|  |  |
| --- | --- |
| **Explanation:**  A Notion Frame consists of a parameter and a range of discrete or non-discrete values for this parameter, such as determined by the sensory capacity of the perceiving party. Possible discrete values will be given between curly brackets {}.  A concept can be a generic concept (starting with C\_ and a name, but no further precisioning). It can also be a specific concept, which references a generic concept, additional Notion Frames and at least one Notion Value. | **Abstract concepts**  Notions can be considered as properties of things; they are represented as parameters. A parameter value can be used for the subdivision of a class of things, in which case it plays the role of discriminator.  Although it is theoretically possible to define everything by means of Notions, it is more convenient to start a model with a given set of abstract concepts, which are subsequently made more precise with Notions. |
|  |  |
| Notion Value(s) may act as discriminator(s) for the specification of specific concepts, i.e. subclasses of the concept. Notion Frames without values can be considered as parameters (variables) of the concept.  Notion Frames and Notion Values for specific concepts are placed between straight brackets [].  Notion parameters and parameter values are given between curved brackets (). | **Generic concepts and Notion Frames**  The sharing of data is only possible if two or more perceptive frames share certain concepts and/or Notion Frames. Several common concepts and Notion Frames will be given in the examples below. |
|  |  |
| **Perceptive Frames**  Every actor in an economical network has its own ‘view’ on a subject matter. Such views, that include specific terminology, are in general reasonably well standardized within the actor’s discipline. A perceptive frame can thus be modelled for each discipline.  A Perceptive Frame consists of one or more Notion Frames. |  |

**Example of a hierarchy of notion frames:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Generic Notion Frames** | **Valid for:** | **Parameter & Value Range** | **Unit** | **Derived from** | **Source, Measurement, Discriminator or Calculus** |
| C\_Person |  |  |  |  | *Generic Concept* |
| NF\_Date (JD) | Generic | Julian\_Date | Day |  | ISO 8601 |
| NF\_Actual\_date (JD) | Generic | Julian\_Date | Day | * NF\_Date (JD) | ISO 8601 |
| NF\_Birth\_day (JD) | C\_Person | Julian\_Date | Day | * NF\_Date (JD) | National Register |
| NF\_Legal\_age (Y) | C\_Person | Duration | Year | * NF\_actual\_date (JD) * NF\_birth\_day (JD) | NF\_Legal\_age = NF\_actual\_date\_JD - NF\_birth\_day\_JD |
| NF\_Legal\_Maturity (m) | C\_Person | m = {Child; Adult} | Logical | * NF\_legal\_age (Y) | If [NF\_legal\_age (Y<18)] then m =‘Child’; else m = ‘Adult’ |
| C\_Child | C\_Person |  |  | * NF\_Legal Maturity (m=Child) | *Derived Concept (subclass)* |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PERCEPTIVE FRAME LEGAL** | |  |  |  |  |
| C\_Natural\_Person |  |  |  |  | *Generic Concept* |
| NF\_ID | C\_Natural\_Person | BSN number | 9 digit integer |  | National Register |
| NF\_Name | C\_Natural\_Person | “Harry Potter” | Text |  | National Register |
| NF\_Birth\_Day (Y) | C\_Natural\_Person | Calender Date | {int(year), int(month), int(day)} |  | National Register |
| NF\_Legal\_age (Y) | C\_Person | Duration | Year | * NF\_actual\_date (JD) * NF\_birth\_day (JD) | NF\_Legal\_age = NF\_actual\_date\_JD - NF\_birth\_day\_JD |
| NF\_Legal\_Maturity (m) | C\_Natural\_Person | m = {immature, mature} | Logical | NF\_legal\_age (Y) | If [NF\_legal\_age (Y<18)] then m =‘immature’; else m = ‘mature’] |
| NF\_TBS (t) | C\_Natural\_Person | t = {verdict, none} | Logical |  |  |
| NF\_Accountable (a) | C\_Natural\_Person | a = {not\_accountable, accountable} | Logical | * NF\_Legal\_age (Y) * NF\_TBS (t) | If [NF\_legal\_age (Y<18)] then a =‘not\_accountable’; else if [NF\_TBS (t) = ‘verdict’] then a = ‘not\_accountable’; else a= ‘accountable’] |
| NF\_Gender (G) | C\_Natural\_Person | G = {Male, Female} |  |  |  |
| NF\_parent\_1 | C\_Natural\_Person | BSN number | 9 digit integer |  |  |
| NF\_parent\_2 | C\_Natural\_Person | BSN number | 9 digit integer |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PERCEPTIVE FRAME MEDICAL** | |  |  |  |  |
| C\_Patient |  |  |  |  | *Generic Concept* |
| NF\_ID | C\_Patient | BSN number | 9 digit integer |  |  |
| NF\_Legal\_Gender (GL) | C\_Patient | GL = {Male, Female} | 16-8-2024 |  |  |
| NF\_Biological\_Gender (GB) | C\_Patient | GB = {Male, Female} | 16-8-2024 |  |  |
| NF\_Insurance\_ID | C\_Patient | Insurance ID number |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **PERCEPTIVE FRAME SOCIAL PLATFORM** | | |  |  |  |
| C\_Person |  |  |  |  | *Generic Concept* |
| NF\_Email\_ID | C\_Person | Text including ‘@’ |  |  |  |
| NF\_Name | C\_Person | First name, family name | Text |  |  |
| NF\_Birth\_Day\_JD | C\_Person | integer | JD |  |  |
| NF\_Social\_Age (Yr) | C\_Person | Integer positive | year |  | NF\_Social\_age = NF\_actual\_date\_JD - NF\_birth\_day\_JD |
| NF\_Social\_Accountability (SA) | C\_Person | SA = {Child, Adult} | Logical |  | If [NF\_social\_age (Y<18)] then SA =‘Child’; else SA = ‘Adult’ |
| NF\_Social\_Gender (SG) | C\_Person | SG = {Male, Female} | Logical |  | Gender given by member of platform (may not be the same as legal gender). |
| C\_Woman | C\_Person |  |  |  | *Derived Concept* Condition: NF\_Social\_Accountability = Adult AND NF\_Social\_Gender = Female. |
| C\_Man | C\_Person |  |  |  | *Derived Concept* Condition: NF\_Social\_Accountability = Adult AND NF\_Social\_Gender = Male. |
| C\_Girl | C\_Person |  |  |  | *Derived Concept* Condition: NF\_Social\_Accountability = Child AND NF\_Social\_Gender = Female. |
| C\_Boy | C\_Person |  |  |  | *Derived Concept* Condition: NF\_Social\_Accountability = Child AND NF\_Social\_Gender = Male. |

