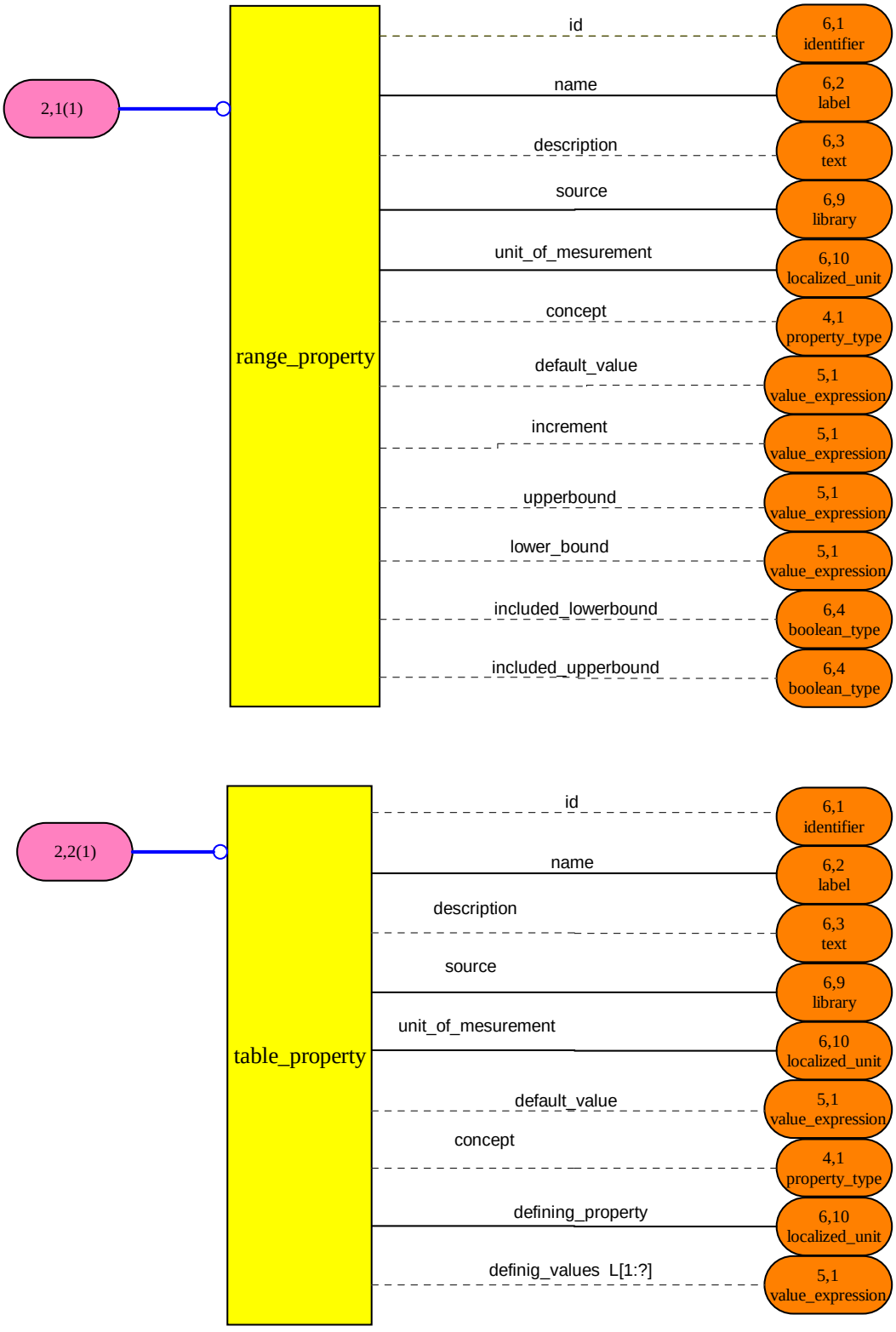


Figure D.2 - MOD_PROPERTY_SCHEMA EXPRESS-G diagram 2 of 6

TLO Holland Controls B.V- 06.05.2004

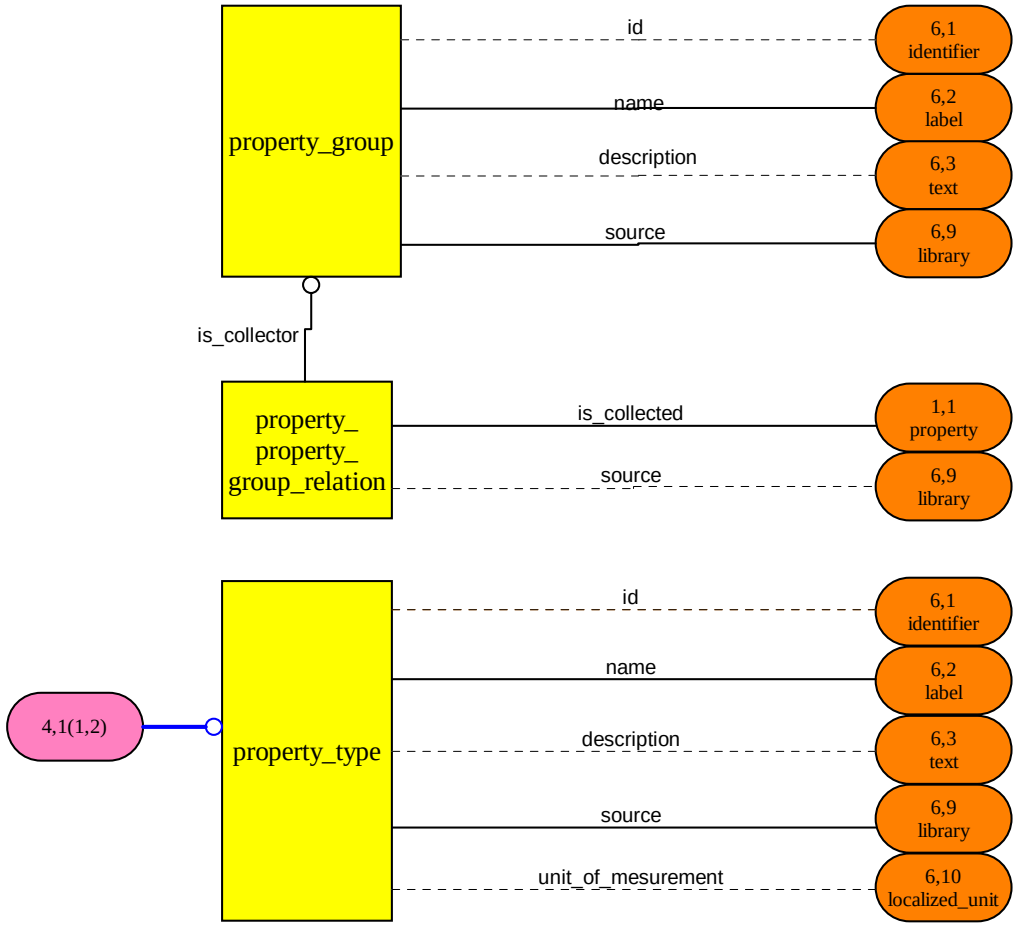


Figure D.4 - MOD_PROPERTY_SCHEMA EXPRESS-G diagram 3 of 6

TLO Holland Controls B.V- 06.05.2004

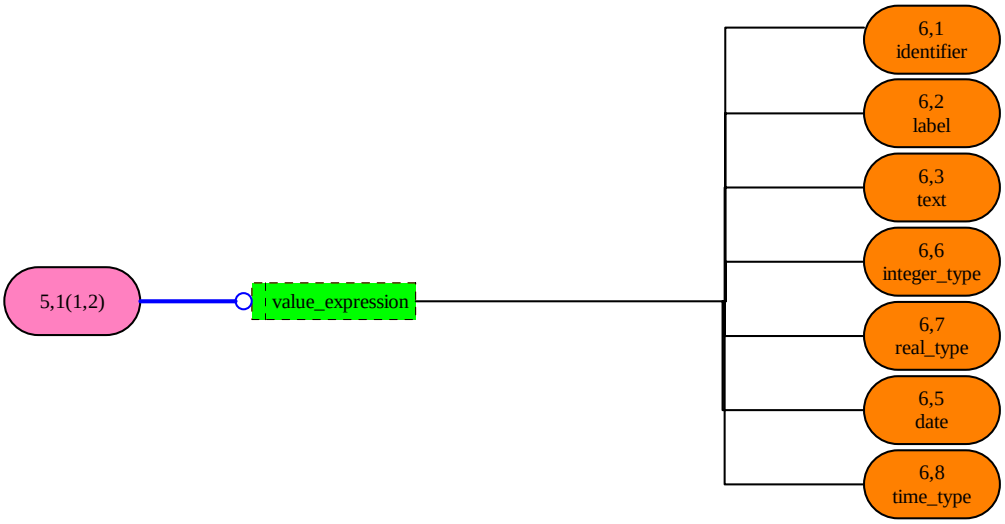


Figure D.5 - MOD_PROPERTY_SCHEMA EXPRESS-G diagram 4 of 6

TLO Holland Controls B.V- 06.05.2004

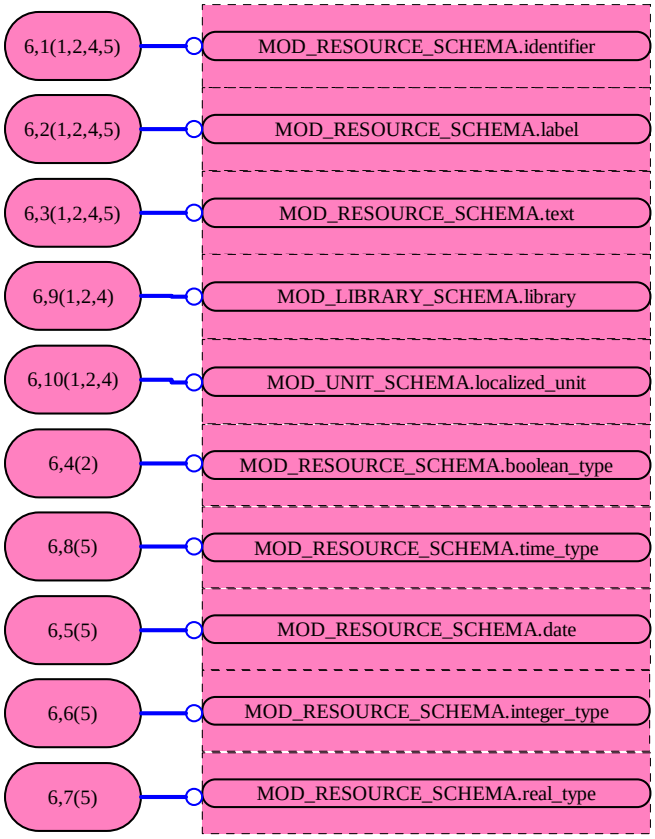


Figure D.6 - MOD_PROPERTY_SCHEMA EXPRESS-G diagram 5 of 6

*)
(*

14. MOD_PROPERTY_SCHEMA

EXPRESS specification:

```
*)
SCHEMA MOD_PROPERTY_SCHEMA;
  REFERENCE FROM MOD_LIBRARY_SCHEMA
    (library);
  REFERENCE FROM MOD_RESOURCE_SCHEMA
    (identifier,
    label,
    text,
    boolean_type,
    date,
    integer_type,
    real_type,
    time_type);
  REFERENCE FROM MOD_UNIT_SCHEMA
    (localized_unit);
(*
```

14.1. enumerated_property

An [enumerated_property] is a [property] which provides the possibility to assign predefined values for the [property].

EXPRESS specification:

```
*)
  ENTITY enumerated_property;
    id                          :OPTIONAL identifier;
    name                        :label;
    description                 :OPTIONAL text;
    source                      :library;
    unit_of_mesurement          :localized_unit;
    default_value               :OPTIONAL value_expression;
    concept                     :OPTIONAL property_type;
    values                      :OPTIONAL LIST [1:?] OF
                                 value_expression;
  END_ENTITY;
(*
```

14.2. property_group

A [property_group] is a collection properties which are somehow alike.

EXPRESS specification:

```
*)
  ENTITY property_group;
    id                          :OPTIONAL identifier;
    name                        :label;
```

```
    description                 :OPTIONAL text;
    source                      :library;
  END_ENTITY;
(*
```

14.3. property_property_group_relation

EXPRESS specification:

```
*)
  ENTITY property_property_group_relation;
    is_collected                :property;
    is_collector                :property_group;
    source                      :OPTIONAL library;
  END_ENTITY;
(*
```

14.4. property_type

A [property_type] is a typical (conceptual) property which is not directly applicable to express properties in.

EXPRESS specification:

```
*)
  ENTITY property_type;
    id                          :OPTIONAL identifier;
    name                        :label;
    description                 :OPTIONAL text;
    source                      :library;
    unit_of_mesurement          :OPTIONAL localized_unit;
  END_ENTITY;
(*
```

14.5. range_property

A [range_property] is a property which provides the possibility to specify a domain restriction to the [property].

EXPRESS specification:

```
*)
  ENTITY range_property;
    id                          :OPTIONAL identifier;
    name                        :label;
    description                 :OPTIONAL text;
    source                      :library;
    unit_of_mesurement          :localized_unit;
    default_value               :OPTIONAL value_expression;
    concept                     :OPTIONAL property_type;
    lower_bound                 :OPTIONAL value_expression;
