Profile diagram

2026-11-19 · `examples/types/uml-profile.json`

2 views · 19 cards · 10 lines

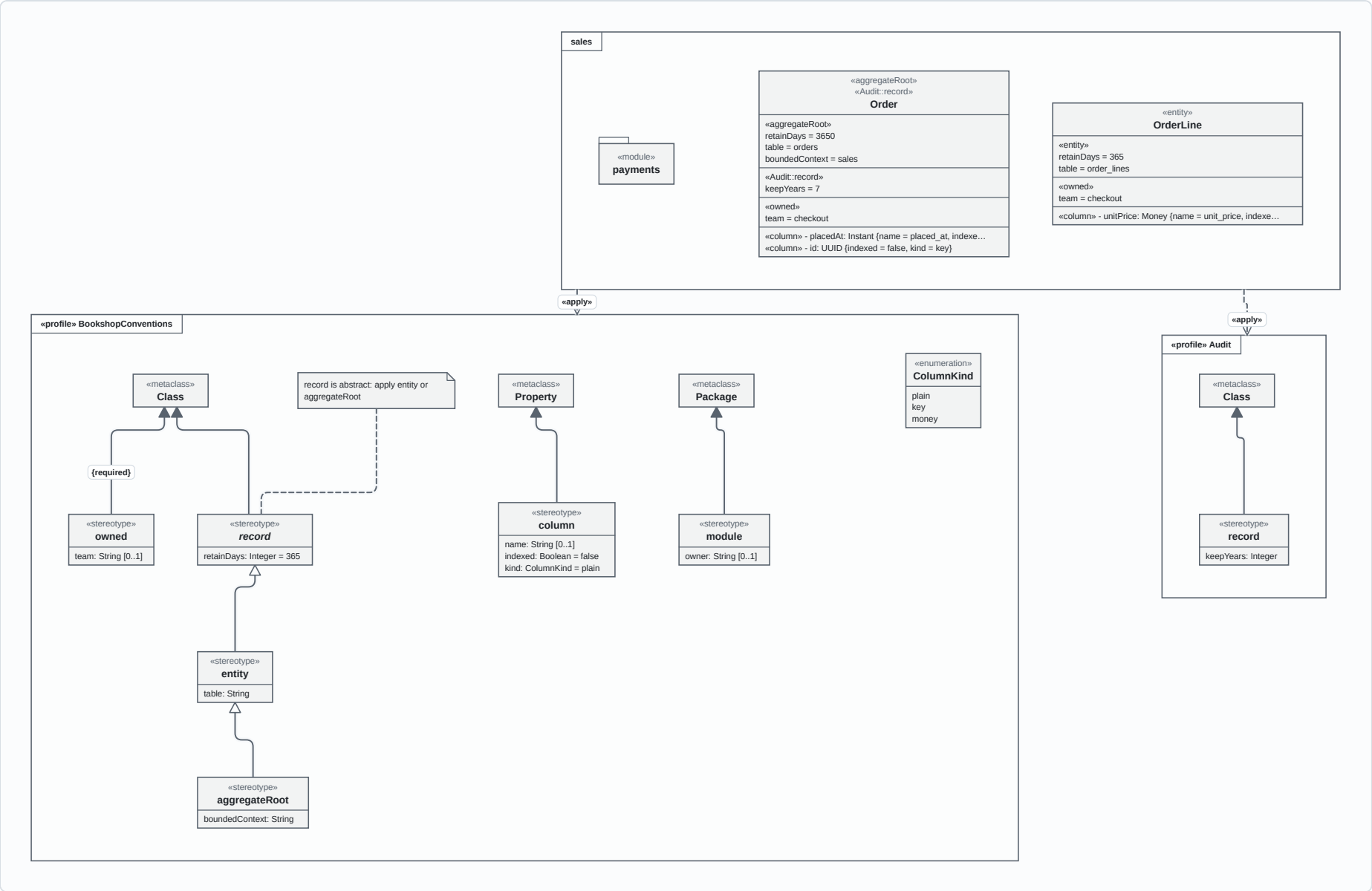
Contents

The bookshop's modelling conventions 2

The same conventions, laid out right 4

The bookshop's modelling conventions

19 cards · 10 lines



Legend **Generalization** is a kind of the one it points to **Comment link** what a comment is about **Extension** the stereotype extends the metaclass it points to **Profile application** «apply»: the package may use the profile's stereotypes

