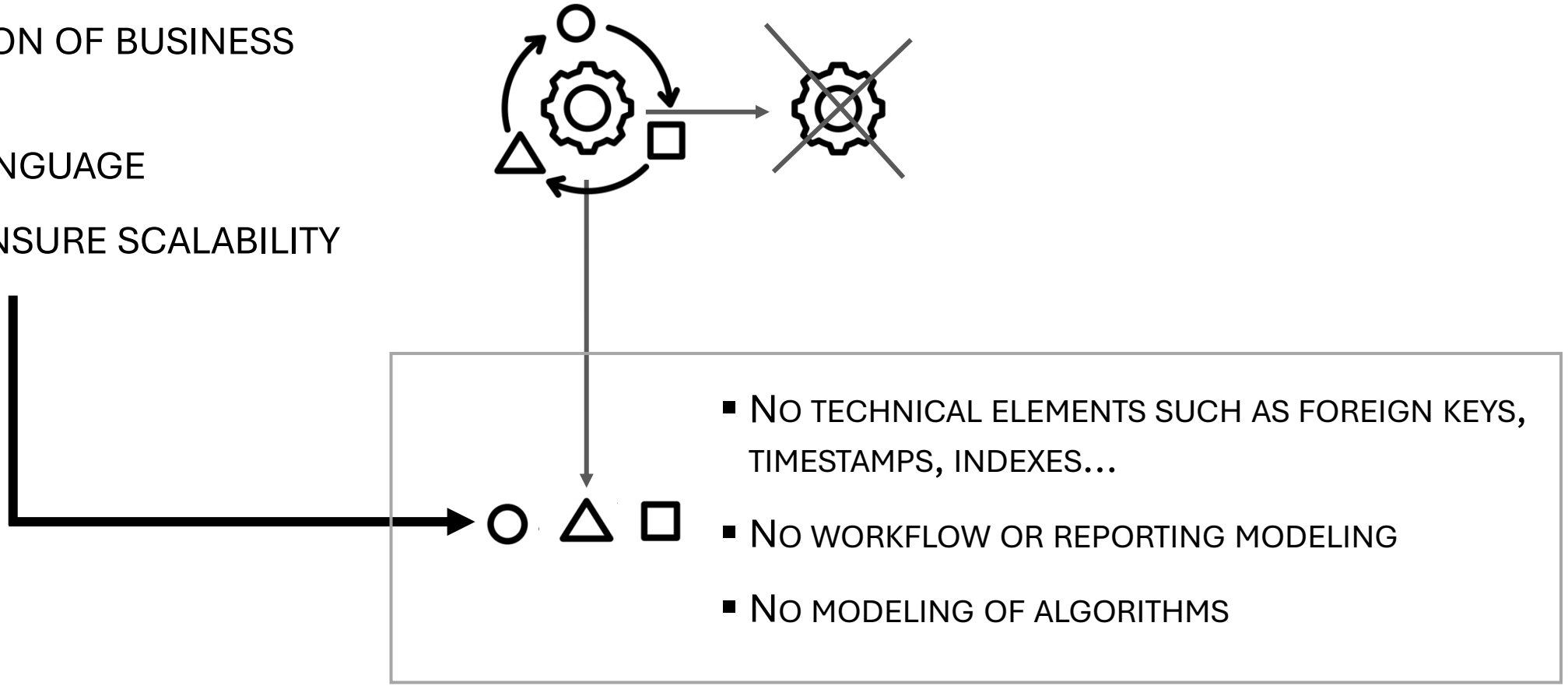
Conceptual Data Modeling

Practices

Mach 17, 2026

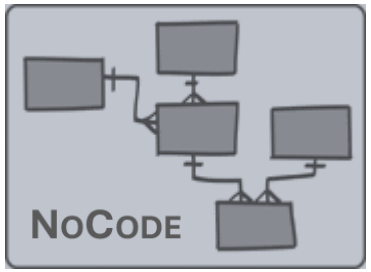
Objectives

- REPRESENTATION OF BUSINESS KNOWLEDGE
- IN NATURAL LANGUAGE
- GENERIC TO ENSURE SCALABILITY



- NO TECHNICAL ELEMENTS SUCH AS FOREIGN KEYS, TIMESTAMPS, INDEXES…
- NO WORKFLOW OR REPORTING MODELING
- NO MODELING OF ALGORITHMS

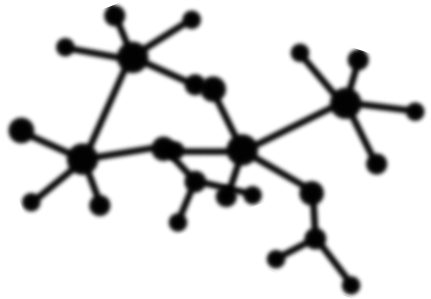
How is it useful?



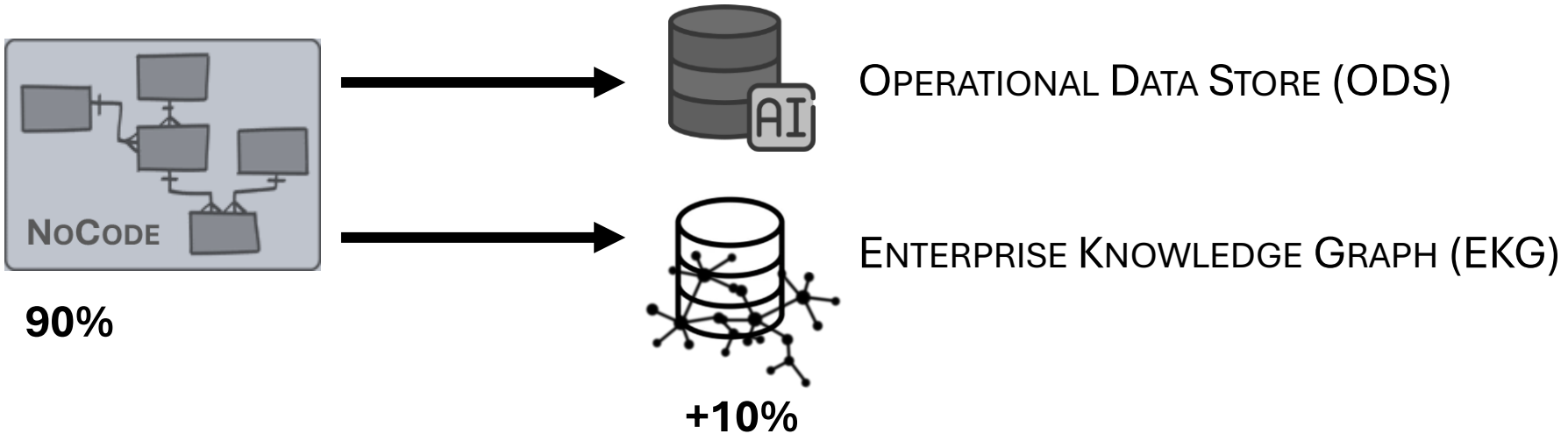
Thanks to NoCode-
LowCode the data model... → ... is translated into a database

Operational Data Store (ODS)
required for AI automations

Ontology



- THE BUSINESS DATA MODEL REPRESENTS 90% OF THE ONTOLOGY
- THE REMAINING 10% IS ADDED WHEN THE DATA MODEL IS PLACED INTO AN ENTERPRISE KNOWLEDGE GRAPH (EKG) WITH HEURISTICS



ODS + EKG

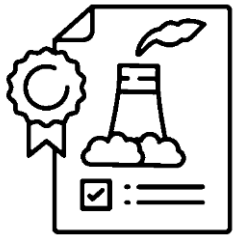
- Both repositories share the same business glossary and the same structural taxonomy (the inheritance hierarchy between concepts). The EKG has its own taxonomy for data values, depending on how we want to manage the granularity of the knowledge graph
- In addition, the business ERD describes relationships between business concepts, which can also be injected into the EKG as needed. It is important to understand that the ODS is based on a data model, whereas the EKG does not systematically require a data model to load information in the form of nodes and relationships
- Thus, we can have an analytics EKG that unfolds the ODS data values to produce a graph for analysis. And we can build an operational EKG from unstructured data sources such as PDF/DOC files, database dumps, support tickets, emails... all tacit knowledge transformed into explicit knowledge
- To properly build the graph of this operational EKG, it is necessary during loading to combine an LLM with the business glossary so that the LLM identifies the correct business tokens (business terms) in the text and turns them into nodes. This graph can then, depending on needs, be structured at the structure level (terms and links) or at the finer level of values (instances of terms and links)

ODS + EKG

- In summary, mastering business ERD modeling + the business glossary is fundamental, as this is the key to creating a shared ontology between the ODS and the EKG
- All of this is justified by the need to provide AI with data and knowledge that is as unambiguous as possible, to intrinsically reduce hallucinations and additionally leverage native RAG-Graph (Graph-QL, Cypher) and RAG-Graph-Vector to counter errors. This does not replace pre- and post-conditions implemented in a formal programming and/or rules language that does not rely on AI (see the LLM+SOA paper published on the Engage-Meta website)

Business description

+



THE DESCRIPTIONS OF THE COMPANY'S ACTIVITY IN THE FORM OF NARRATIVE USE CASES NOT WORKFLOWS

STANDARD MODELING PATTERNS BASED ON ERP, CRM, AND THE COMPANY'S INDUSTRY SECTOR

Business description

- Although there are no formal rules, the following practices can be applied:
  - A high level of writing quality is required, along with a genuine enjoyment of and habit in writing well
  - The aim is to capture the intrinsic value or the raison d'être (DNA) of the context being described, while trying to reduce or tone down organizational descriptions such as workflow, BPMN, etc.
  - Within the entrepreneurial/intrapreneurial mindset framework called META — Motion, Engagement, Treasury, and Assurance — the focus should be on describing the Motion, not the Engagement nor the T and A, of course (see Engage-Meta website)
  - Each use case should remain concise, under 5 pages, ideally 2 or 3. The objective must be clearly understood to write it effectively: it should allow the extraction of business concepts, key data elements and their relationships — in short, the ontology (intrinsic value, DNA, Motion) of the scope

AI as a Game Changer

IMPORTANT

TO HAVE A UNIFIED DATABASE WAS NOT AS CRITICAL FOR A COMPANY

IT WAS A **NICE-TO-HAVE** WHEREAS —AI→ NOW IT IS A **MUST-HAVE**

COMPLEXITY REQUIRED LABORIOUS MANUAL WORK TO ESTABLISH THE BUSINESS GLOSSARY AND MODELING

AI MODELING PROMPTS →

- THE LEVEL OF COMPLEXITY NO LONGER IMPACTS THE DATA MODELING WORKLOAD
- COMBINED WITH NOCODE–LOWCODE DATABASES -> THE TRANSITION FROM THE MODEL TO THE DATABASE IS ACCELERATED

Practices

https://www.visual-paradigm.com/

Vocabulary

ERD = Entity- Relationship Diagram = UML Class Diagram = Data Model

Class = Table

Relation = Association

Recap - Engineering Process

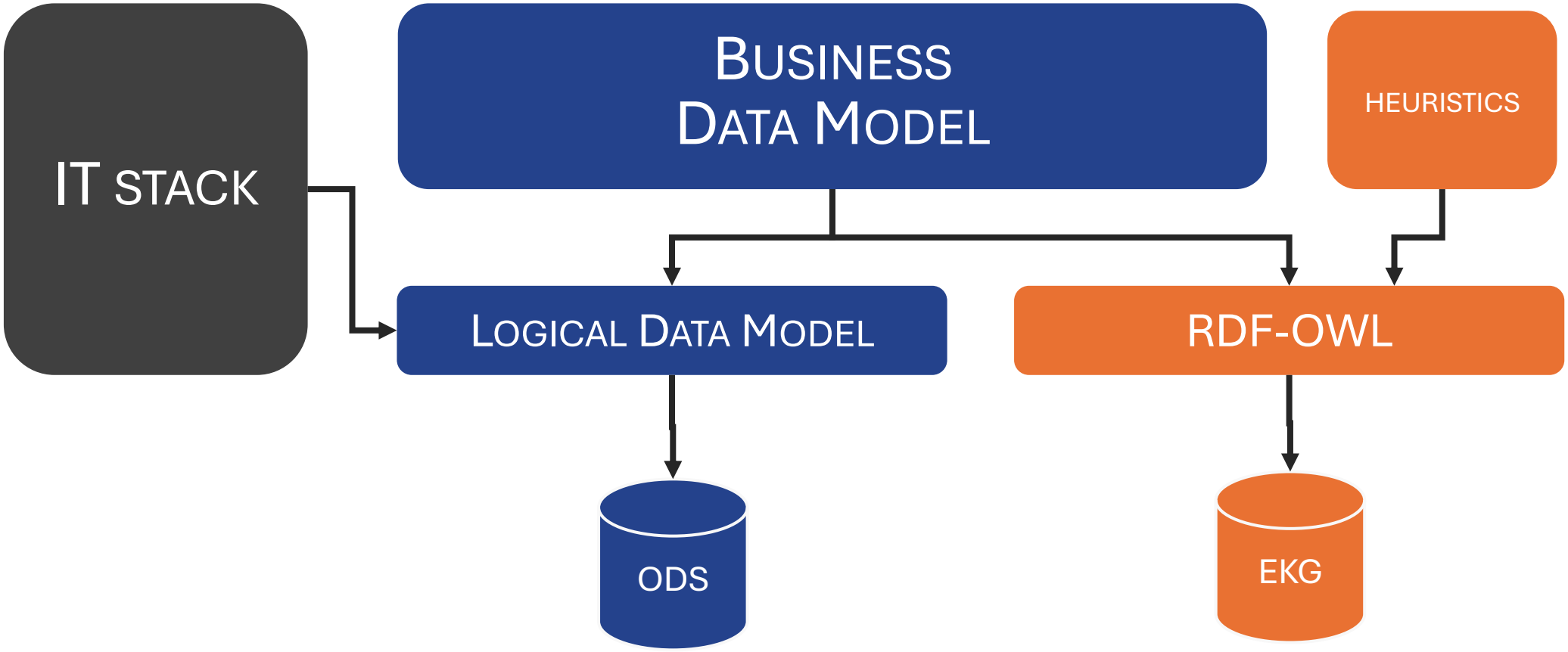


DIAGRAM NAVIGATOR

Provides a flat list of all diagrams created in the project (Class Diagrams, BPMN, ERDs, etc.)

VERSUS

MODEL EXPLORER

Organizes all model elements (classes, attributes, associations, packages, etc.) in a hierarchy

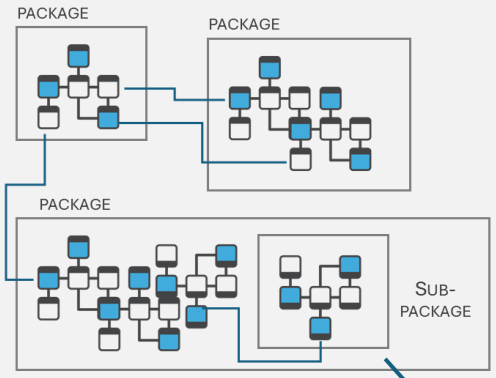
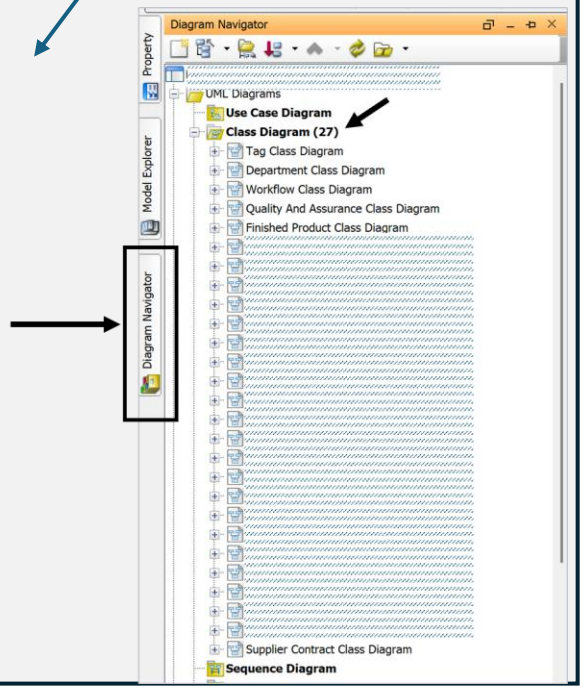


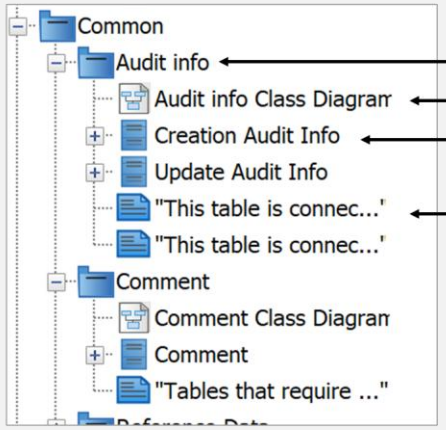
DIAGRAM NAVIGATOR

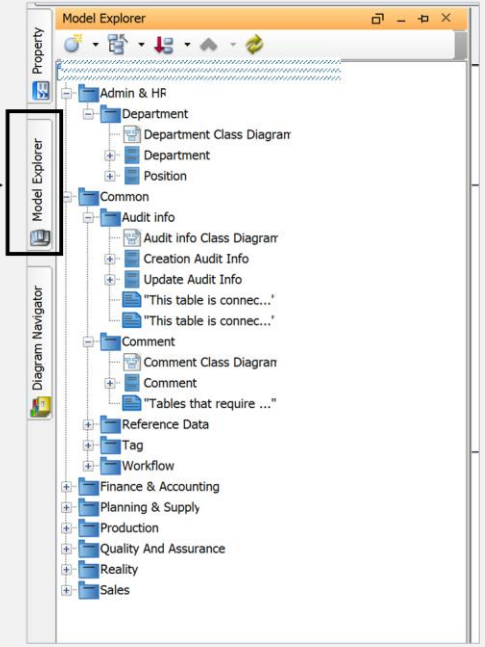
USEFUL WHEN YOU WANT A QUICK ENTRY POINT TO A SPECIFIC DIAGRAM BUT NOT IDEAL FOR MAINTAINING A WELL-ORGANIZED REPOSITORY