    increment                   :OPTIONAL value_expression;
    upperbound                  :OPTIONAL value_expression;
    included_lowerbound         :OPTIONAL boolean_type;
    included_upperbound         :OPTIONAL boolean_type;
  END_ENTITY;
```

(*

14.6. single_property

A [single_property] is a visible of measurable phenomenon of an object.

EXPRESS specification:

```
*)
  ENTITY single_property;
    id                          :OPTIONAL identifier;
    name                        :label;
    description                 :OPTIONAL text;
    source                      :library;
    unit_of_mesurement          :localized_unit;
    default_value               :OPTIONAL value_expression;
    concept                     :OPTIONAL property_type;
    single_value                :OPTIONAL value_expression;
  END_ENTITY;
(*
```

14.7. table_property

A [table_property] is a [property] in which the axis of a two dimensional function can be defined.

EXPRESS specification:

```
*)
  ENTITY table_property;
    id                          :OPTIONAL identifier;
    name                        :label;
    description                 :OPTIONAL text;
    source                      :library;
    unit_of_mesurement          :localized_unit;
    default_value               :OPTIONAL value_expression;
    concept                     :OPTIONAL property_type;
    defining_property           :localized_unit;
    definig_values              :OPTIONAL LIST [1:?] OF
                                 value_expression;
  END_ENTITY;
(*
```

14.8. property

EXPRESS specification:

```
*)
  TYPE property = SELECT
    (single_property,
    enumerated_property,
    range_property,
    table_property);
  END_TYPE;
(*
```

14.9. value_expression

EXPRESS specification:

```
*)
  TYPE value_expression = SELECT
    (identifier,
    label,
    text,
    time_type,
    date,
    integer_type,
    real_type);
  END_TYPE;
END_SCHEMA;
(*
```

EXPRESS-G diagrams MOD_RESOURCE_SCHEMA

TLO Holland Controls B.V- 06.05.2004

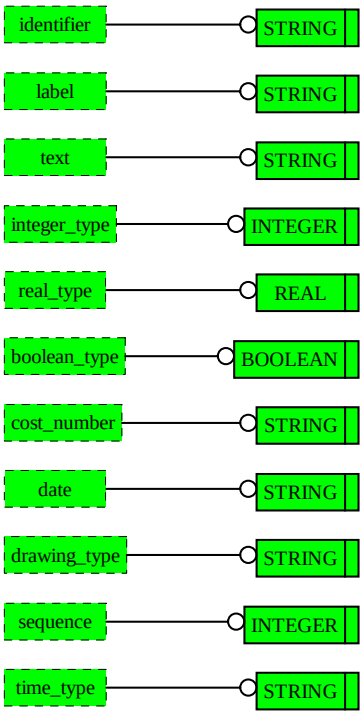


Figure D.1 - MOD_RESOURCE_SCHEMA EXPRESS-G diagram 1 of 1

*)
(*

15. MOD_RESOURCE_SCHEMA

EXPRESS specification:

*)
SCHEMA MOD_RESOURCE_SCHEMA;
(*

15.1. boolean_type

EXPRESS specification:

*)
```
  TYPE boolean_type = BOOLEAN;
  END_TYPE;
```
(*

15.2. cost_number

EXPRESS specification:

*)
```
  TYPE cost_number = STRING;
  END_TYPE;
```
(*

15.3. date

EXPRESS specification:

*)
```
  TYPE date = STRING;
  END_TYPE;
```
(*

15.4. drawing_type

EXPRESS specification:

*)
```
  TYPE drawing_type = STRING;
  END_TYPE;
```
(*

15.5. identifier

EXPRESS specification:

*)
```
  TYPE identifier = STRING;
  END_TYPE;
```
(*

15.6. integer_type

EXPRESS specification:

```
*)
  TYPE integer_type = INTEGER;
  END_TYPE;
(*
```

15.7. label

EXPRESS specification:

```
*)
  TYPE label = STRING;
  END_TYPE;
(*
```

15.8. real_type

EXPRESS specification:

```
*)
  TYPE real_type = REAL;
  END_TYPE;
(*
```

15.9. sequence

EXPRESS specification:

```
*)
  TYPE sequence = INTEGER;
  END_TYPE;
(*
```

15.10. text

EXPRESS specification:

```
*)
  TYPE text = STRING;
  END_TYPE;
(*
```

15.11. time_type

EXPRESS specification:

```
*)
  TYPE time_type = STRING;
  END_TYPE;
END_SCHEMA;
(*
```