**Example of concepts used for modelling graphs:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Generic Notion Frame** | **Valid for:** | **Parameter & Value Range** | **Unit** | **Derived from** | **Measurement, Discriminator or Calculus** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Graph Notion Frames** | | | | | |
| NF\_Orientation (o) |  | o = {start; end} |  |  |  |
| NF\_Predicate (L) |  | L = {true, false} |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Graph - Symmetrical** | | | | | |
| C\_Node |  |  |  | *Generic Concept* |  |
| C\_Arc |  |  |  | *Generic Concept* |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Graph - Oriented** | | | | | |
| C\_Source\_Node |  | = C\_Node [NF\_Orientation (o=start)] |  | *Derived Concept* |  |
| C\_Target\_Node |  | = C\_Node [NF\_Orientation (o=end)] |  | *Derived Concept* |  |
| C\_Arc\_Directed |  | = C\_Arc [NF\_Orientation(o)] |  | *Derived Concept* |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Graph Predicate Logic (1)** | | | |
| C\_Logic\_Object |  | = C\_Source\_Node [NF\_ Predicate (T)] | *Derived Concept* |
| C\_Logic\_Subject |  | = C\_Target\_Node [NF\_ Predicate (T)] | *Derived Concept* |
| C\_Logic\_Predicate |  | = C\_Arc\_Directed [NF\_ Predicate (T)] | *Derived Concept* |

|  |  |  |  |
| --- | --- | --- | --- |
| **Graph Predicate Logic (2) [is same as previous]** | | | |
| C\_Logic\_Object | = C\_Node [NF\_Orientation (o=start), NF\_ Predicate (T)] | *Derived Concept* |  |
| C\_Logic\_Subject | = C\_Node [NF\_Orientation (o=end), NF\_ Predicate (T)] | *Derived Concept* |  |
| C\_Logic\_Predicate | = C\_Arc [NF\_Orientation(o), NF\_ Predicate (T)] | *Derived Concept* |  |