MODEL EXPLORER

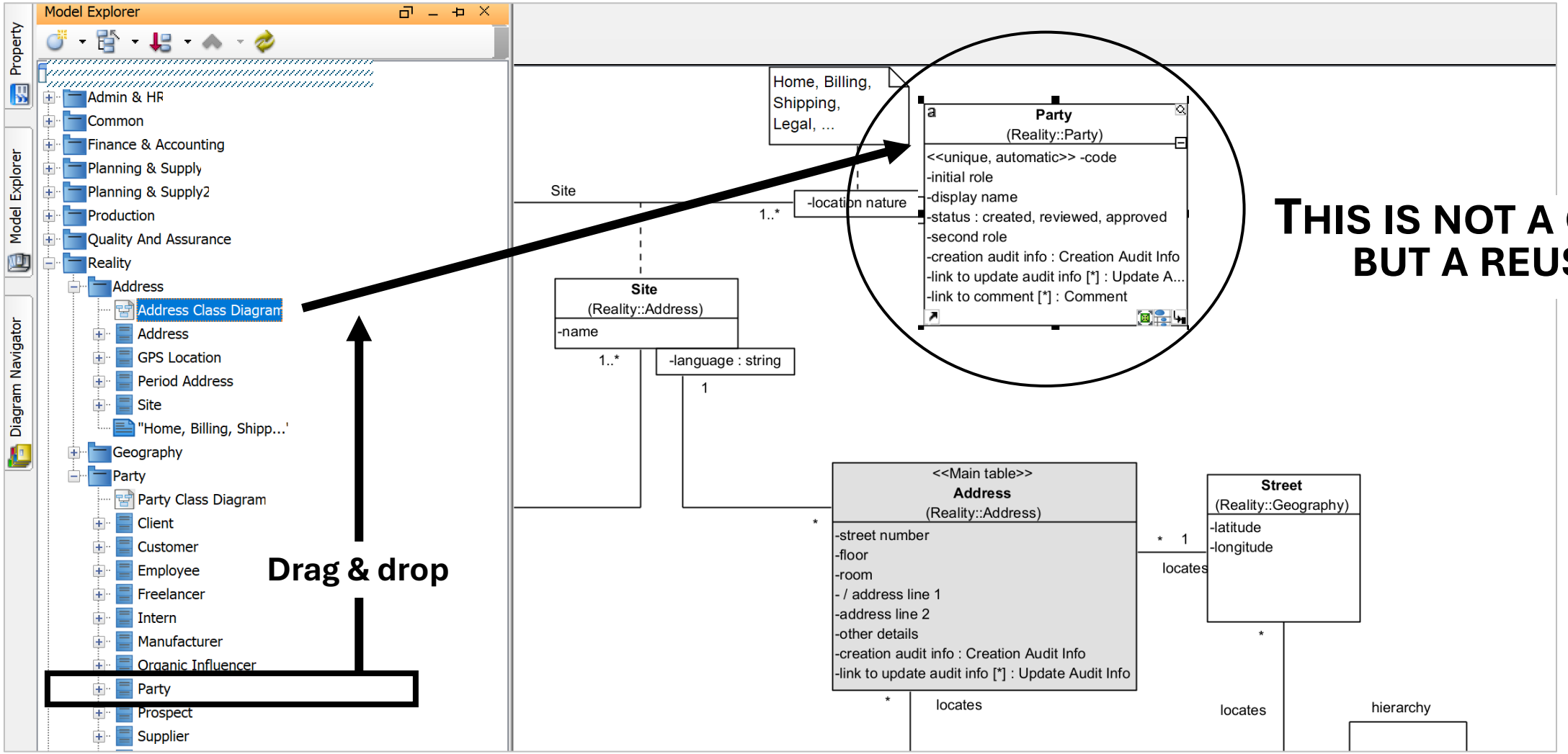
RECOMMENDED FOR PROJECTS WHERE THE GOAL IS TO BUILD A LONG-TERM, STRUCTURED REPOSITORY RATHER THAN AD-HOC DRAWINGS





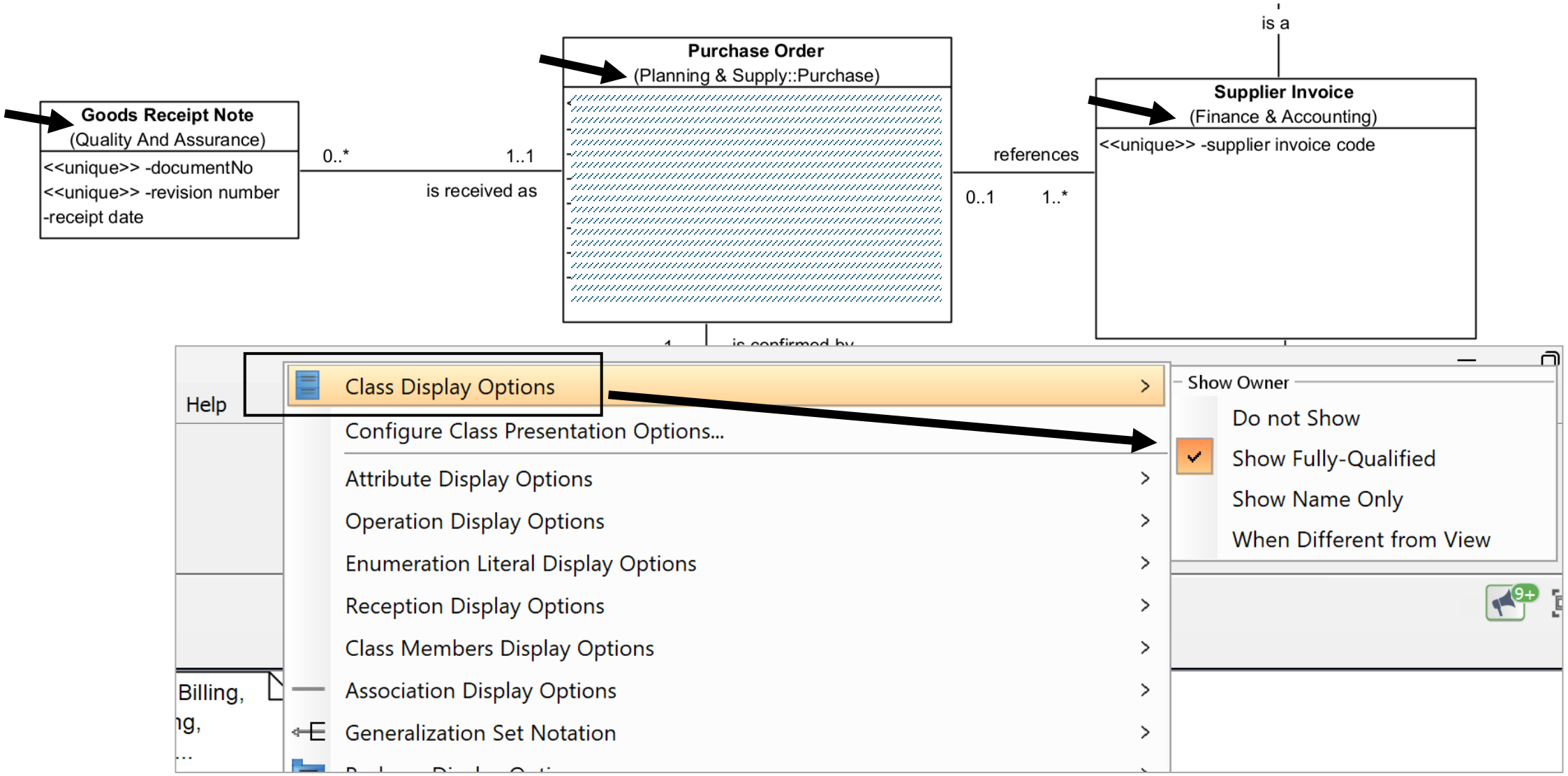
ENGAGE META

REUSING TABLES ACROSS PACKAGES



THIS IS NOT A COPY, BUT A REUSE

NAMING WITH PACKAGE LOCATION OF THE TABLE OWNER



ENGAGE META

How packages are determined

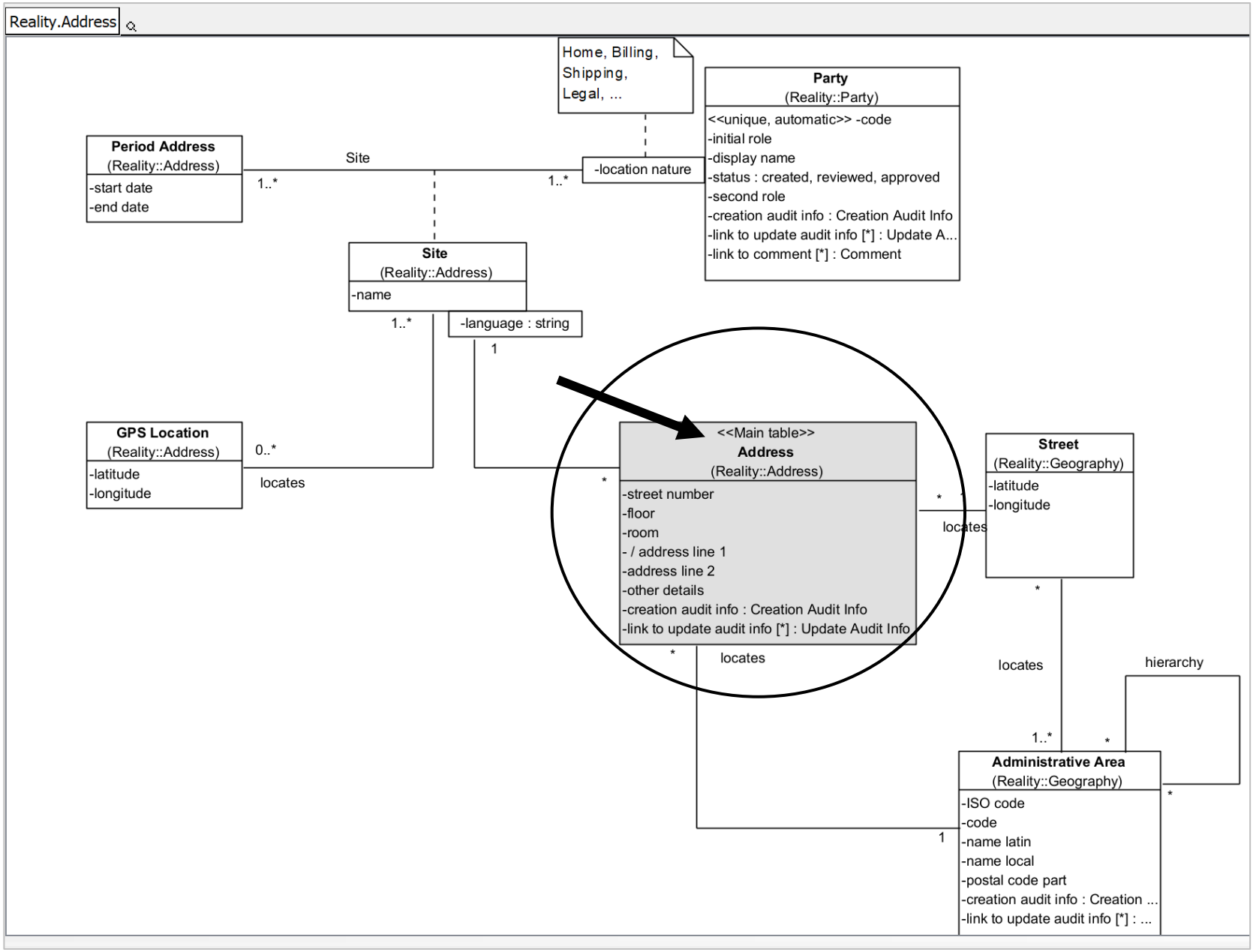
MAIN TABLE

A MAIN TABLE REPRESENTS A **BUSINESS CONCEPT** OF STRONG SEMANTIC SIGNIFICANCE, TYPICALLY MODELED WITH NO MORE THAN **TWENTY TABLES**

SYNONYM: CATEGORY

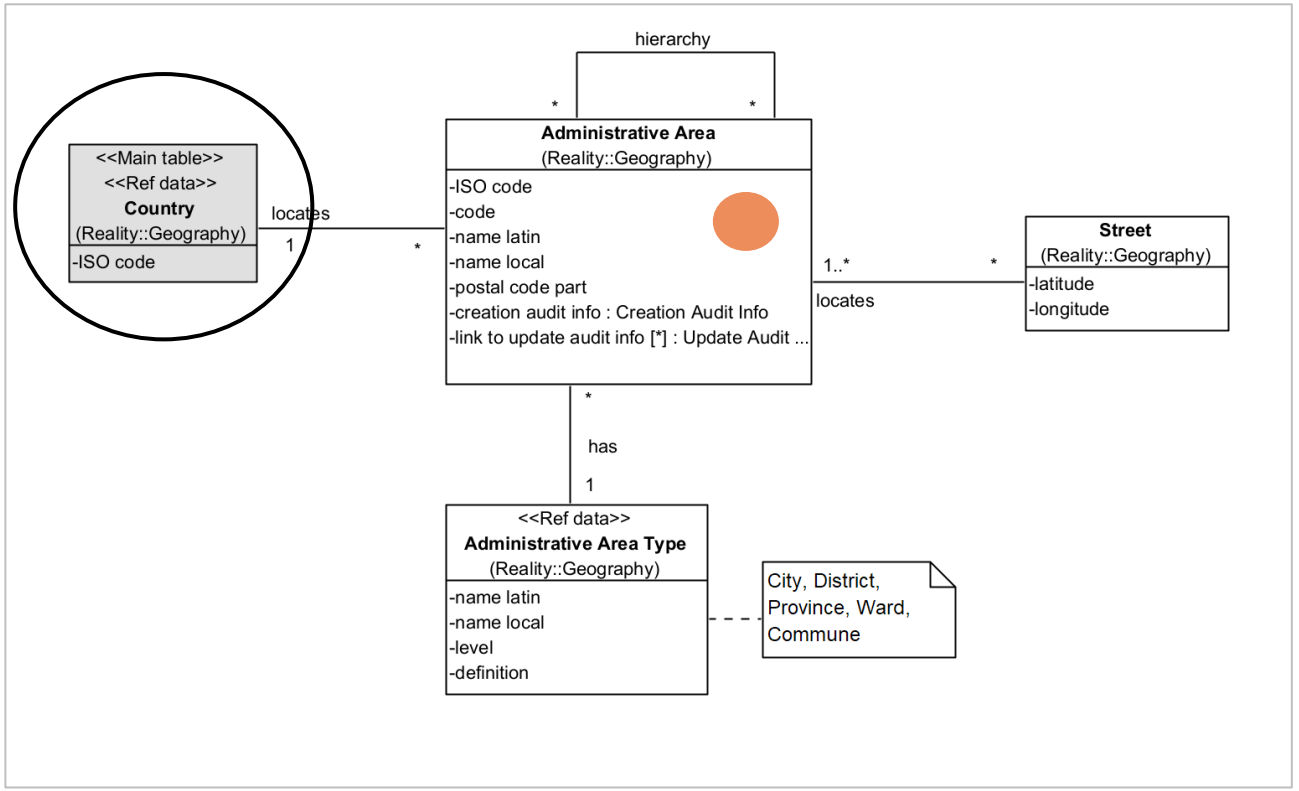
A CATEGORY (IN BOOCH'S UML) IS A CONCEPTUAL GROUPING OF CLASSES THAT SHARE A COMMON PURPOSE, USED TO PARTITION A SYSTEM'S MODEL INTO MEANINGFUL SUBSETS. IN MODERN UML, YOU WOULD NORMALLY USE A PACKAGE FOR THIS ROLE