EXPRESS-G diagrams
MOD_SHIP_GROUPING_SCHEMA

TLO Holland Controls B.V- 06.05.2004

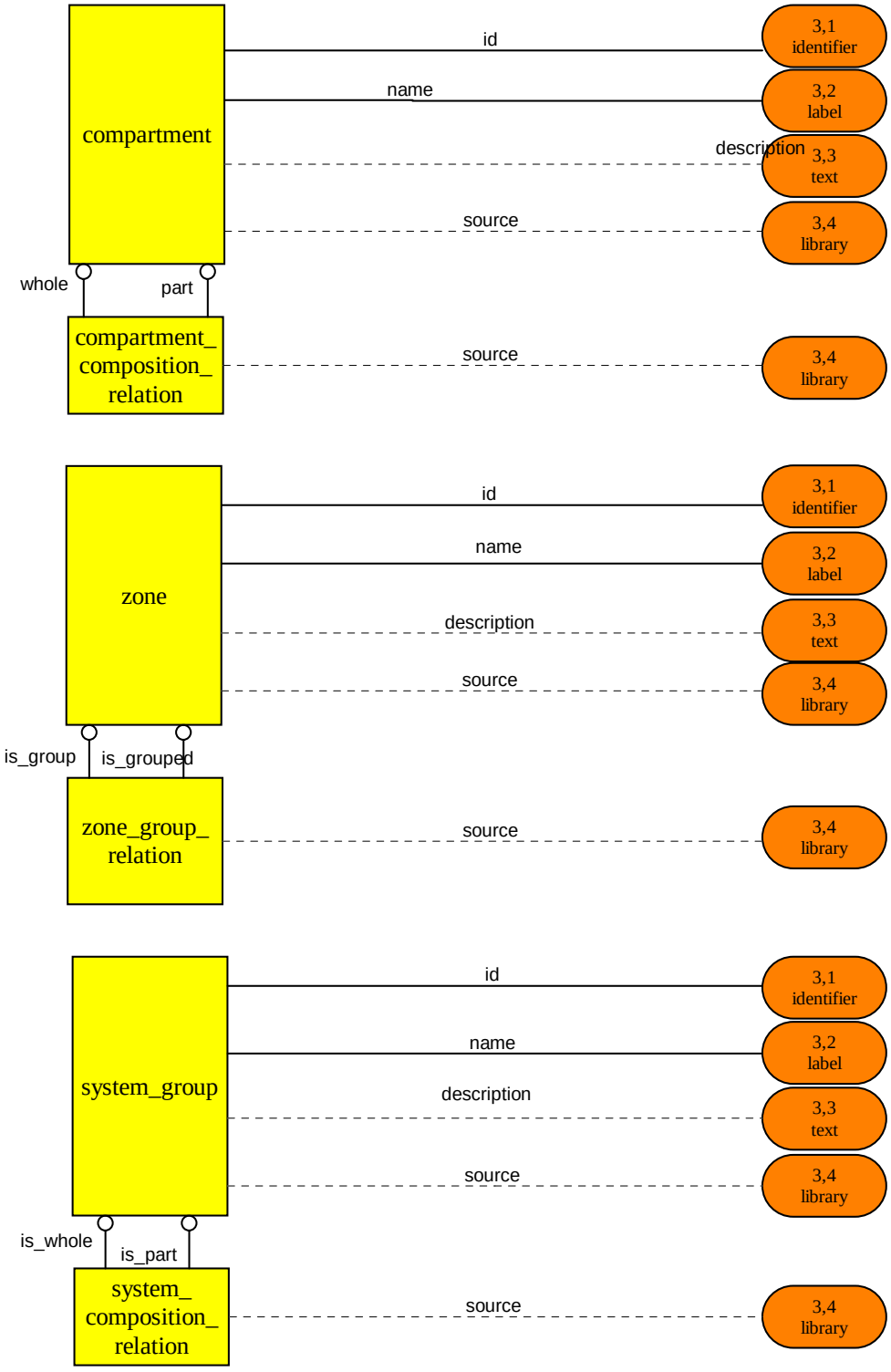


Figure D.1 - MOD_SHIP_GROUPING_SCHEMA EXPRESS-G diagram 1 of 3

TLO Holland Controls B.V- 06.05.2004

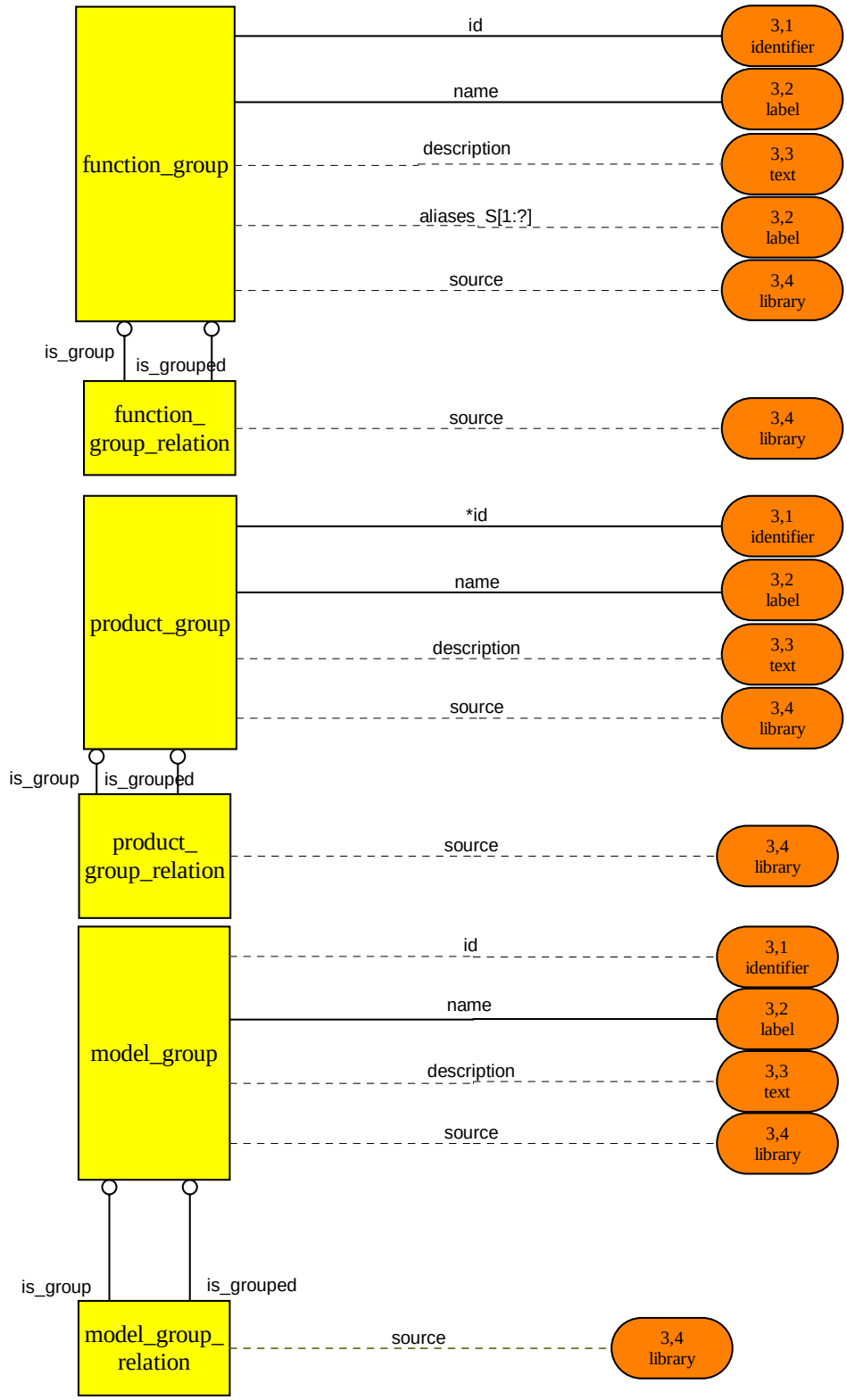


Figure D.2 - MOD_SHIP_GROUPING_SCHEMA EXPRESS-G diagram 2 of 3

TLO Holland Controls B.V- 06.05.2004

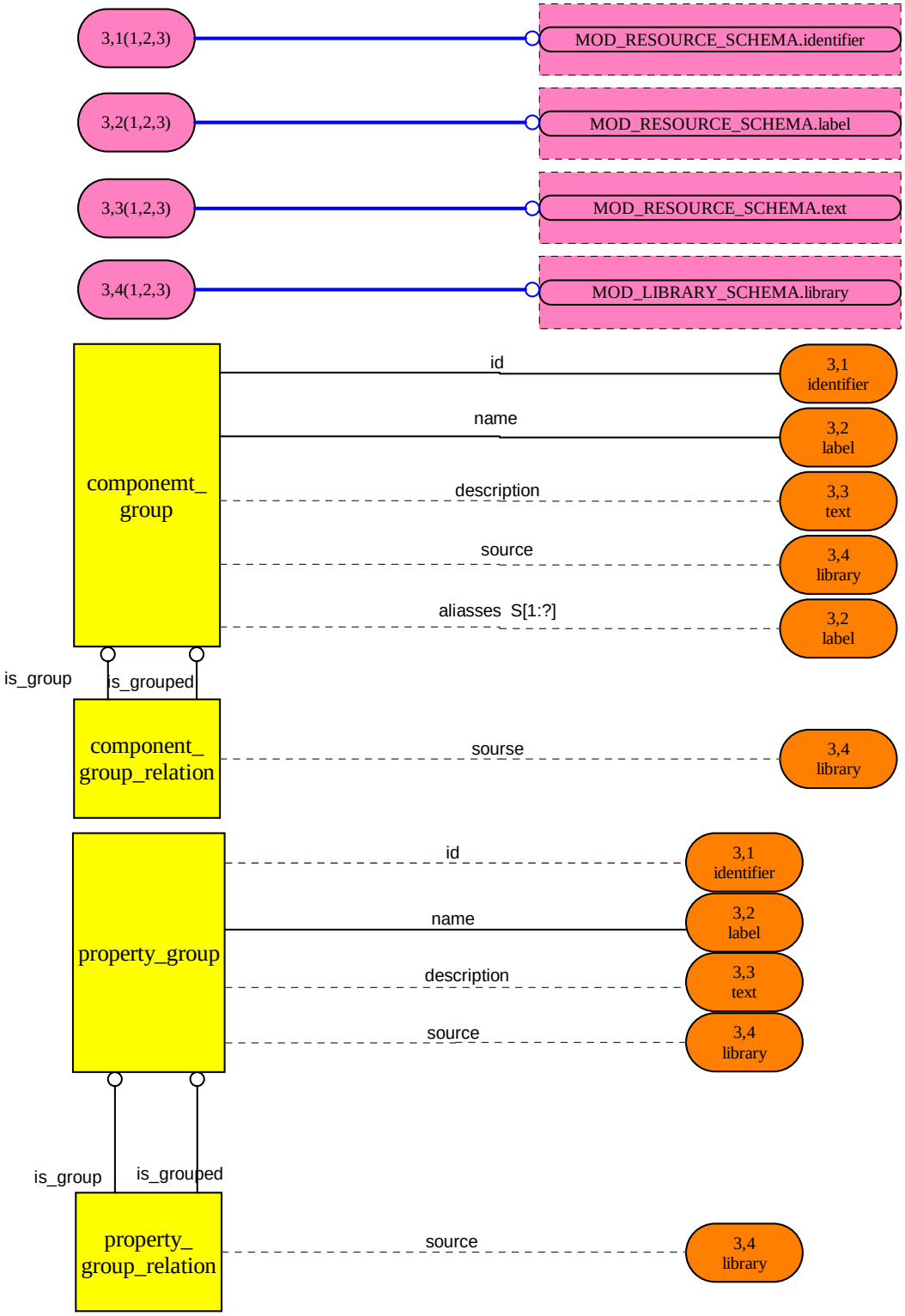


Figure D.3 - MOD_SHIP_GROUPING_SCHEMA EXPRESS-G diagram 3 of 3

*)
(*

16. MOD_SHIP_GROUPING_SCHEMA

EXPRESS specification:

```
*)
SCHEMA MOD_SHIP_GROUPING_SCHEMA;
  REFERENCE FROM MOD_LIBRARY_SCHEMA
    (library);
  REFERENCE FROM MOD_RESOURCE_SCHEMA
    (identifier,
    label,
    text);
(*
```

16.1. compartment

A [compartment] is a space bordered by steel plating.

EXPRESS specification:

```
*)
  ENTITY compartment;
    id                          :identifier;
    name                        :label;
    description                 :OPTIONAL text;
    source                      :OPTIONAL library;
  END_ENTITY;
(*
```

16.2. compartment_composition_relation

EXPRESS specification:

```
*)
  ENTITY compartment_composition_relation;
    whole                       :compartment;
    part                        :compartment;
    source                      :OPTIONAL library;
  END_ENTITY;
(*
```

16.3. componemt_group

A [compnent_group] is a grouping abstract of cmponent to a certain viewpoint.

EXPRESS specification:

```
*)
  ENTITY componemt_group;
    id                          :identifier;
    name                        :label;
    description                 :OPTIONAL text;
    source                      :OPTIONAL library;
```

```
    aliasses                    :OPTIONAL SET [1:?] OF label;
  END_ENTITY;
(*
```

16.4. component_group_relation

EXPRESS specification:

```
*)
  ENTITY component_group_relation;
    is_group                    :componemt_group;
    is_grouped                  :componemt_group;
    sourse                      :OPTIONAL library;
  END_ENTITY;
(*
```

16.5. function_group

A [function_group] is a grouping abstract of functions a certain viewpoint.

EXPRESS specification:

```
*)
  ENTITY function_group;
    id                          :identifier;
    name                        :label;
    description                 :OPTIONAL text;
    aliases                     :OPTIONAL SET [1:?] OF label;
    source                      :OPTIONAL library;
  END_ENTITY;
(*
```

16.6. function_group_relation

EXPRESS specification:

```
*)
  ENTITY function_group_relation;
    is_group                    :function_group;
    is_grouped                  :function_group;