The bookshop's modelling conventions (continued)

Elements 19

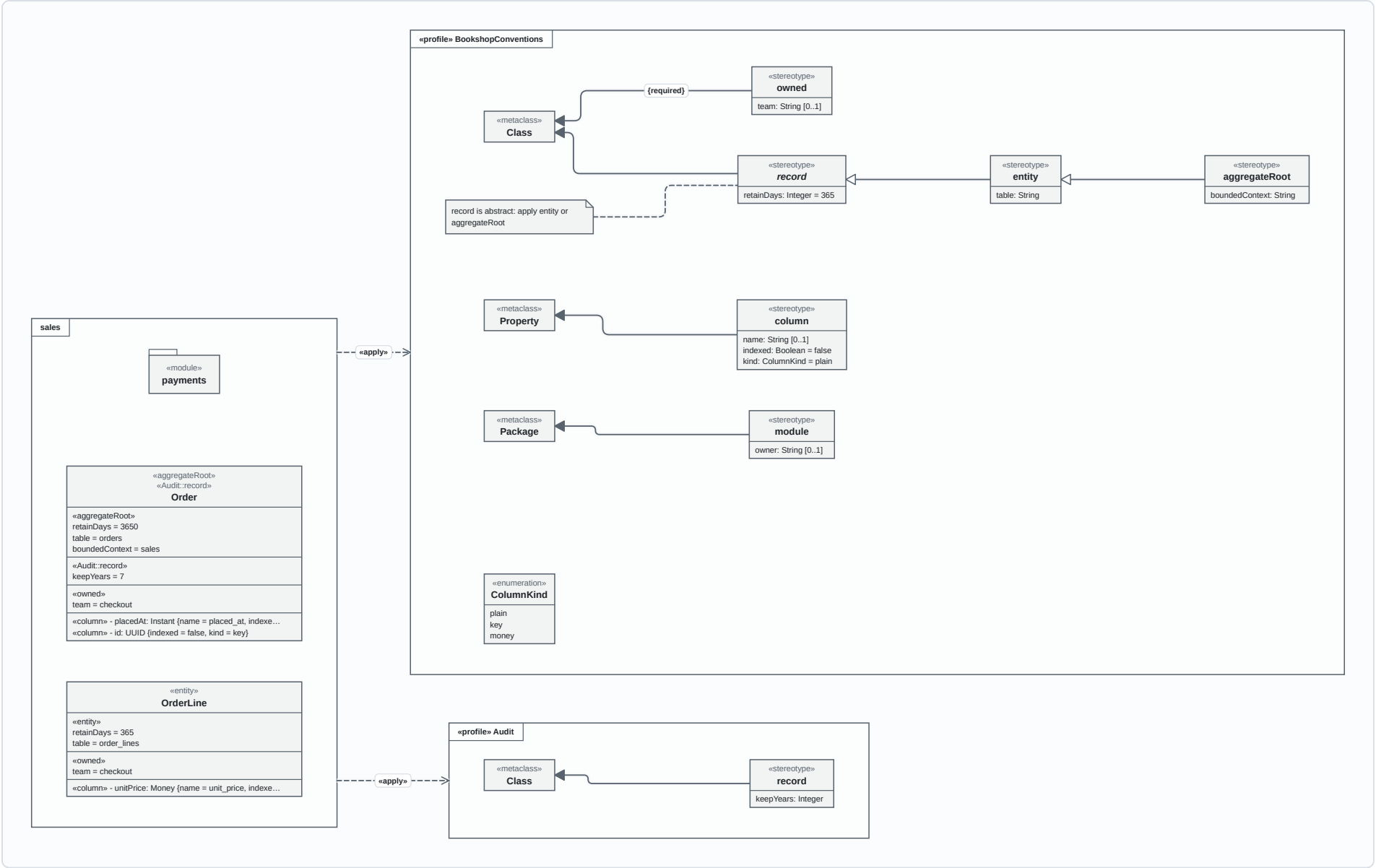
Card	Kind	Members
BookshopConventions	Profile	metaclass Class metaclass Property metaclass Package «record» «entity» «aggregateRoot» «owned» «column» «module» applied to sales
Class	Metaclass	extended by «record» extended by «owned»
Property	Metaclass	extended by «column»
Package	Metaclass	extended by «module»
record	Stereotype (abstract)	extends Class retainDays: Integer = 365
entity	Stereotype	a kind of «record» table: String
aggregateRoot	Stereotype	a kind of «entity» boundedContext: String
owned	Stereotype	extends Class {required} team: String [0..1]
column	Stereotype	extends Property name: String [0..1] indexed: Boolean = false kind: ColumnKind = plain
ColumnKind	Enumeration	plain key money
module	Stereotype	extends Package owner: String [0..1]
Audit	Profile	metaclass Class «record» applied to sales
Class	Metaclass	extended by «record»