EACH PACKAGE CONTAINS ONE AND ONLY ONE MAIN TABLE

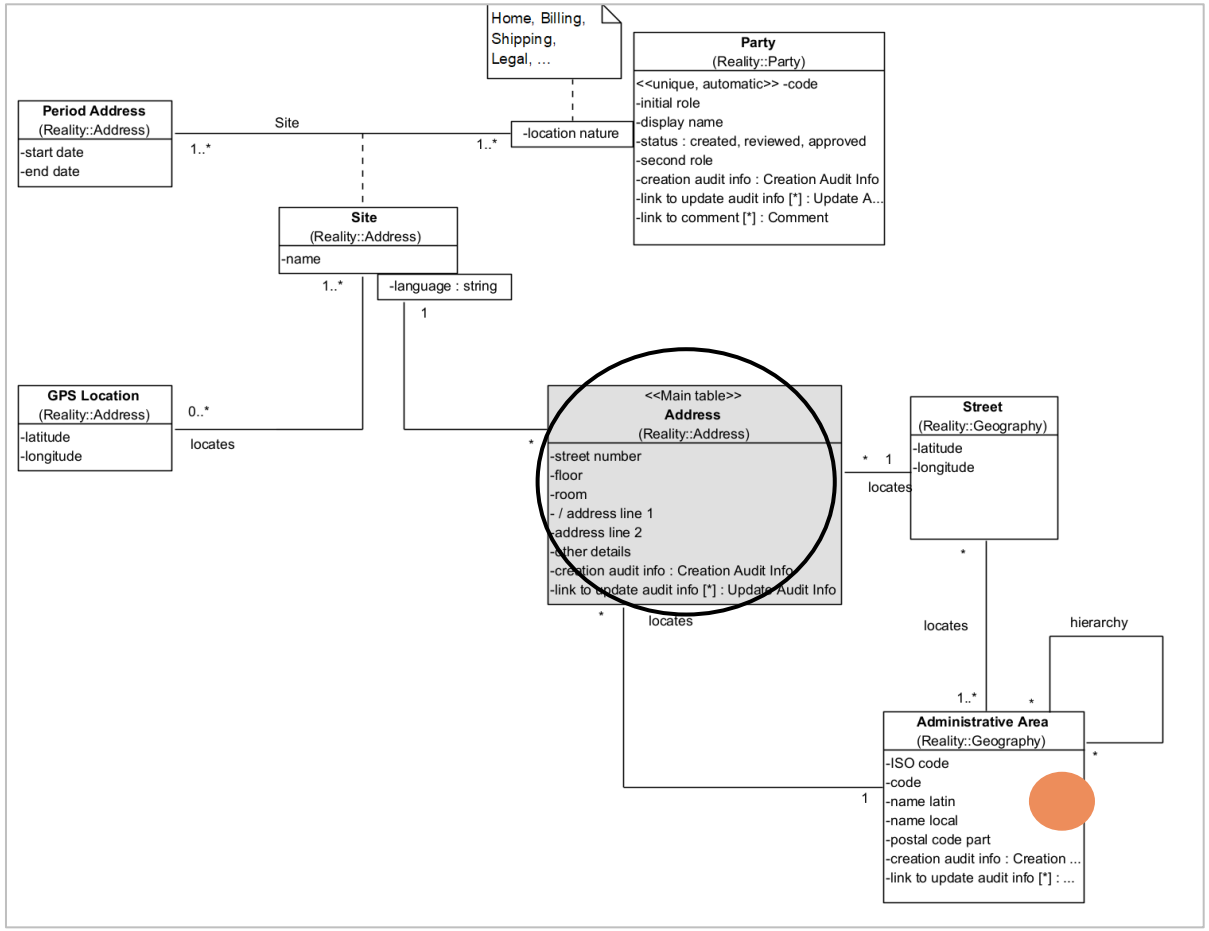


EACH PACKAGE CONTAINS ONE AND ONLY ONE MAIN TABLE

PACKAGE: GEOGRAPHY -> MAIN TABLE "COUNTRY"



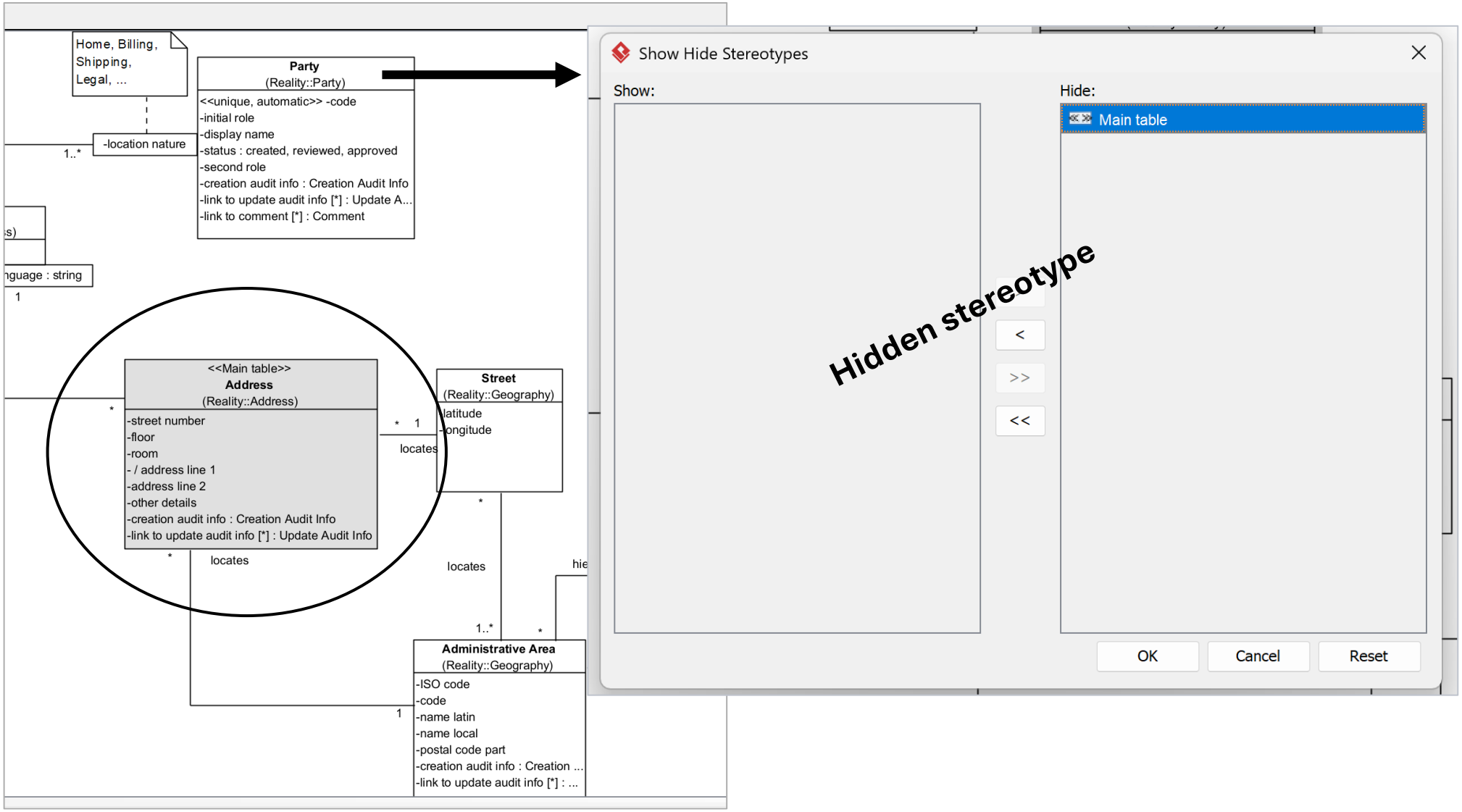
PACKAGE: ADDRESS -> MAIN TABLE "ADDRESS"



THE JUNCTION POINT BETWEEN THE TWO DATA MODELS

ENGAGE META

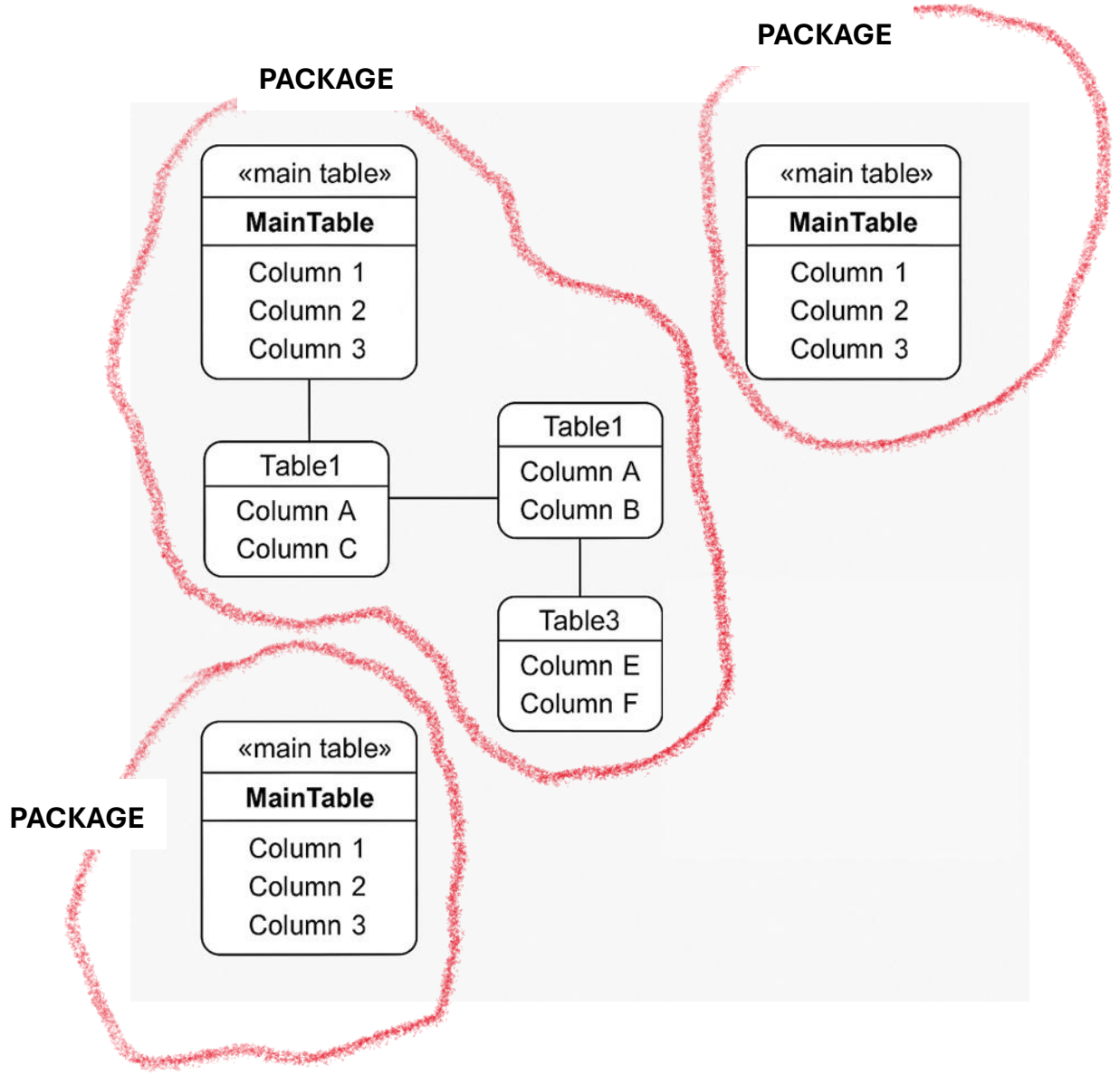
WHEN REUSING A MAIN TABLE FROM ANOTHER PACKAGE ITS STEREOTYPE IS NOT DISPLAYED



ENGAGE META

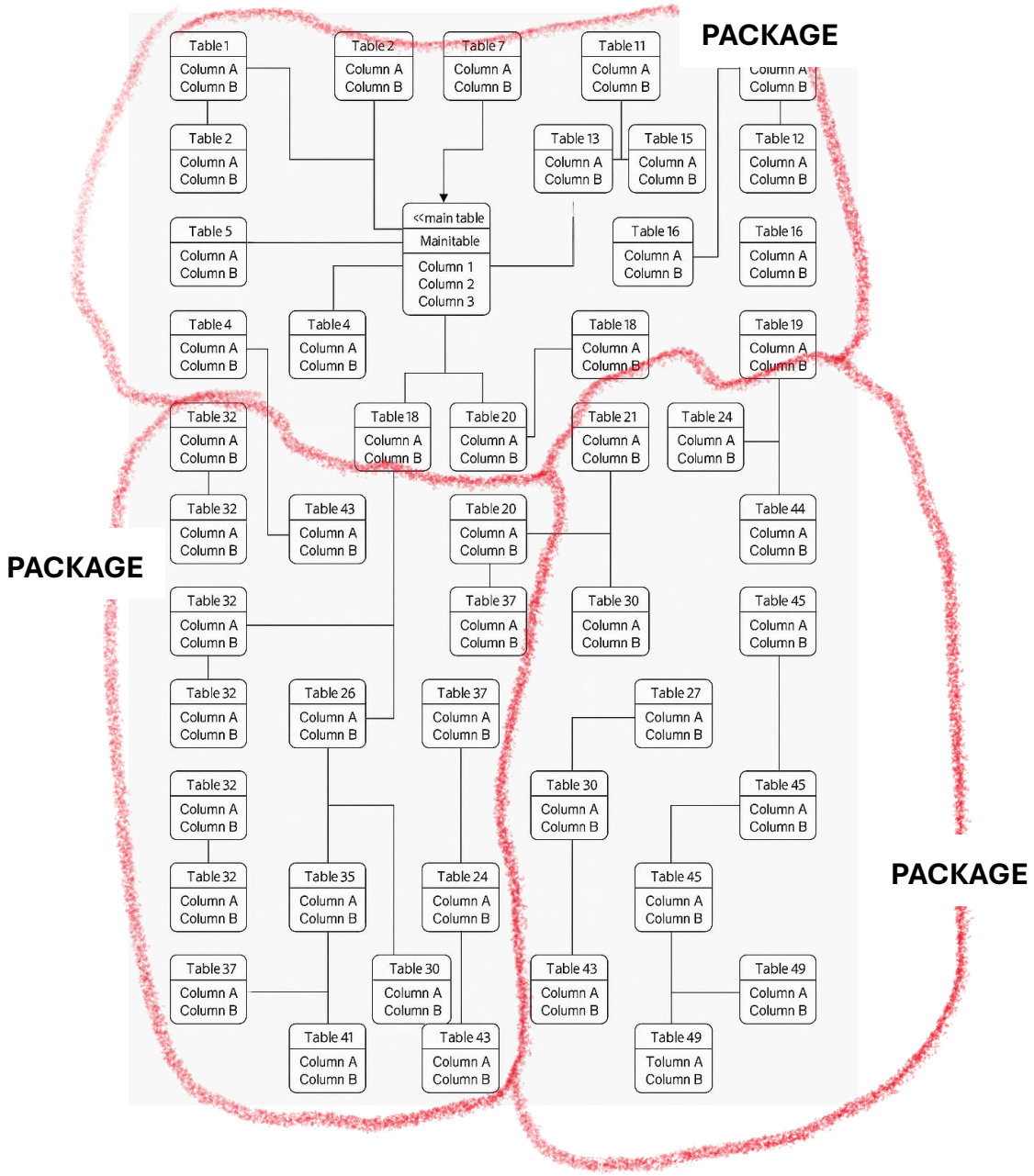
MAIN TABLE – BEST PRACTICES

WHEN A DATA MODEL CONTAINS TABLES THAT ARE ISOLATED FROM THE REST OF THE DESIGN, IT USUALLY INDICATES THE PRESENCE OF SEVERAL BUSINESS CONCEPTS, AND THEREFORE MULTIPLE MAIN TABLES



MAIN TABLE – BEST PRACTICES

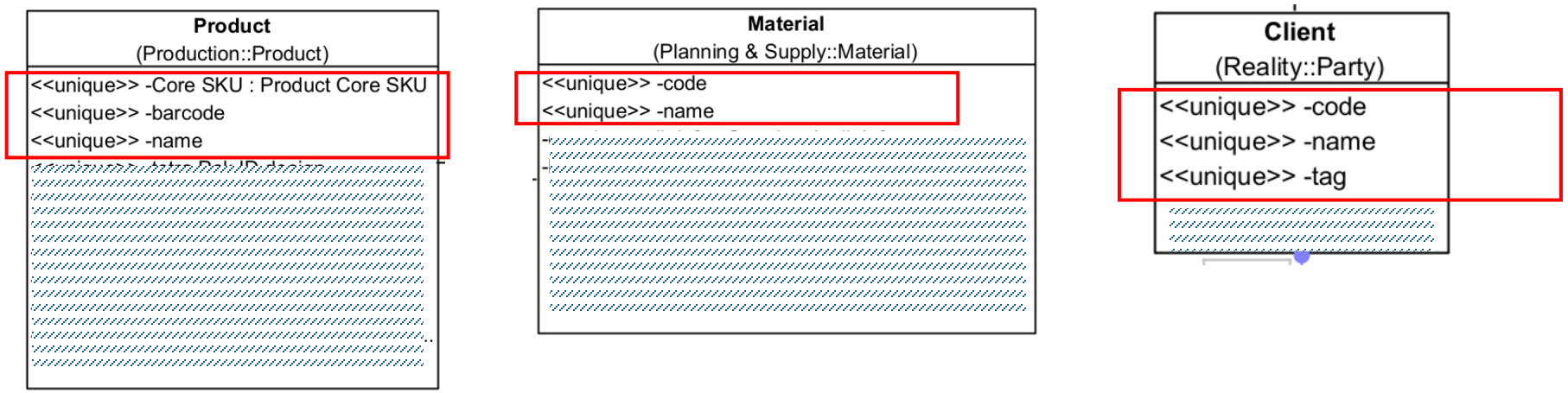
WHEN A DATA MODEL HAS TOO MANY TABLES (MORE THAN 20), IT SHOULD BE DIVIDED INTO PACKAGES, AS THIS USUALLY INDICATES THE PRESENCE OF SEVERAL IMPORTANT BUSINESS CONCEPTS, EACH CORRESPONDING TO A MAIN TABLE