    source                      :OPTIONAL library;
  END_ENTITY;
(*
```

16.7. model_group

A [model_group] is a grouping abstract of models to a certain viewpoint.

EXPRESS specification:

```
*)
  ENTITY model_group;
    id                          :OPTIONAL identifier;
    name                        :label;
    description                 :OPTIONAL text;
    source                      :OPTIONAL library;
```

```
  END_ENTITY;
(*
```

16.8. model_group_relation

EXPRESS specification:

```
*)
  ENTITY model_group_relation;
    is_group                    :model_group;
    is_grouped                  :model_group;
    source                      :OPTIONAL library;
  END_ENTITY;
(*
```

16.9. product_group

A [product_group] is a grouping abstract of products to a certain viewpoint.

EXPRESS specification:

```
*)
  ENTITY product_group;
    id                          :identifier;
    name                        :label;
    description                 :OPTIONAL text;
    source                      :OPTIONAL library;
  UNIQUE
    U1       :id;
  END_ENTITY;
(*
```

16.10. product_group_relation

EXPRESS specification:

```
*)
  ENTITY product_group_relation;
    is_grouped                  :product_group;
    is_group                    :product_group;
    source                      :OPTIONAL library;
  END_ENTITY;
(*
```

16.11. property_group

A [property_group] is a grouping abstract of properties to a certain viewpoint.

EXPRESS specification:

```
*)
  ENTITY property_group;
    id                          :OPTIONAL identifier;
    name                        :label;
    description                 :OPTIONAL text;
    source                      :OPTIONAL library;
```

```
  END_ENTITY;
(*
```

16.12. property_group_relation

EXPRESS specification:

```
*)
  ENTITY property_group_relation;
    is_group                  :property_group;
    is_grouped                :property_group;
    source                    :OPTIONAL library;
  END_ENTITY;
(*
```

16.13. system_composition_relation

EXPRESS specification:

```
*)
  ENTITY system_composition_relation;
    is_whole                  :system_group;
    is_part                   :system_group;
    source                    :OPTIONAL library;
  END_ENTITY;
(*
```

16.14. system_group

A [system_group] is a goup of componenets that as a whole fullfill a function.

EXPRESS specification:

```
*)
  ENTITY system_group;
    id                        :identifier;
    name                      :label;
    description               :OPTIONAL text;
    source                    :OPTIONAL library;
  END_ENTITY;
(*
```

16.15. zone

A [zone] is an area where certain restrictions apply.