**Example of concepts used for modelling topology:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Topology Notion Frames** | | | | |
| NF\_Topology\_Dimension (n) | n = {0; 1; 2; 3} | Integer (0-3) |  |  |
| NF\_Topology\_Location (l) | l = {Inside; (Border / Boundary); Outside} | Logical |  | Border is equivalent to Boundary |
| NF\_Orientation (o) | o = {start; end} | Logical |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Topological Graph 1 (non-oriented, dimension implicit)** | | | |
| C\_Topological\_Node = | = C\_Node [NF\_Topology\_Dimension (n)] |  | *Derived Concept* |
| C\_Topological\_Arc = | = C\_Arc [NF\_Topology\_Dimension (n)] |  | *Derived Concept.* Top\_Arc is equivalent to Top\_Reference |

|  |  |  |  |
| --- | --- | --- | --- |
| **Topological Graph 2 (oriented, dimension implicit)** | | | |
| C\_n\_Top\_Source | = C\_Node [NF\_Topology\_Dimension (n), NF\_Orientation (o=source)] |  | *Derived Concept.* |
| C\_n\_Top\_Target | = C\_Node [NF\_Topology\_Dimension (n-1), NF\_Orientation (o=target)] |  | *Derived Concept.* |
| C\_n\_Top\_Reference | = C\_Arc [NF\_Orientation(o)] |  | *Derived Concept.* |