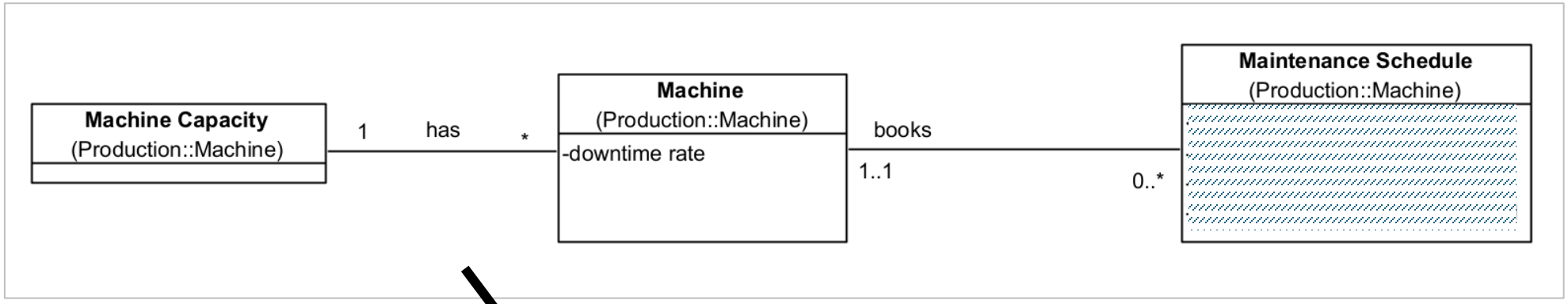
Business Attributes

ENSURES THAT THE MODEL COMMUNICATES CLEAR SEMANTICS TO BUSINESS STAKEHOLDERS WHILE AVOIDING UNNECESSARY TECHNICAL COMPLEXITY

- Always represent the business code (e.g., SKU, customer code, contract number)
- Do not include technical identifiers such as database IDs, UUIDs, or system-generated keys
- Use a tag attribute as a mnemonic whenever a long business name needs to be shortened. This is especially useful for product labels, client B2B identifiers, or any entity that requires a concise reference



Naming Convention & Business Glossary



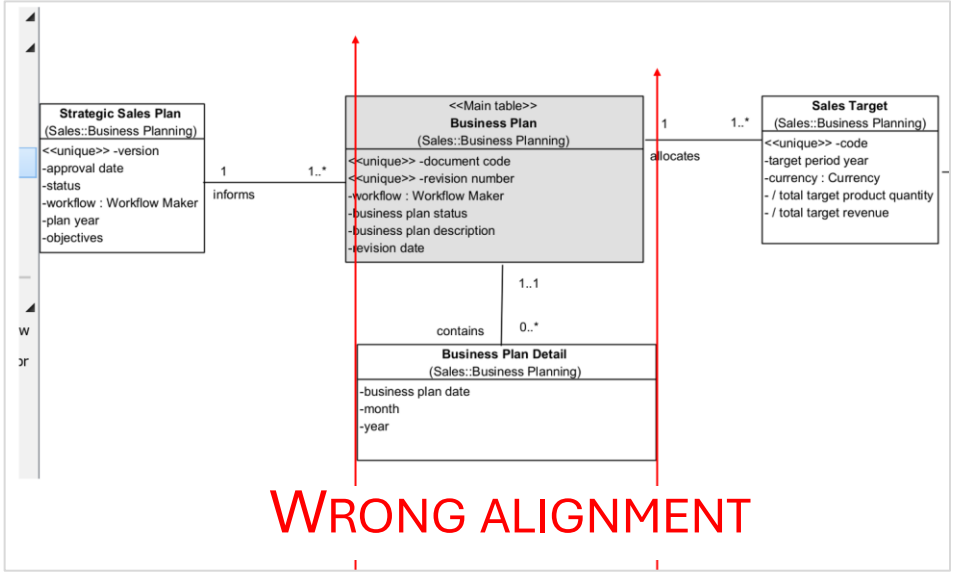
The names of the tables, key attributes, and relationships are defined in the business glossary, which forms the foundation of the ontology

Business Glossary

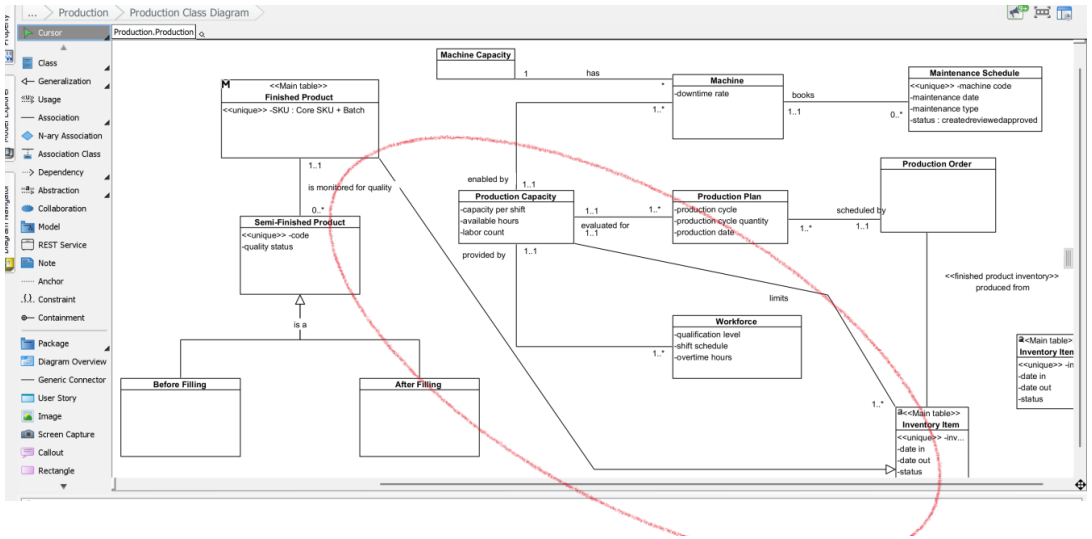
#	Business Term	Category *To add manually*	Definition	
67	[redacted]			Materia
68	Machine	Table	[redacted]	Produ
69	Machine Capacity	Table	Machine Capacity is the maximum production or output rate that a filling, packaging, or processing line at [redacted] can achieve within a given time period under standard operating conditions. It defines the upper technical and operational limit of production lines and is used to ensure that Order-Based Production Plans do not exceed feasible manufacturing capability.	
70	Maintenance Schedule	Table	Timetable for preventive maintenance	

Spatial Layout

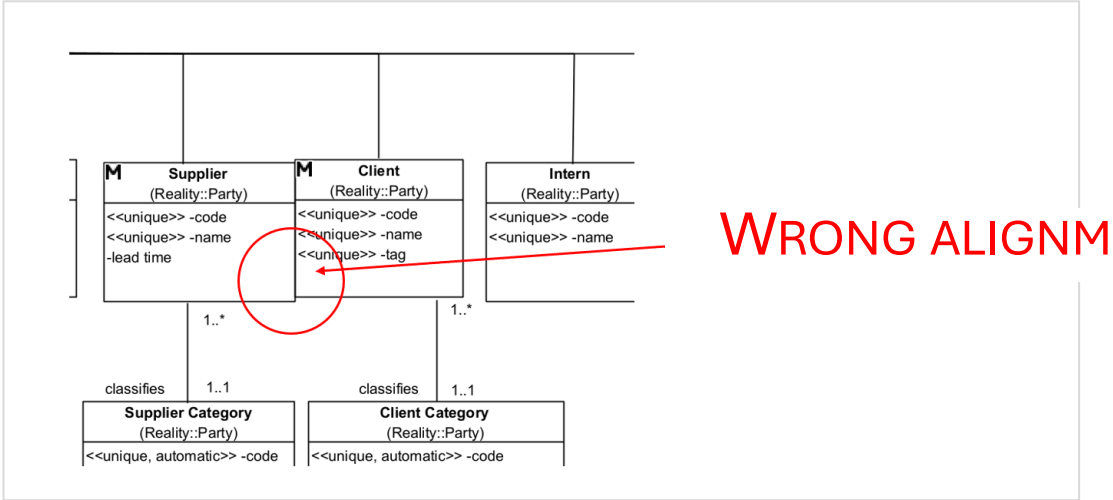
- Center the diagram in the window by default
- Minimize crossing lines
- Avoid forcing scrolling



Wrong alignment



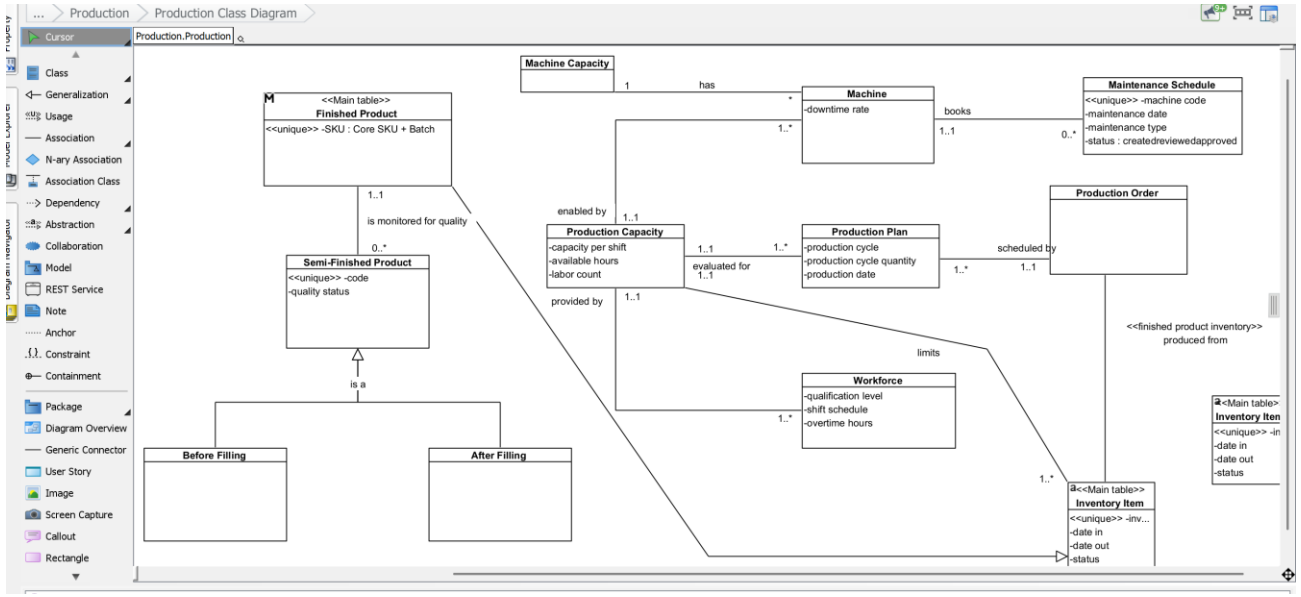
Crossing Lines

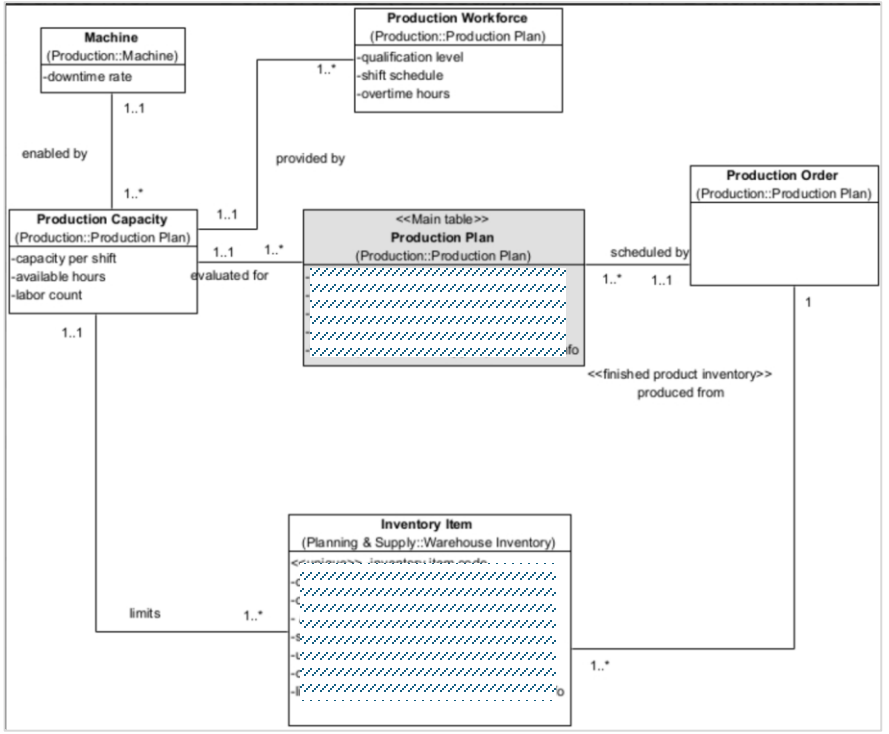


Wrong alignment

ENGAGE META

Spatial Layout

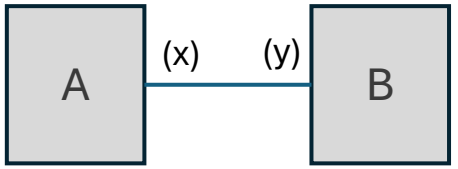


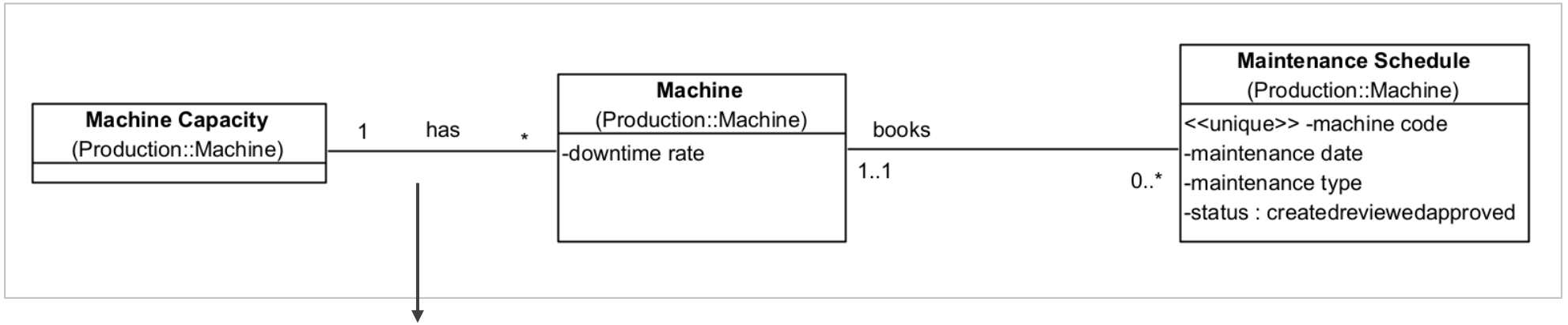


Reorganization of the presentation of tables, relationships, and packages

Cardinality Reading

- When you model associations A – B
  - The multiplicity shown near Class A (x) indicates how many instances of A can be linked to one instance of B
  - **Always read multiplicity from the opposite side**





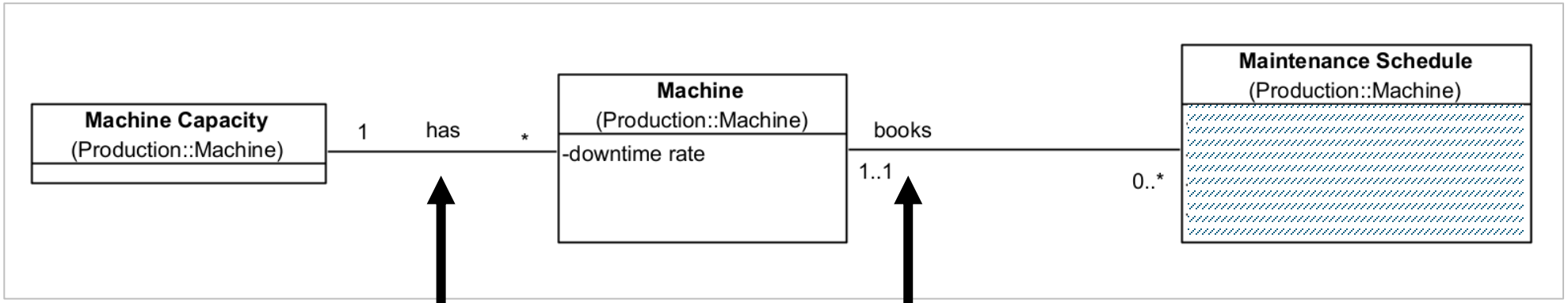
One “Machine Capacity” can be used by zero to many “Machines”

One “Machine” is associated with one and only one “Machine Capacity”

This reading rule is defined by OMG UML (2.x / 3.x specifications)

Reading Of Association

Standard UML representation: formal presence of cardinalities expressed as an (min, max) pair



The name of the association is either placed in the middle of the connection: Two possible directions of reading...