EXPRESS specification:

```
*)
  ENTITY zone;
    id                        :identifier;
    name                      :label;
    description               :OPTIONAL text;
    source                    :OPTIONAL library;
  END_ENTITY;
(*
```

16.16. zone_group_relation

EXPRESS specification:

```
*)
  ENTITY zone_group_relation;
    is_group                  :zone;
    is_grouped                :zone;
    source                    :OPTIONAL library;
  END_ENTITY;
END_SCHEMA;
(*
```

EXPRESS-G diagrams MOD_SPECIFICATION_SCHEMA

TLO Holland Controls B.V- 06.05.2004

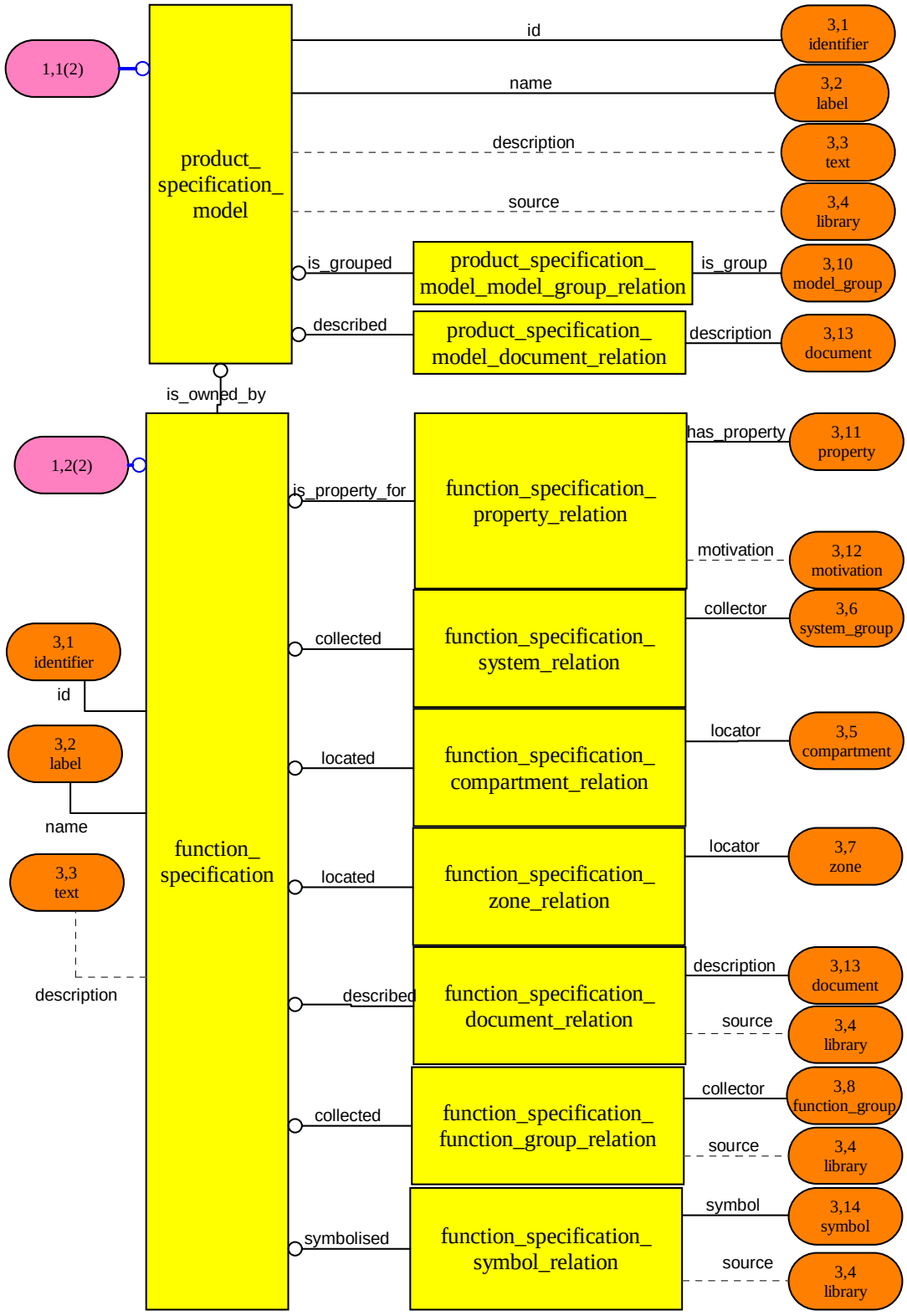


Figure D.1 - MOD_SPECIFICATION_SCHEMA EXPRESS-G diagram 1 of 3

TLO Holland Controls B.V- 06.05.2004

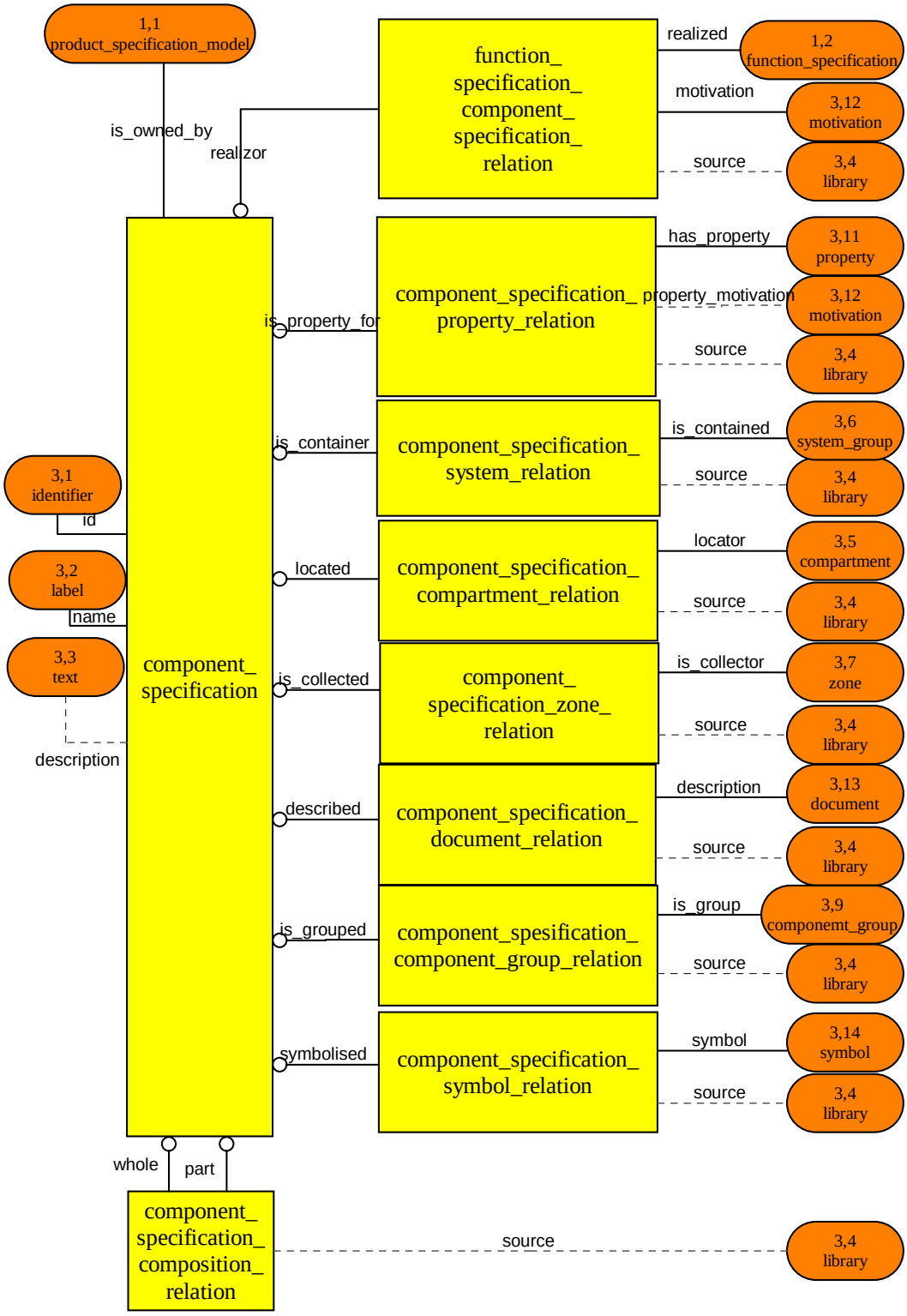


Figure D.2 - MOD_SPECIFICATION_SCHEMA EXPRESS-G diagram 2 of 3

TLO Holland Controls B.V- 06.05.2004

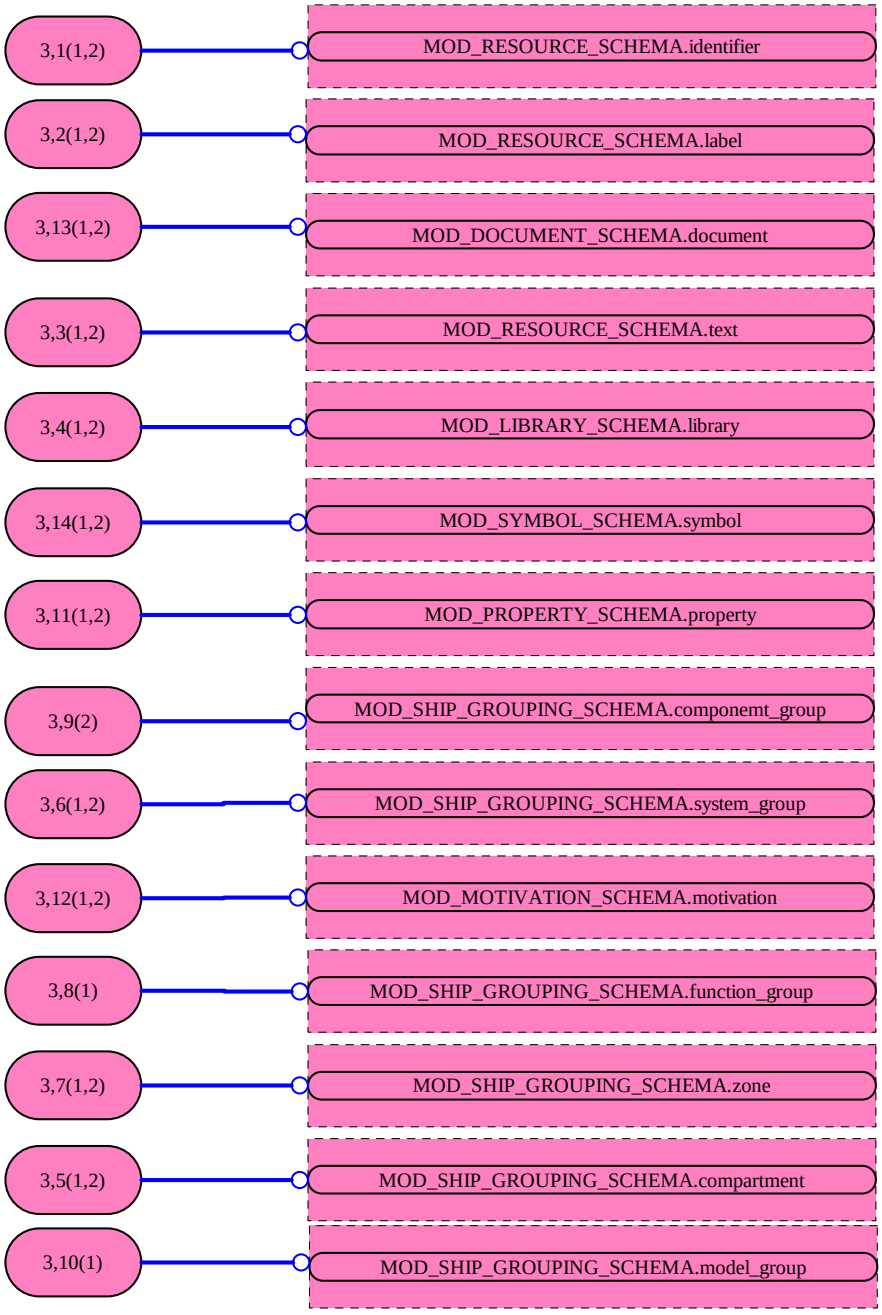


Figure D.3 - MOD_SPECIFICATION_SCHEMA EXPRESS-G diagram 3 of 3

*)
(*

17. MOD_SPECIFICATION_SCHEMA

EXPRESS specification:

```
*)
SCHEMA MOD_SPECIFICATION_SCHEMA;
  REFERENCE FROM MOD_DOCUMENT_SCHEMA
    (document);
  REFERENCE FROM MOD_LIBRARY_SCHEMA
    (library);
  REFERENCE FROM MOD_MOTIVATION_SCHEMA
    (motivation);
  REFERENCE FROM MOD_PROPERTY_SCHEMA
    (property);
  REFERENCE FROM MOD_RESOURCE_SCHEMA
    (identifier,
    label,
    text);
  REFERENCE FROM MOD_SHIP_GROUPING_SCHEMA
    (system_group,
    compartment,
    zone,
    function_group,
    componemt_group,
    model_group);
  REFERENCE FROM MOD_SYMBOL_SCHEMA
    (symbol);
(*
```

17.1. component_specification

A [component_design] is a description of the physical properties of a designed part of a system.

EXPRESS specification:

```
*)
  ENTITY component_specification;
    is_owned_by                :product_specification_model;
    id                         :identifier;
    name                       :label;
    description                :OPTIONAL text;
  END_ENTITY;
(*
```

17.2. component_specification_compartment_relation

EXPRESS specification:

```
*)
  ENTITY component_specification_compartment_relation;
    located                    :component_specification;
    locator                    :compartment;
```

```
    source                    :OPTIONAL library;
  END_ENTITY;
(*
```

17.3. component_specification_composition_relation

EXPRESS specification:

```
*)
  ENTITY component_specification_composition_relation;
    whole                     :component_specification;
    part                      :component_specification;
    source                    :OPTIONAL library;
  END_ENTITY;
(*
```

17.4. component_specification_document_relation

EXPRESS specification:

```
*)
  ENTITY component_specification_document_relation;
    described                 :component_specification;
    description               :document;
    source                    :OPTIONAL library;
  END_ENTITY;
(*
```

17.5. component_specification_property_relation

EXPRESS specification:

```
*)
  ENTITY component_specification_property_relation;
    is_property_for           :component_specification;
    has_property              :property;
    source                    :OPTIONAL library;
    property_motivation       :OPTIONAL motivation;
  END_ENTITY;
(*
```

17.6. component_specification_symbol_relation

EXPRESS specification:

```
*)
  ENTITY component_specification_symbol_relation;
    symbolised                :component_specification;
    symbol                    :symbol;
    source                    :OPTIONAL library;
  END_ENTITY;
(*
```

17.7. component_specification_system_relation

EXPRESS specification:

```
*)
  ENTITY component_specification_system_relation;