Elements 19 (continued)

Card	Kind	Members
record	Stereotype	extends Class keepYears: Integer
sales	Package	
payments	Package	«module» owner = finance
Order	Class	«aggregateRoot» retainDays = 3650, table = orders, boundedContext = sales «Audit::record» keepYears = 7 «owned» required team = checkout - placedAt: Instant: «column» name = placed_at, indexed = true, kind = plain - id: UUID: «column» indexed = false, kind = key
OrderLine	Class	«entity» retainDays = 365, table = order_lines «owned» required team = checkout - unitPrice: Money: «column» name = unit_price, indexed = false, kind = money
record is abstract: apply entity or aggregateRoot	Comment	

Lines 10

From → to	Label	Kind
record → Class		Extension
owned → Class		Extension
column → Property		Extension
module → Package		Extension
entity → record		Generalization
aggregateRoot → entity		Generalization
record → Class		Extension
sales → BookshopConventions		Profile application
sales → Audit		Profile application
record is abstract: apply entity or aggregateRoot → record		Comment link



Legend **Generalization** is a kind of the one it points to **Comment link** what a comment is about **Extension** the stereotype extends the metaclass it points to **Profile application** «apply»: the package may use the profile's stereotypes

Elements 19

Card	Kind	Members
BookshopConventions	Profile	metaclass Class metaclass Property metaclass Package «record» «entity» «aggregateRoot» «owned» «column» «module» applied to sales
Class	Metaclass	extended by «record» extended by «owned»
Property	Metaclass	extended by «column»
Package	Metaclass	extended by «module»
record	Stereotype (abstract)	extends Class retainDays: Integer = 365
entity	Stereotype	a kind of «record» table: String
aggregateRoot	Stereotype	a kind of «entity» boundedContext: String
owned	Stereotype	extends Class {required} team: String [0..1]
column	Stereotype	extends Property name: String [0..1] indexed: Boolean = false kind: ColumnKind = plain
ColumnKind	Enumeration	plain key money
module	Stereotype	extends Package owner: String [0..1]
Audit	Profile	metaclass Class «record» applied to sales
Class	Metaclass	extended by «record»

Elements 19 (continued)

Card	Kind	Members
record	Stereotype	extends Class keepYears: Integer
sales	Package	
payments	Package	«module» owner = finance
Order	Class	«aggregateRoot» retainDays = 3650, table = orders, boundedContext = sales «Audit::record» keepYears = 7 «owned» required team = checkout - placedAt: Instant: «column» name = placed_at, indexed = true, kind = plain - id: UUID: «column» indexed = false, kind = key
OrderLine	Class	«entity» retainDays = 365, table = order_lines «owned» required team = checkout - unitPrice: Money: «column» name = unit_price, indexed = false, kind = money
record is abstract: apply entity or aggregateRoot	Comment	

Lines 10

From → to	Label	Kind
record → Class		Extension
owned → Class		Extension
column → Property		Extension
module → Package		Extension
entity → record		Generalization
aggregateRoot → entity		Generalization
record → Class		Extension
sales → BookshopConventions		Profile application
sales → Audit		Profile application
record is abstract: apply entity or aggregateRoot → record		Comment link