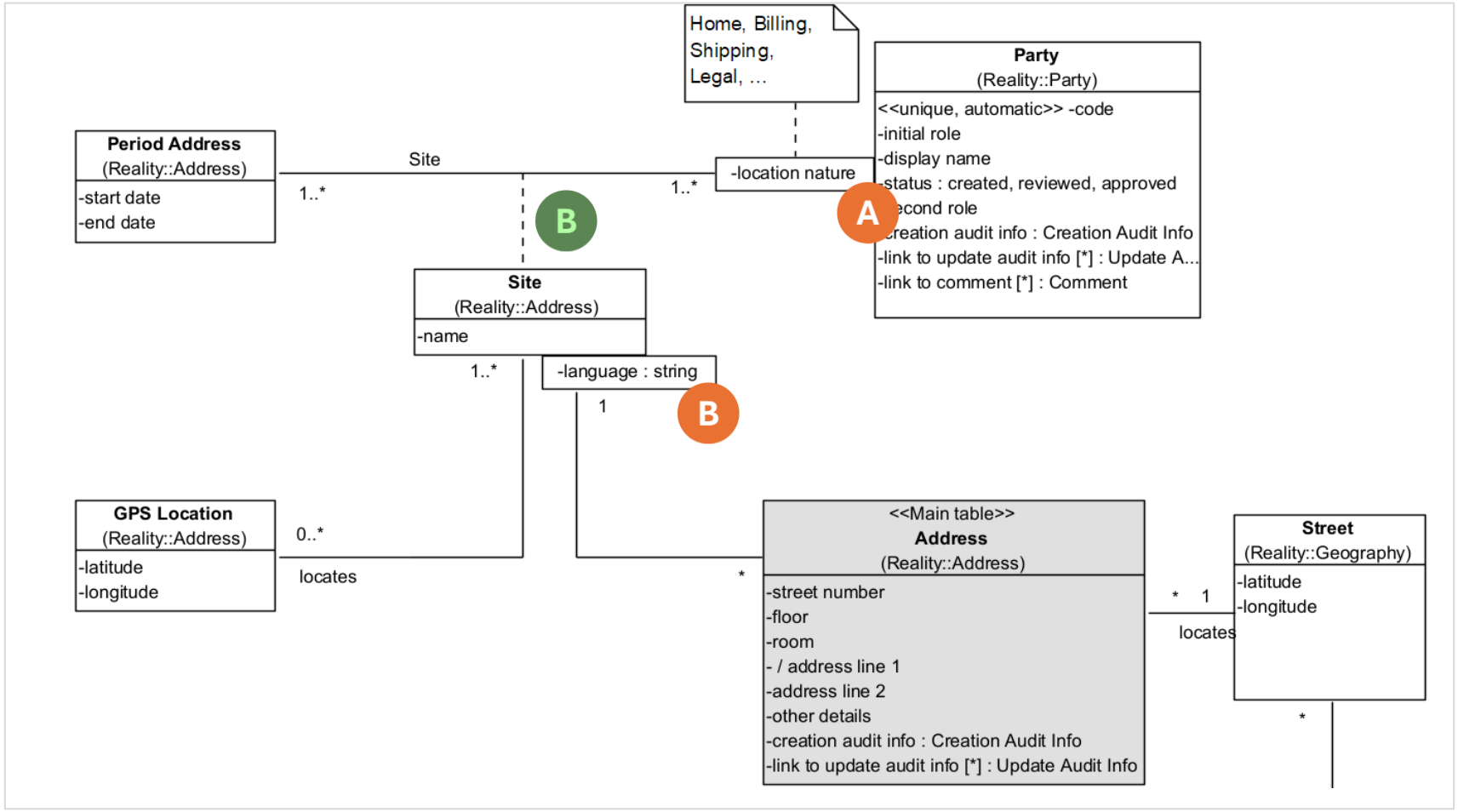
...or at the edge of the table where the reading begins. For example, in this case it should be read as “Machine” books “Maintenance Schedule

Types Of Associations

- Qualifier Attribute
- Associative Class
- Ternary Relation

Qualifier Attribute + Associative Class

Qualifier attribute A Associative class B



Party: ABC Company

Site: French Headquarters

Location nature = Legal Address

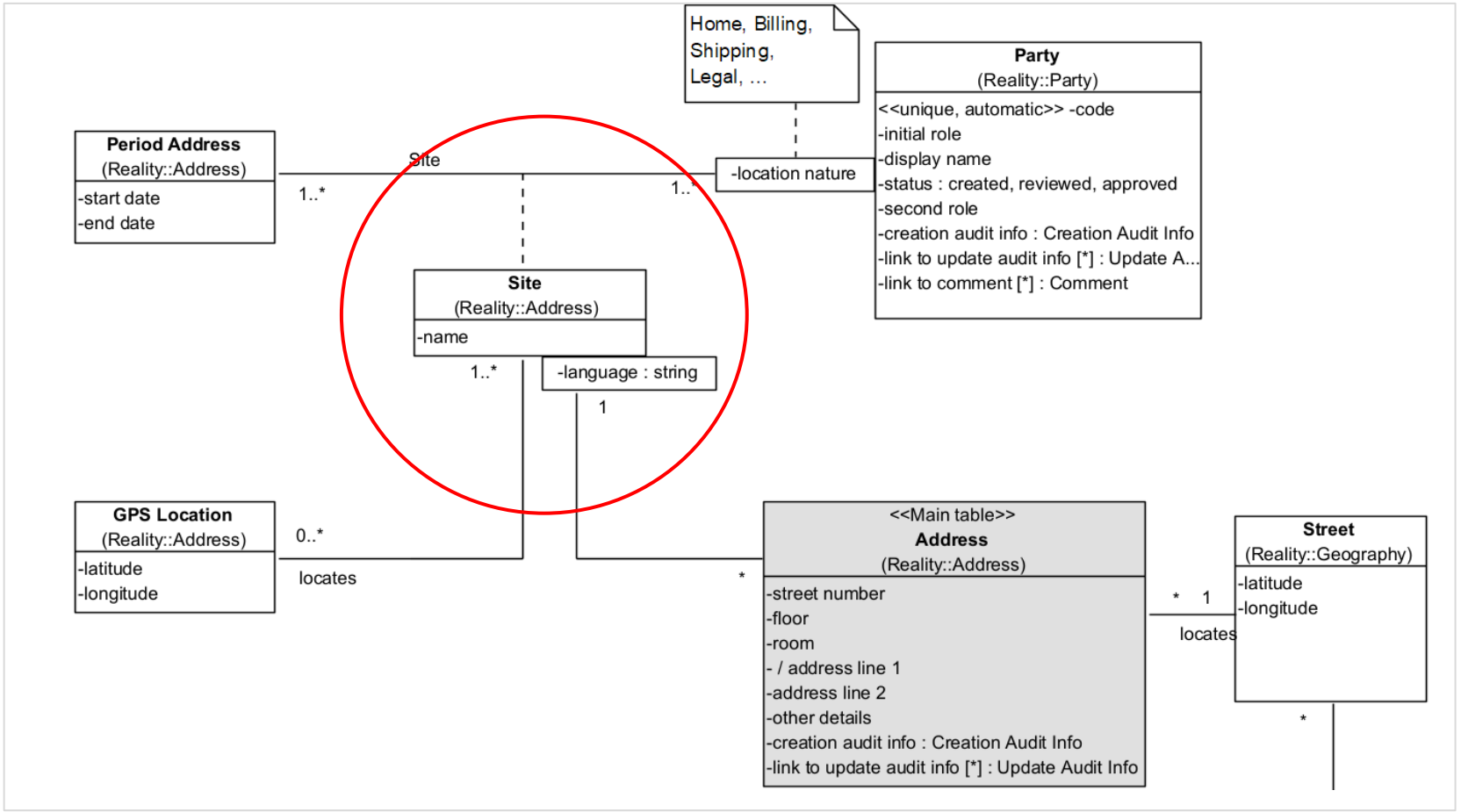
Address Language = English

Address Language = Japanese

Explanation of the business concept of Site

Explanation of the business concept of Site

- A **Site** is not just a postal address but a named place with a business meaning
- It groups one or multiple addresses and provides a semantic/business label

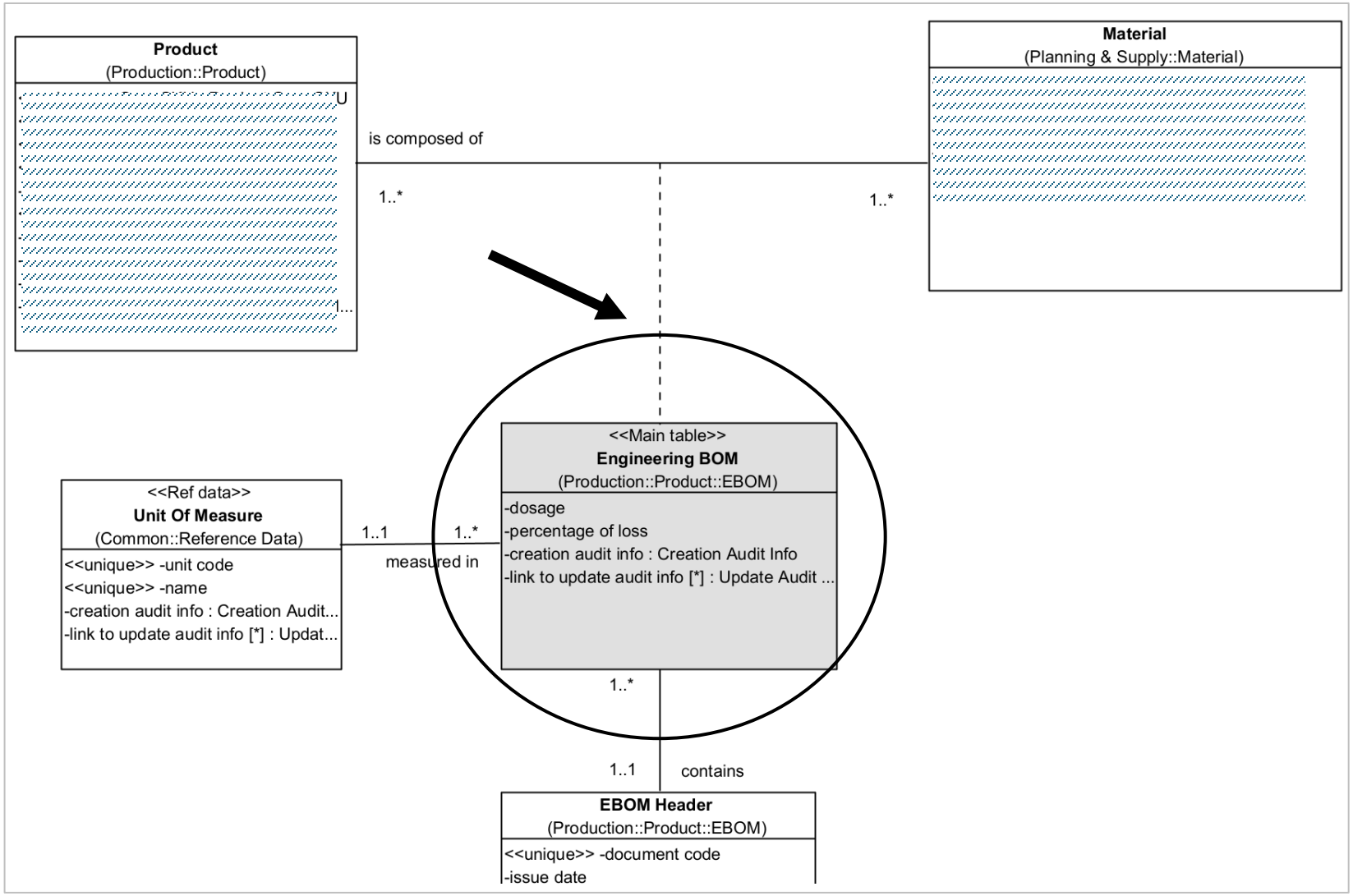


- **Party**
  - An actor (legal entity or person) that can play roles (customer, supplier, employee, etc.)
  - It has rights, obligations, and contracts
  - Examples: Company DYD, Jonh Kramer
  - **Party = who (actor with legal or role-playing capacity)**

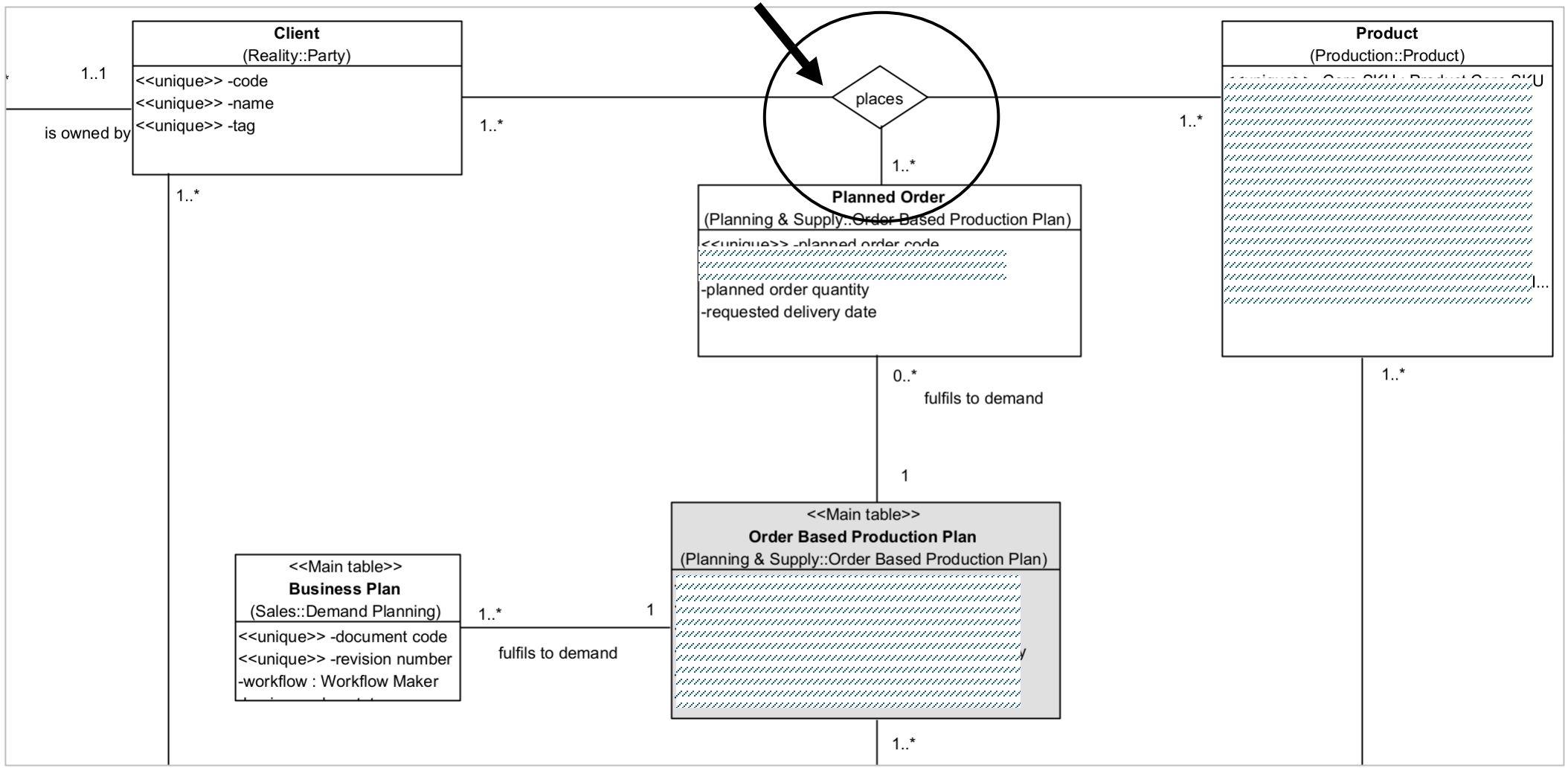
- **Site**
  - A location owned or used by a Party
  - It has no legal autonomy on its own
  - It exists only as a place of activity
  - Examples: Company DYD Plant #2 in Ba Vi, Company DODO Training Center in Paris
  - **Site = where (location of that actor's activity)**