|  |  |  |  |
| --- | --- | --- | --- |
| **Topological Graph 3 (non-oriented, dimension explicit, non-located)** | | |  |
| C\_Vertex = | C\_Topological\_Node [NF\_Topology\_Dimension (0)] |  | *Derived Concept.* |
| C\_Edge = | C\_Topological\_Node [[NF\_Topology\_Dimension (1)] --> 2\*C\_Vertex] | Edge refers to 2 vertices | *Derived Concept.* |
| C\_Connected\_Edges = | C\_Topological\_Node [n\*[NF\_Topology\_Dimension (1)]] --> (n+1)\*C\_Vertex] | Connected Edges = Open Loop; has common vertices. V-E=1. | *Derived Concept.* |
| C\_Loop | C\_Topological\_Node [n\*[NF\_Topology\_Dimension (1)]] --> n\*C\_Vertex] | Closed Loop. E=V | *Derived Concept.* |
|  |  |  |  |
| C\_Face = | C\_Topological\_Node [NF\_Topology\_Dimension (2)] |  | *Derived Concept.* |
| C\_Connected\_Faces = | C\_Topological\_Node [NF\_Topology\_Dimension (2)] | Connected Faces = Open Shell; has common Edges. | *Derived Concept.* |
| C\_Shell = | C\_Topological\_Node [NF\_Topology\_Dimension (2)] | Closed Shell. V-E+F=2 (convex polyhedron) | *Derived Concept.* |
| C\_Volume = | C\_Topological\_Node [NF\_Topology\_Dimension (3)] --> C\_Shell [NF\_Topology\_Location(border)] |  | *Derived Concept.* |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Topological Graph 4 (non-oriented, dimension explicit, located)** | | |  |
| C\_Edge\_Bounding\_Vertex = | C\_Topological\_Node [NF\_Topology\_Dimension (1), NF\_Topology\_Dimension (0), NF\_Topology\_Location (Border)] |  | *Derived Concept.* |
| C\_Face\_Bounding\_Edge = | C\_Topological\_Node [NF\_Topology\_Dimension (2), NF\_Topology\_Dimension (1), NF\_Topology\_Location (Border)] |  | *Derived Concept.* |
| C\_Volume\_Bounding\_Face = | C\_Topological\_Node [NF\_Topology\_Dimension (3), NF\_Topology\_Dimension (2), NF\_Topology\_Location (Border)] |  | *Derived Concept.* |
| C\_Inner\_Loop = | C\_Topological\_Node [[n\*[NF\_Topology\_Dimension (1)]] --> n\*C\_Vertex, [NF\_Topology\_Location (inside)] | Inner Loop = Hole | *Derived Concept.* |
| C\_Inner\_Shell = | C\_Topological\_Node [[n\*[NF\_Topology\_Dimension (2)]] --> n\*C\_Edge, [NF\_Topology\_Location (inside)]] | Inner Shell = Void | *Derived Concept.* |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Topological Graph 5 (cellular)** | | | |
| C\_Edge\_Bounding\_Vertex = | C\_Topological\_Node [NF\_Topology\_Dimension (1) --> NF\_Topology\_Dimension (0), NF\_Topology\_Location (Border)] |  | *Derived Concept.* |
| C\_Face\_Bounding\_Edge = | C\_Topological\_Node [NF\_Topology\_Dimension (2) --> NF\_Topology\_Dimension (1), NF\_Topology\_Location (Border)] |  | *Derived Concept.* |
| C\_Cell\_Bounding\_Face = | C\_Topological\_Node [NF\_Topology\_Dimension (3) --> NF\_Topology\_Dimension (2), NF\_Topology\_Location (Border)] |  | *Derived Concept.* |
| C\_Cell = | C\_Topological\_Node [NF\_Topology\_Dimension (3)] |  | *Derived Concept.* |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Topological Graph 6 (Enclosure Topology)** | | | |
| C\_Vertex\_on\_Vertex = | C\_Topological\_Node [NF\_Topology\_Dimension (0) --> NF\_Topology\_Dimension (0), NF\_Topology\_Location (Inside)] |  | *Derived Concept.* |
| C\_Vertex\_on\_Edge = | C\_Topological\_Node [NF\_Topology\_Dimension (0) --> NF\_Topology\_Dimension (1), NF\_Topology\_Location (Inside)] |  | *Derived Concept.* |
| C\_Vertex\_on\_Face = | C\_Topological\_Node [NF\_Topology\_Dimension (0) --> NF\_Topology\_Dimension (2), NF\_Topology\_Location (Inside)] |  | *Derived Concept.* |
| C\_Vertex\_in\_Volume = | C\_Topological\_Node [NF\_Topology\_Dimension (0) --> NF\_Topology\_Dimension (3), NF\_Topology\_Location (Inside)] |  | *Derived Concept.* |
| C\_Edge\_on\_Edge = | C\_Topological\_Node [NF\_Topology\_Dimension (1) --> NF\_Topology\_Dimension (1), NF\_Topology\_Location (Inside)] |  | *Derived Concept.* |
| C\_Edge \_on\_Face = | C\_Topological\_Node [NF\_Topology\_Dimension (1) --> NF\_Topology\_Dimension (2), NF\_Topology\_Location (Inside)] |  | *Derived Concept.* |
| C\_Edge\_in\_Volume = | C\_Topological\_Node [NF\_Topology\_Dimension (1) --> NF\_Topology\_Dimension (3), NF\_Topology\_Location (Inside)] |  | *Derived Concept.* |
| C\_Face\_on\_Face = | C\_Topological\_Node [NF\_Topology\_Dimension (2) --> NF\_Topology\_Dimension (2), NF\_Topology\_Location (Inside)] |  | *Derived Concept.* |
| C\_Face\_in\_Volume = | C\_Topological\_Node [NF\_Topology\_Dimension (2) --> NF\_Topology\_Dimension (3), NF\_Topology\_Location (Inside)] |  | *Derived Concept.* |
| C\_Volume\_in\_Volume = | C\_Topological\_Node [NF\_Topology\_Dimension (3) --> NF\_Topology\_Dimension (3), NF\_Topology\_Location (Inside)] |  | *Derived Concept.* |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Functional Design** | | | |
| ***Separation*** | |  |  |
| NF\_Separation\_thermal | NF\_Topology\_Location (Boundary), NF\_Thermal\_Resistance (Rc) | kg-1⋅m-2⋅s3⋅K |  |
| NF\_Separation\_accessability | NF\_Topology\_Location (Boundary), NF\_Human\_Access\_Function {door, gateway, corridor, etc} |  |  |
| NF\_Separation\_security | NF\_Topology\_Location (Boundary), NF\_Security\_Rating (Sr) | Unit = ? |  |
| NF\_Separation\_air\_circulation | ? |  |  |
| NF\_Separation\_noise | NF\_Topology\_Location (Boundary), NF\_Noise\_Resistance (Nr) | Unit = dB |  |
| NF\_Separation\_fire | NF\_Topology\_Location (Boundary), NF\_Fire\_Retardance (Rc) | Unit = hour |  |
| ***Connection*** | |  |  |
| NF\_Connection\_ventilation | NF\_Topology\_Location (Boundary), NF\_Air\_Renewal\_Rate (ARR) | Hour-1 |  |
| NF\_Connection\_visual | NF\_Topology\_Location (Boundary), NF\_Visual\_connectivity\_factor (VCF) | ? |  |
| NF\_Connection\_natural\_light | NF\_Topology\_Location (Boundary), NF\_Daylight\_Factor (Df) | % |  |
| NF\_Connection\_solar radiation | NF\_Topology\_Location (Boundary), NF\_Daylight\_Factor (DF) | % |  |
| NF\_Connection\_escape\_routes | NF\_Topology\_Location (Boundary), NF\_Escape\_Route\_Rating (ERR) | s |  |
| ***Containment*** | |  |  |
| NF\_Containment\_sound | NF\_Topology\_Location (Boundary), NF\_Reverberation\_Time (s) | s |  |
| NF\_Containment\_smoke | NF\_Topology\_Location (Boundary), NF\_Smoke\_Disposal\_Rate (SDR) | s |  |
| NF\_Containment\_Thermal\_Gradient | NF\_C\_Volume, NF\_Thermal\_Gradient (TG) | K⋅m-1 |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Horizontal / Vertical** | | | **Explanation** |
| NF\_Horizontal\_Tolerance = | NF\_Tilt (h) | % |  |
| NF\_Vertical\_Tolerance = | NF\_Tilt (v) | % |  |
| NF\_Flat\_Plane\_Horizontal = | [(NF\_Unit\_Normal\_Vector\_Plane) • (NF\_Unit\_Vector\_Direction\_of\_Gravitational\_Force)] < 1-NF\_Horizontal\_Tolerance(h) | Logical (true or false). | A flat plane is horizontal iff the dot product of its unit normal vector and the direction of gravitational force equals 1. |
| NF\_Flat\_Plane\_Vertical = | (NF\_Unit\_Normal\_Vector\_Plane) • (NF\_Unit\_Vector\_Direction\_of\_Gravitational\_Force) > NF\_Vertical\_Tolerance(v) | Logical (true or false). | A flat plane is vertical iff the dot product of its unit normal vector and the direction of gravitational force equals 0. |
| NF\_Curved\_Plane\_Vertical = | (NF\_Unit\_Normal\_Vector\_Plane) • (NF\_Unit\_Vector\_Direction\_of\_Gravitational\_Force) > NF\_Vertical\_Tolerance(v) | Logical (true or false). | A curved plane is vertical iff the dot product of a normal vector on any part of the curved surface and the direction of gravitational force equals 0. |
| NF\_Flat\_Plane\_Tilted = | (NF\_Unit\_Normal\_Vector\_Plane) • (NF\_Unit\_Vector\_Direction\_of\_Gravitational\_Force) > [1 - NF\_Horizontal\_Tolerance(h)] AND < NF\_Vertical\_Tolerance(v) | % |  |