    is_contained                :system_group;
    is_container                :component_specification;
    source                      :OPTIONAL library;
  END_ENTITY;
(*
```

17.8. component_specification_zone_relation

EXPRESS specification:

```
*)
  ENTITY component_specification_zone_relation;
    is_collector                :zone;
    is_collected                :component_specification;
    source                      :OPTIONAL library;
  END_ENTITY;
(*
```

17.9. component_spesification_component_group_relation

EXPRESS specification:

```
*)
  ENTITY component_spesification_component_group_relation;
    is_group                    :componemt_group;
    is_grouped                  :component_specification;
    source                      :OPTIONAL library;
  END_ENTITY;
(*
```

17.10. function_specification

A [function_specification] is a set of functional behaviours to which a component_design should adhere.

EXPRESS specification:

```
*)
  ENTITY function_specification;
    is_owned_by                 :product_specification_model;
    id                          :identifier;
    name                        :label;
    description                 :OPTIONAL text;
  END_ENTITY;
(*
```

17.11. function_specification_compartment_relation

EXPRESS specification:

```
*)
  ENTITY function_specification_compartment_relation;
    located                     :function_specification;
    locator                     :compartment;
```

```
  END_ENTITY;
(*
```

17.12. function_specification_component_specification_relation

EXPRESS specification:

```
*)
  ENTITY function_specification_component_specification_relation;
    realized                  :function_specification;
    realizor                  :component_specification;
    motivation                :motivation;
    source                    :OPTIONAL library;
  END_ENTITY;
(*
```

17.13. function_specification_document_relation

EXPRESS specification:

```
*)
  ENTITY function_specification_document_relation;
    described                 :function_specification;
    description               :document;
    source                    :OPTIONAL library;
  END_ENTITY;
(*
```

17.14. function_specification_function_group_relation

EXPRESS specification:

```
*)
  ENTITY function_specification_function_group_relation;
    collected                 :function_specification;
    collector                 :function_group;
    source                    :OPTIONAL library;
  END_ENTITY;
(*
```

17.15. function_specification_property_relation

EXPRESS specification:

```
*)
  ENTITY function_specification_property_relation;
    is_property_for           :function_specification;
    has_property              :property;
    motivation                :OPTIONAL motivation;
  END_ENTITY;
(*
```

17.16. function_specification_symbol_relation

EXPRESS specification:

```
*)
  ENTITY function_specification_symbol_relation;
    symbolised                  :function_specification;
    symbol                      :symbol;
    source                      :OPTIONAL library;
  END_ENTITY;
(*
```

17.17. function_specification_system_relation

EXPRESS specification:

```
*)
  ENTITY function_specification_system_relation;
    collected                   :function_specification;
    collector                   :system_group;
  END_ENTITY;
(*
```

17.18. function_specification_zone_relation

EXPRESS specification:

```
*)
  ENTITY function_specification_zone_relation;
    located                     :function_specification;
    locator                     :zone;
  END_ENTITY;
(*
```

17.19. product_specification_model

A [product_specification_model] is a container which holds all the functional and physical specifications.