ASSOCIATIVE CLASS

EACH RECORD IN THE ASSOCIATIVE CLASS CORRESPONDS TO **ONE UNIQUE PAIR OF RECORDS FROM THE TWO RELATED CLASSES**

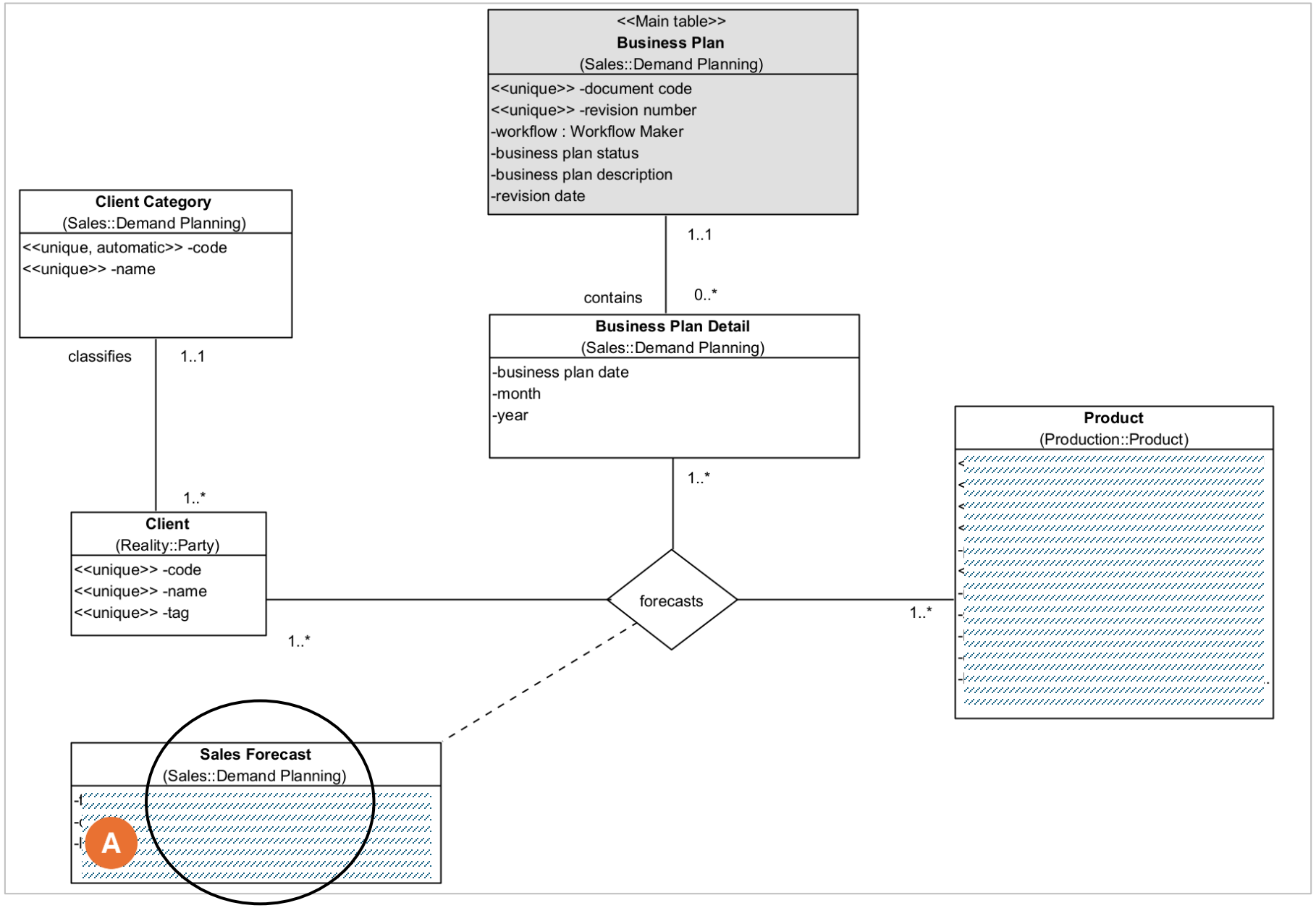


TERNARY ASSOCIATION



Ternary Association + Associative Class

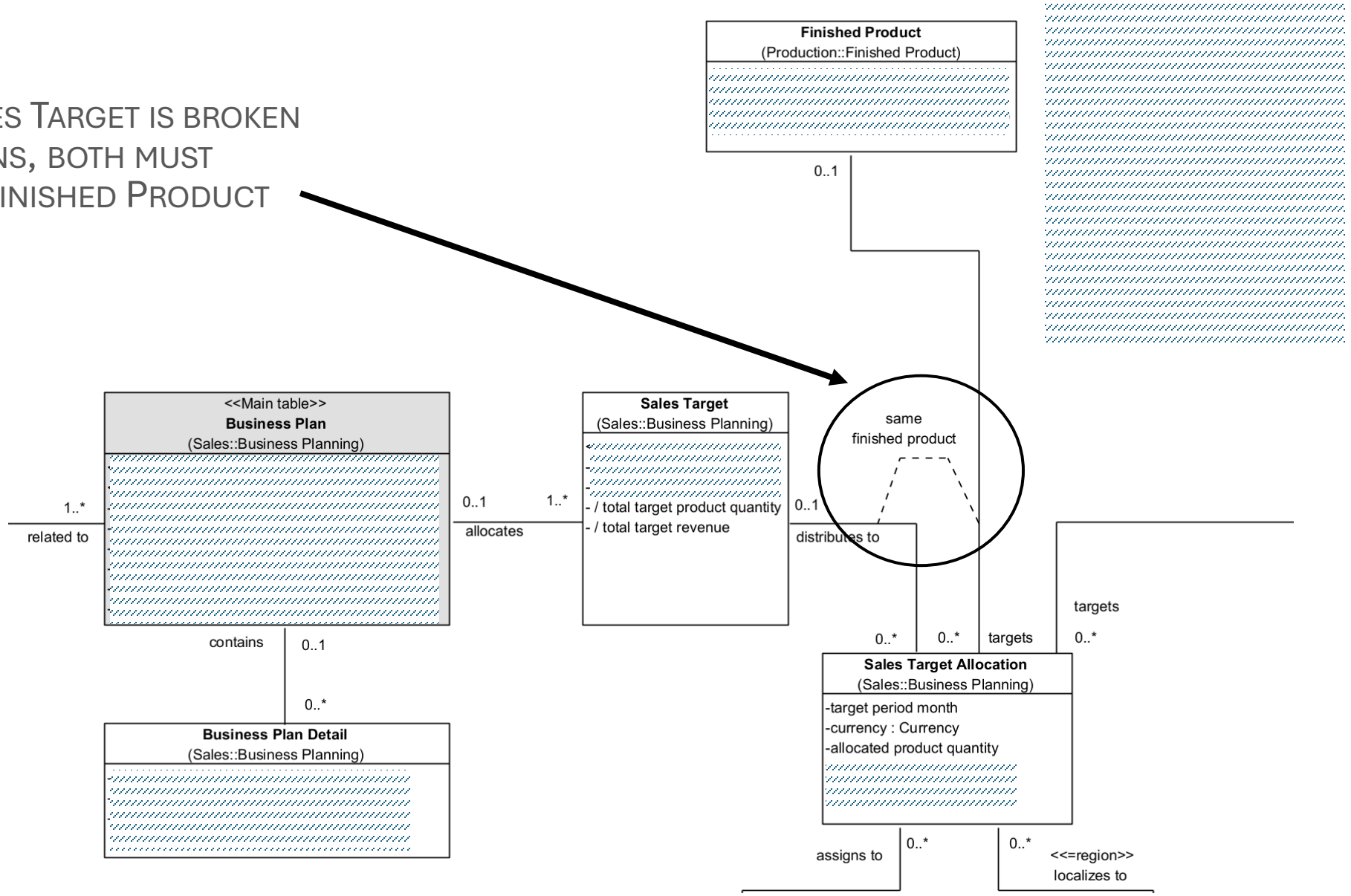
- Attributes are cleanly separated in their own class
- Association class can be referenced by other relationships
- Can be reused across multiple triples (different Business Plan Detail × Client × Product combinations)



ENGAGE META

Referential Integrity Rule

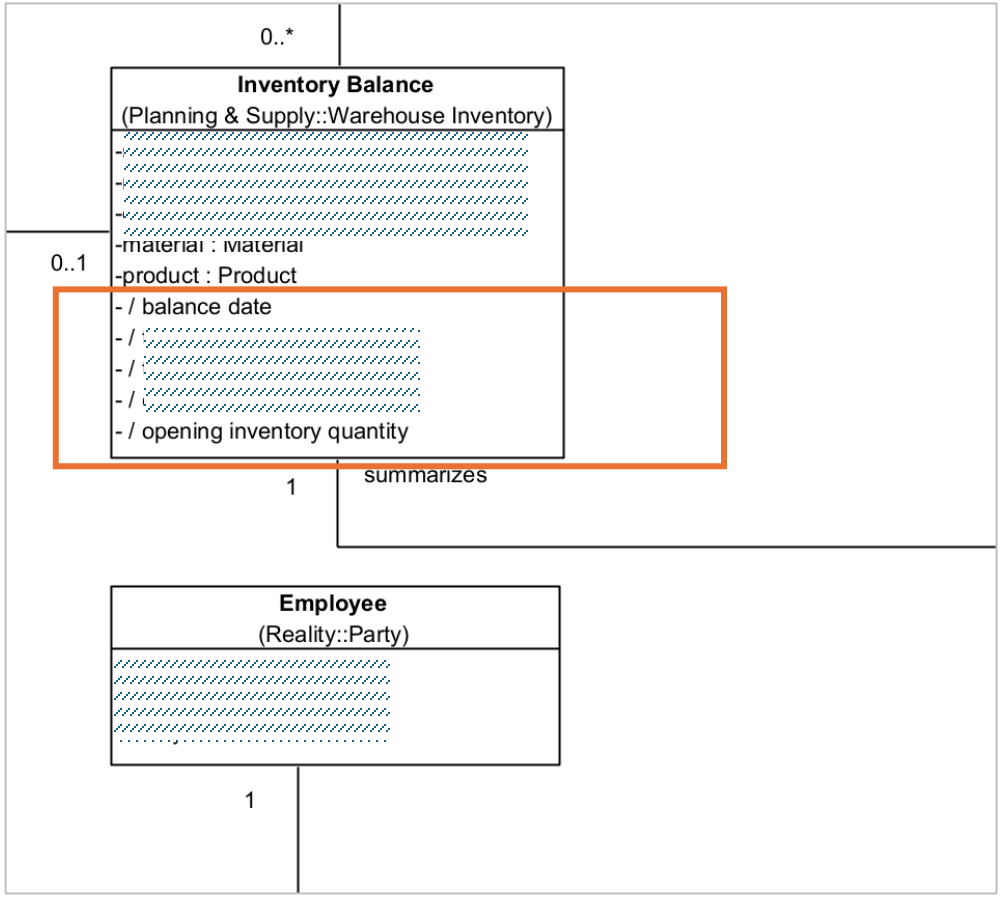
- Example: when a Sales Target is broken down into allocations, both must reference the same Finished Product



ENGAGE META

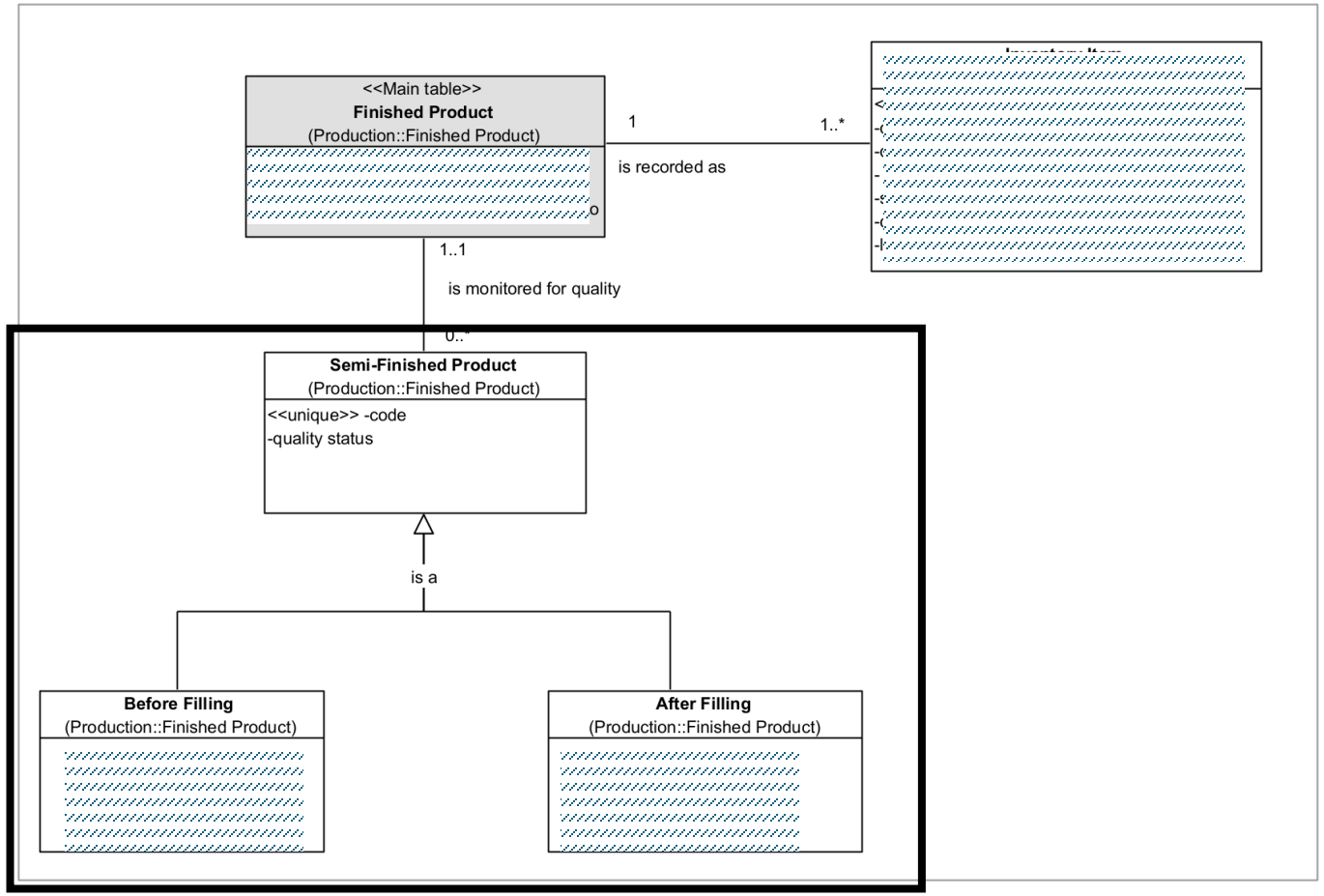
Derived attribute is calculated by the system

If a table consists only of "derived" attributes, then it relates solely to a consolidation or reporting need