|  |  |  |
| --- | --- | --- |
| **Abstraction 1** | | |
| C\_Unit\_of\_Knowledge |  | *Generic Concept.* Knowledge that consists of mathematical models of an existing or planned physical reality. |
| C\_Generic\_UoK | = UoK [x\*p] | *Derived Concept.* UoK that has only variables (parameters). P = parameter, X = number of parameters. How to model this as a Notion Frame? |
| NF\_Abstraction = | {generalization, specialization} |  |
| NF\_Abstraction (Specialization) = | C\_UoK (2) is a specialization of C\_UoK (1) if it has the same set of parameters, plus one more. | Extending the number of parameters. |
| NF\_Abstraction (Generalization) = | C\_UoK (1) is a generalization of C\_UoK (2) if it has the same set of parameters, but one less. | Reducing the number of parameters. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Specification 1** | | | |
| C\_Unit\_of\_Knowledge |  |  | *Generic Concept.* Knowledge that consists of mathematical models of an existing or planned physical reality. |
| C\_Generic\_UoK | = UoK [x\*p] |  | *Derived Concept.* UoK that has only variables (parameters). P = parameter, X = number of parameters. How to model this as a Notion Frame? |
| NF\_Specified = |  |  | One or more parameters of the UoK are specified. |
| NF\_Specific = |  |  | All parameters of the UoK, except uniquess, location and time, are specified. |
| NF\_Individual = | {ID} |  | UoK of an unique phenomenon (i.e. poduct or process). |
| NF\_Occurrence = | * Time * Time reference |  | Time (JD or s)  Zero point for measuring time |
| NF\_Location and orientation = | * Location and orientation * Location and orientation reference |  | Location (cartesian, spherical, polar, …) and orientation (polar)  Zero point for measuring location and orientation. |

|  |  |  |
| --- | --- | --- |
| **Mereology 1 - composition** | | |
| C\_Module |  | *Generic Concept.* |
| NF\_Composition (is\_part, is\_whole) | part = reference to module B (link), whole = reference to module A (link) |  |
| NF\_Selection (choice) | Choice = {none, selected, rejected, option} |  |

|  |  |  |
| --- | --- | --- |
| **Mereology 2 – functional composition** | | |
| NF\_Function (F) | Default value is ‘Functional\_Unit’. It can be replaced by a domain specific function. |  |
| NF\_Functional Module (F) | C\_Module, NF\_Function (F) |  |
| NF\_Functional Module Composition (F) | C\_Module, NF\_Function (F), NF\_Composition (is\_part, is\_whole), NF\_Selection (choice) |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **Mereology 3 (systems engineering)** | | |
| C\_Module |  | *Generic Concept.* |
| NF\_Problem\_Solving (problem, solution) |  |  |
| NF\_Selection (choice) | Choice = {none, selected, rejected, option} |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **Mereology 4 (design for unique projects)** | | |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **Mereology 4 (industrial design)** | | |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **Lifecycle 1** | | |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **Lifecycle 2** | | |
|  |  |  |