EXPRESS specification:

```
*)
  ENTITY product_specification_model;
    id                          :identifier;
    name                        :label;
    description                 :OPTIONAL text;
    source                      :OPTIONAL library;
  END_ENTITY;
(*
```

17.20. product_specification_model_document_relation

EXPRESS specification:

```
*)
  ENTITY product_specification_model_document_relation;
    described                   :product_specification_model;
    description                 :document;
  END_ENTITY;
(*
```

17.21. product_specification_model_model_group_relation

EXPRESS specification:

```
*)
  ENTITY product_specification_model_model_group_relation;
    is_grouped                 :product_specification_model;
    is_group                   :model_group;
  END_ENTITY;
END_SCHEMA;
(*
```

EXPRESS-G diagrams MOD_SYMBOL_SCHEMA

TLO Holland Controls B.V- 06.05.2004

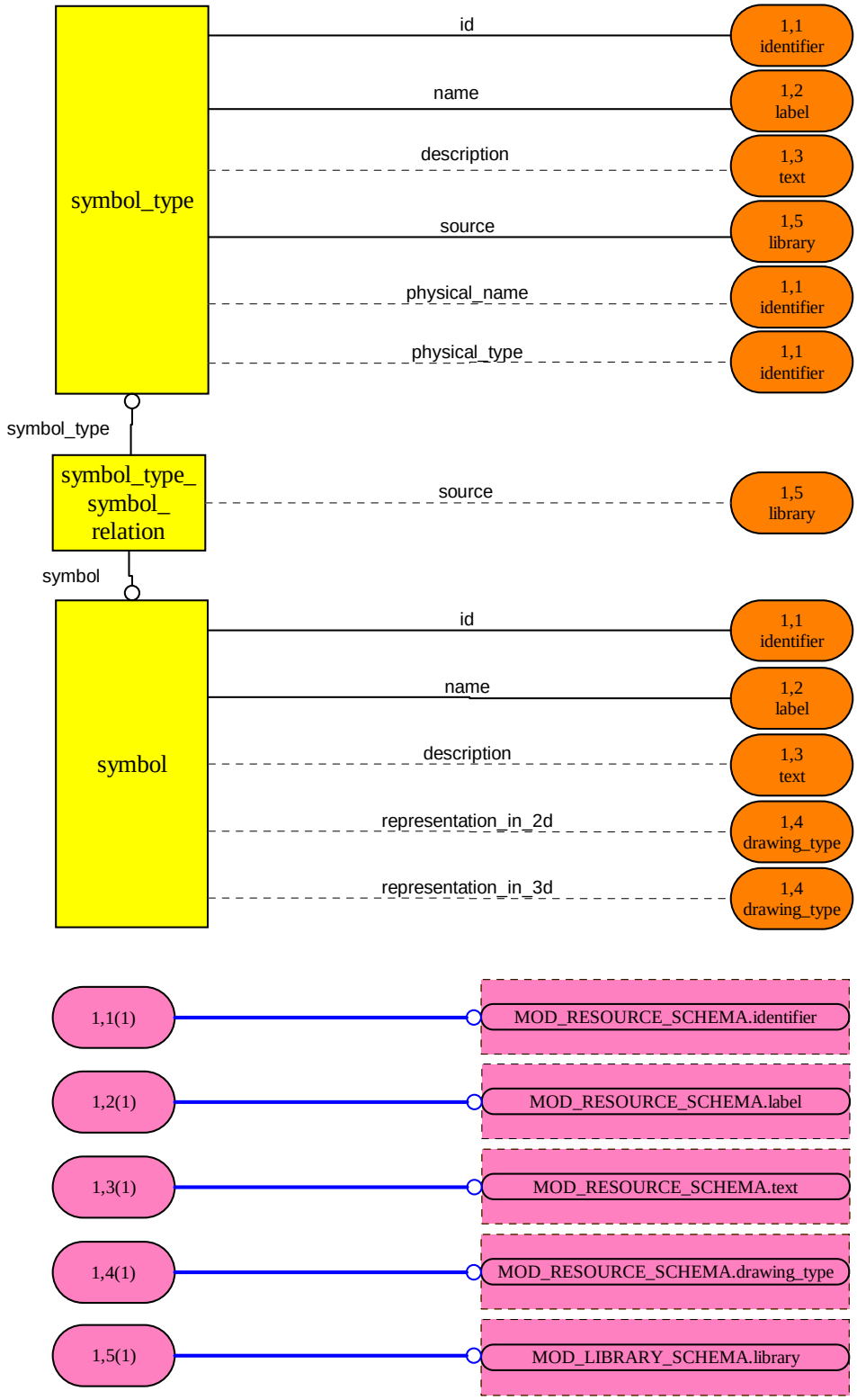


Figure D.1 - MOD_SYMBOL_SCHEMA EXPRESS-G diagram 1 of 1

*)
(*

18. MOD_SYMBOL_SCHEMA

EXPRESS specification:

```
*)
SCHEMA MOD_SYMBOL_SCHEMA;
  REFERENCE FROM MOD_LIBRARY_SCHEMA
    (library);
  REFERENCE FROM MOD_RESOURCE_SCHEMA
    (identifier,
    label,
    text,
    drawing_type);
(*
```

18.1. symbol

A [symbol] is a graphical representation of a product or component.

EXPRESS specification:

```
*)
  ENTITY symbol;
    id                          :identifier;
    name                        :label;
    description                 :OPTIONAL text;
    representation_in_2d        :OPTIONAL drawing_type;
    representation_in_3d        :OPTIONAL drawing_type;
  END_ENTITY;
(*
```

18.2. symbol_type

A [symbol_type] is a typical symbol definition which is not instantiated in a drawing or design model.

EXPRESS specification:

```
*)
  ENTITY symbol_type;
    id                          :identifier;
    name                        :label;
    description                 :OPTIONAL text;
    physical_name               :OPTIONAL identifier;
    physical_type               :OPTIONAL identifier;
    source                      :library;
  END_ENTITY;
(*
```

18.3. symbol_type_symbol_relation

EXPRESS specification:

```
*)
  ENTITY symbol_type_symbol_relation;
    symbol_type                :symbol_type;
    symbol                     :symbol;
    source                     :OPTIONAL library;
  END_ENTITY;
END_SCHEMA;
(*
```

EXPRESS-G diagrams MOD_TERMS_AND_SYNONYMS

TLO Holland Controls B.V- 06.05.2004

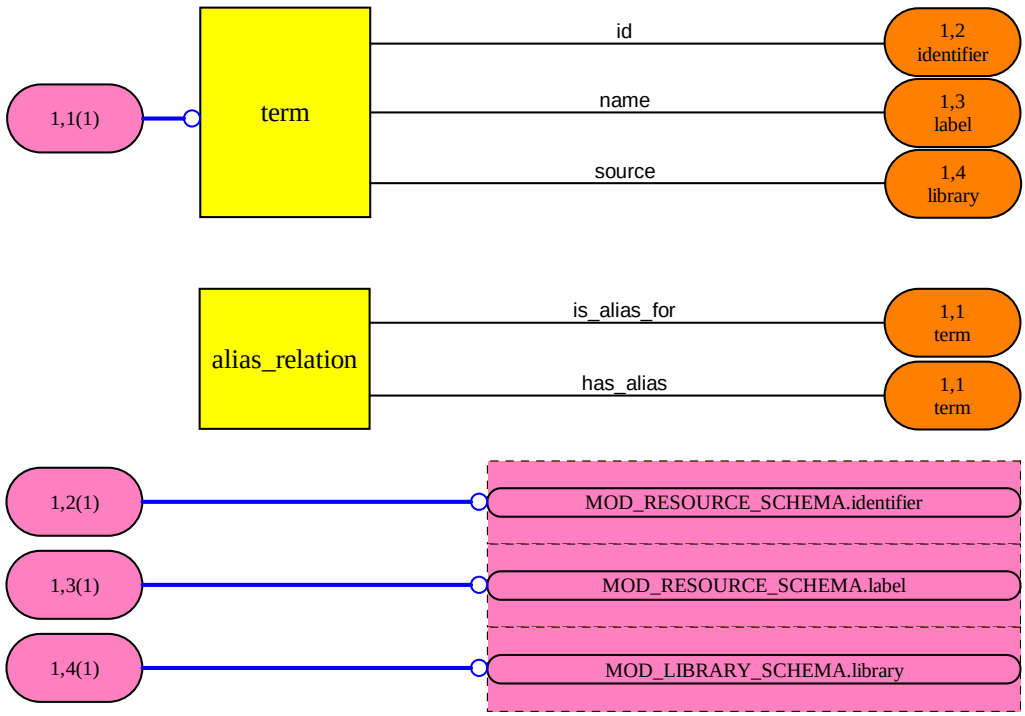


Figure D.1 - MOD_TERMS_AND_SYNONYMS EXPRESS-G diagram 1 of 1

```
*)
(*
```

19. MOD_TERMS_AND_SYNONYMS

EXPRESS specification:

```
*)
SCHEMA MOD_TERMS_AND_SYNONYMS;
  REFERENCE FROM MOD_LIBRARY_SCHEMA;
  REFERENCE FROM MOD_RESOURCE_SCHEMA
    (identifier,
    label);
(*
```

19.1. alias_relation

EXPRESS specification:

```
*)
  ENTITY alias_relation;
    is_alias_for              :term;
    has_alias                 :term;
  END_ENTITY;
(*
```

19.2. term

A [term] is a name for something.