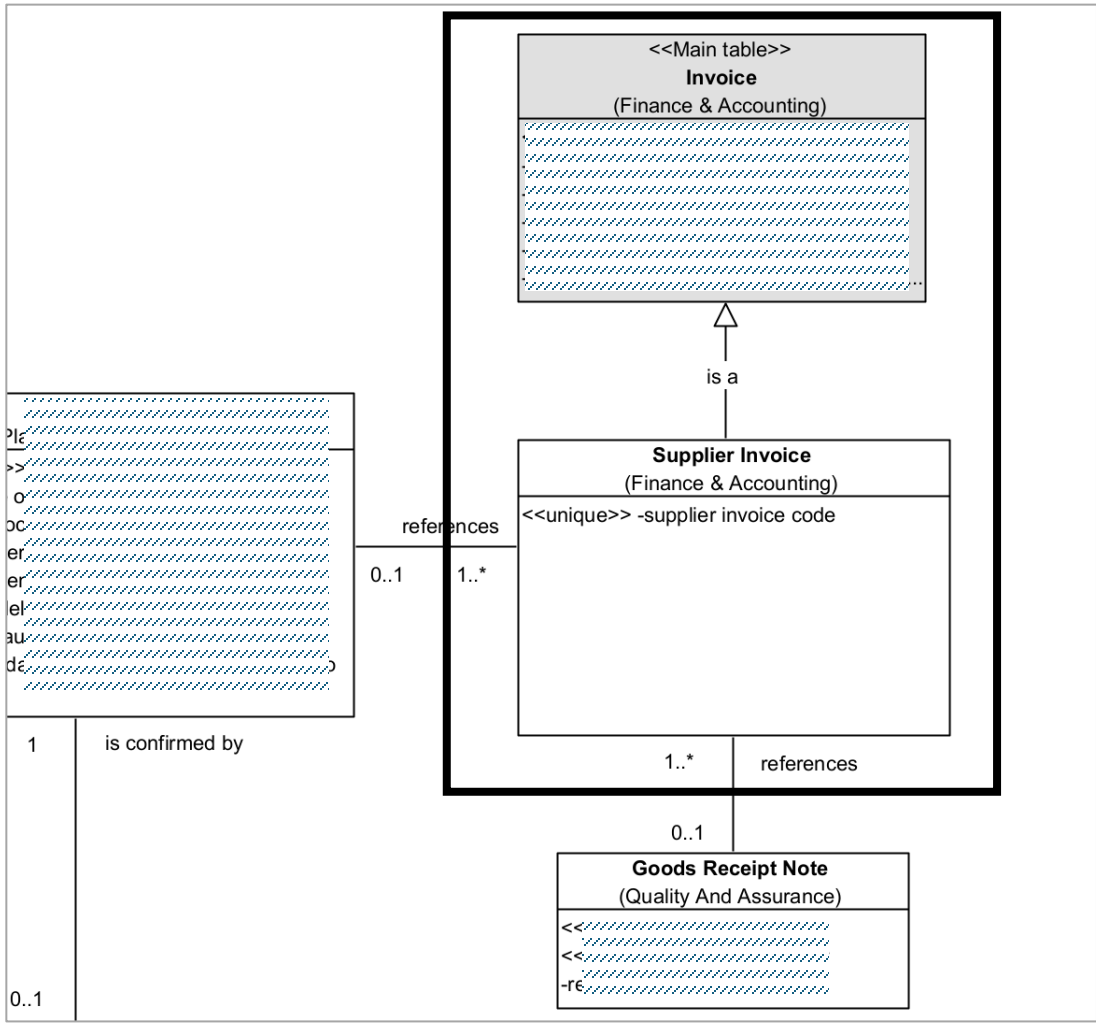


GENERICITY – INHERITANCE

TO COVER AN EXISTING NEED

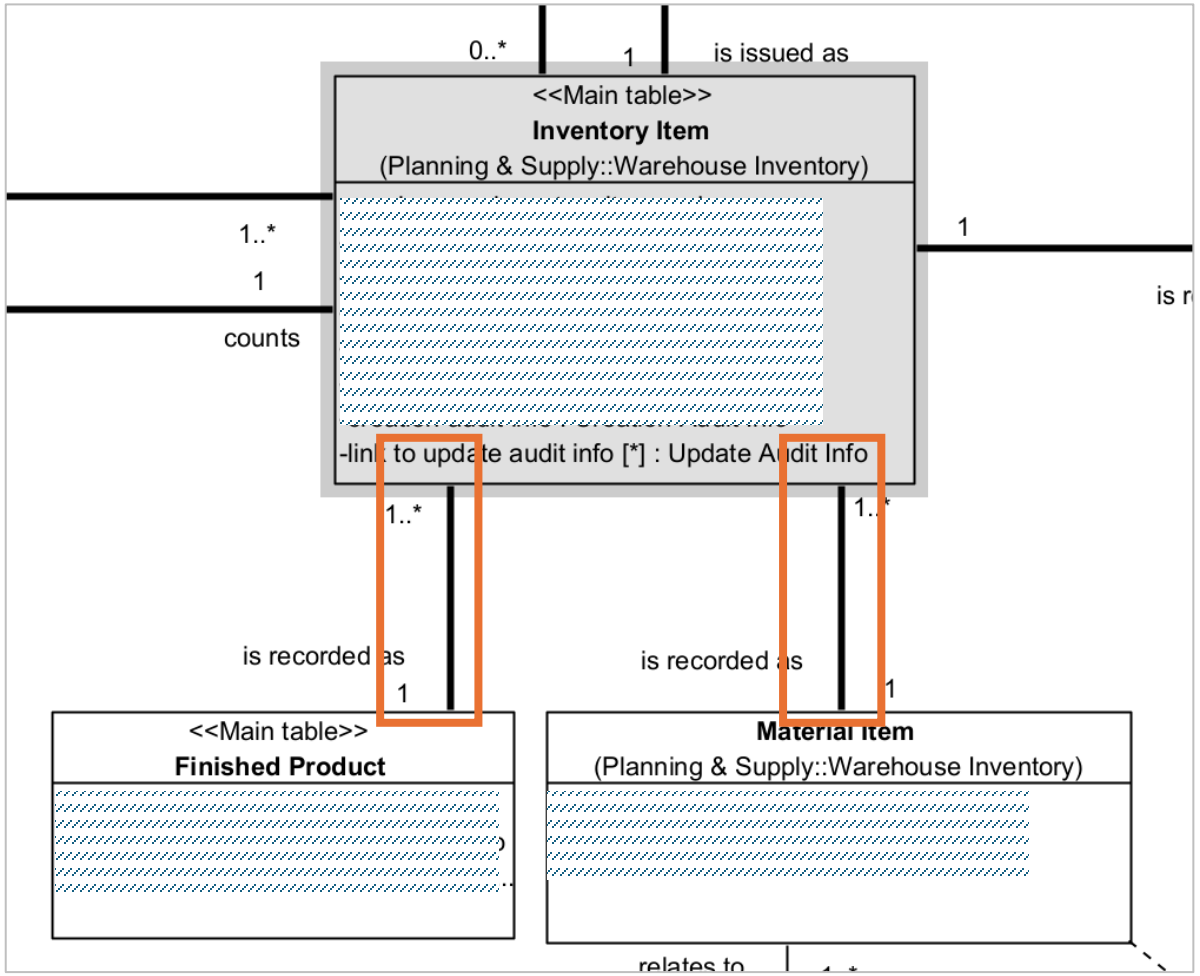


TO ANTICIPATE A FUTURE REQUIREMENT



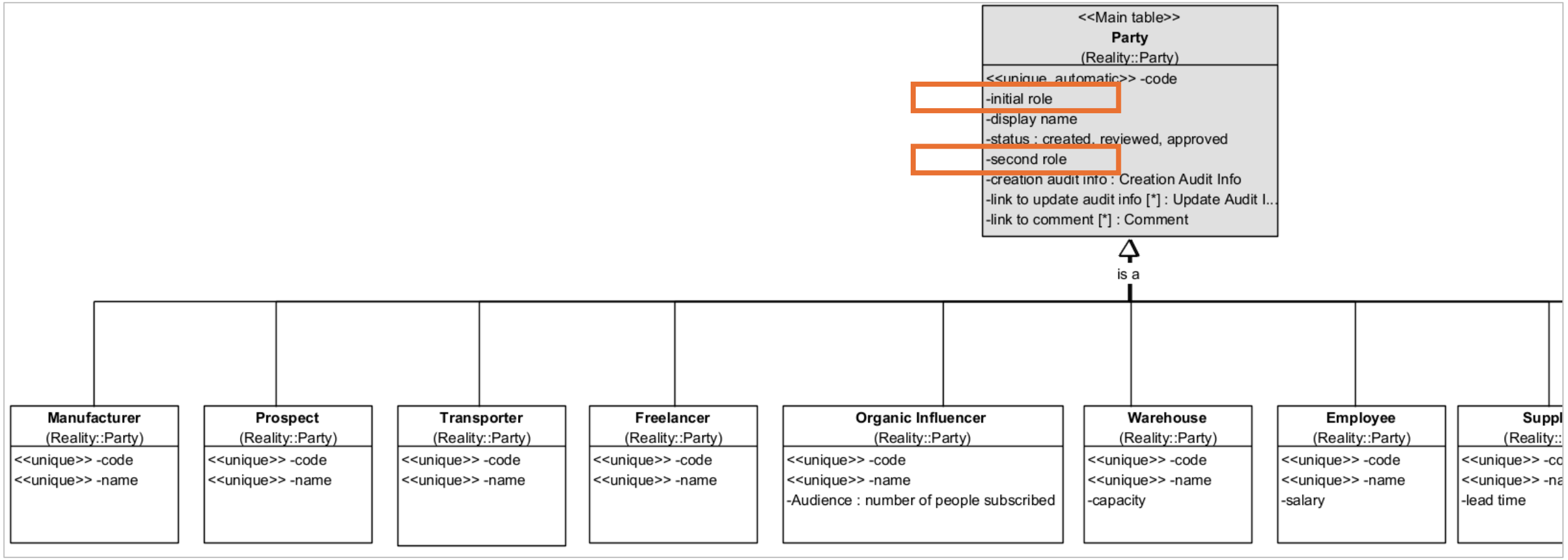
GENERICITY – INHERITANCE

USE INHERITANCE ONLY WHEN THE PARENT–CHILD LINK IS UNIQUE OTHERWISE PREFER AN N-N RELATIONSHIP



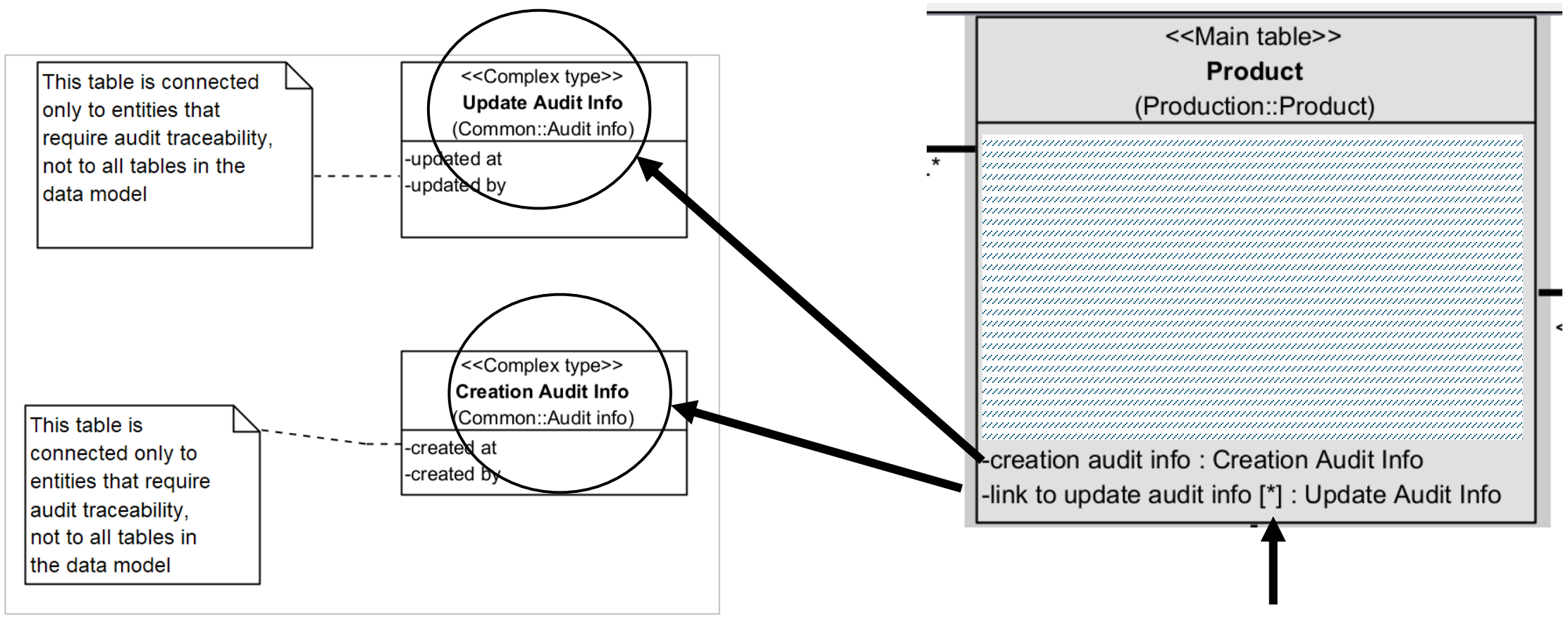
GENERICITY – INHERITANCE

MULTIPLE INHERITANCE CAN BE ENFORCED BECAUSE THE SPECIALIZED TABLES CONTAIN SPECIFIC ATTRIBUTES

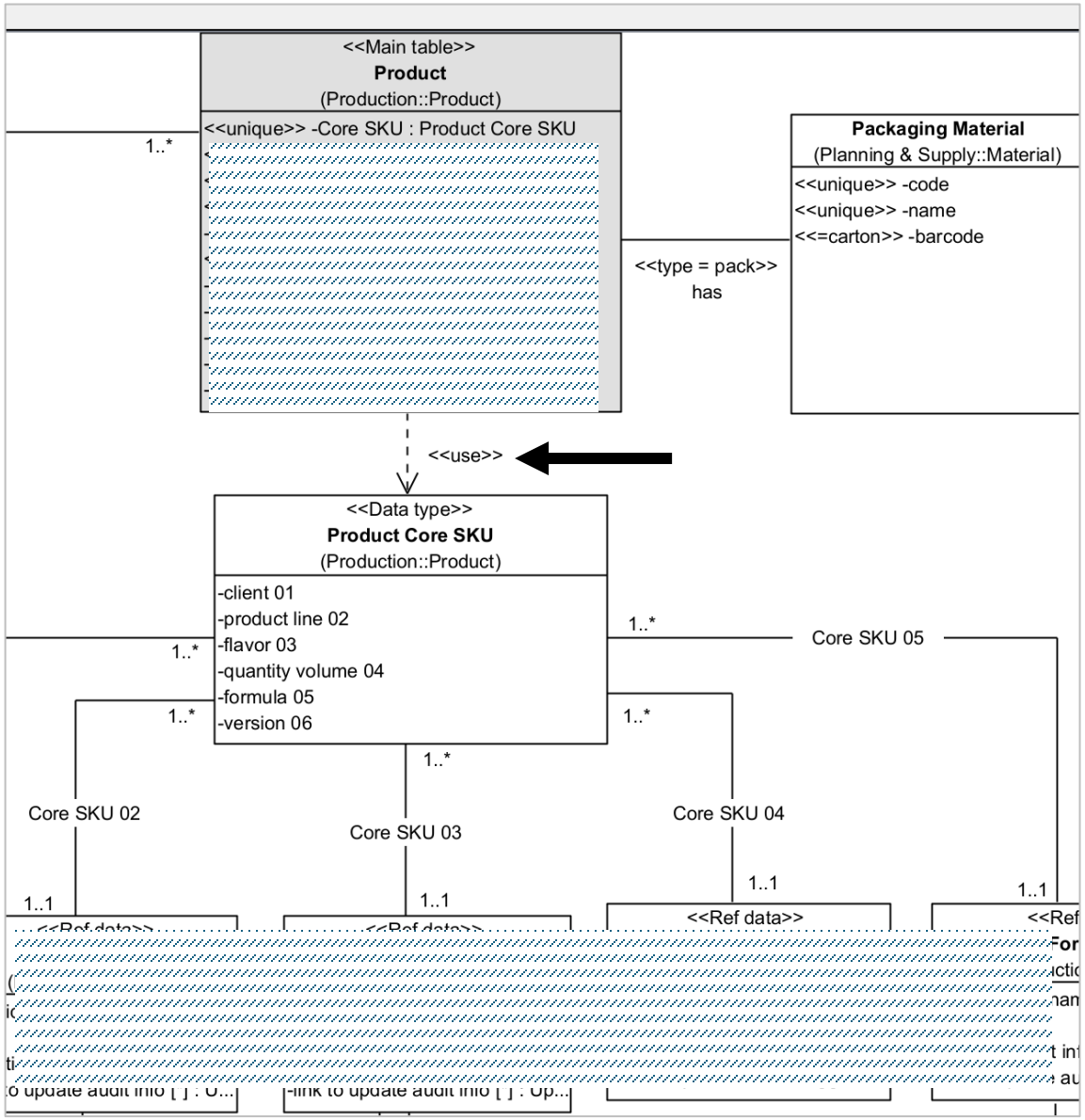


Complex Data Type

Complex Type is used to group multiple related attributes into a reusable structure



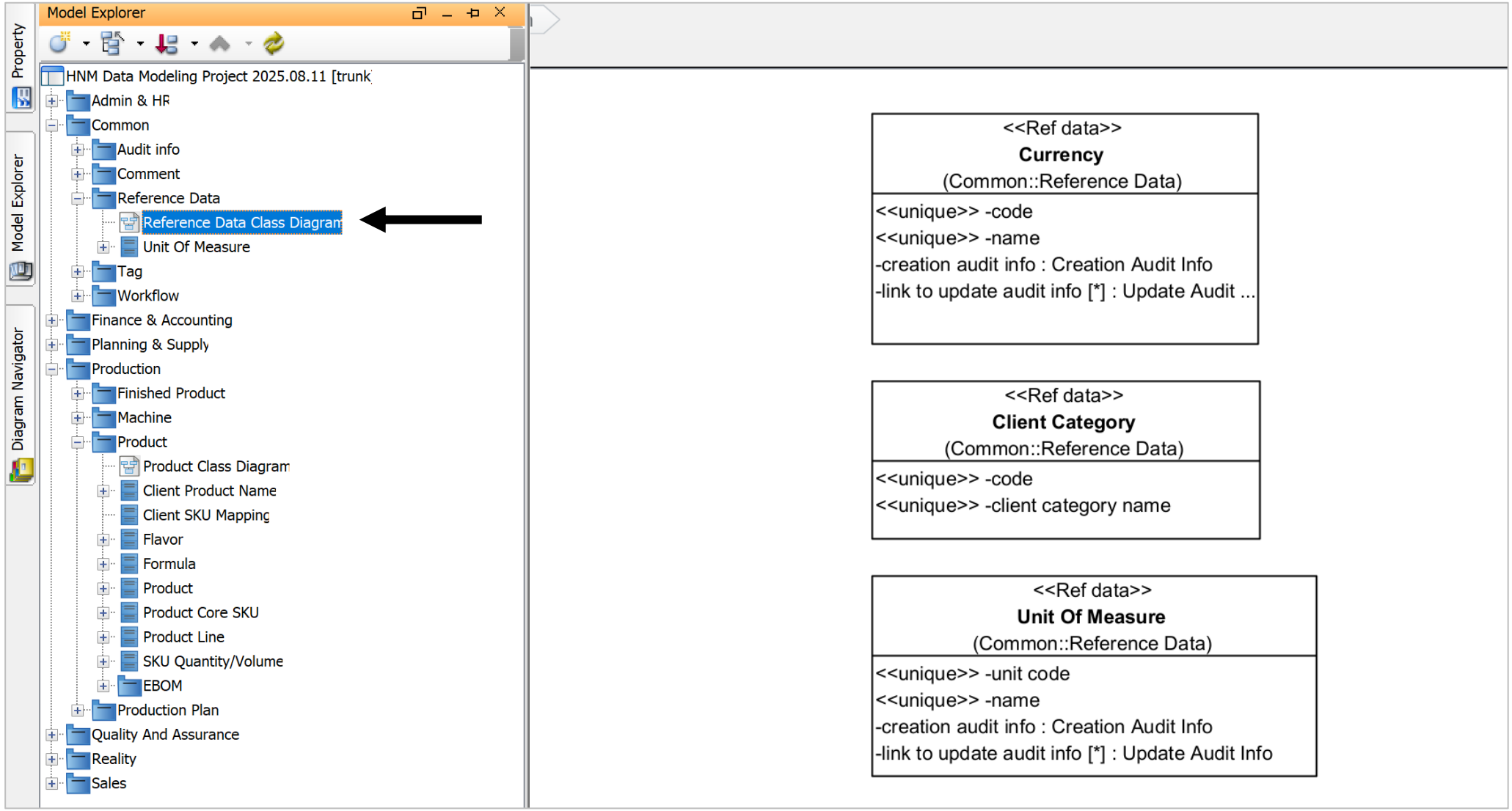
Complex Data Type



Show Complex Type usage with a "uses" dependency when emphasizing model readability

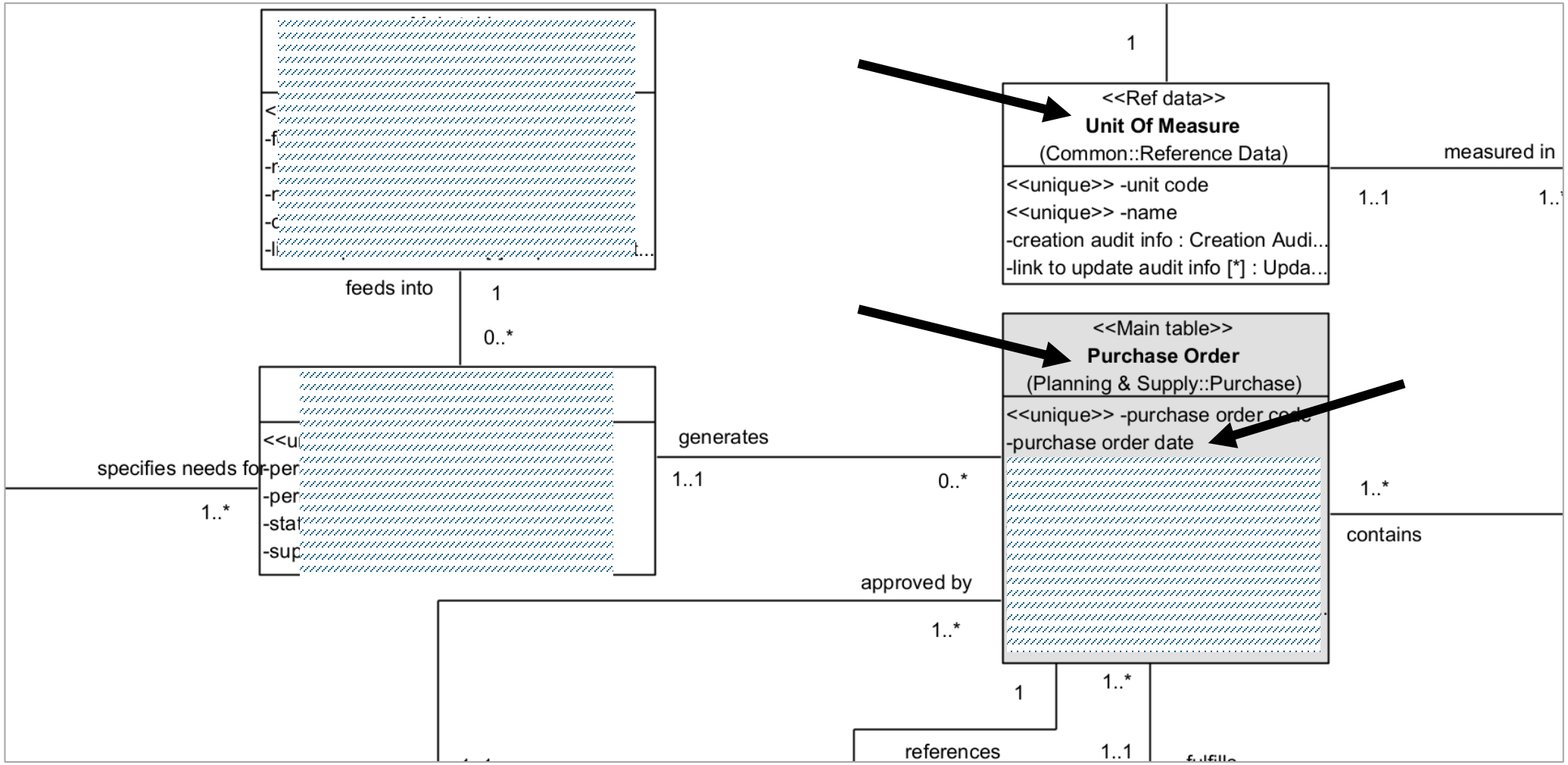
Common Reference Data

Group general reference tables in a dedicated package



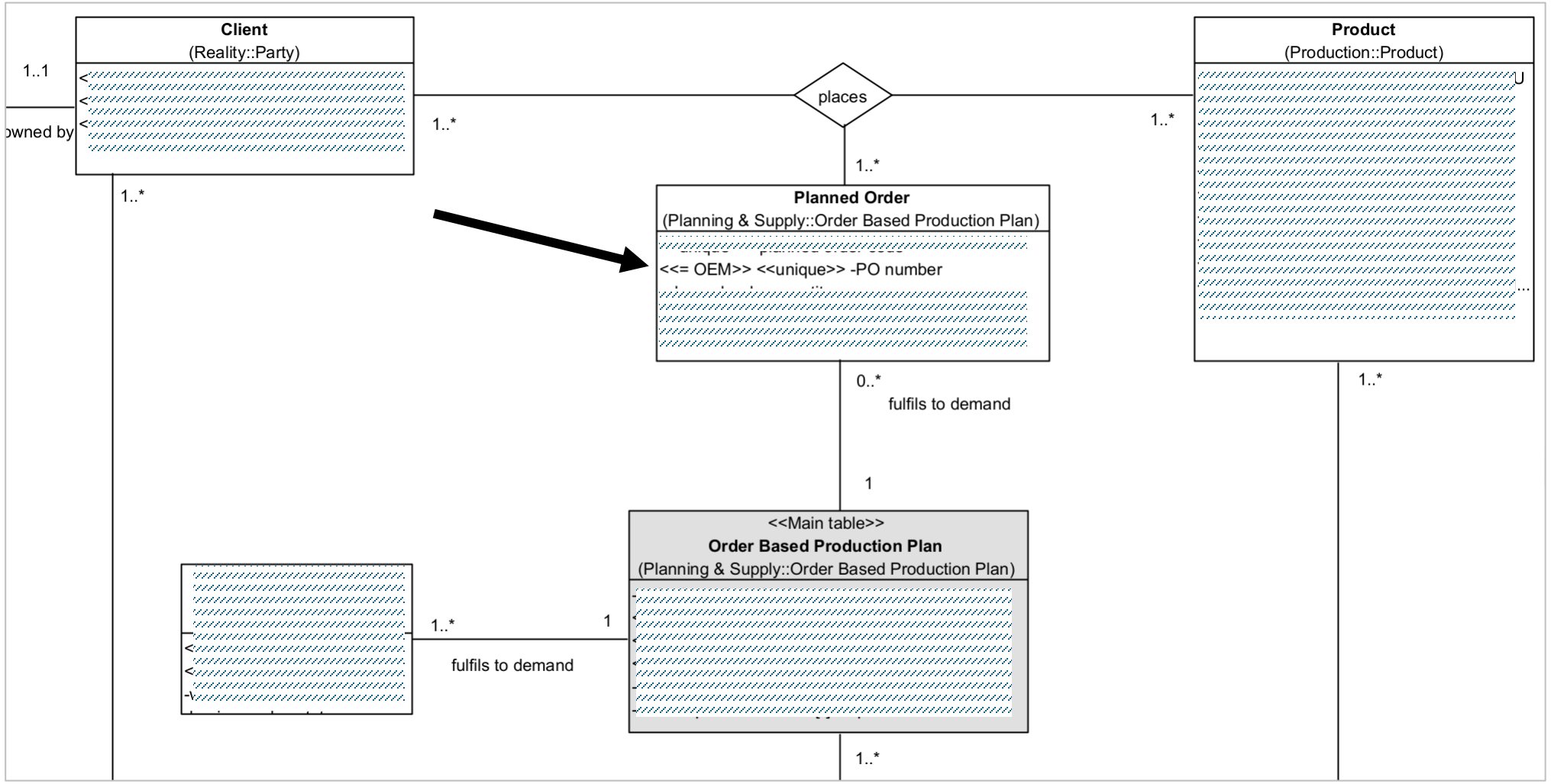
UNAMBIGUOUS NAMING

THE NAME OF EACH TABLE AND EACH ATTRIBUTE MUST FULLY CONVEY ITS SEMANTICS TO AVOID ANY AMBIGUITY



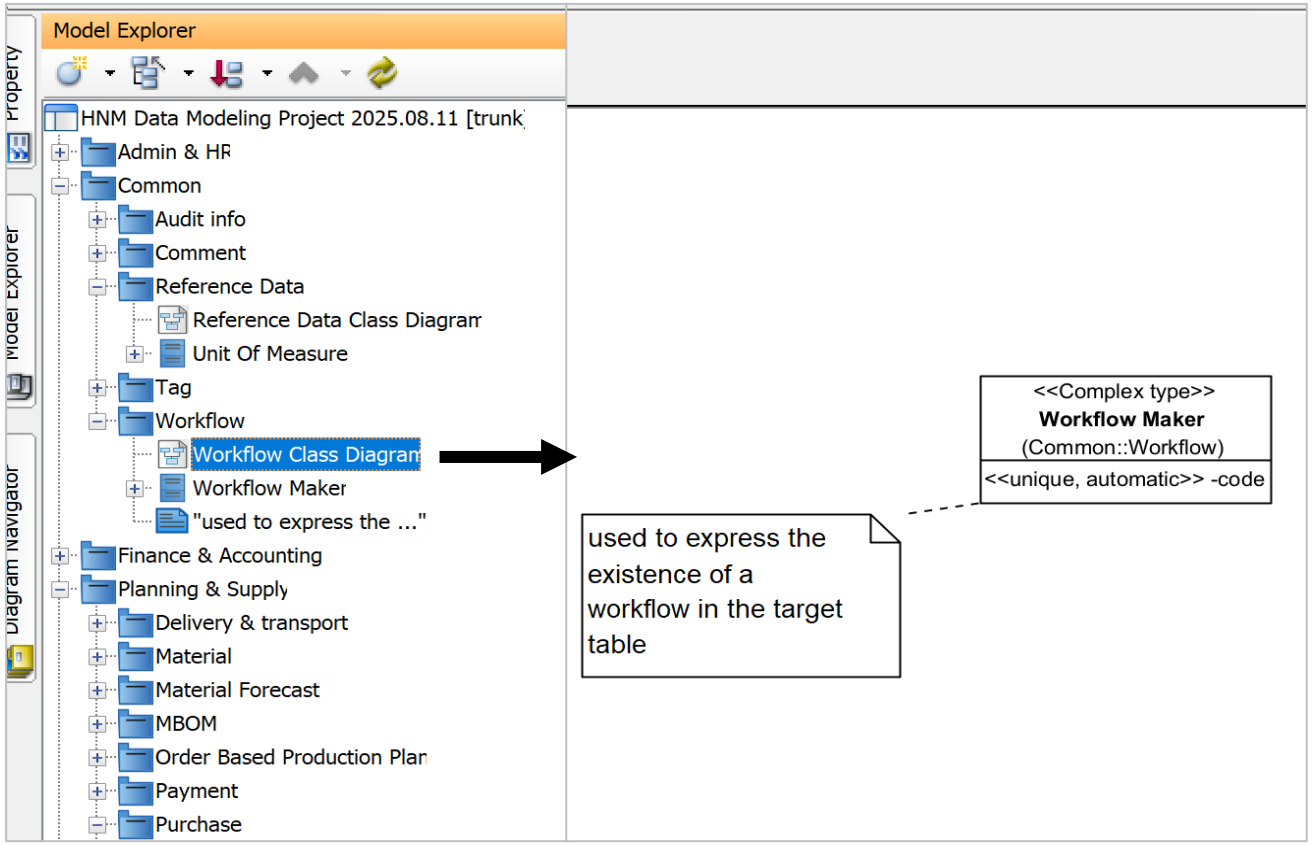
Unambiguous Naming

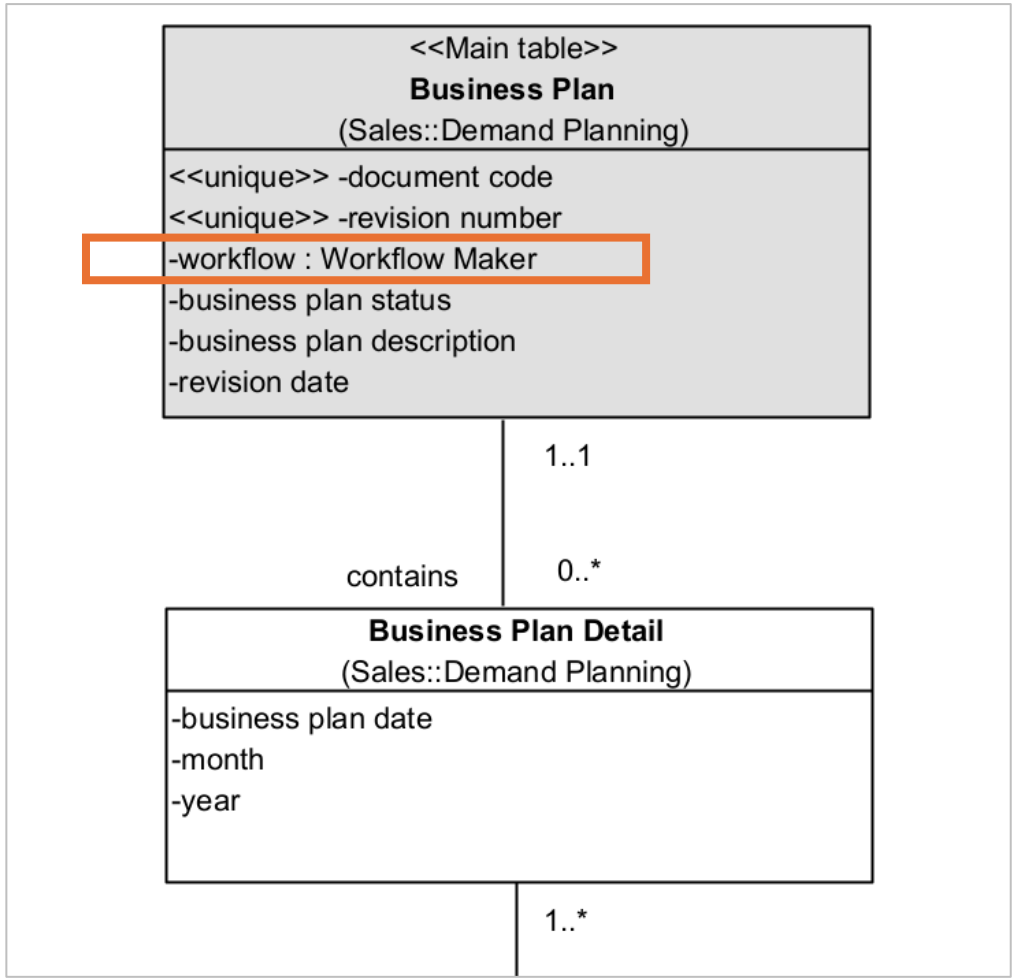
It is possible to use tags to enhance the expressiveness of attributes



WORKFLOW

USE A WORKFLOW COMPLEX TYPE AS AN ANCHOR



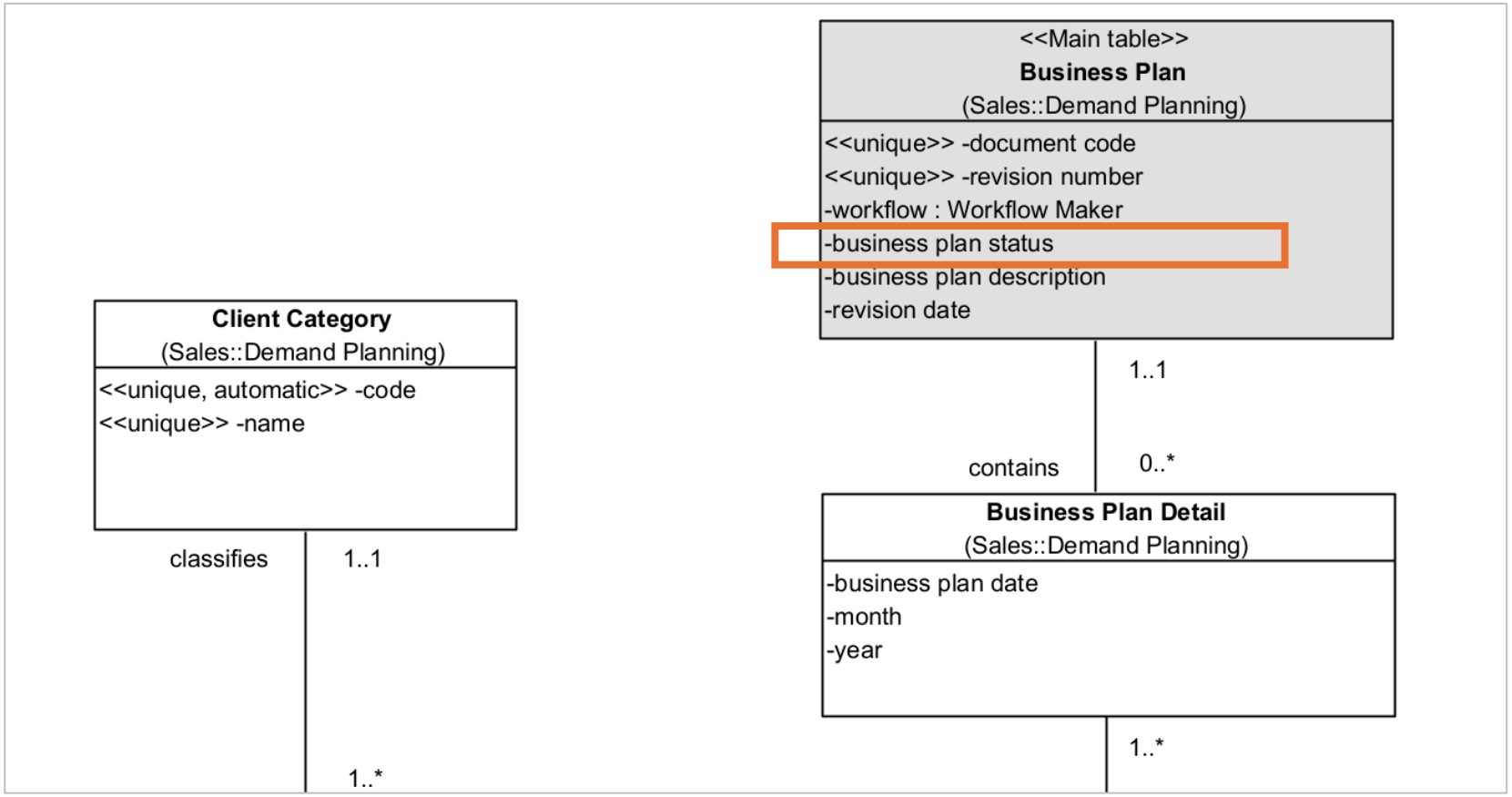


ENGAGE META

STATUS

STATUS REPRESENTS THE BUSINESS STATES A RECORD MAY TAKE OVER TIME, INDEPENDENT OF PROCESS DESIGN

(E.G., CREATED → REVIEWED → APPROVED): **SEE STATE MACHINE DIAGRAM TRAINING**



End