EXPRESS specification:

```
*)
  ENTITY term;
    id                        :identifier;
    name                      :label;
    source                    :library;
  END_ENTITY;
END_SCHEMA;
(*
```

EXPRESS-G diagrams MOD_UNIT_SCHEMA

TLO Holland Controls B.V- 06.05.2004

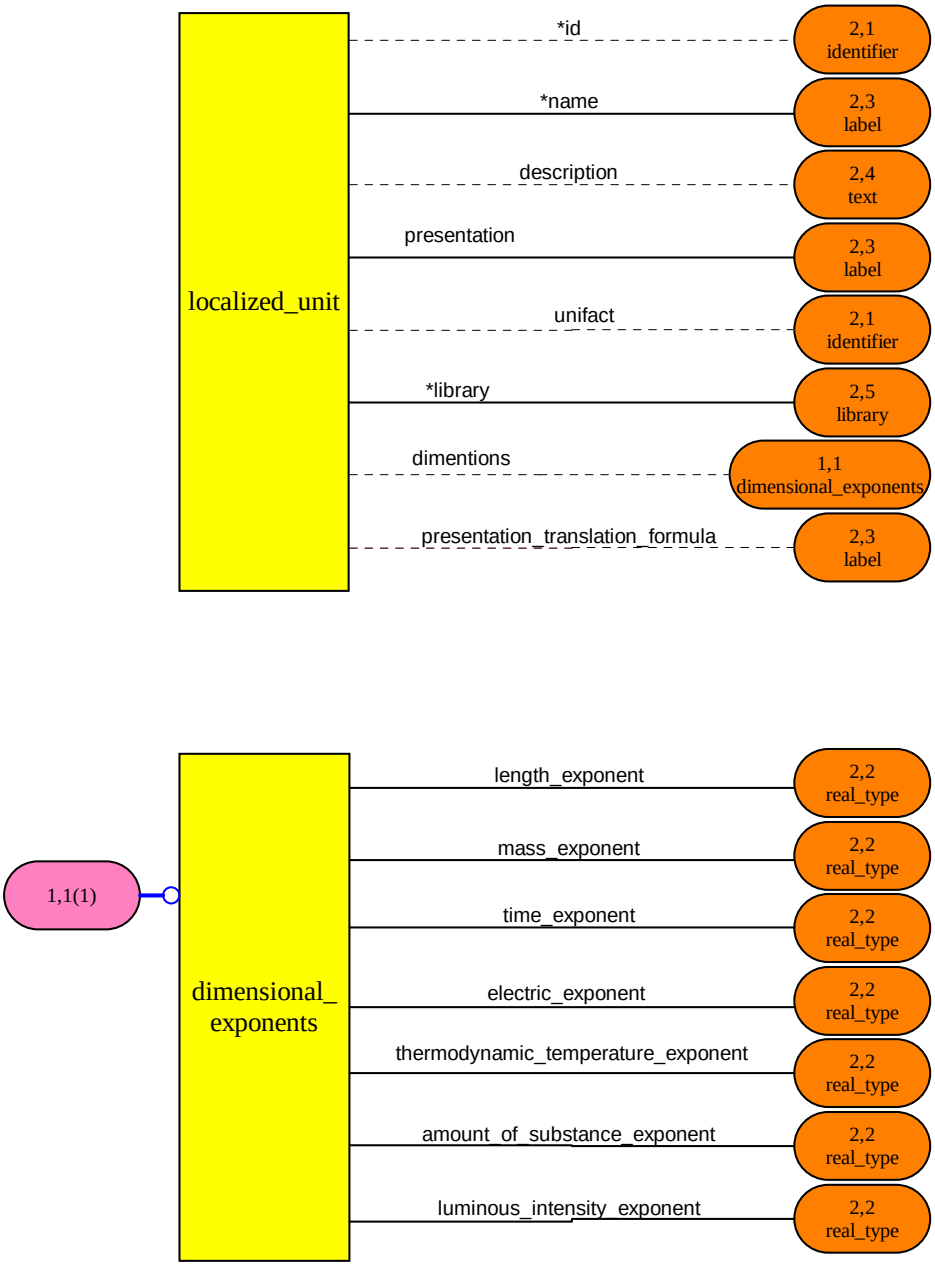


Figure D.1 - MOD_UNIT_SCHEMA EXPRESS-G diagram 1 of 2

TLO Holland Controls B.V- 06.05.2004

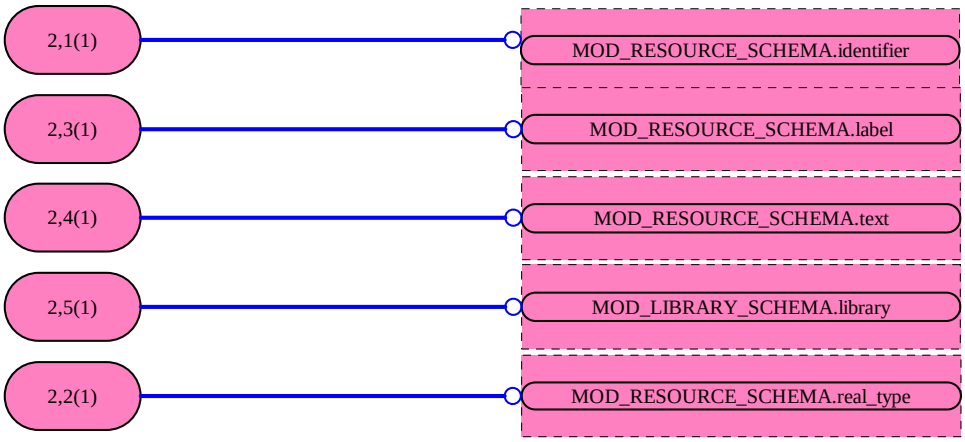


Figure D.2 - MOD_UNIT_SCHEMA EXPRESS-G diagram 2 of 2

*)
(*

20. MOD_UNIT_SCHEMA

EXPRESS specification:

```
*)
SCHEMA MOD_UNIT_SCHEMA;
  REFERENCE FROM MOD_LIBRARY_SCHEMA
    (library);
  REFERENCE FROM MOD_RESOURCE_SCHEMA
    (real_type,
    identifier,
    label,
    text);
(*
```

20.1. dimensional_exponents

The [dimensional_exponents] entity defines the powers of the dimensions of the seven base quantities.

EXPRESS specification:

```
*)
  ENTITY dimensional_exponents;
    mass_exponent              :real_type;
    thermodynamic_temperature_exponent  :    real_type;
    length_exponent            :real_type;
    time_exponent              :real_type;
    electric_exponent          :real_type;
    amount_of_substance_exponent: real_type;
    luminous_intensity_exponent : real_type;
  END_ENTITY;
(*
```

Attribute definitions:

mass_exponent : The power of the mass base quantity.

thermodynamic_temperature_exponent : The power of the thermodynamic temperature base quantity.

length_exponent : The power of the length base quantity.

time_exponent : The power of the time base quantity.

electric_exponent : The power of the electric current base quantity.

amount_of_substance_exponent : The power of the amount of substance base quantity.

luminous_intensity_exponent : The power of the luminous intensity base quantity.

20.2. localized_unit

An [localised unit] is a local administered unit of measurement.

EXPRESS specification:

```
*)
  ENTITY localized_unit;
    id                          :OPTIONAL identifier;
    presentation                :label;
    name                        :label;
    unifact                     :OPTIONAL identifier;
    library                     :library;
    dimentions                  :OPTIONAL dimensional_exponents;
    description                 :OPTIONAL text;
    presentation_translation_formula    :     OPTIONAL label;
  UNIQUE
    id_uniqueness     :     id,
              library;name_uniqueness   :     name,
              library;
  END_ENTITY;
END_SCHEMA;
(*
```